Project-1-Amazon Fine Food Reviews

September 20, 2018

1 Project on classifying whether a review is positive or not for Amazon Fine Foods

```
In [1]: %matplotlib inline

        #insert required modules
        import sqlite3
        import pandas as pd
        import numpy as np
        import nltk
        import string
        import matplotlib.pyplot as plt
        import seaborn as sns
        import re
        import string

        from sklearn.feature_extraction.text import TfidfTransformer
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.metrics import confusion_matrix
        from sklearn import metrics
        from sklearn.metrics import roc_curve,auc
        from sklearn.manifold import TSNE
        from nltk.corpus import stopwords
        from nltk.stem.porter import PorterStemmer
        from nltk.stem import SnowballStemmer
        from nltk.stem.wordnet import WordNetLemmatizer

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

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

```
Out[1]: True
```

```
In [2]: #load the database file
        con=sqlite3.connect('D:\Applied AI Course\database.sqlite')
```

```
In [3]: #query files
        filt_data=pd.read_sql_query("""SELECT * FROM REVIEWS WHERE score!=3""",con)
```

```
In [4]: #check data and shape
        print(filt_data.shape)
        print(filt_data.head())
```

```
(525814, 10)
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"
3   4  B000UA0QIQ  A395BORC6FGVXV                             Karl
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T    Michael D. Bigham "M. Wassir"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600
3                     3                       3      2  1307923200
4                     0                       0      5  1350777600

                 Summary                                               Text
0  Good Quality Dog Food  I have bought several of the Vitality canned d...
1      Not as Advertised  Product arrived labeled as Jumbo Salted Peanut...
2  "Delight" says it all  This is a confection that has been around a fe...
3         Cough Medicine  If you are looking for the secret ingredient i...
4            Great taffy  Great taffy at a great price.  There was a wid...
```

```
In [5]: import pickle

        def savetofile(obj,filename):
            pickle.dump(obj,open(filename+".p","wb"), protocol=4)
        def openfromfile(filename):
            temp = pickle.load(open(filename+".p","rb"))
            return temp
```

```
In [6]: #change the Score field to Review and assign as positive or negative either using lamb
        #using custom function
        def partition(x):
            if x < 3:
                return 'Negative'
            return 'Positive'

        #using lambdas
        #filt_data['Score']=filt_data['Score'].apply(lambda x: 'Positive' if int(x)>3 else 'Ne
In [7]: #change column
        ActScore=filt_data['Score']
        positiveNegative=ActScore.map(partition)
        filt_data['Score']=positiveNegative
```

```
In [146]: filt_data.head(3)
```

```
Out[146]:    Id   ProductId          UserId                      ProfileName  \
          0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
          1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
          2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"

             HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
          0                     1                       1  Positive  1303862400
          1                     0                       0  Negative  1346976000
          2                     1                       1  Positive  1219017600

                           Summary                                               Text
          0  Good Quality Dog Food  I have bought several of the Vitality canned d...
          1      Not as Advertised  Product arrived labeled as Jumbo Salted Peanut...
          2  "Delight" says it all  This is a confection that has been around a fe...
```

```
In [9]: filt_data.shape
```

```
Out[9]: (525814, 10)
```

```
In [10]: import gensim
         from gensim.models import word2vec,KeyedVectors
```

```
D:\Anaconda\lib\site-packages\gensim\utils.py:1209: UserWarning: detected Windows; aliasing chu
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

