In [1]:

```
import numpy as np
import pandas as pd

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/amazon-product-review-spam-and-non-spam/Home_and_Kitchen/Home_and_Kitchen.j
son
/kaggle/input/amazon-product-review-spam-and-non-spam/part.json/part.json
/kaggle/input/amazon-product-review-spam-and-non-spam/Electronics/Electronics.json
/kaggle/input/amazon-product-review-spam-and-non-spam/separate.json/separate.json
/kaggle/input/amazon-product-review-spam-and-non-spam/Clothing_Shoes_and_Jewelry/Clothing
_Shoes_and_Jewelry.json
/kaggle/input/amazon-product-review-spam-and-non-spam/Sports_and_Outdoors/Sports_and_Outd
oors.json
/kaggle/input/amazon-product-review-spam-and-non-spam/Cell_Phones_and_Accessories/Cell_Ph
ones_and_Accessories.json
/kaggle/input/amazon-product-review-spam-and-non-spam/Toys_and_Games/Toys_and_Games.json
```

for importing json file so that we don't get any error in loading.

In [2]:

```
import json
from pandas.io.json import json_normalize
```

In [3]:

```
N = 1000000
with open('../input/amazon-product-review-spam-and-non-spam/Cell_Phones_and_Accessories/C
ell_Phones_and_Accessories.json') as json_file:
    data = [next(json_file) for x in range(N)]
    data = list(map(json.loads, data))
```

For changing the file format like _id-$oid to id and list in helpful to helpfull and not helpfull.

In [4]:

```
for result in data:
    result['id'] = result['_id']['$oid']
    result['helpfull'] = result['helpful'][0]
    result['not_helpfull'] = result['helpful'][1]
    del result['helpful']
    del result['_id']
```

json to pandas dataframe and then dropping the rows which contains null values

In [5]:

```
df = pd.DataFrame(data)
df = df.dropna()
df.head()
```

Out[5]:

	reviewerID	asin	reviewerName	reviewText	overall	summary	unixReviewTime	reviewTime	
0	**A3HVRXV0LVJN7**	**0110400550**	**BiancaNicole**	**Best phone case ever . Everywhere I go I get**	**5.0**	**A++++**	**1358035200**	**01 13, 2013**	**Cell_Phones**

	reviewerID	asin	reviewerName	reviewText	overall	summary	unixReviewTime	reviewTime	a...
1	A1BJGDS0L1IO6I	0110400550	cf "t"	ITEM NOT SENT from Blue Top Company in Hong Ko...	1.0	ITEM NOT SENT!!	1359504000	01 30, 2013	Cell_Phones
2	A1YX2RBMS1L9L	0110400550	Andrea Busch	Saw this same case at a theme park store for 2...	5.0	Great product	1353542400	11 22, 2012	Cell_Phones
3	A180NNPPKWCCU0	0110400550	Aniya pennington	case fits perfectly and I always gets complime...	5.0	Perfect	1374105600	07 18, 2013	Cell_Phones
4	A30P2CYOUYAJM8	0110400550	Gene	I got this for my 14 year old sister. She lov...	4.0	Cool purchase.	1363737600	03 20, 2013	Cell_Phones

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 993759 entries, 0 to 999999
Data columns (total 13 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   reviewerID      993759 non-null  object 
 1   asin            993759 non-null  object 
 2   reviewerName    993759 non-null  object 
 3   reviewText      993759 non-null  object 
 4   overall         993759 non-null  float64
 5   summary         993759 non-null  object 
 6   unixReviewTime  993759 non-null  int64  
 7   reviewTime      993759 non-null  object 
 8   category        993759 non-null  object 
 9   class           993759 non-null  float64
 10  id              993759 non-null  object 
 11  helpfull        993759 non-null  int64  
 12  not_helpfull    993759 non-null  int64  
dtypes: float64(2), int64(3), object(8)
memory usage: 106.1+ MB
```

In [7]:

```
df.corr()
```

Out[7]:

	overall	unixReviewTime	class	helpfull	not_helpfull
overall	1.000000	0.027549	0.910217	-0.009687	-0.024900
unixReviewTime	0.027549	1.000000	0.016142	-0.105645	-0.122660
class	0.910217	0.016142	1.000000	-0.006871	-0.020781
helpfull	-0.009687	-0.105645	-0.006871	1.000000	0.990947
not_helpfull	-0.024900	-0.122660	-0.020781	0.990947	1.000000

In [8]:

```
df[df['not_helpfull']==0].value_counts()
```

Out[8]:

```
reviewerID            asin        reviewerName      reviewText
overall  summary
unixReviewTime  reviewTime   category                    class  id
helpfull  not_helpfull
A0004478EF5NFPHLGCWG  B004MG8KCS  STEPHANIE FIELDS  Just for show, not very durable, or a
of great protection.  Colorful and easy to coordinate, with clothes.  Enjoyed by everyone
.
3.0      Cute
1375056000      07 29, 2013  Cell_Phones_and_Accessories  0.0    5a1321f8741a2384e80e7b02
0         0              1
A3IRRWCAC7L40T        B005GSYOVM  KARI M. PEAK      I wouldn't recommend this to anyone.
When I opened it, it was already scratched and with a little pressure you can rub the pai
nt off.  My disappointing.
1.0      Came scratched and poorly made
1357516800      01 7, 2013   Cell_Phones_and_Accessories  0.0    5a132204741a2384e8125ab9
0         0              1
A3IRPYR5JKH7EB        B005JHIYLG  Chris             Bought this for my IPhone 4S and it f
its perfectly and looks very sharp.Very much worth the price. Actually nicer then covers
I've checked out in the stores going for 10.00 to 20.00 dollars.This is well worth the pu
rchase. Looks good with my Black IPHONE , and I can see it looking just as sharp with a W
hite IPHONE also.You can't go wrong with this item for its price. Yes you will need to be
patient because it is mailed from over seas but its well worth the wait. I'm Very happy w
ith this purchase.  5.0      Sharp and great price
1384300800      11 13, 2013  Cell_Phones_and_Accessories  1.0    5a132205741a2384e812c342
0         0              1
A3IRQ3XHIRPMTC        B0056IKQCS  Mike Augsburger   the moshi ivisor ag screen protector
is one of the best screen savers u can can buy on the market
5.0      phone protection
1365033600      04 4, 2013   Cell_Phones_and_Accessories  1.0    5a132200741a2384e8110ae5
0         0              1
A3IRQPGBHVRQ8X        B00275ZYGG  Invader Zee       The case looks great, but it doesn't
have a screen protector.  It's a good case though, especially for the price...I just used
the screen cover from my old case that was $30 at the AT&T Store...
3.0      Good for the price
1262217600      12 31, 2009  Cell_Phones_and_Accessories  0.0    5a1321e0741a2384e8067bc2
0         0              1

..
A297GYTC2AW2IS        B004T36GCU  daz dillinger     just got to use my phone an so far it
s great,looks good an works even better,good buy for anyone who wants a budget phone
5.0      htc inspire
1338508800      06 1, 2012   Cell_Phones_and_Accessories  1.0    5a1321fa741a2384e80f18f2
0         0              1
A297HIUFBLPEL7        B001R5KQGK  Jay Gillette      Works Great, awsome deal for the mone
y.  Will definetly shop here again.  Same one was for sale at radio shack for 8 times the
price.
5.0      Works great, awsome deal for the money.
1366588800      04 22, 2013  Cell_Phones_and_Accessories  1.0    5a1321df741a2384e8061737
0         0              1
                      B005ERQBW0  Jay Gillette      Battery is great; back piece (door)
is flimsy in less than a month is starting to fall apart.  For the extra battery life, it
's well worth it, will proably have to replace the back after a month or two though.
4.0      Battery is great; back piece (door) is flimsy in less than a month is starting t
o fall apart.  1366588800      04 22, 2013  Cell_Phones_and_Accessories  1.0    5a1322037
41a2384e8120604  0         0              1
A297HKVHRY9PES        B00490Z2J6  Cat "CatPA"       A very useful little gadget. Works sm
oothly and it's compact enough to throw in my purse. I bought one for home as well.
5.0      Great!
1327968000      01 31, 2012  Cell_Phones_and_Accessories  1.0    5a1321f2741a2384e80c6321
0         0              1
AZZZVH7FYD0UR         B0045KJAK2  stalin            I was skeptical about the price, aft
er receiving the product very pleased and happy.It just works fine flawlessly.Thanks
5.0      Works flawlessly
1364515200      03 29, 2013  Cell_Phones_and_Accessories  1.0    5a1321f1741a2384e80bfb8c
0         0              1
Length: 704484, dtype: int64
```

In [9]:

```
df = df.drop(["asin","reviewerName","category","id"],axis = 1)
```

In [10]:

```
df["review"]=df["reviewText"]
df = df.drop(["reviewText","summary"],axis=1)
df.head()
```

Out[10]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review
0	A3HVRXV0LVJN7	5.0	1358035200	01 13, 2013	1.0	4	4	Best phone case ever . Everywhere I go I get a...
1	A1BJGDS0L1IO6I	1.0	1359504000	01 30, 2013	0.0	0	3	ITEM NOT SENT from Blue Top Company in Hong Ko...
2	A1YX2RBMS1L9L	5.0	1353542400	11 22, 2012	1.0	0	0	Saw this same case at a theme park store for 2...
3	A180NNPPKWCCU0	5.0	1374105600	07 18, 2013	1.0	3	3	case fits perfectly and I always gets complime...
4	A30P2CYOUYAJM8	4.0	1363737600	03 20, 2013	1.0	1	1	I got this for my 14 year old sister. She lov...

In [11]:

```
df1 = df.sort_values(['reviewerID','unixReviewTime'], ascending=[False,False])
df1.head()
```

Out[11]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review
553800	AZZZVH7FYD0UR	5.0	1364515200	03 29, 2013	1.0	0	0	I was skeptical about the price, after receivi...
872343	AZZZRS1YZ8HVP	4.0	1316044800	09 15, 2011	1.0	0	0	Pro: Price (local retailer wanted $15.00 for...
774291	AZZZOVIBXHGDR	4.0	1331856000	03 16, 2012	1.0	0	0	This is a good case for the money. It looks go...
637386	AZZZMSZI9LKE6	5.0	1361232000	02 19, 2013	1.0	0	0	I bought this for my daughter. She loves it! S...
286010	AZZZKX0IEBKE0	1.0	1361923200	02 27, 2013	0.0	0	0	The design is so bad and make the audio set us...

In [12]:

