t-SNE-Balanced-AFF-Review

November 21, 2018

1 t-SNE Visualization on Amazon Food Review Dataset

1.1 Import Required Modules

```
In [1]: import os # for file management
        import shutil # for file management
        from pathlib import Path
        import sqlite3
        import pandas as pd
        import numpy as np
        import csv
        from tqdm import tqdm
        import matplotlib.pyplot as plt
        import seaborn as sns
        import time # for time measurement
        import imageio # for GIF creation

        from sklearn.feature_extraction.text import CountVectorizer # for Bag Of Words
        from sklearn.feature_extraction.text import TfidfVectorizer # for text to vector creation
        from gensim.models import Word2Vec

        from sklearn.preprocessing import StandardScaler # for Column Standardization - DO WE NEED THIS?
        from sklearn.manifold import TSNE # for t-SNE
```

```
In [2]: ## Configure Matplotlib for nice image in PDF
        from IPython.display import set_matplotlib_formats
        set_matplotlib_formats('pdf', 'png')
        plt.rcParams['savefig.dpi'] = 75
        plt.rcParams['figure.figsize'] = 10,6
        plt.rcParams['axes.labelsize'] = 18
        plt.rcParams['axes.titlesize'] = 20
        plt.rcParams['font.size'] = 10
        plt.rcParams['lines.linewidth'] = 2.0
        plt.rcParams['lines.markersize'] = 8
```

```
In [3]: output_dir = 'Output'
        if not os.path.exists(output_dir):
            os.makedirs(output_dir)
```

1.2 Load Data

```
In [4]: con = sqlite3.connect('./cleaned.sqlite')

        df = pd.read_sql_query(""" SELECT * from Reviews""", con)
        df.head()
```

```
Out[4]:     index      Id   ProductId          UserId                  ProfileName  \
        0  138706  150524  0006641040   ACITT7DI6IDDL              shari zychinski
        1  138688  150506  0006641040  A2IW4PEEKO2R0U                        Tracy
        2  138689  150507  0006641040  A1S4A3IQ2MU7V4        sally sue "sally sue"
        3  138690  150508  0006641040   AZGXZ2UUK6X  Catherine Hallberg "(Kate)"
        4  138691  150509  0006641040  A3CMRKGE0P909G                       Teresa

           HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
        0                     0                       0      1   939340800
        1                     1                       1      1  1194739200
        2                     1                       1      1  1191456000
        3                     1                       1      1  1076025600
        4                     3                       4      1  1018396800

                                      Summary  \
        0                EVERY book is educational
        1  Love the book, miss the hard cover version
        2                 chicken soup with rice months
        3      a good swingy rhythm for reading aloud
        4             A great way to learn the months
```

```
                                                Text  \
0  this witty little book makes my son laugh at l...   
1  I grew up reading these Sendak books, and watc...   
2  This is a fun way for children to learn their ...   
3  This is a great little book to read aloud- it ...   
4  This is a book of poetry about the months of t...   

                                         CleanedText  
0  b'witti littl book make son laugh loud recit c...  
1  b'grew read sendak book watch realli rosi movi...  
2  b'fun way children learn month year learn poem...  
3  b'great littl book read nice rhythm well good ...  
4  b'book poetri month year goe month cute littl ...  
```

```
In [5]: df.describe()
```

```
Out[5]:                 index             Id  HelpfulnessNumerator  \
        count  364106.000000  364106.000000         364106.000000   
        mean   261221.056821  282777.564772              1.738411   
        std    152361.122483  164601.735167              6.716471   
        min         0.000000       1.000000              0.000000   
        25%    129625.250000  140699.250000              0.000000   
        50%    257307.500000  278947.500000              0.000000   
        75%    396338.750000  428557.750000              2.000000   
        max    525813.000000  568454.000000            866.000000   

               HelpfulnessDenominator          Score          Time  
        count           364106.000000  364106.000000  3.641060e+05  
        mean                 2.186231       0.843164  1.296157e+09  
        std                  7.339767       0.363647  4.859821e+07  
        min                  0.000000       0.000000  9.393408e+08  
        25%                  0.000000       1.000000  1.270858e+09  
        50%                  1.000000       1.000000  1.311379e+09  
        75%                  2.000000       1.000000  1.332893e+09  
        max                878.000000       1.000000  1.351210e+09  
```

```
In [6]: df.dtypes
```

```
Out[6]: index                      int64
        Id                         int64
        ProductId                 object
        UserId                    object
        ProfileName               object
        HelpfulnessNumerator       int64
        HelpfulnessDenominator     int64
        Score                      int64
        Time                       int64
        Summary                   object
        Text                      object
        CleanedText               object
        dtype: object
```

```
In [7]: # Split data
        # positive review score, negative review score and review text as seperate dataframes
        df_text = df['CleanedText']
        print(df_text.shape)
        df_text.head()
```

```
(364106,)
```

```
Out[7]: 0    b'witti littl book make son laugh loud recit c...
        1    b'grew read sendak book watch realli rosi movi...
        2    b'fun way children learn month year learn poem...
        3    b'great littl book read nice rhythm well good ...
        4    b'book poetri month year goe month cute littl ...
        Name: CleanedText, dtype: object
```

```
In [8]: def genTSNEGif(std_data, ndp, p, itr_list, file_prefix, closePlt=False):
            '''
            Fuction which genrate t-SNE visualtion for each itr_list using given ndp and p
            Generates a GIF and stores it under '{img_name}.gif'
            Where:
                std_data - Column Standardized Data
                ndp - Number of Data Points to consider in std_data
                p - Perplexity
```

```
        itr_list - List of iterations, each iteration will be a frame in GIF
        file_prefix - Prefix to the name of GIF image
        closePlt - If you do not want to display the generated image in Notebook
    '''
    image_name = '{0}_tsne_ndp_{1}_p_{2}.gif'.format(file_prefix,ndp,p)
    print('No.Of Data Points - {0}, Perplexity - {1}, Iterations - {2}, ImageName - {3}'.format(
            ndp, p, itr_list, image_name))

    # list to hold the frames
    frames = []
    p_data = std_data
    p_labels = final_reviews_scores[0:ndp]

    #print('t-SNE Data Points {0} and its Labels {1}'.format(p_data.shape, p_labels.shape))
    for itr_val in itr_list:
        img_title = '{0}-ndp={1} p={2} itr={3}'.format(file_prefix, ndp, p, itr_val)

        time_start = time.time()

        model = TSNE(n_components=2,random_state=0,perplexity=p,n_iter=itr_val) # ,verbose=2
        tsne_data = model.fit_transform(p_data)
        time_elapsed = time.time() - time_start
        print('{0} ==> t-SNE done! Time elapsed: {1} seconds'.format(img_title, time.time() - time_start))

        tsne_data = np.vstack((tsne_data.T,p_labels)).T
        #print(tsne_data.shape)
        #tsne_data[:4]
        tsne_df = pd.DataFrame(tsne_data,columns=['Dim_1','Dim_2','Score'])
        #tsne_df.head()
        g = sns.FacetGrid(tsne_df,hue='Score',height=10).map(plt.scatter, 'Dim_1', 'Dim_2').add_legend();
        g.fig.suptitle(img_title);
        g.fig.canvas.draw();
        image = np.frombuffer(g.fig.canvas.tostring_rgb(), dtype='uint8')
        image = image.reshape(g.fig.canvas.get_width_height()[::-1] + (3,))
        frames.append(image)

        if (closePlt == True):
            plt.close()

    kwargs_write = {'fps':1.0, 'quantizer':'nq'}
    imageio.mimsave(Path.cwd() / output_dir / image_name, frames, fps=1)

    return
```

