In [1]:

```python
import pandas as pd
import numpy as np
import nltk
nltk.download('wordnet')
nltk.download('stopwords')
from nltk.corpus import stopwords
import re
from bs4 import BeautifulSoup
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/apoorvasharma/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/apoorvasharma/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [2]:

```python
#! pip install bs4 # in case you don't have it installed

# Dataset: https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Kitchen_v1_00.tsv.gz
```

Read Data

In [3]:

```python
#loading dataset
# data = pd.read_table('amazon_reviews_us_Kitchen_v1_00.tsv')

df = pd.read_csv('amazon_reviews_us_Kitchen_v1_00.tsv', sep='\t', error_bad_lines=False)
```

```
b'Skipping line 16148: expected 15 fields, saw 22\nSkipping line 20100: expected 15 fields, saw 22\nSkipping line 45178: expected 15 fields, saw 22\nSkipping line 48700: expected 15 fields, saw 22\nSkipping line 63331: expected 15 fields, saw 22\n'
b'Skipping line 86053: expected 15 fields, saw 22\nSkipping line 88858: expected 15 fields, saw 22\nSkipping line 115017: expected 15 fields, saw 22\n'
b'Skipping line 137366: expected 15 fields, saw 22\nSkipping line 139110: expected 15 fields, saw 22\nSkipping line 165540: expected 15 fields, saw 22\nSkipping line 171813: expected 15 fields, saw 22\n'
b'Skipping line 203723: expected 15 fields, saw 22\nSkipping line 209366: expected 15 fields, saw 22\nSkipping line 211310: expected 15 fields, saw 22\nSkipping line 246351: expected 15 fields, saw 22\nSkipping line 252364: expected 15 fields, saw 22\n'
b'Skipping line 267003: expected 15 fields, saw 22\nSkipping line 268957: expected 15 fields, saw 22\nSkipping line 303336: expected 15 fields, saw 22\nSkipping line 306021: expected 15 fields, saw 22\nSkipping line 311569: expected 15 fields, saw 22\nSkipping line 316767: expected 15 fields, saw 22\nSkipping line 324009: expected 15 fields, saw 22\n'
b'Skipping line 359107: expected 15 fields, saw 22\nSkipping line 368367: expected 15 fields, saw 22\nSkipping line 381180: expected 15 fields, saw 22\nSkipping line 390453: expected 15 fields, saw 22\n'
b'Skipping line 412243: expected 15 fields, saw 22\nSkipping line 419342: expected 15 fields, saw 22\nSkipping line 457388: expected 15 fields, saw 22\n'
b'Skipping line 459935: expected 15 fields, saw 22\nSkipping line 460167: expected 15 fields, saw 22\nSkipping line 466460: expected 15 fields, saw 22\nSkipping line 500314: expected 15 fields, saw 22\nSkipping line 500339: expected 15 fields, saw 22\nSkipping line 505396: expected 15 fields, saw 22\nSkipping line 507760: expected 15 fields, saw 22\nSkipping line 513626: expected 15 fields, saw 22\n'
b'Skipping line 527638: expected 15 fields, saw 22\nSkipping line 534209: expected 15 fields, saw 22\nSkipping line 535687: expected 15 fields, saw 22\nSkipping line 547671: expected 15 fields, saw 22\nSkipping line 549054: expected 15 fields, saw 22\n'
b'Skipping line 599929: expected 15 fields, saw 22\nSkipping line 604776: expected 15 fields, saw 22\nSkipping line 609937: expected 15 fields, saw 22\nSkipping line 632059: expected 15 fields, saw 22\nSkipping line 638546: expected 15 fields, saw 22\n'
b'Skipping line 665017: expected 15 fields, saw 22\nSkipping line 677680: expected 15 fields, saw 22\nSkipping line 684370: expected 15 fields, saw 22\nSkipping line 720217: expected 15 fields, saw 29\n'
b'Skipping line 723240: expected 15 fields, saw 22\nSkipping line 723433: expected 15 fields, saw 22\nSkipping line 763891: expected 15 fields, saw 22\n'