2 Data cleaning-removing duplicate entries

```
In [11]: dup_data=pd.read_sql_query("""SELECT * FROM REVIEWS WHERE score!=3 ORDER BY ProductId
```

```
In [12]: dup_data.head(3)
```

```
Out[12]:             Id   ProductId          UserId  \
         0       150493  0006641040   AMX0PJKV4PPNJ
         1       150494  0006641040   AYZ0PR5QZROD1
         2       150496  0006641040  A3KKR87BJ0C595
         3       150497  0006641040  A1HKYQOFC8ZZCH
         4       150498  0006641040  A3SJWISOCP31TR
         5       150499  0006641040  A3E7R866M94L0C
         6       150500  0006641040  A1IJKK6Q1GTEAY
         7       150501  0006641040   AJ46FKXOVC7NR
         8       150502  0006641040   AVFMJ50HNO21J
```

```
9       150503  0006641040  A3R5XMPFU8YZ4D
10      150504  0006641040   AQEYF1AXARWJZ
11      150505  0006641040  A2PTSM496CF40Z
12      150506  0006641040  A2IW4PEEKO2R0U
13      150507  0006641040  A1S4A3IQ2MU7V4
14      150508  0006641040     AZGXZ2UUK6X
15      150509  0006641040  A3CMRKGE0P909G
16      150510  0006641040   AM1MNZMYMS7D8
17      150511  0006641040  A1C9K534BCI9GO
18      150512  0006641040  A1DJXZA5V5FFVA
19      150513  0006641040   ASH0DZQQF6AIZ
20      150514  0006641040   A2ONB6ZA292PA
21      150515  0006641040  A2RTT81R6Y3R7X
22      150516  0006641040  A3OI7ZGH6WZJ5G
23      150517  0006641040   ABW4IC5G5G8B5
24      150518  0006641040   AK1L4EJBA23JF
25      150519  0006641040  A12HY5OZ2QNK4N
26      150520  0006641040   ADBFSA9KTQANE
27      150521  0006641040  A3RMCRB2NDTDYP
28      150522  0006641040  A1S3C5OFU508P3
29      150523  0006641040  A2P4F2UO0UMP8C
...        ...         ...             ...
525784  193171  B009RSR8HO   AH2FVNP7Z6PZH
525785  193172  B009RSR8HO  A3JJTHP8T7A8LY
525786  193173  B009RSR8HO  A34TVEXPHSSPBV
525787  193174  B009RSR8HO   A4P6AN2L435PV
525788  193175  B009RSR8HO  A1A0PMN417S4V9
525789  193176  B009RSR8HO   A76WHW051R3KV
525790  204271  B009SA5NNW  A133WGB2RLKB1T
525791  204272  B009SA5NNW   AWFA8N9IXELVH
525792  204273  B009SA5NNW   AG4YGLLIE8BWP
525793  204274  B009SA5NNW  A379KV6EQ66ZJR
525794  204275  B009SA5NNW  A1XPE0WCC6RYVO
525795  204276  B009SA5NNW  A3U0YIPTZX8DZ4
525796  204277  B009SA5NNW  A2TWDT92R8VPTI
525797  204278  B009SA5NNW   A3M922QSBYYXR
525798  204279  B009SA5NNW  A1PVBIUKEDNGVP
525799  204280  B009SA5NNW   AI1G344L7R1TN
525800  204281  B009SA5NNW  A3CBCI8ZU6A9XM
525801  204282  B009SA5NNW  A373QMETEUKMS7
525802  204283  B009SA5NNW  A2QXG1QOV4MTVL
525803  204284  B009SA5NNW  A2SB8DPH72UOM7
525804  204285  B009SA5NNW  A2XNO53D6J6322
525805  204286  B009SA5NNW   AVRU1Z8N59UZV
525806  188389  B009SF0TN6  A1L0GWGRK4BYPT
525807  226019  B009SMKESO  A35K4XT7T1ZIFU
525808  221795  B009SR4OQ2  A32A6X5KCP7ARG
525809  191721  B009UOFTUI   AJVB004EB0MVK
```

```
525810    1478  B009UOFU20   AJVB004EB0MVK
525811  328482  B009UUS05I   ARL20DSHGVM1Y
525812    5703  B009WSNWC4   AMP7K1O84DH1T
525813  327601  B009WVB40S  A3ME78KVX31T21

                                             ProfileName  \
0                               E. R. Bird "Ramseelbird"
1                                      Mother of 3 girls
2          Gretchen Goodfellow "Lover of children's lit"
3                             Maria Apolloni "lanarossa"
4                                            R. J. Wells
5                                 L. Barker "simienwolf"
6                                             A Customer
7                                     Nicholas A Mesiano
8                                               Jane Doe
9                          Her Royal Motherliness "Nana"
10                             Les Sinclair "book maven"
11      Jason A. Teeple "Nobody made a greater mistak...
12                                                 Tracy
13                                 sally sue "sally sue"
14                           Catherine Hallberg "(Kate)"
15                                                Teresa
16                                  Dr. Joshua  Grossman
17                                    Laura Purdie Salas
18                                             A. Conway
19                                              tessarat
20                                      Rosalind Matzner
21                                               Lindylu
22                           Mary Jane Rogers "Maedchen"
23                                           kevin clark
24                                           L. M. Kraus
25                                 Elizabeth H. Roessner
26                        James L. Hammock "Pucks Buddy"
27                                      Carol Carruthers
28                                     Charles Ashbacher
29                 Elizabeth A. Curry "Lovely Librarian"
...                                                  ...
525784                                    Marty Campbell
525785                            Joanne Eklund "Joanne"
525786                                              Beth
525787                                            romarc
525788                               mamaelle "mamaelle"
525789                                     Shawn "Shawn"
525790                                     Temple Gordon
525791                                       No Pen Name
525792                                          Miwintee
525793                                             Craig
525794                                          AnthonyT
```

```
525795                                                vee
525796  d wilson "Visitor from a Perpendicular Universe"
525797                                    Jeannie Jordahl
525798                                            Steve L
525799                                 Brian M. Schissler
525800                                            Cody B.
525801                                       Rebecca Wade
525802                                Wordup "Wordup2you"
525803                                             Tim C.
525804                                       dragenfli254
525805                                        LIsa Fresch
525806                                      Bety Robinson
525807                                        Inez Rivera
525808                                            sicamar
525809                                  D. Christofferson
525810                                  D. Christofferson
525811                                              Jamie
525812                                               ESTY
525813                                               K'la

        HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                         71                      72      4  1096416000
1                          3                       3      5  1173312000
2                          3                       3      5  1111363200
3                          2                       2      1  1334707200
4                          2                       2      5  1176336000
5                          2                       2      5  1065830400
6                          2                       2      5  1009324800
7                          2                       2      5   940809600
8                          1                       1      4  1324944000
9                          1                       1      5  1233964800
10                         1                       1      4  1212278400
11                         1                       1      4  1210809600
12                         1                       1      4  1194739200
13                         1                       1      4  1191456000
14                         1                       1      5  1076025600
15                         3                       4      5  1018396800
16                         0                       0      5  1348358400
17                         0                       0      4  1344211200
18                         0                       0      5  1338249600
19                         0                       0      5  1325721600
20                         0                       0      5  1313884800
21                         0                       0      5  1303171200
22                         0                       0      5  1293840000
23                         0                       0      5  1291075200
24                         0                       0      5  1288224000
25                         0                       0      5  1256774400
26                         0                       0      5  1256688000
```

```
27                         0                       0      5  1243468800
28                         0                       0      4  1219536000
29                         0                       0      4  1096675200
...                      ...                     ...    ...         ...
525784                     0                       0      5  1350432000
525785                     0                       0      5  1350432000
525786                     0                       0      5  1350432000
525787                     0                       0      5  1350432000
525788                     0                       0      5  1350432000
525789                     0                       0      5  1350432000
525790                     1                       1      4  1321228800
525791                     0                       0      1  1351123200
525792                     0                       0      5  1351123200
525793                     0                       0      5  1347062400
525794                     0                       0      5  1344988800
525795                     0                       0      4  1339977600
525796                     0                       0      5  1337904000
525797                     0                       0      5  1335744000
525798                     0                       0      5  1333843200
525799                     0                       0      4  1332979200
525800                     0                       0      5  1328486400
525801                     0                       0      4  1327017600
525802                     0                       0      1  1321920000
525803                     0                       0      4  1317600000
525804                     0                       0      5  1317081600
525805                     0                       1      1  1349654400
525806                     0                       0      5  1350518400
525807                     0                       1      4  1304985600
525808                     1                       1      5  1350604800
525809                     0                       0      1  1345852800
525810                     0                       0      1  1345852800
525811                     0                       0      5  1331856000
525812                     0                       0      5  1351209600
525813                     0                       0      5  1351123200

                                                  Summary  \
0       Read it once. Read it twice. Reading Chicken S...
1                                         Family favorite
2                 You'll use it once, you'll use it twice
3       The story is great, the softcover book is disa...
4                                         A Gem of a Book
5                                       Can't explain why
6                                      It Was a favorite!
7       This whole series is great way to spend time w...
8              Tiny little book, Wonderful little rhymes.
9                                          so fun to read
10                                  Chicken Soup with Rice
11                                              A classic
```

```
12             Love the book, miss the hard cover version
13                          chicken soup with rice months
14                 a good swingy rhythm for reading aloud
15                        A great way to learn the months
16                                 Professional Mentoring
17                                 Charming and childlike
18                                             Must have.
19                                              A classic
20                                 Chicken soup with Rice
21                     One of our family's favorite books
22                                               Darling!
23                                      good for children
24                                         love this book
25                                     It's a great book!
26                                             Great Gift
27                                    This book is great!
28      Children will find it entertaining and a gener...
29                                  MMMM chicken soup... 
...                                                   ...
525784          The BEST sugar replacement on the market!
525785                                               Zero
525786                                           Love it!
525787                                     LOVE!!  LOVE!!
525788                                 YAY! No Dextrose!!
525789                          My #1 Sweetener of choice
525790                         Walkers smkey bacon crisps
525791                              Deceptive description
525792               Makes me drool just thinking of them
525793  Awesome Crisps!!! Arrived in just 8 days in Te...
525794                                          Excellent
525795                                          Re-Rating
525796                            Tastes just like bacon!
525797                                These were amazing!
525798                                   One Word, "YUM!"
525799                                           WOW...  
525800                                            Cody B.
525801                                         Excellent!
525802                                        Stale Chips
525803                                               Yum!
525804                                            Delish!
525805                              Walkers Crisps 6 pack
525806              Amazing!! Great sauce for everything!
525807                                 Not a bad product.
525808                                      Awesome Taste
525809  weak coffee not good for a premium product and...
525810  weak coffee not good for a premium product and...
525811                                            Perfect
525812                                          DELICIOUS
```

```
525813                                          Tasty!

                                                      Text
0       These days, when a person says, "chicken soup"...
1       All of my children love this book.  My first g...
2       One of my earliest memories is of this book.  ...
3       I give five stars to the Maurice Sendak story... 
4       This is a wonderful little book. I loved it 40...
5       This book has been a favorite of mine since I ...
6       This was a favorite book of mine when I was a ...
7       I can remember seeing the show when it aired o...
8       This copy is smaller than I expected (mostly b...
9       This is my grand daughter's and my favorite bo...
10      A very entertaining rhyming story--cleaver and...
11      Get the movie or sound track and sing along wi...
12      I grew up reading these Sendak books, and watc...
13      This is a fun way for children to learn their ...
14      This is a great little book to read aloud- it ...
15      This is a book of poetry about the months of t...
16      TITLE: Chicken Soup with Rice<br />AUTHOR: Mau...
17      A charming, rhyming book that describes the ci...
18      I set aside at least an hour each day to read ...
19      I remembered this book from my childhood and g...
20      It's a great book with adorable illustrations... 
21      This book is a family favorite and was read to...
22      The same author wrote "Where the Wild Things A...
23      Classic children's book, can't go wrong. I rea...
24      Great book, perfect condition arrived in a sho...
25      I've always loved chicken soup and rice. My la...
26      This book was purchased as a birthday gift for...
27      My 7 year old daughter brought this book home ...
28      This book contains a collection of twelve shor...
29      Summary:  A young boy describes the usefulness...
...                                                  ...
525784  I've been using Fat to Skinny Zero since it wa...
525785  FTS Zero is the best sweetener I have ever tri...
525786  I love this sweetener.  I use it to replace su...
525787  LOVE, LOVE this sweetener!!  I use it in all m...
525788  Packets of powdered sweeteners usually have a ...
525789  What a wonderful product! It's perfect to use ...
525790  These are amazing chips but they just cost too...
525791  On Oct 9 I ordered from a different vendor the...
525792  The Brit's have out done us. The flavor is sup...
525793  These crisps are my favorite.  I ordered these...
525794  These are the best flavor chips, my daughter a...
525795  Okay, I jumped the gun, because they were send...
525796  I had a bag of these during a trip to London. ...
525797  This chips kind of reminded me of bacon bits. ...
```

```
525798  If you like salt and vinegar crisps (chips), b...
525799  This could possibly be the best tasting chip I...
525800  I loved the chips they were AWESOME!!! but tha...
525801  The crisps are awesome. Give me English crisps...
525802  Item came promptly however the crisps were 3 m...
525803  Bought these the other day while I was in Cana...
525804  I had these wonderful chips in Ireland a few y...
525805  I ordered this product on Amazon to get some o...
525806  You have to try this sauce to believe it! It s...
525807  This review is for the boneless ham. A little ...
525808  I bought this Hazelnut Paste (Nocciola Spread)...
525809  This coffee supposedly is premium, it tastes w...
525810  This coffee supposedly is premium, it tastes w...
525811  The basket was the perfect sympathy gift when ...
525812  Purchased this product at a local store in NY ...
525813  I purchased this to send to my son who's away ...

[525814 rows x 10 columns]
```