```
df1["year"] = df1['reviewTime'].str.split("," , n =1 , expand = True)[1]
df1['month'] = df1['reviewTime'].str.split("," , n =1 , expand = True)[0].str.split(" "
, n =1 , expand = True)[0]
df1['day'] = df1['reviewTime'].str.split("," , n =1 , expand = True)[0].str.split(" " ,
n =1 , expand = True)[1]
```

In [13]:

```
df1.reset_index(drop = True, inplace = True)
```

In [14]:

```
df1.head()
```

Out[14]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day
0	AZZZVH7FYD0UR	5.0	1364515200	03 29, 2013	1.0	0	0	I was skeptical about the price, after receivi...	2013	03	29

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day
1	AZZZRS1YZ8HVP	4.0	1316044800	09 15, 2011	1.0	0	0	Pro: Price (local retailer wanted $15.00 for...	2011	09	15
2	AZZZOVIBXHGDR	4.0	1331856000	03 16, 2012	1.0	0	0	This is a good case for the money. It looks go...	2012	03	16
3	AZZZMSZI9LKE6	5.0	1361232000	02 19, 2013	1.0	0	0	I bought this for my daughter. She loves it! S...	2013	02	19
4	AZZZKX0IEBKE0	1.0	1361923200	02 27, 2013	0.0	0	0	The design is so bad and make the audio set us...	2013	02	27

In [15]:

```
df1['year'] = df1['year'].astype(float)
df1['month'] = df1['month'].astype(float)
df1['day'] = df1['day'].astype(float)
```

In [16]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 993759 entries, 0 to 993758
Data columns (total 11 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   reviewerID      993759 non-null  object 
 1   overall         993759 non-null  float64
 2   unixReviewTime  993759 non-null  int64  
 3   reviewTime      993759 non-null  object 
 4   class           993759 non-null  float64
 5   helpfull        993759 non-null  int64  
 6   not_helpfull    993759 non-null  int64  
 7   review          993759 non-null  object 
 8   year            993759 non-null  float64
 9   month           993759 non-null  float64
 10  day             993759 non-null  float64
dtypes: float64(5), int64(3), object(3)
memory usage: 83.4+ MB
```

In [17]:

```
df1.corr()
```

Out[17]:

	overall	unixReviewTime	class	helpfull	not_helpfull	year	month	day
overall	1.000000	0.027549	0.910217	-0.009687	-0.024900	0.028471	-0.008873	0.001129
unixReviewTime	0.027549	1.000000	0.016142	-0.105645	-0.122660	0.985275	-0.032849	-0.007920
class	0.910217	0.016142	1.000000	-0.006871	-0.020781	0.017061	-0.007455	0.000871
helpfull	-0.009687	-0.105645	-0.006871	1.000000	0.990947	-0.105265	0.010467	0.000516
not_helpfull	-0.024900	-0.122660	-0.020781	0.990947	1.000000	-0.122155	0.011768	0.000706
year	0.028471	0.985275	0.017061	-0.105265	-0.122155	1.000000	-0.202709	-0.028009
month	-0.008873	-0.032849	-0.007455	0.010467	0.011768	-0.202709	1.000000	0.039253
day	0.001129	-0.007920	0.000871	0.000516	0.000706	-0.028009	0.039253	1.000000

In [18]:

```
df3 = df1
```

In [19]:

```
df3.head()
```

Out[19]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day
0	AZZZVH7FYD0UR	5.0	1364515200	03 29, 2013	1.0	0	0	I was skeptical about the price, after receivi...	2013.0	3.0	29.0
1	AZZZRS1YZ8HVP	4.0	1316044800	09 15, 2011	1.0	0	0	Pro: Price (local retailer wanted $15.00 for...	2011.0	9.0	15.0
2	AZZZOVIBXHGDR	4.0	1331856000	03 16, 2012	1.0	0	0	This is a good case for the money. It looks go...	2012.0	3.0	16.0
3	AZZZMSZI9LKE6	5.0	1361232000	02 19, 2013	1.0	0	0	I bought this for my daughter. She loves it! S...	2013.0	2.0	19.0
4	AZZZKX0IEBKE0	1.0	1361923200	02 27, 2013	0.0	0	0	The design is so bad and make the audio set us...	2013.0	2.0	27.0

In [20]:

```
df1 = df3
```

In [21]:

```
df1.head()
```

Out[21]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day
0	AZZZVH7FYD0UR	5.0	1364515200	03 29, 2013	1.0	0	0	I was skeptical about the price, after receivi...	2013.0	3.0	29.0
1	AZZZRS1YZ8HVP	4.0	1316044800	09 15, 2011	1.0	0	0	Pro: Price (local retailer wanted $15.00 for...	2011.0	9.0	15.0
2	AZZZOVIBXHGDR	4.0	1331856000	03 16, 2012	1.0	0	0	This is a good case for the money. It looks go...	2012.0	3.0	16.0
3	AZZZMSZI9LKE6	5.0	1361232000	02 19, 2013	1.0	0	0	I bought this for my daughter. She loves it! S...	2013.0	2.0	19.0
4	AZZZKX0IEBKE0	1.0	1361923200	02 27, 2013	0.0	0	0	The design is so bad and make the audio set us...	2013.0	2.0	27.0

In [22]:

```
try:
    df1['helpfullness'] = df1['helpfull']/df1['not_helpfull']
except ZeroDivisionError:
    df1['helpfullness'] = -1
```

In [23]:

```
df4 = df1[df1['class']==1]
df5 = df1[df1['class']==0]
```

In [24]:

```
df4.head()
```

Out[24]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day	helpfu
0	**AZZZVH7FYD0UR**	**5.0**	**1364515200**	**03 29, 2013**	**1.0**	**0**	**0**	**I was skeptical about the price, after receivi...**	**2013.0**	**3.0**	**29.0**	
1	**AZZZRS1YZ8HVP**	**4.0**	**1316044800**	**09 15, 2011**	**1.0**	**0**	**0**	**Pro: Price (local retailer wanted $15.00 for...**	**2011.0**	**9.0**	**15.0**	
2	**AZZZOVIBXHGDR**	**4.0**	**1331856000**	**03 16, 2012**	**1.0**	**0**	**0**	**This is a good case for the money. It looks go...**	**2012.0**	**3.0**	**16.0**	
3	**AZZZMSZI9LKE6**	**5.0**	**1361232000**	**02 19, 2013**	**1.0**	**0**	**0**	**I bought this for my daughter. She loves it! S...**	**2013.0**	**2.0**	**19.0**	
5	**AZZZ159U3Q5OO**	**5.0**	**1161993600**	**10 28, 2006**	**1.0**	**1**	**2**	**The first three seasons of Scrubs were the fun...**	**2006.0**	**10.0**	**28.0**	

In [25]:

```
mean_value1 = df4['helpfullness'].mean()
mean_value2 = df5['helpfullness'].mean()
```

In [26]:

```
print(mean_value1)
print(mean_value2)
```

```
0.815548939980668
0.6567437745740992
```

In [27]:

```
from numpy import nan
```

In [28]:

```
arr = np.zeros(len(df1))
k1=0
```

```
i=0
while i < len(df1):
   a = df1.iloc[i]
   if a['helpfullness'] ==nan:
       if a['class'] == 1:
         arr[i] = mean_value1
       else:
         arr[i] = mean_value2
   else:
       arr[i] = a['helpfullness']
   i = i+1
```

In [29]:

```
df1['helpfullness'] = arr
```

In [30]:

```
df1.head()
```

Out[30]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day	helpfu
0	**AZZZVH7FYD0UR**	**5.0**	**1364515200**	**03 29, 2013**	**1.0**	**0**	**0**	**I was skeptical about the price, after receivi...**	**2013.0**	**3.0**	**29.0**	
1	**AZZZRS1YZ8HVP**	**4.0**	**1316044800**	**09 15, 2011**	**1.0**	**0**	**0**	**Pro: Price (local retailer wanted $15.00 for...**	**2011.0**	**9.0**	**15.0**	
2	**AZZZOVIBXHGDR**	**4.0**	**1331856000**	**03 16, 2012**	**1.0**	**0**	**0**	**This is a good case for the money. It looks go...**	**2012.0**	**3.0**	**16.0**	
3	**AZZZMSZI9LKE6**	**5.0**	**1361232000**	**02 19, 2013**	**1.0**	**0**	**0**	**I bought this for my daughter. She loves it! S...**	**2013.0**	**2.0**	**19.0**	
4	**AZZZKX0IEBKE0**	**1.0**	**1361923200**	**02 27, 2013**	**0.0**	**0**	**0**	**The design is so bad and make the audio set us...**	**2013.0**	**2.0**	**27.0**	

In [31]:

```
df1.corr()
```

Out[31]:

	overall	unixReviewTime	class	helpfull	not_helpfull	year	month	day	helpfullness
overall	1.000000	0.027549	0.910217	-0.009687	-0.024900	0.028471	-0.008873	0.001129	0.211723
unixReviewTime	0.027549	1.000000	0.016142	-0.105645	-0.122660	0.985275	-0.032849	-0.007920	-0.063510

	overall	unixReviewTime	class	helpfull	not_helpfull	year	month	day	helpfullness
class	0.910217	0.016142	1.000000	-0.006871	-0.020781	0.017061	-0.007453	0.000871	0.210453
helpfull	-0.009687	-0.105645	-0.006871	1.000000	0.990947	-0.105265	0.010467	0.000516	0.082990
not_helpfull	-0.024900	-0.122660	-0.020781	0.990947	1.000000	-0.122155	0.011768	0.000706	0.034675
year	0.028471	0.985275	0.017061	-0.105265	-0.122155	1.000000	-0.202709	-0.028009	-0.062990
month	-0.008873	-0.032849	-0.007455	0.010467	0.011768	-0.202709	1.000000	0.039253	0.001024
day	0.001129	-0.007920	0.000871	0.000516	0.000706	-0.028009	0.039253	1.000000	0.000866
helpfullness	0.211723	-0.063510	0.210453	0.082990	0.034675	-0.062990	0.001024	0.000866	1.000000