b'Skipping line 800288: expected 15 fields, saw 22\nSkipping line 802942: expected 15 fields, saw 22\nSkipping line 803379: expected 15 fields, saw 22\nSkipping line 805122: expected 15 fields, saw 22\nSkipping line 821899: expected 15 fields, saw 22\nSkipping line 831707: expected 15 fields, saw 22\nSkipping line 842829: expected 15 fields, saw 22\nSkipping line 843604: expected 15 fields, saw 22\n'
b'Skipping line 863904: expected 15 fields, saw 22\nSkipping line 875655: expected 15 fields, saw 22\nSkipping line 886796: expected 15 fields, saw 22\nSkipping line 892299: expected 15 fields, saw 22\nSkipping line 902518: expected 15 fields, saw 22\nSkipping line 903079: expected 15 fields, saw 22\nSkipping line 912678: expected 15 fields, saw 22\n'
b'Skipping line 932953: expected 15 fields, saw 22\nSkipping line 936838: expected 15 fields, saw 22\nSkipping line 937177: expected 15 fields, saw 22\nSkipping line 947695: expected 15 fields, saw 22\nSkipping line 960713: expected 15 fields, saw 22\nSkipping line 965225: expected 15 fields, saw 22\nSkipping line 980776: expected 15 fields, saw 22\n'
b'Skipping line 999318: expected 15 fields, saw 22\nSkipping line 1007247: expected 15 fields, saw 22\nSkipping line 1015987: expected 15 fields, saw 22\nSkipping line 1018984: expected 15 fields, saw 22\nSkipping line 1028671: expected 15 fields, saw 22\n'
b'Skipping line 1063360: expected 15 fields, saw 22\nSkipping line 1066195: expected 15 fields, saw 22\nSkipping line 1066578: expected 15 fields, saw 22\nSkipping line 1066869: expected 15 fields, saw 22\nSkipping line 1068809: expected 15 fields, saw 22\nSkipping line 1069505: expected 15 fields, saw 22\nSkipping line 1087983: expected 15 fields, saw 22\nSkipping line 1108184: expected 15 fields, saw 22\n'
b'Skipping line 1118137: expected 15 fields, saw 22\nSkipping line 1142723: expected 15 fields, saw 22\nSkipping line 1152492: expected 15 fields, saw 22\nSkipping line 1156947: expected 15 fields, saw 22\nSkipping line 1172563: expected 15 fields, saw 22\n'
b'Skipping line 1209254: expected 15 fields, saw 22\nSkipping line 1212966: expected 15 fields, saw 22\nSkipping line 1236533: expected 15 fields, saw 22\nSkipping line 1237598: expected 15 fields, saw 22\n'
b'Skipping line 1273825: expected 15 fields, saw 22\nSkipping line 1277898: expected 15 fields, saw 22\nSkipping line 1283654: expected 15 fields, saw 22\nSkipping line 1286023: expected 15 fields, saw 22\nSkipping line 1302038: expected 15 fields, saw 22\nSkipping line 1305179: expected 15 fields, saw 22\n'
b'Skipping line 1326022: expected 15 fields, saw 22\nSkipping line 1338120: expected 15 fields, saw 22\nSkipping line 1338503: expected 15 fields, saw 22\nSkipping line 1338849: expected 15 fields, saw 22\nSkipping line 1341513: expected 15 fields, saw 22\nSkipping line 1346493: expected 15 fields, saw 22\nSkipping line 1373127: expected 15 fields, saw 22\n'
b'Skipping line 1389508: expected 15 fields, saw 22\nSkipping line 1413951: expected 15 fields, saw 22\nSkipping line 1433626: expected 15 fields, saw 22\n'
b'Skipping line 1442698: expected 15 fields, saw 22\nSkipping line 1472982: expected 15 fields, saw 22\nSkipping line 1482282: expected 15 fields, saw 22\nSkipping line 1487808: expected 15 fields, saw 22\nSkipping line 1500636: expected 15 fields, saw 22\n'
b'Skipping line 1511479: expected 15 fields, saw 22\nSkipping line 1532302: expected 15 fields, saw 22\nSkipping line 1537952: expected 15 fields, saw 22\nSkipping line 1539951: expected 15 fields, saw 22\nSkipping line 1541020: expected 15 fields, saw 22\n'
b'Skipping line 1594217: expected 15 fields, saw 22\nSkipping line 1612264: expected 15 fields, saw 22\nSkipping line 1615907: expected 15 fields, saw 22\nSkipping line 1621859: expected 15 fields, saw 22\n'