In [145]:
```
#the product id 0006641040 is a book and not a fine food and hence to be removed
sort_data=filt_data.sort_values('ProductId',axis=0,ascending=True)
sort_data.head(5)
```

Out[145]:
```
            Id   ProductId          UserId                  ProfileName  \
138706  150524  0006641040   ACITT7DI6IDDL              shari zychinski
138688  150506  0006641040  A2IW4PEEKO2ROU                        Tracy
138689  150507  0006641040  A1S4A3IQ2MU7V4        sally sue "sally sue"
138690  150508  0006641040     AZGXZ2UUK6X  Catherine Hallberg "(Kate)"
138691  150509  0006641040  A3CMRKGE0P909G                       Teresa

        HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
138706                     0                       0  Positive   939340800
138688                     1                       1  Positive  1194739200
138689                     1                       1  Positive  1191456000
138690                     1                       1  Positive  1076025600
138691                     3                       4  Positive  1018396800

                                           Summary  \
138706                   EVERY book is educational
138688  Love the book, miss the hard cover version
138689               chicken soup with rice months
138690      a good swingy rhythm for reading aloud
138691             A great way to learn the months

                                                     Text
138706  this witty little book makes my son laugh at l...
138688  I grew up reading these Sendak books, and watc...
138689  This is a fun way for children to learn their ...
```

```
          138690  This is a great little book to read aloud- it ...
          138691  This is a book of poetry about the months of t...
```

```
In [14]: final_data=sort_data.drop_duplicates(subset={"UserId","ProfileName","Time","Text"},kee
```

```
In [15]: percent=(final_data['Id'].size*1.0 / filt_data['Id'].size*1.0) *100
         print(percent)
```

```
69.25890143662969
```

```
In [16]: final_data["Score"].value_counts()
```

```
Out[16]: Positive    307063
         Negative     57110
         Name: Score, dtype: int64
```

```
In [143]: final_data.head(3)
```

```
Out[143]:             Id   ProductId          UserId            ProfileName  \
          138706  150524  0006641040   ACITT7DI6IDDL        shari zychinski
          138688  150506  0006641040  A2IW4PEEKO2R0U                  Tracy
          138689  150507  0006641040  A1S4A3IQ2MU7V4  sally sue "sally sue"

                  HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
          138706                     0                       0  Positive   939340800
          138688                     1                       1  Positive  1194739200
          138689                     1                       1  Positive  1191456000

                                                   Summary  \
          138706                    EVERY book is educational
          138688  Love the book, miss the hard cover version
          138689                chicken soup with rice months

                                                              Text  \
          138706  this witty little book makes my son laugh at l...
          138688  I grew up reading these Sendak books, and watc...
          138689  This is a fun way for children to learn their ...

                                                       CleanedText
          138706  witti littl book make son laugh loud recit car...
          138688  grew read sendak book watch realli rosi movi i...
          138689  fun way children learn month year learn poem t...
```

```
In [18]: dup_data1=pd.read_sql_query("""SELECT DISTINCT ProductId,UserId FROM REVIEWS WHERE sc
```

```
In [19]: dup_data1.shape
```

```
Out[19]: (34, 2)
```

```
In [22]: labels=final_data['Score']

In [24]: labels.head(3)

Out[24]: 138706    Positive
         138688    Positive
         138689    Positive
         Name: Score, dtype: object
```