In [32]:

```
df1.head()
```

Out[32]:

	reviewerID	overall	unixReviewTime	reviewTime	class	helpfull	not_helpfull	review	year	month	day	helpfu
0	AZZZVH7FYD0UR	5.0	1364515200	03 29, 2013	1.0	0	0	I was skeptical about the price, after receivi...	2013.0	3.0	29.0	
1	AZZZRS1YZ8HVP	4.0	1316044800	09 15, 2011	1.0	0	0	Pro: Price (local retailer wanted $15.00 for...	2011.0	9.0	15.0	
2	AZZZOVIBXHGDR	4.0	1331856000	03 16, 2012	1.0	0	0	This is a good case for the money. It looks go...	2012.0	3.0	16.0	
3	AZZZMSZI9LKE6	5.0	1361232000	02 19, 2013	1.0	0	0	I bought this for my daughter. She loves it! S...	2013.0	2.0	19.0	
4	AZZZKX0IEBKE0	1.0	1361923200	02 27, 2013	0.0	0	0	The design is so bad and make the audio set us...	2013.0	2.0	27.0	

In [33]:

```
df1 = df1.drop(["helpfull","not_helpfull","reviewTime"],axis =1)
```

In [34]:

```
df1['Uppercase'] = df1['review'].str.findall(r'[A-Z]').str.len()
df1['Lowercase'] = df1['review'].str.findall(r'[a-z]').str.len()
```

In [35]:

```
df1.corr()
```

Out[35]:

	overall	unixReviewTime	class	year	month	day	helpfullness	Uppercase	Lowercase
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	-0.024751	-0.017607
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.121132	-0.210649
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	-0.016985	-0.011882
year	0.028471	0.985275	0.017061	1.000000	-0.202709	-0.028009	-0.062990	-0.119969	-0.208717
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.007570	0.013665
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.002975	0.005408
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.022916	0.105254
Uppercase	-0.024751	-0.121132	-0.016985	-0.119969	0.007570	0.002975	0.022916	1.000000	0.542774
Lowercase	-0.017607	-0.210649	-0.011882	-0.208717	0.013665	0.005408	0.105254	0.542774	1.000000

In [36]:

```
df1["upper_in_tot"] = df1['Uppercase']/(df1['Uppercase']+df1['Lowercase'])
```

In [37]:

```
df1.corr()
```

Out[37]:

	overall	unixReviewTime	class	year	month	day	helpfullness	Uppercase	Lowercase	upp
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	-0.024751	-0.017607	-
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.121132	-0.210649	
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	-0.016985	-0.011882	-
year	0.028471	0.985275	0.017061	1.000000	-0.202709	-0.028009	-0.062990	-0.119969	-0.208717	
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.007570	0.013665	-
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.002975	0.005408	-
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.022916	0.105254	-
Uppercase	-0.024751	-0.121132	-0.016985	-0.119969	0.007570	0.002975	0.022916	1.000000	0.542774	
Lowercase	-0.017607	-0.210649	-0.011882	-0.208717	0.013665	0.005408	0.105254	0.542774	1.000000	-
upper_in_tot	-0.011022	0.003188	-0.006535	0.003319	-0.001068	-0.000670	-0.064891	0.568758	-0.065749	

In [38]:

```
df1 = df1.drop(['Lowercase','Uppercase','upper_in_tot'],axis = 1)
```

In [39]:

```
df1.head()
```

Out[39]:

	reviewerID	overall	unixReviewTime	class			review	year	month	day	helpfullness

	reviewerID	overall	unixReviewTime	class	review	year	month	day	helpfullness
0	AZZZVH7FYD0UR	5.0	1364515200	1.0	I was skeptical about the price, after receivi...	2013.0	3.0	29.0	NaN
1	AZZZRS1YZ8HVP	4.0	1316044800	1.0	Pro: Price (local retailer wanted $15.00 for...	2011.0	9.0	15.0	NaN
2	AZZZOVIBXHGDR	4.0	1331856000	1.0	This is a good case for the money. It looks go...	2012.0	3.0	16.0	NaN
3	AZZZMSZI9LKE6	5.0	1361232000	1.0	I bought this for my daughter. She loves it! S...	2013.0	2.0	19.0	NaN
4	AZZZKX0IEBKE0	1.0	1361923200	0.0	The design is so bad and make the audio set us...	2013.0	2.0	27.0	NaN

In [40]:

```
import nltk
import re
import string
from wordcloud import WordCloud,STOPWORDS
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [41]:

```
def review_cleaning(text):
    '''Make text lowercase, remove text in square brackets,remove links,remove punctuation
    and remove words containing numbers.'''
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

In [42]:

```
df1['review']=df1['review'].apply(lambda x:review_cleaning(x))
```

In [43]:

```
stop_words= ['yourselves', 'between', 'whom', 'itself', 'is', "she's", 'up', 'herself',
'here', 'your', 'each',
 'we', 'he', 'my', "you've", 'having', 'in', 'both', 'for', 'themselves', 'are', 'them',
'other',
 'and', 'an', 'during', 'their', 'can', 'yourself', 'she', 'until', 'so', 'these', 'ours
', 'above',
 'what', 'while', 'have', 're', 'more', 'only', "needn't", 'when', 'just', 'that', 'were
', "don't",
 'very', 'should', 'any', 'y', 'isn', 'who', 'a', 'they', 'to', 'too', "should've", 'has
', 'before',
 'into', 'yours', "it's", 'do', 'against', 'on', 'now', 'her', 've', 'd', 'by', 'am', 'f
rom',
 'about', 'further', "that'll", "you'd", 'you', 'as', 'how', 'been', 'the', 'or', 'doing
', 'such',
 'his', 'himself', 'ourselves', 'was', 'through', 'out', 'below', 'own', 'myself', 'thei
rs',
 'me', 'why', 'once', 'him', 'than', 'be', 'most', "you'll", 'same', 'some', 'with', 'fe
w', 'it',
 'at', 'after', 'its', 'which', 'there','our', 'this', 'hers', 'being', 'did', 'of', 'ha
d', 'under',
 'over','again', 'where', 'those', 'then', "you're", 'i', 'because', 'does', 'all']
```

In [44]:

```
df1['review'] = df1['review'].apply(lambda x: ' '.join([word for word in x.split() if wo
rd not in (stop_words)]))
```

```
df1.head()
```

Out[44]:

	reviewerID	overall	unixReviewTime	class	review	year	month	day	helpfullness
0	AZZZVH7FYD0UR	5.0	1364515200	1.0	skeptical price receiving product pleased happ...	2013.0	3.0	29.0	NaN
1	AZZZRS1YZ8HVP	4.0	1316044800	1.0	pro price local retailer wanted usb alonecon c...	2011.0	9.0	15.0	NaN
2	AZZZOVIBXHGDR	4.0	1331856000	1.0	good case money looks good leaves ports open u...	2012.0	3.0	16.0	NaN
3	AZZZMSZI9LKE6	5.0	1361232000	1.0	bought daughter loves says best bluetooth ever...	2013.0	2.0	19.0	NaN
4	AZZZKX0IEBKE0	1.0	1361923200	0.0	design bad make audio set useless cave man not...	2013.0	2.0	27.0	NaN

In [45]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tf_idf_vectorizer = TfidfVectorizer(max_features=5000,ngram_range=(2,2))
text = tf_idf_vectorizer.fit_transform(df['review'])
tf_idf_vectorizer.vocabulary_
```

Out[45]:

```
{'best phone': 539,
 'phone case': 2972,
 'of compliments': 2649,
 'compliments on': 892,
 'on it': 2737,
 'it it': 1991,
 'it was': 2070,
 'was in': 4573,
 'in perfect': 1741,
 'perfect condition': 2942,
 'as well': 401,
 'and it': 210,
 'it been': 1934,
 'two months': 4394,
 'do not': 1012,
 'not use': 2617,
 'use this': 4445,
 'this company': 4092,
 'not happy': 2586,
 'at all': 405,
 'saw this': 3301,
 'case at': 755,
 'this is': 4105,
 'is very': 1898,
 'very good': 4497,
 'good quality': 1459,
 'quality for': 3202,
 'for great': 1286,
 'great price': 1495,
 'case fits': 765,
 'fits perfectly': 1247,
 'perfectly and': 2947,
 'and always': 142,
 'it its': 1992,
 'when dropped': 4695,
 'dropped it': 1073,
 'got this': 1472,
 'this for': 4097,
 'for my': 1305,
 'year old': 4923,
 'she loves': 3364,
 'loves it': 2282,
 'it really': 2035,
 'really don': 3224,
```

```
'don have': 1047,
'have any': 1568,
'this case': 4089,
'case is': 773,
'is extremely': 1834,
've dropped': 4470,
'dropped my': 1074,
'my phone': 2484,
'in the': 1749,
'the case': 3679,
'case my': 779,
'my favorite': 2440,
'about this': 21,
'is that': 1889,
'that the': 3610,
'the plastic': 3880,
'over the': 2915,
'the front': 3747,
'front of': 1378,
'of my': 2665,
'phone so': 3017,
'so no': 3433,
'no matter': 2545,
'matter how': 2314,
'how many': 1673,
'many times': 2311,
'times it': 4189,
'face down': 1156,
'down the': 1062,
'the screen': 3920,
'screen is': 3318,
'is good': 1841,
'good but': 1440,
'but the': 647,
'the two': 3967,
'two pieces': 4396,
'not fit': 2578,
'all the': 78,
'the way': 3982,
'it is': 1989,
'is slightly': 1880,
'off and': 2694,
'and no': 234,
'how much': 1674,
'the side': 3927,
'it wouldn': 2089,
'very very': 4522,
'very little': 4506,
'and you': 308,
'you can': 4936,
'it the': 2054,
'the bottom': 3666,
'bottom of': 574,
'of the': 2677,
'case has': 769,
'has never': 1554,
'off so': 2706,
'deal with': 963,
'with it': 4803,
'it for': 1971,
'for the': 1329,
'case cover': 760,
'received was': 3244,
'sure the': 3521,
'the quality': 3900,
'quality of': 3205,
'the product': 3893,
'product is': 3139,
'is fine': 1838,
'fine the': 1212,
'the color': 3693,
'color is': 866,
```