b'Skipping line 1653542: expected 15 fields, saw 22\nSkipping line 1671537: expected 15 fields, saw 22\nSkipping line 1672879: expected 15 fields, saw 22\nSkipping line 1674523: expected 15 fields, saw 22\nSkipping line 1677355: expected 15 fields, saw 22\nSkipping line 1703907: expected 15 fields, saw 22\n'
b'Skipping line 1713046: expected 15 fields, saw 22\nSkipping line 1722982: expected 15 fields, saw 22\nSkipping line 1727290: expected 15 fields, saw 22\nSkipping line 1744482: expected 15 fields, saw 22\n'
b'Skipping line 1803858: expected 15 fields, saw 22\nSkipping line 1810069: expected 15 fields, saw 22\nSkipping line 1829751: expected 15 fields, saw 22\nSkipping line 1831699: expected 15 fields, saw 22\n'
b'Skipping line 1863131: expected 15 fields, saw 22\nSkipping line 1867917: expected 15 fields, saw 22\nSkipping line 1874790: expected 15 fields, saw 22\nSkipping line 1879952: expected 15 fields, saw 22\nSkipping line 1880501: expected 15 fields, saw 22\nSkipping line 1886655: expected 15 fields, saw 22\nSkipping line 1887888: expected 15 fields, saw 22\nSkipping line 1894286: expected 15 fields, saw 22\nSkipping line 1895400: expected 15 fields, saw 22\n'
b'Skipping line 1904040: expected 15 fields, saw 22\nSkipping line 1907604: expected 15 fields, saw 22\nSkipping line 1915739: expected 15 fields, saw 22\nSkipping line 1921514: expected 15 fields, saw 22\nSkipping line 1939428: expected 15 fields, saw 22\nSkipping line 1944342: expected 15 fields, saw 22\nSkipping line 1949699: expected 15 fields, saw 22\nSkipping line 1961872: expected 15 fields, saw 22\n'
b'Skipping line 1968846: expected 15 fields, saw 22\nSkipping line 1999941: expected 15 fields, saw 22\nSkipping line 2001492: expected 15 fields, saw 22\nSkipping line 2011204: expected 15 fields, saw 22\nSkipping line 2025295: expected 15 fields, saw 22\n'
b'Skipping line 2041266: expected 15 fields, saw 22\nSkipping line 2073314: expected 15 fields, saw 22\nSkipping line 2080133: expected 15 fields, saw 22\nSkipping line 2088521: expected 15 fields, saw 22\n'
b'Skipping line 2103490: expected 15 fields, saw 22\nSkipping line 2115278: expected 15 fields, saw 22\nSkipping line 2153174: expected 15 fields, saw 22\nSkipping line 2161731: expected 15 fields, saw 22\n'
b'Skipping line 2165250: expected 15 fields, saw 22\nSkipping line 2175132: expected 15 fields, saw 22\nSkipping line 2206817: expected 15 fields, saw 22\nSkipping line 2215848: expected 15 fields, saw 22\nSkipping line 2223811: expected 15 fields, saw 22\n'
b'Skipping line 2257265: expected 15 fields, saw 22\nSkipping line 2259163: expected 15 fields, saw 22\nSkipping line 2263291: expected 15 fields, saw 22\n'
b'Skipping line 2301943: expected 15 fields, saw 22\nSkipping line 2304371: expected 15 fields, saw 22\nSkipping line 2306015: expected 15 fields, saw 22\nSkipping line 2312186: expected 15 fields, saw 22\nSkipping line 2314740: expected 15 fields, saw 22\nSkipping line 2317754: expected 15 fields, saw 22\n'
b'Skipping line 2383514: expected 15 fields, saw 22\n'
b'Skipping line 2449763: expected 15 fields, saw 22\n'
b'Skipping line 2589323: expected 15 fields, saw 22\n'
b'Skipping line 2775036: expected 15 fields, saw 22\n'
b'Skipping line 2935174: expected 15 fields, saw 22\n'
b'Skipping line 3078830: expected 15 fields, saw 22\n'
b'Skipping line 3123091: expected 15 fields, saw 22\n'
b'Skipping line 3185533: expected 15 fields, saw 22\n'
b'Skipping line 4150395: expected 15 fields, saw 22\n'
b'Skipping line 4748401: expected 15 fields, saw 22\n'
```

Keep Reviews and Ratings

In [7]:

```python
data = df[["review_id","review_body", "star_rating"]]
# print(data.info())
data=data.dropna(subset=['review_body'])
# print(data.info())
```

Labelling Reviews:

The reviews with rating 4,5 are labelled to be 1 and 1,2 are labelled as 0. Discard the reviews with rating 3'

In [8]:

```python
discarded_count = len(data[data['star_rating'] == 3.0])
# print("Neutral Reviews: ", discarded_count)
```

```
Neutral Reviews:  349539
```

In [9]:

```python
data.drop(data[data['star_rating'] == 3.0].index, inplace = True)
# print(data.info())

data['label'] = np.where(data['star_rating']>=4.0, 1, 0)
# data.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4525105 entries, 0 to 4874889
Data columns (total 3 columns):
 #   Column       Dtype  
---  ------       -----  
 0   review_id    object 
 1   review_body  object 
 2   star_rating  float64
dtypes: float64(1), object(2)
memory usage: 138.1+ MB
None
```

We select 200000 reviews randomly with 100,000 positive and 100,000 negative reviews.

In [10]:

```python
pos_data = data[data['label']==1]
neg_data = data[data['label']==0]

# print("Positive Reviews: ", len(pos_data))
# print("Negative Reviews: ", len(neg_data))

pos = pos_data.sample(n=100000, random_state=1)
neg = neg_data.sample(n=100000, random_state=1)

data_both = pd.concat([pos, neg], ignore_index=True)
data_both = data_both.sample(frac=1).reset_index(drop=True)
# data_both.head(10)
```

```
Positive Reviews:  3856296
Negative Reviews:  668809
```

In [12]:

```python
from statistics import mean

data_both['review_body'] = data_both['review_body'].astype(str)

avg_len1 = mean([len(s) for s in data_both["review_body"]])

#before cleaing and pre-processing
# print("BEFORE CLEANING")
# print(*[rev+", length: "+str(len(rev))+"\n" for rev in data_both["review_body"].head(3)])
# print("Average review length before processing: "+ str(avg_len1))
```

```
BEFORE CLEANING
It is as pictured.  It is cool, unique gift idea.  I also purchased &#34;The Best Ice Mold&#34; (Amazon) and the round ice ball fits perfectly in this glass.  This was a birthday themed idea that I put together for my husband.  I also purchased  a bottle of Markers Mark whiskey (local) and a MM glass and coaster (Amazon).  He really liked it.  FYI: I also bought some real good cigars (locally) and a travel humidor (Amazon). to round out the whole theme....because lets face it, buying a surprise birthday gift for my husband of 30 years is a challenge LoL but a fun one.  I really scored with this theme. :-), length: 612
 The tags are satin finish and very hard to write on. Either the ink wipes off (sharpie) or is hard to write on (ball point)., length: 124
 This was really cheap made. It serves the purpose but tore very easily, length: 70

Average review length before processing: 323.274835
```

Data Cleaning

In [13]:

```python
#regexs for expressions to be removed
replace_url = re.compile(r'https?:\/\/.*[\r\n]*')
remove_href = re.compile(r'\<a href')
remove_amp = re.compile(r'&amp;')
remove_sym = re.compile(r'[_"\-;%()|+&=*%.,~!?:#$\[\]/]')
remove_br = re.compile(r'<br />')
remove_space = re.compile(r' +')
replace_brackets = re.compile('[/(){}\[\]\|,;+_-]')
remove_numbers = re.compile("[\d]")


def cleaning_text(text):
    text = text.lower() # lowercase text
    text = replace_url.sub("", text) # delete URLs from text
    text = remove_numbers.sub("", text) # delete numbers tags from text
    text = remove_href.sub('', text) # delete HTML tags from text
    text = remove_amp.sub('', text) # delete "&" from text
    text = remove_sym.sub(' ', text)# replace "remove_sym" symbols by space in text
    text = remove_br.sub(' ', text) # delete HTML tags from text
    text = replace_brackets.sub(' ', text) # replace "replace_brackets" symbols by space in text
    text = remove_space.sub(' ', text)