1.3 Training Data for Visualization - 3K Points

```
In [9]: # we can't process all 364K revies, selecting a subset of it
        total_data_set_size = 500

        # Create a Balanced dataset having both +ive and -ive reviews
        df_positive_reviews = df[df.Score == 1].sample(int(total_data_set_size/2))
        df_negative_reviews = df[df.Score == 0].sample(int(total_data_set_size/2))

        final_reviews = pd.concat([df_positive_reviews, df_negative_reviews])
        final_reviews_scores = final_reviews['Score']

        print('Shape of Training Data {0}'.format(final_reviews.shape))
        print('Shape of Training Label {0}'.format(final_reviews_scores.shape))
```

```
Shape of Training Data (500, 12)
Shape of Training Label (500,)
```

```
In [10]: final_reviews.head()
```

```
Out[10]:          index      Id   ProductId          UserId  \
         146239   49378   53627  B0016687F2   A56T9S9XCZI80
         200081  250182  271258  B001IZHZGS  A2ZKXGAW9W0C40
         2935    238287  258517  B0000CNTZI  A1Q2AD6OYW04S8
         191631   52669   57206  B001EYUE2A  A2N9B0XGETY0L0
         316739  432232  467431  B004M8KV6Y  A30Q054KF9C1K8

                                    ProfileName  HelpfulnessNumerator  \
         146239                           cathy                     0
         200081  Christopher A. Dowling "tintinet"                     3
         2935            chad-roscoe "chad-roscoe"                     0
         191631                        M. ZELLARS                     1
```

```
316739                      averagepunter                    10

        HelpfulnessDenominator  Score        Time  \
146239                       0      1  1344297600
200081                       3      1  1283644800
2935                         0      1  1293408000
191631                       1      1  1311206400
316739                      12      1  1304553600

                                                Summary  \
146239                            Was delivered in 1 day!!!
200081                   Best gum currently available, IMO.
2935                                 The scent of heaven
191631                                              Coffee
316739  Might just save my Tassimo from the scrapheap ...

                                                   Text  \
146239  I purchased this for my Marine Corps husband t...
200081  The perfect size, nice, long lasting flavor.  ...
2935    I first had this tea at a vegetarian Vietnames...
191631  This is my favorite coffee for my Keurig.  Thi...
316739  The demise of the relationship between Kraft a...

                                            CleanedText
146239  b'purchas marin corp husband put care packag d...
200081  b'perfect size nice long last flavor coat does...
2935    b'first tea vegetarian vietnames restaur order...
191631  b'favorit coffe keurig blend smooth pleas high...
316739  b'demis relationship kraft starbuck threaten m...
```