```
'ordered this': 2867,
'this as': 4080,
'my sister': 2494,
'it arrived': 1928,
'arrived quickly': 367,
'but was': 657,
'was very': 4611,
'very disappointed': 4492,
'the item': 3784,
'item was': 2098,
'the cover': 3705,
'tried to': 4376,
'phone the': 3021,
'didn even': 991,
'the picture': 3875,
'is one': 1870,
'one of': 2793,
'the worst': 3992,
've ever': 4471,
'no reason': 2552,
'reason for': 3237,
'of this': 2681,
'this item': 4109,
'item is': 2095,
'is completely': 1823,
'like the': 2208,
'case for': 766,
'for its': 1293,
'where it': 4728,
'it should': 2041,
'onto the': 2835,
'comes off': 877,
'off easily': 2697,
'as was': 400,
'was to': 4606,
'to purchase': 4289,
'purchase this': 3182,
'the day': 3711,
'that it': 3595,
'me on': 2334,
'no more': 2546,
'phone and': 2959,
'it however': 1985,
'case was': 792,
'was not': 4585,
'not worth': 2625,
'worth the': 4893,
'the money': 3830,
'the top': 3962,
'did not': 989,
'with the': 4819,
'right off': 3279,
'in my': 1735,
'my hands': 2449,
'my iphone': 2457,
'hit the': 1647,
'the middle': 3825,
'purchasing this': 3189,
'got the': 1469,
'case very': 791,
'quick and': 3210,
'before the': 525,
'it came': 1942,
'perfect and': 2941,
'and is': 209,
'is really': 1876,
'really easy': 3225,
'easy to': 1101,
'to put': 4291,
'also have': 92,
'have dropped': 1572,
'and my': 229,
```

```
'phone has': 2987,
'has not': 1557,
'not one': 2601,
'way better': 4619,
'better than': 545,
'than an': 3547,
'an otterbox': 137,
'bought it': 578,
'my wife': 2500,
'but it': 628,
'was really': 4594,
'really hard': 3228,
'hard to': 1545,
'to take': 4321,
'take off': 3529,
'way too': 4624,
'too tight': 4364,
'the phone': 3872,
'phone when': 3033,
'when put': 4713,
'put it': 3192,
'it on': 2021,
'on for': 2732,
'for her': 1287,
'screen protector': 3322,
'thought was': 4155,
'end up': 1106,
'it when': 2075,
'when trying': 4720,
'trying to': 4382,
'to remove': 4297,
'remove it': 3253,
'liked the': 2215,
'the look': 3810,
'look of': 2244,
'of it': 2659,
'it but': 1940,
'but that': 646,
'about it': 15,
'this and': 4079,
'and received': 256,
'received it': 3240,
'two weeks': 4397,
'weeks the': 4644,
'into the': 1776,
'and the': 274,
'the whole': 3986,
'case does': 762,
'does not': 1026,
'not come': 2572,
'it great': 1978,
'great phone': 1494,
'case and': 753,
'and would': 307,
'would definitely': 4896,
'definitely buy': 966,
'buy it': 677,
'it again': 1916,
'came in': 703,
'ok but': 2716,
'but there': 649,
'there was': 4022,
'on the': 2751,
'the left': 3801,
'left side': 2175,
'also it': 93,
'easily but': 1096,
'it may': 2009,
'pretty good': 3099,
'good the': 1461,
'the first': 3740,
'couple of': 922,
```

```
'of weeks': 2688,
'then it': 4012,
'it started': 2047,
'started to': 3485,
'to slide': 4308,
'off it': 2701,
'one day': 2777,
'down and': 1060,
'quality was': 3209,
'was nice': 4583,
'was expecting': 4561,
'case that': 786,
'fit my': 1235,
'my verizon': 2498,
'verizon iphone': 4482,
'iphone 4s': 1781,
'got it': 1466,
'it in': 1987,
'the mail': 3813,
'and was': 293,
'was so': 4599,
'excited to': 1146,
'to try': 4334,
'try it': 4379,
'on my': 2741,
'phone it': 2994,
'fits the': 1249,
'the button': 3670,
'fit the': 1239,
'the buttons': 3671,
'thing is': 4064,
'is the': 1890,
'near the': 2503,
'my screen': 2492,
'with this': 4823,
'and so': 264,
'it doesn': 1957,
'doesn fit': 1034,
'if you': 1702,
'you have': 4948,
'have the': 1597,
'the verizon': 3975,
'very pretty': 4513,
'pretty and': 3098,
'but its': 629,
'its not': 2103,
'case on': 783,
'is super': 1886,
'durable and': 1078,
'the delivery': 3713,
'delivery was': 970,
'case it': 774,
'is well': 1900,
'well made': 4658,
'since it': 3384,
'it isn': 1990,
'piece of': 3056,
'phone before': 2966,
'put on': 3194,
'if it': 1692,
'the colors': 3694,
'colors are': 870,
'exactly as': 1138,
'pictures and': 3054,
'since the': 3386,
'the white': 3985,
'part is': 2932,
'than the': 3556,
'the pink': 3878,
'cheap plastic': 847,
'is not': 1863,
'waste of': 4616,
```

```
'of money': 2663,
'will not': 4773,
'use the': 4442,
'it took': 2063,
'over month': 2913,
'month and': 2364,
'and half': 196,
'for this': 1333,
'this product': 4124,
'product to': 3143,
'to be': 4211,
'to me': 4261,
'not only': 2602,
'that but': 3570,
'but once': 638,
'plastic case': 3067,
'is quite': 1875,
'some time': 3455,
'time it': 4175,
'to grip': 4241,
'would not': 4907,
'not recommend': 2608,
'received this': 3243,
'very quickly': 4514,
'seems to': 3343,
'top and': 4369,
'and bottom': 153,
'but do': 614,
'in place': 1743,
'using it': 4461,
'out of': 2899,
'my case': 2423,
'tons of': 4352,
'what paid': 4685,
'paid for': 2926,
'is terrible': 1888,
'use it': 4438,
'it just': 1994,
'off the': 2708,
'nice case': 2530,
'and got': 193,
'lot of': 2265,
'cheap but': 845,
'but guess': 621,
'worth it': 4892,
'love this': 2277,
'case ve': 790,
've had': 4474,
'had it': 1517,
'for few': 1282,
'few years': 1198,
'years now': 4926,
'now and': 2631,
'and ve': 288,
've never': 4475,
'never had': 2515,
'had any': 1513,
'any problems': 320,
'problems with': 3124,
'snaps on': 3405,
'just right': 2132,
'exactly what': 1141,
'what wanted': 4689,
'and more': 227,
'came with': 707,
'very impressed': 4503,
'cute and': 942,
'and light': 216,
'light weight': 2191,
'the big': 3658,
'drawback is': 1065,
'is it': 1848,
```

```
'at my': 415,
'my work': 2501,
'work the': 4853,
'is also': 1805,
'am in': 105,
'the back': 3648,
'my house': 2453,
'is in': 1847,
'3g and': 4,
'or so': 2854,
'don know': 1048,
'know what': 2159,
'times and': 4187,
'it does': 1956,
'loved the': 2280,
'and how': 205,
'how the': 1675,
'the design': 3715,
'up on': 4418,
'bought this': 584,
'4s and': 5,
'unfortunately it': 4403,
'stay on': 3491,
'it comes': 1947,
'comes in': 876,
'in two': 1756,
'piece that': 3057,
'that makes': 3598,
'all and': 67,
'so much': 3431,
'much to': 2405,
'to keep': 4251,
'on but': 2729,
'but also': 606,
'due to': 1077,
'that doesn': 3579,
'the price': 3889,
'price it': 3106,
'but really': 642,
'going to': 1434,
'keep it': 2142,
'my new': 2471,
'love it': 2270,
'its very': 2107,
'very cute': 4490,
'on easily': 2731,
'easily and': 1095,
'and comes': 164,
'some of': 3450,
'my other': 2480,
'other cases': 2872,
'like how': 2197,
'so my': 3432,
'my charger': 2426,
'fit perfectly': 1237,
'perfectly the': 2951,
'the only': 3849,
'side is': 3374,
'other than': 2884,
'from scratches': 1370,
'so if': 3425,
'you drop': 4943,
'drop your': 1071,
'your phone': 4996,
'phone lot': 3000,
'wouldn recommend': 4918,
'recommend this': 3251,
'case but': 758,
'but if': 625,
'you don': 4941,
'think it': 4069,
'great value': 1505,
```

```
'can not': 729,
'not so': 2610,
'great for': 1483,
'protecting the': 3154,
'phone but': 2967,
'but for': 619,
'look and': 2240,
'and feel': 181,
'feel of': 1180,
'was sent': 4596,
'sent the': 3351,
'the wrong': 3993,
'cover for': 927,
'for another': 1266,
'type of': 4399,
'this cover': 4094,
'cover is': 928,
'very well': 4523,
'made and': 2285,
'and also': 141,
'to use': 4338,
'is durable': 1828,
'and protects': 251,
'protects my': 3173,
'very easy': 4495,
'remove the': 3254,
'from the': 1371,
'like this': 2210,
'had the': 1525,
'the regular': 3907,
'it would': 2088,
'and didn': 170,
'didn like': 997,
'like it': 2199,
'it as': 1929,
'as much': 383,
'much the': 2404,
'pretty much': 3100,
'hands free': 1534,
'found the': 1354,
'it fits': 1970,
'fits my': 1244,
'use and': 4433,
'highly recommend': 1644,
'case this': 788,
'case will': 795,
'on its': 2738,
'its own': 2104,
'while you': 4748,
'very sturdy': 4519,
'sturdy and': 3510,
'and of': 238,
'of good': 2654,
'received the': 3242,
'the one': 3847,
'one that': 2800,
'instead of': 1770,
'this one': 4117,
'for replacement': 1320,
'and they': 278,
'that they': 3613,
'they didn': 4042,
'didn have': 994,
'have that': 1596,
'and that': 273,
'something that': 3460,
'keep the': 2144,
'they sent': 4055,
'sent me': 3350,
'cover and': 924,
'protection but': 3156,
'good case': 1442,
```