    return text

#applying all the above cleaning to the dataset
data_both['review_body'] = data_both['review_body'].apply(cleaning_text)
# data_both.head(10)
```

Out[13]:

	review_id	review_body	star_rating	label
0	RZMWGGF37ISTO	it is as pictured it is cool unique gift idea ...	4.0	1
1	R3BN73A6RT3LFP	the tags are satin finish and very hard to wri...	1.0	0
2	R1QRZ3W59ZMREX	this was really cheap made it serves the purpo...	2.0	0
3	R1T1KAXBW2HJEC	this product was shipped as a present to carry...	1.0	0
4	R1IP7DH28CEL72	unacceptable for my cooktop which requires tha...	1.0	0
5	R2W5RCO3L4VBHH	the only downside i see is that it takes up a ...	5.0	1
6	R1NIVXSOO7SH3J	i'm shocked this has good reviews i just got i...	2.0	0
7	R1RSNTGPQMLK2U	really like this pan perfect for large dishes ...	4.0	1
8	R3QFR9AAC4DGWQ	i purchased these when they were advertised as...	2.0	0
9	R256OUIH0I1QXW	quite large and a little cheap looking i have ...	2.0	0

perform contractions on the reviews.

In [14]:

```python
def contractionfunction(s):

    s = re.sub(r"won\'t", "will not", s)
    s = re.sub(r"can\'t", "can not", s)
    s = re.sub(r"ain\'t", "are not", s)
    s = re.sub(r"\'cause","because",s)
    s = re.sub(r"let\'s", "let us",s)
    s = re.sub(r"o\'clock", "of the clock", s)
    s = re.sub(r"ma\'am", "madam",s)

    s = re.sub(r"n\'t", " not", s)
    s = re.sub(r"\'re", " are", s)
    s = re.sub(r"\'s", " is", s)
    s = re.sub(r"\'d", " would", s)
    s = re.sub(r"\'ll", " will", s)
    s = re.sub(r"\'t", " not", s)
    s = re.sub(r"\'ve", " have", s)
    s = re.sub(r"\'m", " am", s)
    return s

data_both['review_body'] = data_both['review_body'].apply(contractionfunction)
# data_both.head(10)
```

Out[14]:

	review_id	review_body	star_rating	label
0	RZMWGGF37ISTO	it is as pictured it is cool unique gift idea ...	4.0	1
1	R3BN73A6RT3LFP	the tags are satin finish and very hard to wri...	1.0	0
2	R1QRZ3W59ZMREX	this was really cheap made it serves the purpo...	2.0	0
3	R1T1KAXBW2HJEC	this product was shipped as a present to carry...	1.0	0
4	R1IP7DH28CEL72	unacceptable for my cooktop which requires tha...	1.0	0
5	R2W5RCO3L4VBHH	the only downside i see is that it takes up a ...	5.0	1
6	R1NIVXSOO7SH3J	i am shocked this has good reviews i just got ...	2.0	0
7	R1RSNTGPQMLK2U	really like this pan perfect for large dishes ...	4.0	1
8	R3QFR9AAC4DGWQ	i purchased these when they were advertised as...	2.0	0
9	R256OUIH0I1QXW	quite large and a little cheap looking i have ...	2.0	0

In [15]:

```python
avg_len2 = mean([len(s) for s in data_both["review_body"]])

# #before cleaing and pre-processing
# print("AFTER CLEANING AND BEFORE PREPROCESSING")
# print(*[rev+", length: "+str(len(rev))+"\n" for rev in data_both["review_body"].head(3)])
# print("Average review length after cleaning and before processing: "+ str(avg_len2))
```

```
AFTER CLEANING AND BEFORE PREPROCESSING
it is as pictured it is cool unique gift idea i also purchased the best ice mold amazon and the round ice ball fits perfectly in this glass this was a birthday themed idea that i put together for my husband i also purchased a bottle of markers mark whiskey local and a mm glass and coaster amazon he really liked it fyi i also bought some real good cigars locally and a travel humidor amazon to round out the whole theme because lets face it buying a surprise birthday gift for my husband of years is a challenge lol but a fun one i really scored with this theme , length: 563
 the tags are satin finish and very hard to write on either the ink wipes off sharpie or is hard to write on ball point , length: 119
 this was really cheap made it serves the purpose but tore very easily, length: 69