3 BOW,TFIDF,Word2Vec(Avg-W2Vec,TfIDF-W2Vec) t-SNE plots

Text preprocessing 1) Remove HTML tags present in Text column words 2) remove any punctuation 3) check if word in english and alphanumeric 4) check if length>2 5) convert all words to lowercase 6) remove stopwords

```
In [25]: #helper functions

         stop_word=set(stopwords.words('english'))
         sno=SnowballStemmer('english')

         def cleanhtml(sentence):
             cleanh=re.compile('<.*?>')
             cleantext=re.sub(cleanh,' ',sentence)
             return cleantext

         def cleanpunc(sentence):
             cleaned=re.sub(r'[?|!|\'|"|#]',r'',sentence)
             cleaned=re.sub(r'[.|,|)|(|\|/]',r'',cleaned)
             return cleaned

         print(stop_word)
         print("##################################################")
         print(sno.stem('tasty'))

{'at', 'up', 'into', 'under', 'most', 'don', 'but', "haven't", "you'll", 'himself', 'all', "sha
##################################################
tasti
```

```
In [26]: #code to check for implemented checks above
         i=0
         str1=''
         final_string=[]
         all_pos_words=[]
         all_neg_words=[]
         s=''

         for sent in final_data['Text'].values:
```

```
    filtered_sentences=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                if(cleaned_words.lower() not in stop_word):
                    s=(sno.stem(cleaned_words.lower())).encode('utf8')
                    filtered_sentences.append(s)
                    if (labels.values)[i]=='Positive':
                        all_pos_words.append(s)
                    if (labels.values)[i]=='Negative':
                        all_neg_words.append(s)
                else:
                    continue
            else:
                continue

    str1=b" ".join(filtered_sentences)

    final_string.append(str1)
    i +=1
```

```
In [27]: final_data['CleanedText']=final_string #adding a column of CleanedText which displays
         final_data['CleanedText']=final_data['CleanedText'].str.decode("utf8")
```

```
D:\Anaconda\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  """Entry point for launching an IPython kernel.
D:\Anaconda\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
```

```
In [142]: final_data.head(3)
```

```
Out[142]:             Id   ProductId          UserId            ProfileName  \
          138706  150524  0006641040   ACITT7DI6IDDL       shari zychinski
          138688  150506  0006641040  A2IW4PEEKO2R0U                  Tracy
          138689  150507  0006641040  A1S4A3IQ2MU7V4  sally sue "sally sue"
```

```
        HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
138706                     0                       0  Positive   939340800
138688                     1                       1  Positive  1194739200
138689                     1                       1  Positive  1191456000

                                         Summary  \
138706                  EVERY book is educational
138688  Love the book, miss the hard cover version
138689              chicken soup with rice months

                                                     Text  \
138706  this witty little book makes my son laugh at l...
138688  I grew up reading these Sendak books, and watc...
138689  This is a fun way for children to learn their ...

                                              CleanedText
138706  witti littl book make son laugh loud recit car...
138688  grew read sendak book watch realli rosi movi i...
138689  fun way children learn month year learn poem t...
```

In [29]:
```
#save it to database
conn=sqlite3.connect('final2.sqlite')
c=conn.cursor()
#c.execute("alter table REVIEWS add column '%s'" %labels)
conn.commit()
conn.text_factory=str
final_data.to_sql('Reviews',conn,schema=None,if_exists='replace')
```

In [30]:
```
final_data.head(3)
```

Out[30]:
```
            Id   ProductId          UserId            ProfileName  \
138706  150524  0006641040   ACITT7DI6IDDL        shari zychinski
138688  150506  0006641040  A2IW4PEEKO2R0U                  Tracy
138689  150507  0006641040  A1S4A3IQ2MU7V4  sally sue "sally sue"

        HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
138706                     0                       0  Positive   939340800
138688                     1                       1  Positive  1194739200
138689                     1                       1  Positive  1191456000

                                         Summary  \
138706                  EVERY book is educational
138688  Love the book, miss the hard cover version
138689              chicken soup with rice months

                                                     Text  \
138706  this witty little book makes my son laugh at l...
```

```
138688  I grew up reading these Sendak books, and watc...
138689  This is a fun way for children to learn their ...

                                              CleanedText
138706  witti littl book make son laugh loud recit car...
138688  grew read sendak book watch realli rosi movi i...
138689  fun way children learn month year learn poem t...
```

```
In [119]: n_samples=2000
          test_data=final_data.sample(n_samples)
          label_data=final_data['Score'][0:2000]
```

```
In [141]: test_data.head(5)
```

```
Out[141]:
             Id   ProductId          UserId          ProfileName  \
438074  473737  B000FCI6T0  A3K05R0KCA9BD3       Cynde "cyndec"
326423  353276  B000HEA8Q0  A1LZUDRS218G1R               DMM-NH
50185    54488  B001TLY7A8  A20X2L5P94PZPF        Diana L. Gray
13970    15247  B00503DP00  A1H6SB07R007I8            A. Reader
372411  402730  B0043H35Y0  A3K4TWQ0C43MXX  michelle "michelle"

        HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
438074                     2                       2  Positive  1172707200
326423                     4                       4  Positive  1294617600
50185                      0                       0  Positive  1287100800
13970                      2                       2  Positive  1313625600
372411                     0                       0  Positive  1341187200

                                  Summary  \
438074                      Really fresh!
326423            Amost excellent product
50185              Roxie loves this food!
13970         amazing delicious fantastic
372411  great to get rid of garlic breath

                                                     Text  \
438074  I received this box as a gift from my husband ...
326423  I looked for this product for years. Quite acc...
50185   I've tried numerous canned -grain free- foods ...
13970   Sure it says 'Oat' bar, but make no mistake th...
372411  i tried this for the first time in NY,  and lo...

                                              CleanedText
438074  receiv box gift husband valentin day real trea...
326423  look product year quit accident found tea shop...
50185   ive tri numer can food cat roxi doesnt care op...
13970   sure say oat bar make mistak indulg amaz proab...
372411  tri first time love bought amazon happybut bad...
```

```
In [120]: #bag of words
          count_vect=CountVectorizer()
          final_count=count_vect.fit_transform(test_data['CleanedText'].values)
          type(final_count)
          final_count.get_shape()
          #Bi-grams and n-grams

          freq_dist_pos=nltk.FreqDist(all_pos_words)
          freq_dist_neg=nltk.FreqDist(all_neg_words)
          print("Most common positive words:",freq_dist_pos.most_common(20))
          print("Most common negative words:",freq_dist_neg.most_common(20))

          #Bi-grams
          #count_vect=CountVectorizer(ngram_range=(1,2))
          #final_count=count_vect.fit_transform(test_data['CleanedText'].values)
```

```
Most common positive words: [(b'like', 139075), (b'tast', 128082), (b'good', 112017), (b'flavo
Most common negative words: [(b'tast', 34300), (b'like', 32225), (b'product', 28003), (b'one',
```