```
'for any': 1267,
'case in': 772,
'the photo': 3874,
'looks great': 2254,
'great and': 1475,
'and very': 289,
'however when': 1684,
'when it': 4703,
'scratched up': 3311,
'flimsy and': 1254,
'and not': 235,
'needless to': 2512,
'to say': 4302,
'returned the': 3268,
'purchased the': 3186,
'because it': 507,
'it has': 1981,
'is much': 1858,
'and easy': 177,
'we have': 4630,
'have had': 1579,
'had some': 1524,
'to the': 4327,
'the internet': 3779,
'at times': 423,
'and she': 261,
'is nice': 1861,
'nice and': 2528,
'was easy': 4557,
'was quick': 4593,
'order from': 2861,
'from this': 1373,
'this seller': 4130,
'seller again': 3344,
'well the': 4662,
'product was': 3144,
'was good': 4568,
'the rest': 3909,
'rest of': 3264,
'the body': 3665,
'coming off': 884,
'off but': 2695,
'but they': 651,
'quite bit': 3212,
'bit of': 554,
'found this': 1355,
'case to': 789,
'to fit': 4234,
'my blackberry': 2418,
'blackberry and': 556,
'is just': 1849,
'just like': 2125,
'like new': 2203,
'again the': 61,
'only thing': 2829,
'thing that': 4066,
'that would': 3628,
'would suggest': 4915,
'on how': 2736,
'how to': 1677,
'to install': 4247,
'install the': 1767,
'the new': 3838,
'the package': 3861,
'we had': 4629,
'had to': 1527,
'to and': 4202,
'and then': 275,
'all have': 71,
'have to': 1600,
'say is': 3303,
'is my': 1860,
```

```
'very happy': 4499,
'happy with': 1539,
'with its': 4804,
'the vendor': 3973,
'thank you': 3560,
'and now': 237,
'about the': 20,
'the world': 3991,
'with my': 4809,
'and will': 301,
'had my': 1518,
'my pocket': 2486,
'pocket and': 3085,
'the floor': 3744,
'it and': 1925,
'and broke': 155,
'broke the': 594,
'the outside': 3859,
'and found': 188,
'would be': 4894,
'amazon and': 115,
'color of': 868,
'not have': 2587,
'on their': 2752,
'my original': 2479,
'scratches on': 3313,
'the red': 3906,
'much more': 2401,
'it definitely': 1951,
'my expectations': 2436,
'am very': 114,
'very satisfied': 4516,
'satisfied with': 3297,
'able to': 7,
'to charge': 4218,
'charge my': 815,
'perfect for': 2944,
'this piece': 4120,
'is great': 1842,
'all my': 74,
'my friends': 2445,
'loved it': 2279,
'it works': 2086,
'works fine': 4873,
'one would': 2811,
'would recommend': 4912,
'need to': 2507,
'to have': 4243,
'this little': 4110,
'does the': 1027,
'the job': 3788,
'for me': 1299,
'couldn be': 918,
'without it': 4836,
'works great': 4876,
'the road': 3915,
'this to': 4137,
'to play': 4281,
'from my': 1367,
'to set': 4305,
'set up': 3358,
'up and': 4412,
'and works': 306,
'great it': 1488,
'it also': 1921,
'allows me': 83,
'me to': 2340,
'phone at': 2962,
'at the': 420,
'the same': 3918,
'same time': 3294,
'although it': 98,
```

```
'charger for': 825,
'for phone': 1317,
'micro usb': 2347,
'not the': 2614,
'this phone': 4119,
'just got': 2122,
'new iphone': 2521,
'iphone was': 1795,
'was super': 4601,
'he loves': 1615,
'and to': 282,
'thought this': 4154,
'was perfect': 4590,
'ago and': 63,
'case itself': 775,
'enough to': 1111,
'to last': 4253,
'so the': 3438,
'seem to': 3340,
'but then': 648,
'when opened': 4709,
'opened the': 2840,
'even in': 1119,
'or in': 2846,
'or anything': 2842,
'was just': 4575,
'on amazon': 2723,
'before and': 522,
'it to': 2061,
'at least': 414,
'great the': 1503,
'way the': 4622,
'they just': 4048,
'threw it': 4160,
'sent it': 3349,
'it off': 2020,
'the the': 3954,
'price of': 3107,
'the 34': 3631,
'they did': 4041,
'for use': 1337,
'use in': 4437,
'works and': 4869,
'charger that': 830,
'that is': 3594,
'one thing': 2802,
'needed to': 2511,
'haven had': 1607,
'it very': 2069,
'very long': 4507,
'long and': 2232,
'and this': 280,
'while on': 4743,
'to work': 4344,
'so could': 3413,
'my nexus': 2472,
'it only': 2024,
'works on': 4880,
'so am': 3408,
'am not': 106,
'giving it': 1421,
'right now': 3278,
'and am': 143,
'am still': 111,
'if could': 1688,
'could get': 913,
'get it': 1393,
'work on': 4849,
'my tablet': 2496,
'am happy': 104,
'good for': 1447,
'this charger': 4091,
```

```
'as backup': 370,
'for when': 1342,
'when your': 4727,
'just have': 2123,
'to make': 4260,
'make sure': 2297,
'the battery': 3654,
'battery is': 462,
'and battery': 150,
'will charge': 4759,
'phone from': 2984,
'with out': 4814,
'that was': 3618,
'the charger': 3685,
'usb charger': 4428,
'might have': 2350,
'the title': 3960,
'on this': 2756,
'buy one': 680,
'it keeps': 1995,
'connected to': 896,
'to my': 4266,
'my computer': 2427,
'computer and': 893,
'charging the': 843,
'the light': 3806,
'not working': 2624,
'other one': 2876,
'not work': 2623,
'work with': 4856,
'not compatible': 2573,
'charge the': 816,
'try to': 4381,
'to connect': 4222,
'connect to': 895,
'charge your': 817,
'tell me': 3541,
'me if': 2329,
'can do': 717,
'do it': 1010,
'it have': 1983,
'lost my': 2261,
'my money': 2466,
'in this': 1752,
'up in': 4415,
'it did': 1952,
'the wall': 3980,
'not charge': 2571,
'my kindle': 2460,
'but since': 643,
'not even': 2576,
'even have': 1117,
'usb port': 4432,
'and does': 172,
'the usb': 3970,
'the device': 3716,
'device it': 982,
'it will': 2078,
'will never': 4772,
'the seller': 3923,
'replaced it': 3260,
'it with': 2079,
'with another': 4793,
'another one': 312,
'one and': 2770,
'was the': 4605,
'they don': 4044,
'don work': 1058,
'this thing': 4135,
'worked for': 4859,
'the second': 3921,
've purchased': 4478,
```

```
'purchased for': 3183,
'phone on': 3008,
'just about': 2112,
'with amazon': 4790,
'would like': 4904,
'like to': 2211,
'to add': 4196,
'care of': 748,
'and can': 158,
'say the': 3306,
'doesn make': 1038,
'make this': 2299,
'covers the': 936,
'can get': 721,
'get in': 1392,
'in there': 1751,
'than that': 3555,
'it pretty': 2030,
'and really': 255,
'do like': 1011,
'that didn': 3577,
'like was': 2212,
'hard case': 1540,
'when first': 4696,
'first got': 1218,
'it after': 1915,
'after about': 38,
'day it': 948,
'fell off': 1189,
'who is': 4751,
'just what': 2139,
'thought it': 4151,
'will be': 4756,
'that have': 3589,
'have more': 1582,
'can charge': 715,
'this device': 4095,
'device is': 981,
'that can': 3572,
'plug in': 3075,
'device and': 979,
'have seen': 1594,
'just make': 2127,
'sure that': 3520,
'compatible with': 888,
'with your': 4828,
'but would': 663,
'think the': 4072,
'especially for': 1112,
'the mobile': 3828,
'will probably': 4776,
'access the': 30,
'the full': 3748,
'but just': 630,
'being able': 530,
'to plug': 4282,
'plug into': 3076,
'absolutely love': 28,
'case just': 776,
'makes me': 2301,
'love my': 2272,
'even more': 1120,
'find it': 1202,
'my purse': 2489,
'have gotten': 1578,
'very durable': 4493,
'be careful': 483,
'sort of': 3466,
'up the': 4420,
'it still': 2049,
'also like': 94,
'the fact': 3734,
```

```
'fact that': 1159,
'this was': 4141,
'just wish': 2140,
'wish it': 4786,
'was more': 4579,
'more durable': 2377,
'it broke': 1939,
'that not': 3602,
'it looks': 2004,
'well in': 4656,
'the volume': 3979,
'volume buttons': 4527,
'making it': 2304,
'it hard': 1980,
'to turn': 4335,
'out in': 2896,
'gave it': 1384,
'it stars': 2046,
'stars because': 3479,
'it provides': 2034,
'get this': 1403,
'this if': 4103,
'you like': 4952,
'and if': 206,
'you are': 4934,
'go through': 1427,
'because have': 506,
'my car': 2422,
'car and': 742,
'have not': 1586,
'many people': 2310,
'me about': 2321,
'it like': 1999,
'and as': 147,
'as stated': 395,
'in great': 1727,
'and had': 195,
'had no': 1519,
'no problems': 2551,
'the shipping': 3926,
'these are': 4025,
'stick to': 3494,
'way to': 4623,
'to get': 4238,
'case looks': 778,
'solid and': 3449,
'and fits': 186,
'fits perfect': 1246,
'not know': 2594,
'what else': 4677,
'say about': 3302,
'case not': 780,
'was pleasantly': 4591,
'pleasantly surprised': 3072,
'really like': 3230,
'the packaging': 3862,
'case the': 787,
'screen protectors': 3323,
'well as': 4647,
'as the': 396,
'pleased with': 3073,
'this purchase': 4126,
'good to': 1463,
'to see': 4303,
'see that': 3337,
'that bought': 3569,
'bought the': 582,
'but not': 635,
'think they': 4073,
'they can': 4039,
'it around': 1927,
'around the': 362,
```