Average review length after cleaning and before processing: 312.739695
```

Pre-processing

remove the stop words

In [16]:

```python
from nltk.corpus import stopwords

def remove_stopwords(text):
    STOPWORDS = set(stopwords.words('english'))
    text = ' '.join(word for word in text.split() if word not in STOPWORDS) # delete stopwors from text
    return text

data_both['review_body'] = data_both['review_body'].apply(remove_stopwords)
# data_both.head(10)
```

Out[16]:

	review_id	review_body	star_rating	label
0	RZMWGGF37ISTO	pictured cool unique gift idea also purchased ...	4.0	1
1	R3BN73A6RT3LFP	tags satin finish hard write either ink wipes ...	1.0	0
2	R1QRZ3W59ZMREX	really cheap made serves purpose tore easily	2.0	0
3	R1T1KAXBW2HJEC	product shipped present carry overseas paid da...	1.0	0
4	R1IP7DH28CEL72	unacceptable cooktop requires cookware totally...	1.0	0
5	R2W5RCO3L4VBHH	downside see takes little room kitchen larger ...	5.0	1
6	R1NIVXSOO7SH3J	shocked good reviews got mail today excited us...	2.0	0
7	R1RSNTGPQMLK2U	really like pan perfect large dishes used calp...	4.0	1
8	R3QFR9AAC4DGWQ	purchased advertised popsicle holders pain kno...	2.0	0
9	R256OUIH0I1QXW	quite large little cheap looking seen much nic...	2.0	0

perform tokenization and lemmatization

In [17]:

```python
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import TweetTokenizer
from nltk.stem import PorterStemmer

#creating an tokenizer object
text_tokenizer = TweetTokenizer(strip_handles=True, reduce_len=True)

def tokenize_text(text):
    #tokenize text
    review_tokens = text_tokenizer.tokenize(text)

    return review_tokens

#tokenizing each review in the dataset
data_both['review_body'] = data_both['review_body'].apply(tokenize_text)
# data_both.head(10)
```

Out[17]:

	review_id	review_body	star_rating	label
0	RZMWGGF37ISTO	[pictured, cool, unique, gift, idea, also, pur...	4.0	1
1	R3BN73A6RT3LFP	[tags, satin, finish, hard, write, either, ink...	1.0	0
2	R1QRZ3W59ZMREX	[really, cheap, made, serves, purpose, tore, e...	2.0	0
3	R1T1KAXBW2HJEC	[product, shipped, present, carry, overseas, p...	1.0	0
4	R1IP7DH28CEL72	[unacceptable, cooktop, requires, cookware, to...	1.0	0
5	R2W5RCO3L4VBHH	[downside, see, takes, little, room, kitchen, ...	5.0	1
6	R1NIVXSOO7SH3J	[shocked, good, reviews, got, mail, today, exc...	2.0	0
7	R1RSNTGPQMLK2U	[really, like, pan, perfect, large, dishes, us...	4.0	1
8	R3QFR9AAC4DGWQ	[purchased, advertised, popsicle, holders, pai...	2.0	0
9	R256OUIH0I1QXW	[quite, large, little, cheap, looking, seen, m...	2.0	0

In [18]:

```python
#stemmer object
stemmer=PorterStemmer()

#Lemmatizer object
lemmmatizer=WordNetLemmatizer()

def lemmatize_tokens(tokens):
    #stemming
    stemmed_tokens=[stemmer.stem(t) for t in tokens]

    #Stemming with lemmatization
    lemmatized_tokens = [lemmmatizer.lemmatize(st) for st in stemmed_tokens]
    lemmatize_review = ' '.join(lemmatized_tokens)

    return lemmatize_review

data_both['review_body'] = data_both['review_body'].apply(lemmatize_tokens)
```

In [19]:

```python
from statistics import mean

avg_len3 = mean([len(s) for s in data_both["review_body"]])

# #after cleaing and pre-processing
# print("AFTER PREPROCESSING")
# print(*[rev+", length: "+str(len(rev))+"\n" for rev in data_both["review_body"].head(3)])
# print("Average review lentgh after processing: "+ str(avg_len3))
```