4 Bag Of Words

```
In [32]: #bag of words
         count_vect=CountVectorizer()
         final_count=count_vect.fit_transform(final_data['CleanedText'].values)
         print("the type of count vectorizer is:",type(final_count))
         final_count.get_shape()
```

```
the type of count vectorizer is: <class 'scipy.sparse.csr.csr_matrix'>


Out[32]: (364173, 120724)
```

```
In [33]: final_count.get_shape
```

```
Out[33]: <bound method spmatrix.get_shape of <364173x120724 sparse matrix of type '<class 'num
                 with 11452731 stored elements in Compressed Sparse Row format>>
```

```
In [34]: #t-SNE plot for Bag of words
         #from sklearn.preprocessing import StandardScaler

         #standard_data=StandardScaler(with_mean=False).fit_transform(final_count)
         #standard_data.shape
```

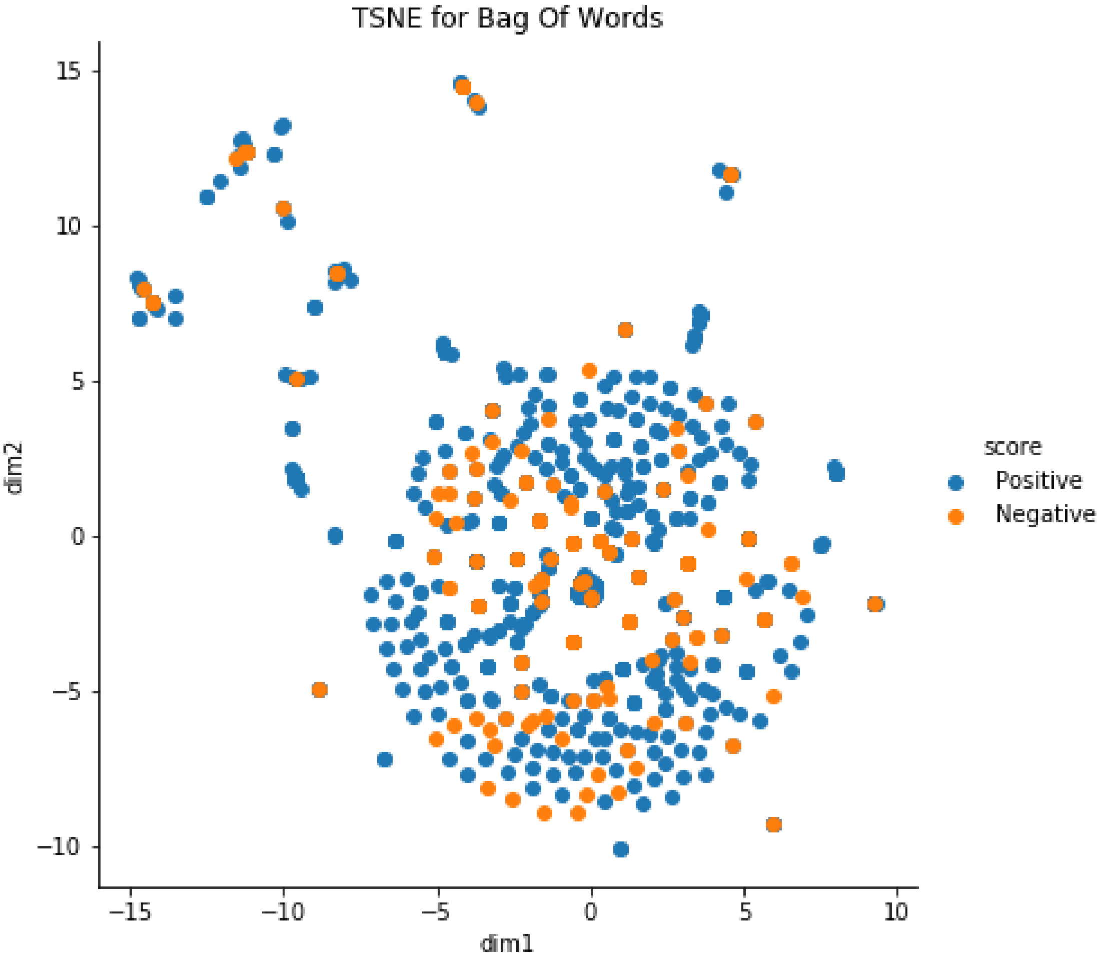
```
In [35]: n_samples=1000
         std_data=final_count[0:n_samples,:n_samples].todense()
         label_data=final_data["Score"][0:n_samples]
```

```
In [36]: std_data.shape
```

```
Out[36]: (1000, 1000)
```

```
In [37]: from sklearn.manifold import TSNE

         tmodel=TSNE(n_components=2,random_state=0,perplexity=30,n_iter=1000)
         tsne_data=tmodel.fit_transform(std_data)

         tsne_data = np.vstack((tsne_data.T, label_data)).T
         tsne_df = pd.DataFrame(data=tsne_data, columns=("dim1", "dim2", "score"))


         sns.FacetGrid(tsne_df, hue="score", size=6).map(plt.scatter, 'dim1', 'dim2').add_leger
         plt.title("TSNE for Bag Of Words")
         plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` paramter has bee
  warnings.warn(msg, UserWarning)
```