```
 'gives me': 1419,
 'the home': 3771,
 'home button': 1663,
 'purchase again': 3177,
 'again and': 57,
 'and recommend': 257,
 'item for': 2094,
 'for anyone': 1268,
 'anyone who': 323,
 'the next': 3839,
 'my husband': 2455,
 'work and': 4839,
 'and there': 276,
 'there are': 4017,
 'all over': 76,
 'over it': 2912,
 'it from': 1972,
 'what it': 4680,
 'it looked': 2003,
 'looked like': 2246,
 ...}
```

In [46]:

```
def generate_ngrams(text, n_gram=1):
 token = [token for token in text.lower().split(" ") if token != "" if token not in STOP
WORDS]
 ngrams = zip(*[token[i:] for i in range(n_gram)])
 return [" ".join(ngram) for ngram in ngrams]
```

In [47]:

```
from collections import defaultdict
```

In [48]:

```
Pos_freq_dict = defaultdict(int)
Neg_freq_dict = defaultdict(int)
for i in range(len(df1)):
    if df1.iloc[i]['class'] == 1:
        for word in generate_ngrams(df1.iloc[i]['review'],2):
            Neg_freq_dict[word]+=1
    else:
        for word in generate_ngrams(df1.iloc[i]['review'],2):
            Pos_freq_dict[word]+=1
```

In [49]:

```
arr = np.zeros(len(df1))
for i in range(len(df1)):
        for word in generate_ngrams(df1.iloc[i]['review'],2):
           arr[i] = arr[i] + Neg_freq_dict[word] - Pos_freq_dict[word]
```

In [50]:

```
df1['review_int'] = arr
```

In [51]:

```
df1['review_int'] = df1['review_int'] - df1['review_int'].min()
```

In [52]:

```
df1['review_int'] = df1['review_int']/df1['review_int'].max()
```

In [53]:

```
df1.head()
```

Out[53]:

	reviewerID	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int
0	AZZZVH7FYD0UR	5.0	1364515200	1.0	skeptical price receiving product pleased happ...	2013.0	3.0	29.0	NaN	0.166501
1	AZZZRS1YZ8HVP	4.0	1316044800	1.0	pro price local retailer wanted usb alonecon c...	2011.0	9.0	15.0	NaN	0.160338
2	AZZZOVIBXHGDR	4.0	1331856000	1.0	good case money looks good leaves ports open u...	2012.0	3.0	16.0	NaN	0.170099
3	AZZZMSZI9LKE6	5.0	1361232000	1.0	bought daughter loves says best bluetooth ever...	2013.0	2.0	19.0	NaN	0.161556
4	AZZZKX0IEBKE0	1.0	1361923200	0.0	design bad make audio set useless cave man not...	2013.0	2.0	27.0	NaN	0.169471

In [54]:

```
df1.corr()
```

Out[54]:

	overall	unixReviewTime	class	year	month	day	helpfullness	review_int
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	0.271779
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.137677
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	0.261538
year	0.028471	0.985275	0.017061	1.000000	-0.202709	-0.028009	-0.062990	-0.136929
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.011924
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.004058
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.155497
review_int	0.271779	-0.137677	0.261538	-0.136929	0.011924	0.004058	0.155497	1.000000

In [55]:

```
arr = np.zeros(len(df1))
k1=0
i=1
while i < len(df1):
   a = df1.iloc[i]
   b = df1.iloc[i-1]
   if a['reviewerID'] != b['reviewerID']:
      k1 = k1+1
   arr[i]=k1
   i = i+1
```

In [56]:

```
df1['id'] = arr
```

In [57]:

```
df1.head()
```

Out[57]:

	reviewerID	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int	id
0	AZZZVH7FYD0UR	5.0	1364515200	1.0	skeptical price receiving product pleased happ...	2013.0	3.0	29.0	NaN	0.166501	0.0
1	AZZZRS1YZ8HVP	4.0	1316044800	1.0	pro price local retailer wanted usb alonecon c	2011.0	9.0	15.0	NaN	0.160338	1.0

	reviewerID	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int	id
2	AZZZOVIBXHGDR	4.0	1331856000	1.0	good case money looks good leaves ports open u...	2012.0	3.0	16.0	NaN	0.170099	2.0
3	AZZZMSZI9LKE6	5.0	1361232000	1.0	bought daughter loves says best bluetooth ever...	2013.0	2.0	19.0	NaN	0.161556	3.0
4	AZZZKX0IEBKE0	1.0	1361923200	0.0	design bad make audio set useless cave man not...	2013.0	2.0	27.0	NaN	0.169471	4.0

In [58]:

```
df1 = df1.drop(["reviewerID"],axis = 1)
```

In [59]:

```
df1.corr()
```

Out[59]:

	overall	unixReviewTime	class	year	month	day	helpfullness	review_int	id
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	0.271779	-0.000131
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.137677	0.002078
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	0.261538	-0.000193
year	0.028471	0.985275	0.017061	1.000000	-0.202709	-0.028009	-0.062990	-0.136929	0.002092
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.011924	-0.000432
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.004058	0.001002
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.155497	-0.000180
review_int	0.271779	-0.137677	0.261538	-0.136929	0.011924	0.004058	0.155497	1.000000	-0.002398
id	-0.000131	0.002078	-0.000193	0.002092	-0.000432	0.001002	-0.000180	-0.002398	1.000000

In [60]:

```
df1.head(20)
```

Out[60]:

	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int	id
0	5.0	1364515200	1.0	skeptical price receiving product pleased happ...	2013.0	3.0	29.0	NaN	0.166501	0.0
1	4.0	1316044800	1.0	pro price local retailer wanted usb alonecon c...	2011.0	9.0	15.0	NaN	0.160338	1.0
2	4.0	1331856000	1.0	good case money looks good leaves ports open u...	2012.0	3.0	16.0	NaN	0.170099	2.0
3	5.0	1361232000	1.0	bought daughter loves says best bluetooth ever...	2013.0	2.0	19.0	NaN	0.161556	3.0
4	1.0	1361923200	0.0	design bad make audio set useless cave man not...	2013.0	2.0	27.0	NaN	0.169471	4.0
5	5.0	1161993600	1.0	first three seasons scrubs funniest ones one f...	2006.0	10.0	28.0	0.500000	0.168490	5.0
6	4.0	1358035200	1.0	works great car charger adapter charged primar...	2013.0	1.0	13.0	NaN	0.269217	6.0
7	5.0	1357776000	1.0	stand great tilted good but fixed angle perfec...	2013.0	1.0	10.0	1.000000	0.169540	6.0
8	5.0	1218240000	1.0	great product fits iphone perfectly keeps secu...	2008.0	8.0	9.0	1.000000	0.198710	6.0
9	5.0	1355270400	1.0	product exactly described several months looks	2012.0	12.0	12.0	NaN	0.175104	7.0

	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int	id
10	3.0	1316390400	0.0	really great pricebut little prongs hold place...	2011.0	9.0	19.0	1.000000	0.171529	7.0
11	1.0	1259020800	0.0	blame maker itemnot company sold itthe item ar...	2009.0	11.0	24.0	NaN	0.164699	7.0
12	1.0	1259020800	0.0	item really bad cover flimsy apologize compari...	2009.0	11.0	24.0	0.000000	0.156553	7.0
13	5.0	1259020800	1.0	screen cover much thicker protects screen bett...	2009.0	11.0	24.0	NaN	0.177190	7.0
14	5.0	1372636800	1.0	worked well fit perfectly great selection prod...	2013.0	7.0	1.0	NaN	0.168285	8.0
15	4.0	1355616000	1.0	battery much better one came phone follow dire...	2012.0	12.0	16.0	NaN	0.184202	9.0
16	5.0	1373500800	1.0	since galaxy note ii received android update v...	2013.0	7.0	11.0	NaN	0.236247	10.0
17	4.0	1331251200	1.0	shipped case better expected doesnt hard plast...	2012.0	3.0	9.0	0.000000	0.240817	10.0
18	4.0	1373241600	1.0	bought gift dad loves main reason wanted one a...	2013.0	7.0	8.0	1.000000	0.159478	11.0
19	5.0	1151452800	1.0	using nokia phones years always great receptio...	2006.0	6.0	28.0	0.944444	0.193879	12.0

In [61]:

```
arr = np.zeros(len(df1))
```

In [62]:

```
i=1
while i < len(df1):
   a = df1.iloc[i]
   b = df1.iloc[i-1]
   if a['unixReviewTime'] == b['unixReviewTime']:
      arr[i]=5
      arr[i-1]=5
   i = i+1
```

In [63]:

```
df1['review_burst'] = arr
```

In [64]:

```
df1.head()
```

Out[64]:

	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int	id	review_burst
0	5.0	1364515200	1.0	skeptical price receiving product pleased happ...	2013.0	3.0	29.0	NaN	0.166501	0.0	0.0
1	4.0	1316044800	1.0	pro price local retailer wanted usb alonecon c...	2011.0	9.0	15.0	NaN	0.160338	1.0	0.0
2	4.0	1331856000	1.0	good case money looks good leaves ports open u...	2012.0	3.0	16.0	NaN	0.170099	2.0	0.0
3	5.0	1361232000	1.0	bought daughter loves says best bluetooth ever...	2013.0	2.0	19.0	NaN	0.161556	3.0	0.0
4	1.0	1361923200	0.0	design bad make audio set useless cave man not...	2013.0	2.0	27.0	NaN	0.169471	4.0	0.0

In [65]:

In [65]:

```
df1.corr()
```

Out[65]:

	overall	unixReviewTime	class	year	month	day	helpfullness	review_int	id	review
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	0.271779	-0.000131	0.0
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.137677	0.002078	0.0
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	0.261538	-0.000193	0.0
year	0.028471	0.985275	0.017061	1.000000	-0.202709	-0.028009	-0.062990	-0.136929	0.002092	0.0
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.011924	-0.000432	-0.0
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.004058	0.001002	0.0
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.155497	-0.000180	0.0
review_int	0.271779	-0.137677	0.261538	-0.136929	0.011924	0.004058	0.155497	1.000000	-0.002398	-0.0
id	-0.000131	0.002078	-0.000193	0.002092	-0.000432	0.001002	-0.000180	-0.002398	1.000000	0.0
review_burst	0.064998	0.049326	0.049139	0.048900	-0.003585	0.000753	0.007883	-0.020942	0.001758	1.0

In [66]:

```
arr = np.zeros(len(df1))
```

In [67]:

```
i=1
first_time = df1.iloc[0]['unixReviewTime']
while i < len(df1):
   a = df1.iloc[i]
   b = df1.iloc[i-1]
   if a['unixReviewTime'] == b['unixReviewTime']:
      arr[i] = first_time - a['unixReviewTime']
   else:
      first_time=last_time=a['unixReviewTime']
   i = i+1
```

In [68]:

```
df1['Activity1'] = arr
```

In [69]:

```
df1.corr()
```

Out[69]:

	overall	unixReviewTime	class	year	month	day	helpfullness	review_int	id	review
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	0.271779	-0.000131	0.0
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.137677	0.002078	0.0
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	0.261538	-0.000193	0.0

	overall	unixReviewTime	class	year	month	day	helpfullness	review_int	id	review
year	0.028471	0.985275	0.017061	1.000000	0.202709	0.028009	-0.062990	-0.136929	0.002092	0.0
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.011924	-0.000432	-0.0
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.004058	0.001002	0.0
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.155497	-0.000180	0.0
review_int	0.271779	-0.137677	0.261538	-0.136929	0.011924	0.004058	0.155497	1.000000	-0.002398	-0.0
id	-0.000131	0.002078	-0.000193	0.002092	-0.000432	0.001002	-0.000180	-0.002398	1.000000	0.0
review_burst	0.064998	0.049326	0.049139	0.048900	-0.003585	0.000753	0.007883	-0.020942	0.001758	1.0
Activity1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [70]:

```
arr = np.zeros(len(df1))
arr1 = np.zeros(len(df1))
arr2 = np.zeros(len(df1))
```

In [71]:

```
i=0

while i < len(df1):
   count_review_by_author = 0
   postive_review= 0      # rating count of 4 and 5
   negative_review = 0     # rating count of 1 and 2
   j = i
   while j < len(df1):
     if df1.iloc[j]['id'] == df1.iloc[i]['id']:
            count_review_by_author = count_review_by_author + 1
            if df1.iloc[j]['overall'] >= 4:
                postive_review =  postive_review + 1
            elif df1.iloc[j]['overall'] <= 2:
                negative_review = negative_review + 1
     j = j + 1
   k = i
   while k < j:
        arr[k] = count_review_by_author
        arr1[k] = postive_review / count_review_by_author
        arr2[k] = negative_review / count_review_by_author
        k = k + 1
   i = j
```

In [72]:

```
df1['r_count'] = arr
df1['pos_rev'] = arr1
df1['neg_rev'] = arr2
```

In [73]:

```
df1.head()
```

Out[73]:

	overall	unixReviewTime	class	review	year	month	day	helpfullness	review_int	id	review_burst	Activity1	r_cou
0	5.0	1364515200	1.0	skeptical price receiving product	2013.0	3.0	29.0	NaN	0.166501	0.0	0.0	0.0	1.

	overall	unixReviewTime	class	pleased review happ...	year	month	day	helpfullness	review_int	id	review_burst	Activity1	r_cour
1	4.0	1316044800	1.0	pro price local retailer wanted usb alonecon c...	2011.0	9.0	15.0	NaN	0.160338	1.0	0.0	0.0	1.
2	4.0	1331856000	1.0	good case money looks good leaves ports open u...	2012.0	3.0	16.0	NaN	0.170099	2.0	0.0	0.0	1.
3	5.0	1361232000	1.0	bought daughter loves says best bluetooth ever...	2013.0	2.0	19.0	NaN	0.161556	3.0	0.0	0.0	1.
4	1.0	1361923200	0.0	design bad make audio set useless cave man not...	2013.0	2.0	27.0	NaN	0.169471	4.0	0.0	0.0	1.

In [74]:

```
df1.corr()
```

Out[74]:

	overall	unixReviewTime	class	year	month	day	helpfullness	review_int	id	review
overall	1.000000	0.027549	0.910217	0.028471	-0.008873	0.001129	0.211723	0.271779	-0.000131	0.0
unixReviewTime	0.027549	1.000000	0.016142	0.985275	-0.032849	-0.007920	-0.063510	-0.137677	0.002078	0.0
class	0.910217	0.016142	1.000000	0.017061	-0.007455	0.000871	0.210453	0.261538	-0.000193	0.0
year	0.028471	0.985275	0.017061	1.000000	-0.202709	-0.028009	-0.062990	-0.136929	0.002092	0.0
month	-0.008873	-0.032849	-0.007455	-0.202709	1.000000	0.039253	0.001024	0.011924	-0.000432	-0.0
day	0.001129	-0.007920	0.000871	-0.028009	0.039253	1.000000	0.000866	0.004058	0.001002	0.0
helpfullness	0.211723	-0.063510	0.210453	-0.062990	0.001024	0.000866	1.000000	0.155497	-0.000180	0.0
review_int	0.271779	-0.137677	0.261538	-0.136929	0.011924	0.004058	0.155497	1.000000	-0.002398	-0.0
id	-0.000131	0.002078	-0.000193	0.002092	-0.000432	0.001002	-0.000180	-0.002398	1.000000	0.0
review_burst	0.064998	0.049326	0.049139	0.048900	-0.003585	0.000753	0.007883	-0.020942	0.001758	1.0
Activity1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
r_count	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
pos_rev	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

neg_rev	NaN	overall	NaN	unixReviewTime	NaN	class	NaN	year	NaN	month	NaN	day	NaN	helpfullness	NaN	review_int	NaN	id	review

In [75]:

```
df2 = df1.drop(['unixReviewTime','year','month','day','review','id'],axis = 1)
```

In [76]:

```
df2.head()
```

Out[76]:

	overall	class	helpfullness	review_int	review_burst	Activity1	r_count	pos_rev	neg_rev
0	5.0	1.0	NaN	0.166501	0.0	0.0	1.0	1.0	0.0
1	4.0	1.0	NaN	0.160338	0.0	0.0	1.0	1.0	0.0
2	4.0	1.0	NaN	0.170099	0.0	0.0	1.0	1.0	0.0
3	5.0	1.0	NaN	0.161556	0.0	0.0	1.0	1.0	0.0
4	1.0	0.0	NaN	0.169471	0.0	0.0	1.0	1.0	0.0

In [77]:

```
df2 = df2.dropna()
```

In [78]:

```
df2['r_count']=df2['r_count'] /df2['r_count'].max()
```

In [79]:

```
df2.corr()
```

Out[79]:

	overall	class	helpfullness	review_int	review_burst	Activity1	r_count	pos_rev	neg_rev
overall	1.000000	0.917932	0.211723	0.289026	0.071037	NaN	NaN	NaN	NaN
class	0.917932	1.000000	0.210453	0.278584	0.060159	NaN	NaN	NaN	NaN
helpfullness	0.211723	0.210453	1.000000	0.155497	0.007883	NaN	NaN	NaN	NaN
review_int	0.289026	0.278584	0.155497	1.000000	-0.008526	NaN	NaN	NaN	NaN
review_burst	0.071037	0.060159	0.007883	-0.008526	1.000000	NaN	NaN	NaN	NaN
Activity1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
r_count	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
pos_rev	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
neg_rev	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [110]:

```
X = df2.drop(['r_count','pos_rev','neg_rev','helpfullness','review_int','review_burst'],a
xis = 1)
y = df2['class']
```

In [81]:

```
print(len(X))
```

```
289275
```

In [111]:

```
X.corr()
```

Out[111]:

	overall	class	Activity1
overall	1.000000	0.917932	NaN
class	0.917932	1.000000	NaN
Activity1	NaN	NaN	NaN

In [82]:

```
X.head()
```

Out[82]:

	overall	Activity1
5	5.0	0.0
7	5.0	0.0
8	5.0	0.0
10	3.0	0.0
12	1.0	0.0

In [83]:

```
# from imblearn.over_sampling import SMOTE
# smote = SMOTE(random_state=42)
# X, y = smote.fit_resample(X, y)
```

In [84]:

```
# from sklearn.preprocessing import StandardScaler
# st_x= StandardScaler()
# X= st_x.fit_transform(X)
# y= st_x.transform(y)
```

In [85]:

```
from sklearn.model_selection import train_test_split

train_X,test_X,train_y,test_y = train_test_split(X,y,test_size=0.3,random_state=42)
```

In [86]:

```
from sklearn.metrics import mean_squared_error
```

In [87]:

```
import matplotlib.pyplot as plt
```

In [88]:

```
def plot_confusion_matrix(cm, classes,
     normalize=False,
     title='Confusion matrix',
     cmap=plt.cm.Blues):
     """
     This function prints and plots the confusion matrix.
     Normalization can be applied by setting `normalize=True`.
     """

     plt.imshow(cm, interpolation='nearest', cmap=cmap)
     plt.title(title)
     plt.colorbar()
     tick_marks = np.arange(len(classes))
     plt.xticks(tick_marks, classes, rotation=45)
     plt.yticks(tick_marks, classes)
```

```
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')
    thresh = cm.max() / 2.
    for i in range (cm.shape[0]):
       for j in range (cm.shape[1]):
            plt.text(j, i, cm[i, j],
                horizontalalignment="center",
                color="white" if cm[i, j] > thresh else "black")
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