```
AFTER PREPROCESSING
pictur cool uniqu gift idea also purchas best ice mold amazon round ice ball fit perfectli glass birthday theme idea put togeth husband also purchas bottl marker mark whiskey local mm glass coaster amazon realli like fyi also bought real good cigar local travel humidor amazon round whole theme let face buy surpris birthday gift husband year challeng lol fun one realli score theme, length: 382
 tag satin finish hard write either ink wipe sharpi hard write ball point, length: 72
 realli cheap made serv purpos tore easili, length: 41

Average review lentgh after processing: 176.841055
```

In [20]:

```python
from sklearn.model_selection import train_test_split

#splitting the dataset into train and test sets
X = data_both.review_body
y = data_both.label
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Perceptron

In [22]:

```python
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.linear_model import Perceptron

#perceptron for reviews
perceptron = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', Perceptron(tol=1e-3, random_state=0)),
               ])
#training the model
perceptron.fit(X_train, y_train)

# print("---PERCEPTRON---")
#predicting the classes on train
y_pred_perceptron_train = perceptron.predict(X_train)

# #metrics
# print('Train Accuracy: %s' % accuracy_score(y_pred_perceptron_train, y_train))
# print('Train Precision: %s' % precision_score(y_pred_perceptron_train, y_train))
# print('Train Recall: %s' % recall_score(y_pred_perceptron_train, y_train))
# print('Train F1: %s \n' % f1_score(y_pred_perceptron_train, y_train))

#predicting the classes on test
y_pred_perceptron = perceptron.predict(X_test)

#metrics
# print('Test Accuracy: %s' % accuracy_score(y_pred_perceptron, y_test))
# print('Test Precision: %s' % precision_score(y_pred_perceptron, y_test))
# print('Test Recall: %s' % recall_score(y_pred_perceptron, y_test))
# print('Test F1: %s' % f1_score(y_pred_perceptron, y_test))
```

```
---PERCEPTRON---
Train Accuracy: 0.88214375
Train Precision: 0.910613269396471
Train Recall: 0.8617466320018908
Train F1: 0.885506287227002

Test Accuracy: 0.84735
Test Precision: 0.8781185683449626
Test Recall: 0.8262717868782768
Test F1: 0.8514065998247835
```

SVM

In [23]:

```python
#SVM classifier for reviews
from sklearn.linear_model import SGDClassifier

svm_classifier = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', SGDClassifier(loss='hinge', penalty='l2',alpha=1e-5, max_iter=100, tol=None)),
               ])
#training the model
svm_classifier.fit(X_train, y_train)

# print("---SVM---")
#predicting the classes on train
y_pred_svm_train = svm_classifier.predict(X_train)

#metrics
# print('Train Accuracy: %s' % accuracy_score(y_pred_svm_train, y_train))
# print('Train Precision: %s' % precision_score(y_pred_svm_train, y_train))
# print('Train Recall: %s' % recall_score(y_pred_svm_train, y_train))
# print('Train F1: %s \n' % f1_score(y_pred_svm_train, y_train))

#predicting the classes on test
y_pred_svm = svm_classifier.predict(X_test)

#metrics
# print('Test Accuracy: %s' % accuracy_score(y_pred_svm, y_test))
# print('Test Precision: %s' % precision_score(y_pred_svm, y_test))
# print('Test Recall: %s' % recall_score(y_pred_svm, y_test))
# print('Test F1: %s' % f1_score(y_pred_svm, y_test))
```

```
---SVM---
Train Accuracy: 0.91041875
Train Precision: 0.9053559609885238
Train Recall: 0.9147803265450324
Train F1: 0.9100437448613908

Test Accuracy: 0.893275
Test Precision: 0.8891119923698609
Test Recall: 0.8958122597612785
Test F1: 0.8924495502985413
```