5 TF-IDF

```
In [38]: #tf_idf_vect=TfidfVectorizer(ngram_range=(1,2))
         tf_idf_vect=TfidfVectorizer()
         final_tf_idf_vect=tf_idf_vect.fit_transform(final_data["CleanedText"].values)
         final_tf_idf_vect.get_shape()
         #get features
         features=tf_idf_vect.get_feature_names()
         print(len(features))
         print("type of count vectorizer :",type(final_tf_idf_vect))
```

```
120724
type of count vectorizer : <class 'scipy.sparse.csr.csr_matrix'>
```

```
In [39]: #top tdf-idf features code taken from https://buhrmann.github.io/tfidf-analysis.html
         def top_tfidf_feats(row, features, top_n=25):
             ''' Get top n tfidf values in row and return them with their corresponding featur
             topn_ids = np.argsort(row)[::-1][:top_n]
             top_feats = [(features[i], row[i]) for i in topn_ids]
             df = pd.DataFrame(top_feats)
             df.columns = ['feature', 'tfidf']
             return df

         top_tfidf = top_tfidf_feats(final_tf_idf_vect[1,:].toarray()[0],features,25)
```

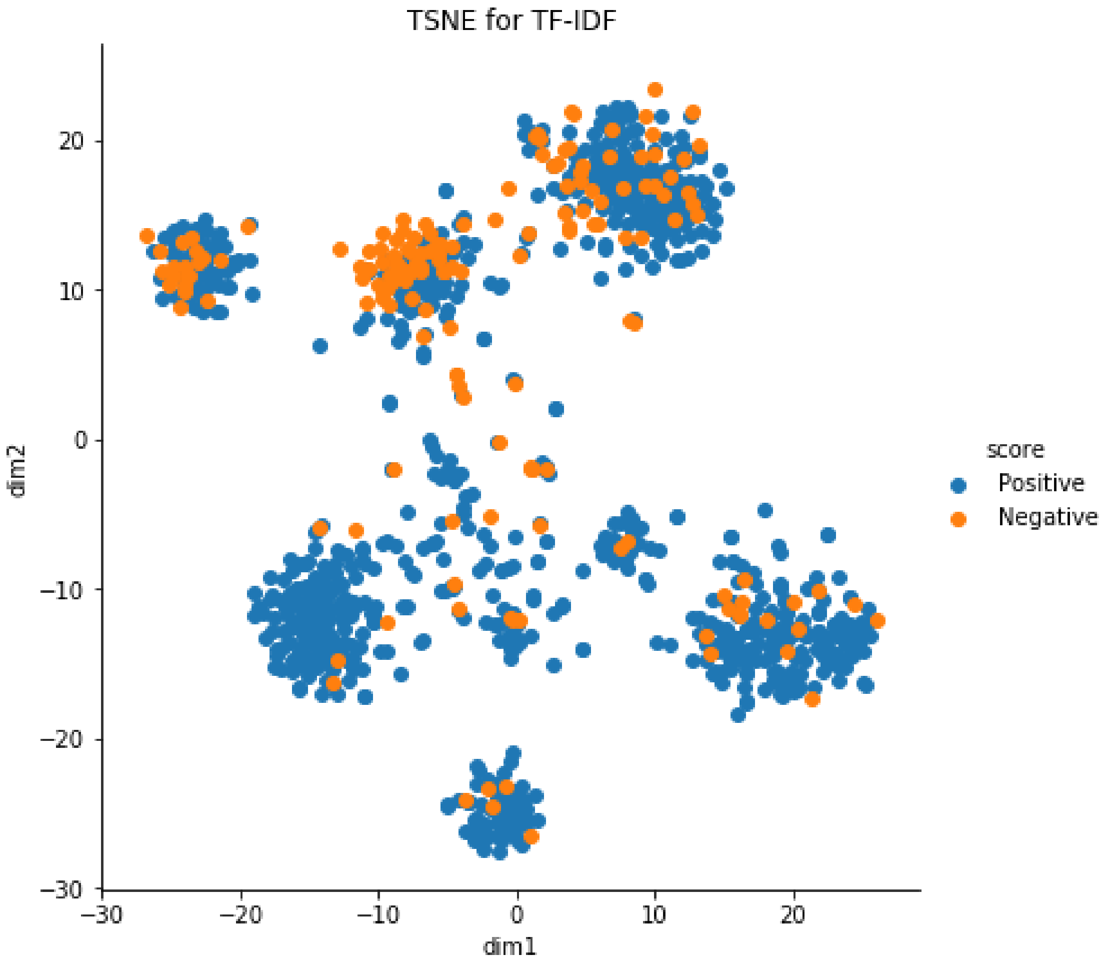
```
In [40]: top_tfidf
```

```
Out[40]:       feature     tfidf
         0      sendak  0.359946
         1   paperback  0.348872
         2        rosi  0.320880
         3      flimsi  0.259566
         4    incorpor  0.247205
         5        page  0.222703
         6        movi  0.212070
         7        book  0.202037
         8        grew  0.195787
         9       cover  0.176995
         10      watch  0.175478
         11       miss  0.171671
         12    version  0.160795
         13        son  0.159014
         14       love  0.149731
         15       hand  0.146462
         16       read  0.142371
         17       kind  0.140196
         18       open  0.132674
         19      howev  0.128820
```

```
20       hard  0.126456
21       seem  0.123736
22       take  0.121284
23       keep  0.119093
24        two  0.116942
```

```
In [41]: #t-SNE visualization for tf-idf
         n_samples=1000
         std_data=final_tf_idf_vect[0:n_samples,:].todense()
         label_data=final_data["Score"][0:n_samples]

         #from sklearn.manifold import TSNE

         tmodel=TSNE(n_components=2,random_state=0,perplexity=40,n_iter=1000)
         tsne_data=tmodel.fit_transform(std_data)

         tsne_data = np.vstack((tsne_data.T, label_data)).T
         tsne_df = pd.DataFrame(data=tsne_data, columns=("dim1", "dim2", "score"))


         sns.FacetGrid(tsne_df, hue="score", size=6).map(plt.scatter, 'dim1', 'dim2').add_legen
         plt.title("TSNE for TF-IDF")
         plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` paramter has bee
  warnings.warn(msg, UserWarning)
```



```
In [121]: savetofile(final_tf_idf_vect,"tfidf")
```