In [89]:

```
from sklearn.linear_model import LogisticRegression
model_2= LogisticRegression(random_state=42)
model_2.fit(train_X,train_y)
pred_y = model_2.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

```
0.0
```

In [90]:

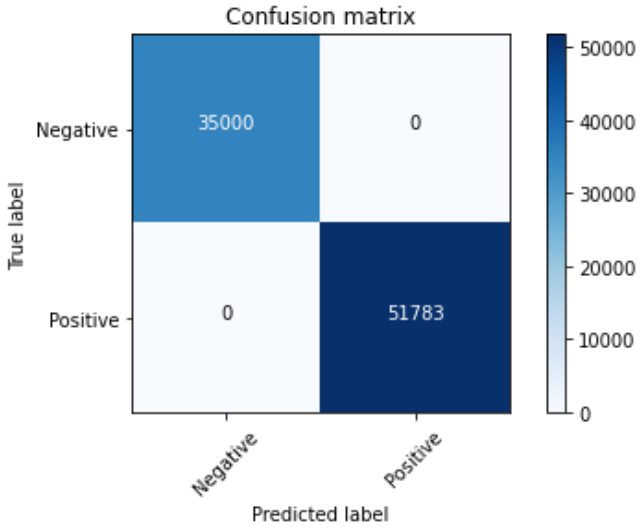
```
print(len(pred_y))
```

```
86783
```

In [91]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```



In [92]:

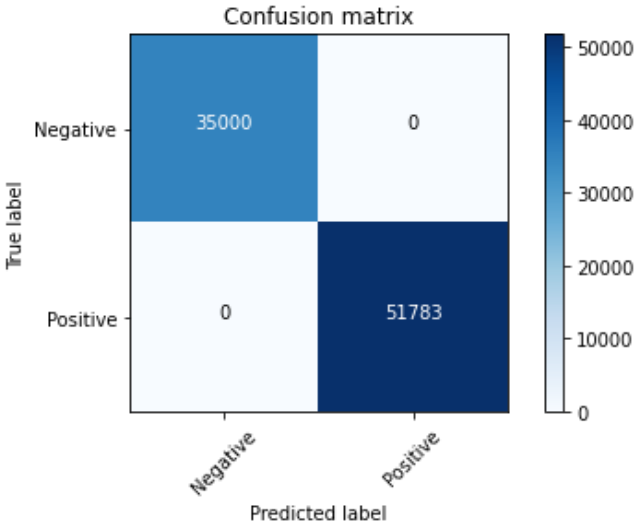
```
from sklearn.naive_bayes import GaussianNB
model_3 = GaussianNB()
model_3.fit(train_X,train_y)
pred_y = model_3.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

```
0.0
```

In [93]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```



In [94]:

```
from sklearn.linear_model import SGDClassifier
model_4 = SGDClassifier(loss="hinge", penalty="l2", max_iter=5)
model_4.fit(train_X,train_y)
pred_y = model_4.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

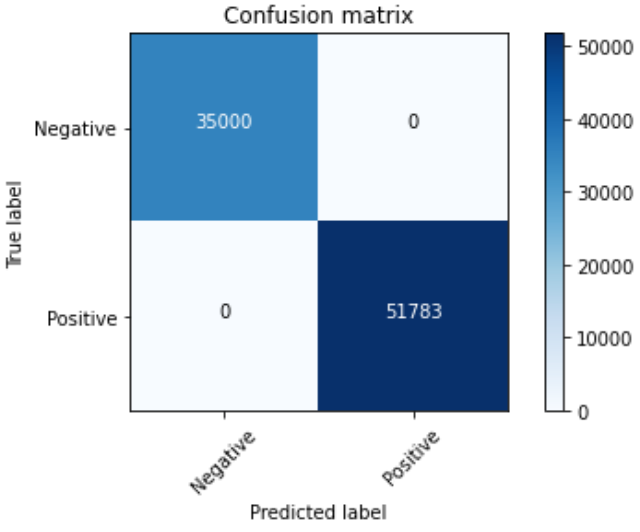
```
0.0
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_stochastic_gradient.py:573:
ConvergenceWarning: Maximum number of iteration reached before convergence. Consider incr
easing max_iter to improve the fit.
  ConvergenceWarning)
```

In [95]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```



In [96]:

```
from sklearn.tree import DecisionTreeClassifier
```

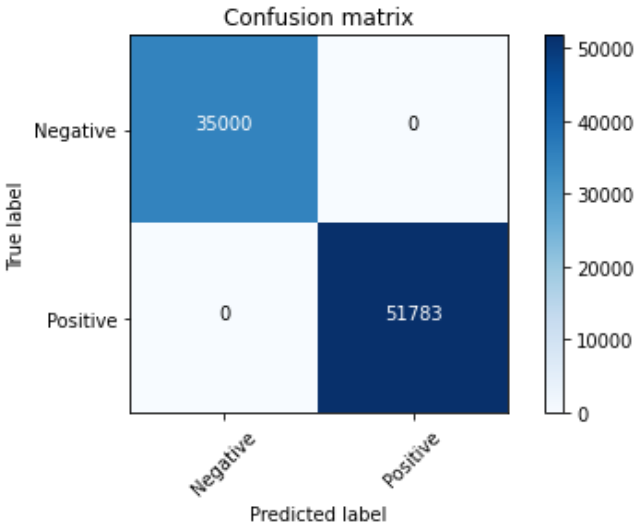
```
model_5 = DecisionTreeClassifier()
model_5.fit(train_X,train_y)
pred_y = model_5.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

```
0.0
```

In [97]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```



In [98]:

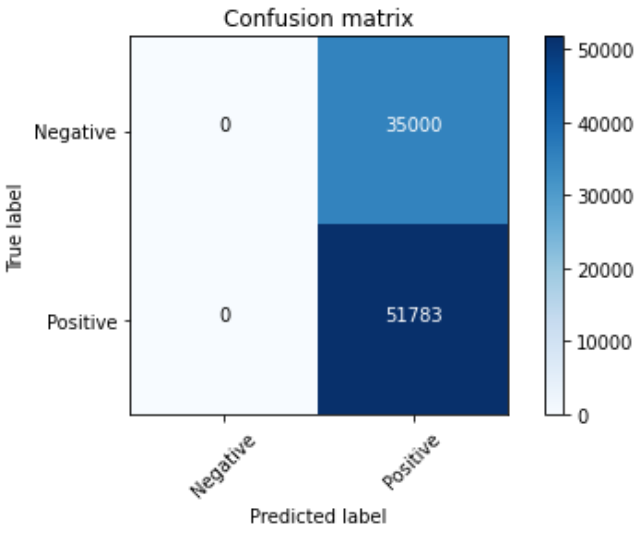
```
from sklearn.naive_bayes import BernoulliNB
model_6 = BernoulliNB()
model_6.fit(train_X,train_y)
pred_y = model_6.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

```
0.6350628273784095
```

In [99]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```



In [100]:

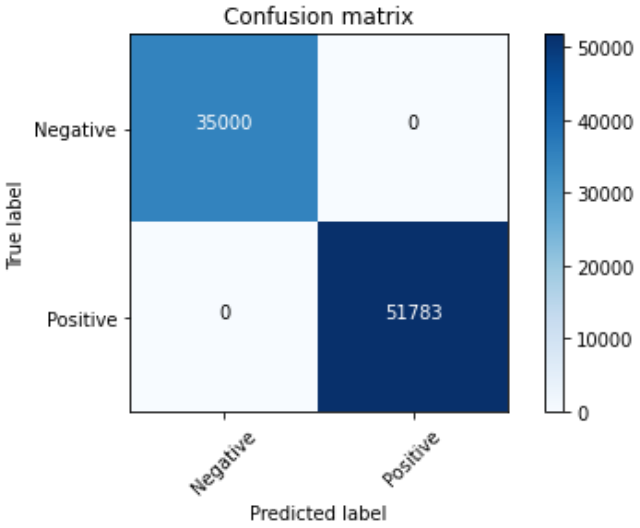
```
from sklearn.neighbors import KNeighborsClassifier
model_7= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
model_7.fit(X, y)
pred_y = model_7.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

```
0.0
```

In [101]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```



In [102]:

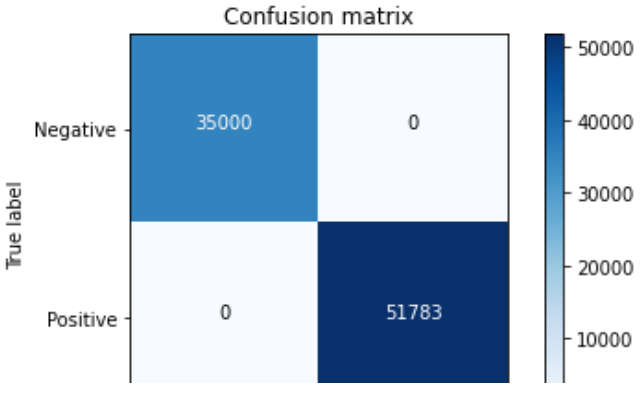
```
from sklearn.neighbors import KNeighborsClassifier
model_7= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
model_7.fit(train_X, train_y)
pred_y = model_7.predict(test_X)
error = mean_squared_error(pred_y,test_y)
rmse = np.sqrt(error)
print(rmse)
```

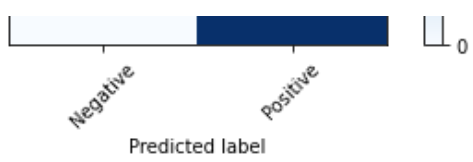
```
0.0
```

In [103]:

```
from sklearn import metrics
cm = metrics.confusion_matrix(test_y, pred_y)
plot_confusion_matrix(cm, classes=['Negative','Positive'])
```

```
Confusion matrix, without normalization
```





In [105]:

```
from sklearn.model_selection import cross_val_score
```

In [106]:

```
model_2= LogisticRegression(random_state=42)
scores = cross_val_score(model_2, X, y, cv=5, scoring='f1_macro')
print(scores)
```

```
[1. 1. 1. 1. 1.]
```