Logistic Regression

In [24]:

```python
#logistic regression model for reviews
from sklearn.linear_model import LogisticRegression

log_reg = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', LogisticRegression(n_jobs=5, C=1e6)),
               ])
#training the model
log_reg.fit(X_train, y_train)

# print("---LR---")

#predicting the classes on train
y_pred_lr_train = log_reg.predict(X_train)

#metrics
# print('Train Accuracy: %s' % accuracy_score(y_pred_lr_train, y_train))
# print('Train Precision: %s' % precision_score(y_pred_lr_train, y_train))
# print('Train Recall: %s' % recall_score(y_pred_lr_train, y_train))
# print('Train F1: %s \n' % f1_score(y_pred_lr_train, y_train))

#predicting the classes on test
y_pred_lr = log_reg.predict(X_test)

#metrics
# print('Test Accuracy: %s' % accuracy_score(y_pred_lr, y_test))
# print('Test Precision: %s' % precision_score(y_pred_lr, y_test))
# print('Test Recall: %s' % recall_score(y_pred_lr, y_test))
# print('Test F1: %s' % f1_score(y_pred_lr, y_test))
```

```
---LR---
Train Accuracy: 0.9239125
Train Precision: 0.9226264064236567
Train Recall: 0.9251449393320895
Train F1: 0.9238839564836815

Test Accuracy: 0.881875
Test Precision: 0.8807790773555544
Test Recall: 0.8818858061921995
Test F1: 0.8813320943315669
```

Naive Bayes

In [25]:

```python
#Multinomial Naive Bayes Classifier for reviews
from sklearn.naive_bayes import MultinomialNB

multi_naive_bayes = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf', MultinomialNB()),
              ])
#training the classifier
multi_naive_bayes.fit(X_train, y_train)

# print("---NB---")

#predicting the classes on train
y_pred_nb_train = multi_naive_bayes.predict(X_train)

#metrics
# print('Train Accuracy: %s' % accuracy_score(y_pred_nb_train, y_train))
# print('Train Precision: %s' % precision_score(y_pred_nb_train, y_train))
# print('Train Recall: %s' % recall_score(y_pred_nb_train, y_train))
# print('Train F1: %s \n' % f1_score(y_pred_nb_train, y_train))

#predicting the classes on test
y_pred_nb = multi_naive_bayes.predict(X_test)

#metrics
# print('Test Accuracy: %s' % accuracy_score(y_pred_nb, y_test))
# print('Test Precision: %s' % precision_score(y_pred_nb, y_test))
# print('Test Recall: %s' % recall_score(y_pred_nb, y_test))
# print('Test F1: %s' % f1_score(y_pred_nb, y_test))
```

```
---NB---
Train Accuracy: 0.87893125
Train Precision: 0.8714269658712022
Train Recall: 0.884919729133379
Train F1: 0.8781215198474868

Test Accuracy: 0.867075
Test Precision: 0.8590934190050701
Test Recall: 0.8720953933958419
Test F1: 0.8655455809836895
```

In [37]:

```python
#final output

print("\nNeutral Reviews:", discarded_count, "Positive Reviews:", len(pos_data), "Negative Reviews:", len(neg_data), "\n")

print(str(avg_len1)+","+ str(avg_len2))
print(str(avg_len2)+","+ str(avg_len3)+"\n")

print(str(accuracy_score(y_pred_perceptron_train, y_train))+","+str(precision_score(y_pred_perceptron_train, y_train))+","+ str(recall_score(y_pred
print(str(accuracy_score(y_pred_svm_train, y_train))+","+str(precision_score(y_pred_svm_train, y_train))+","+ str(recall_score(y_pred_svm_train, y_
print(str(accuracy_score(y_pred_lr_train, y_train))+","+str(precision_score(y_pred_lr_train, y_train))+","+ str(recall_score(y_pred_lr_train, y_tra
print(str(accuracy_score(y_pred_nb_train, y_train))+","+str(precision_score(y_pred_nb_train, y_train))+","+ str(recall_score(y_pred_nb_train, y_tra
```

```
Neutral Reviews: 349539 Positive Reviews: 3856296 Negative Reviews: 668809

323.274835,312.739695
312.739695,176.841055

0.88214375,0.910613269396471,0.8617466320018908,0.885506287227002,0.84735,0.8781185683449626,0.8262717868782768,0.8514065998247835

0.91041875,0.9053559609885238,0.9147803265450324,0.9100437448613908,0.893275,0.8891119923698609,0.8958122597612785,0.8924495502985413

0.9239125,0.9226264064236567,0.9251449393320895,0.9238839564836815,0.881875,0.8807790773555544,0.8818858061921995,0.8813320943315669

0.87893125,0.8714269658712022,0.884919729133379,0.8781215198474868,0.867075,0.8590934190050701,0.8720953933958419,0.8655455809836895
```