6 Word2Vec

```
In [65]: pwd
```

```
Out[65]: 'C:\\Users\\Tejas'
```

```
In [123]: ##Create own word2vec model

          i=0
          list_of_sentence=[]
          for sent in test_data['CleanedText'].values:
              list_of_sentence.append(sent.split())
              #sent=cleanhtml(sent)
              #for w in sent.split():
                  #for cleaned in cleanpunc(w).split():
                      #if(cleaned.isalpha()):
                          #filtered_sentence.append(cleaned.lower())
                      #else:
                          #continue
                  #list_of_sentence.append(filtered_sentence)
          print(test_data['CleanedText'].values[0])
          print('###########')
          print(list_of_sentence[0])
          w2v_model=gensim.models.Word2Vec(list_of_sentence,min_count=5,size=50,workers=4)

          words=list(w2v_model.wv.vocab)
          print(len(words))
```

```
receiv box gift husband valentin day real treat browni arriv fresh handl perfect chewi tasti ea
###########
['receiv', 'box', 'gift', 'husband', 'valentin', 'day', 'real', 'treat', 'browni', 'arriv', 'fi
1929
```

```
In [128]: w2v_model.save('w2vmodel')
```

```
In [129]: print(w2v_model)
```

```
Word2Vec(vocab=1929, size=50, alpha=0.025)
```

```
In [130]: w2v_model.wv.most_similar('tasti')
```

```
D:\Anaconda\lib\site-packages\gensim\matutils.py:737: FutureWarning: Conversion of the second a
  if np.issubdtype(vec.dtype, np.int):
```

```
Out[130]: [('stuff', 0.9998751282691956),
           ('come', 0.9998654127120972),
           ('cereal', 0.9998612999916077),
           ('away', 0.9998562335968018),
           ('seed', 0.9998495578765869),
           ('dri', 0.9998475909233093),
           ('also', 0.9998428225517273),
           ('pleas', 0.9998418092727661),
           ('healthi', 0.999841570854187),
           ('meal', 0.9998407959938049)]
```

7 Avg W2V

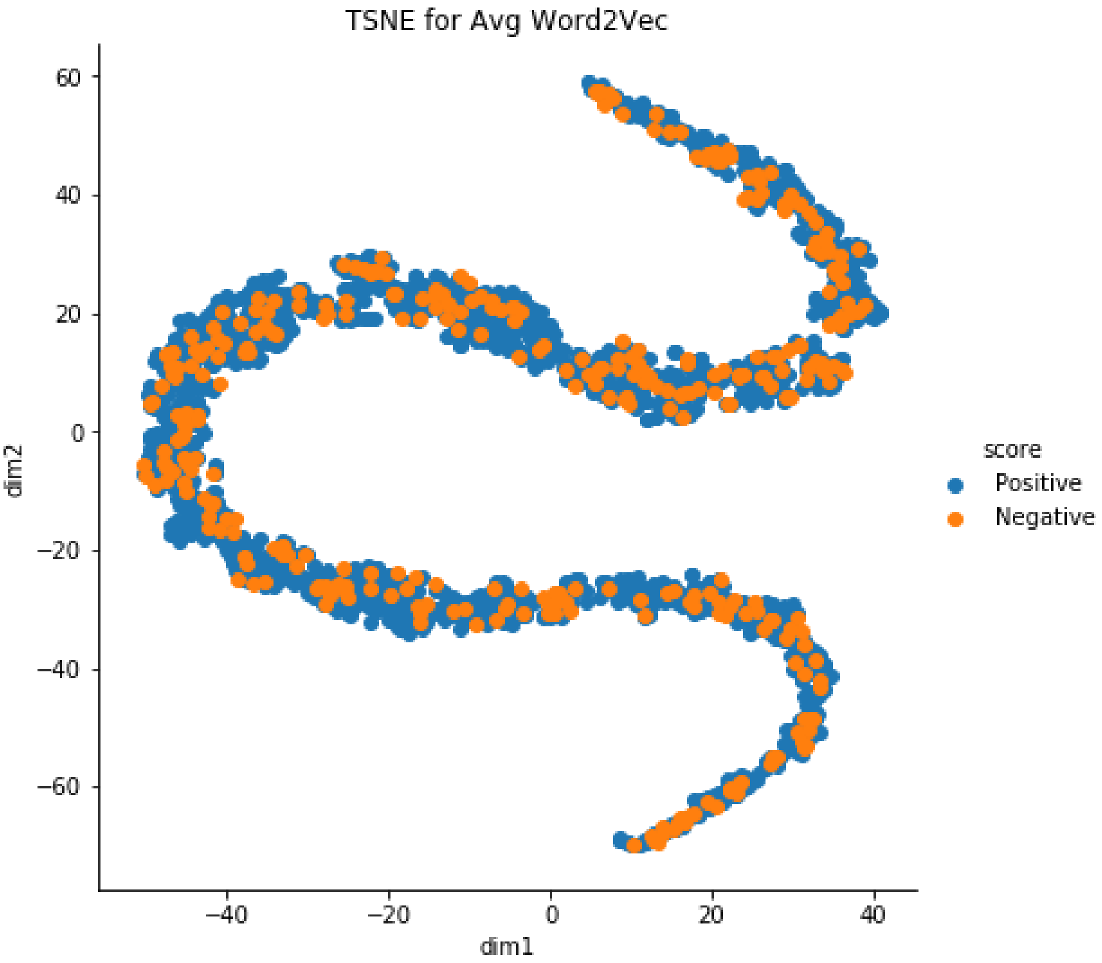
```
In [132]: #average word2vec
          sent_vectors = []
          for sent in list_of_sentence: # for each review/sentence
```

```
    sent_vec = np.zeros(50) # as word vectors are of zero length
    cnt_words =0 # num of words with a valid vector in the sentence/review
    for word in sent: # for each word in a review/sentence
        if word in words:
            vec = w2v_model.wv[word]
            sent_vec += vec
            cnt_words += 1
    if cnt_words != 0:
        sent_vec /= cnt_words
    sent_vectors.append(sent_vec)
print(len(sent_vectors))
print(len(sent_vectors[0]))

vec_avg=np.array(sent_vectors)
```

```
2000
50
```

```
In [134]: #n_samples=1000
          std_data=vec_avg
          #label_data=final_data["Score"][0:n_samples]

          #from sklearn.manifold import TSNE

          tmodel=TSNE(n_components=2,random_state=0,perplexity=30,n_iter=1000)
          tsne_data=tmodel.fit_transform(std_data)

          tsne_data = np.vstack((tsne_data.T, label_data)).T
          tsne_df = pd.DataFrame(data=tsne_data, columns=("dim1", "dim2", "score"))


          sns.FacetGrid(tsne_df, hue="score", size=6).map(plt.scatter, 'dim1', 'dim2').add_lege
          plt.title("TSNE for Avg Word2Vec")
          plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` paramter has bee
  warnings.warn(msg, UserWarning)
```



Maybe need to increase the sample size to get more correct design or dimension for the t-SNE

8 TF-IDF Word2Vec t-SNE

```
In [135]: tf_idf_vect=TfidfVectorizer(ngram_range=(1,2))
          final_tf_idf_vect=tf_idf_vect.fit_transform(test_data["CleanedText"].values)
          final_tf_idf_vect.get_shape()
          #get features
          features=tf_idf_vect.get_feature_names()
          print(len(features))
          print("type of count vectorizer :",type(final_tf_idf_vect))
```

```
62875
type of count vectorizer : <class 'scipy.sparse.csr.csr_matrix'>
```

```
In [95]: tf_idf=openfromfile('tfidf')
         tf_idf
```

```
Out[95]: <364173x120724 sparse matrix of type '<class 'numpy.float64'>'
                with 11452731 stored elements in Compressed Sparse Row format>
```

```
In [136]: print("shape:",final_tf_idf_vect.get_shape())
```

```
shape: (2000, 62875)
```

```
In [137]: from sklearn.decomposition import TruncatedSVD
          s=TruncatedSVD(n_components=5, n_iter=7, random_state=42)
          sample_feat_vect=s.fit_transform(final_tf_idf_vect)
```

```
In [115]: sample_feat_vect
```

```
Out[115]: array([[ 0.02481256, -0.00722239, -0.00572286,  0.0020652 ,  0.00238161],
                 [ 0.04156559, -0.01452842, -0.01195302,  0.00715577, -0.00262578],
                 [ 0.02501641, -0.00705836, -0.00415058,  0.00195987, -0.0012734 ],
                 ...,
                 [ 0.0285171 , -0.00470295,  0.00358799, -0.0105155 ,  0.01320058],
                 [ 0.07065738, -0.02995533, -0.01761324, -0.00174938,  0.05572974],
                 [ 0.05689007, -0.02061095, -0.01080435, -0.0037914 ,  0.03055602]])
```

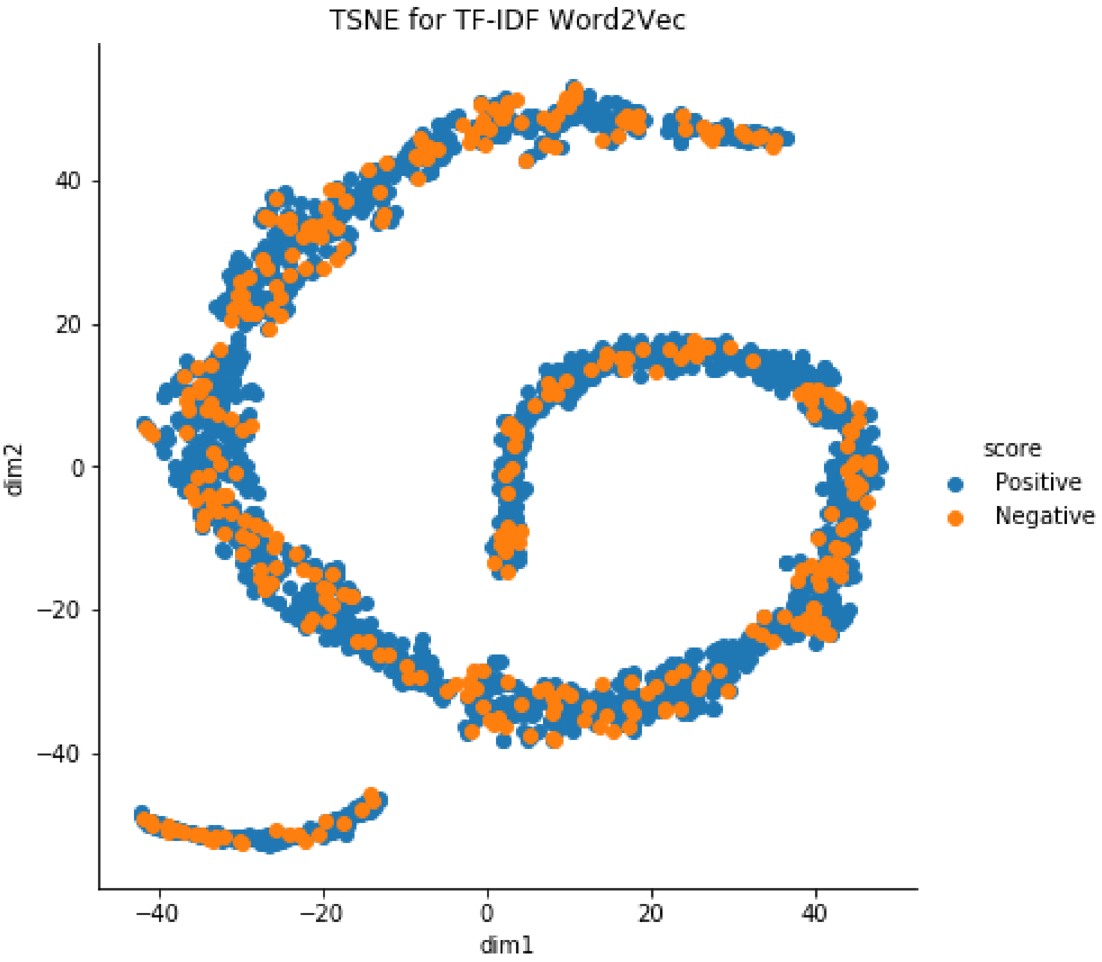
```
In [138]: # TF-IDF weighted Word2Vec
          tf_idf_features = tf_idf_vect.get_feature_names() # tfidf words/col-names
          # final_tf_idf is the sparse matrix with row= sentence, col=word and cell_val = tfid

          tfidf_sent_vectors = [] # the tfidf-w2v for each sentence/review is stored in this l
          row=0
          for sent in list_of_sentence: # for each review/sentence
              sent_vec = np.zeros(50) # as word vectors are of zero length
              weight_sum =0 # num of words with a valid vector in the sentence/review
              for word in sent: # for each word in a review/sentence
                  if word in words:
                      vec = w2v_model.wv[word]
                      # obtain the tf_idfidf of a word in a sentence/review
                      tf_idf = final_tf_idf_vect[row, tf_idf_features.index(word)]
                      sent_vec += (vec * tf_idf)
                      weight_sum += tf_idf
              if weight_sum != 0:
                  sent_vec /= weight_sum
              tfidf_sent_vectors.append(sent_vec)
              row += 1
```

```
In [139]: tf_vec_avg=np.array(tfidf_sent_vectors)
```

```
In [140]: std_data=tf_vec_avg


          #from sklearn.manifold import TSNE
```

```
tmodel=TSNE(n_components=2,random_state=0,perplexity=30,n_iter=1000)
tsne_data=tmodel.fit_transform(std_data)

tsne_data = np.vstack((tsne_data.T, label_data)).T
tsne_df = pd.DataFrame(data=tsne_data, columns=("dim1", "dim2", "score"))


sns.FacetGrid(tsne_df, hue="score", size=6).map(plt.scatter, 'dim1', 'dim2').add_lege
plt.title("TSNE for TF-IDF Word2Vec")
plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` paramter has bee
  warnings.warn(msg, UserWarning)
```



From the above diagrams we can not be able to separate the positive or negative reviews clearly. Even though some of the plots need more working on since sample set size is just 2000