2 Bag of Words (BoW)

```
In [11]: # Create Vectors
         count_vect = CountVectorizer(ngram_range=(1,2)) # create an instance
         final_counts = count_vect.fit_transform(final_reviews['CleanedText'].values)
         print('Shape of BoW Vectorizer: ', final_counts.get_shape())
         print('Total no.of unique words: ', final_counts.get_shape()[1])

         # Standardize the Data
         standardized_data = StandardScaler().fit_transform(final_counts.toarray().astype(np.float64)) #, with_mean=False
         print('Shape of Standardized data', standardized_data.shape)
```

```
Shape of BoW Vectorizer:  (500, 19908)
Total no.of unique words:  19908
Shape of Standardized data (500, 19908)
```

```
In [12]: genTSNEGif(standardized_data, len(standardized_data), 30, range(1000,3001,1000), 'BoW-std',closePlt=True)
         dense_mat = final_counts.toarray().astype(np.float64)

         for p in range(10, 101, 10):
             genTSNEGif(dense_mat, len(dense_mat), p, range(1000,5001,1000), 'BoW',closePlt=True)
```

```
No.Of Data Points - 500, Perplexity - 30, Iterations - range(1000, 3001, 1000), ImageName - BoW-std_tsne_ndp_500_p_30.gif
BoW-std-ndp=500 p=30 itr=1000 ==> t-SNE done! Time elapsed: 16.620235443115234 seconds
BoW-std-ndp=500 p=30 itr=2000 ==> t-SNE done! Time elapsed: 20.292940855026245 seconds
BoW-std-ndp=500 p=30 itr=3000 ==> t-SNE done! Time elapsed: 21.018538236618042 seconds
No.Of Data Points - 500, Perplexity - 10, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_10.gif
BoW-ndp=500 p=10 itr=1000 ==> t-SNE done! Time elapsed: 14.674026012420654 seconds
BoW-ndp=500 p=10 itr=2000 ==> t-SNE done! Time elapsed: 18.46445941925049 seconds
BoW-ndp=500 p=10 itr=3000 ==> t-SNE done! Time elapsed: 18.840811252593994 seconds
BoW-ndp=500 p=10 itr=4000 ==> t-SNE done! Time elapsed: 18.71005940437317 seconds
BoW-ndp=500 p=10 itr=5000 ==> t-SNE done! Time elapsed: 19.041726112365723 seconds
No.Of Data Points - 500, Perplexity - 20, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_20.gif
BoW-ndp=500 p=20 itr=1000 ==> t-SNE done! Time elapsed: 15.636509895324707 seconds
BoW-ndp=500 p=20 itr=2000 ==> t-SNE done! Time elapsed: 18.317270278930664 seconds
BoW-ndp=500 p=20 itr=3000 ==> t-SNE done! Time elapsed: 18.23276996612549 seconds
BoW-ndp=500 p=20 itr=4000 ==> t-SNE done! Time elapsed: 18.256227016448975 seconds
BoW-ndp=500 p=20 itr=5000 ==> t-SNE done! Time elapsed: 18.391199350357056 seconds
No.Of Data Points - 500, Perplexity - 30, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_30.gif
BoW-ndp=500 p=30 itr=1000 ==> t-SNE done! Time elapsed: 15.787344932556152 seconds
BoW-ndp=500 p=30 itr=2000 ==> t-SNE done! Time elapsed: 20.322612285614014 seconds
BoW-ndp=500 p=30 itr=3000 ==> t-SNE done! Time elapsed: 25.33496403694153 seconds
BoW-ndp=500 p=30 itr=4000 ==> t-SNE done! Time elapsed: 30.575119733810425 seconds
BoW-ndp=500 p=30 itr=5000 ==> t-SNE done! Time elapsed: 33.14903998374939 seconds
No.Of Data Points - 500, Perplexity - 40, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_40.gif
```

```
BoW-ndp=500 p=40 itr=1000 ==> t-SNE done! Time elapsed: 15.516915321350098 seconds
BoW-ndp=500 p=40 itr=2000 ==> t-SNE done! Time elapsed: 20.360701322555542 seconds
BoW-ndp=500 p=40 itr=3000 ==> t-SNE done! Time elapsed: 25.25319814682007 seconds
BoW-ndp=500 p=40 itr=4000 ==> t-SNE done! Time elapsed: 25.988220691680908 seconds
BoW-ndp=500 p=40 itr=5000 ==> t-SNE done! Time elapsed: 26.792847633361816 seconds
No.Of Data Points - 500, Perplexity - 50, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_50.gif
BoW-ndp=500 p=50 itr=1000 ==> t-SNE done! Time elapsed: 16.622135877609253 seconds
BoW-ndp=500 p=50 itr=2000 ==> t-SNE done! Time elapsed: 17.505221128463745 seconds
BoW-ndp=500 p=50 itr=3000 ==> t-SNE done! Time elapsed: 17.501485586166382 seconds
BoW-ndp=500 p=50 itr=4000 ==> t-SNE done! Time elapsed: 17.496704816818237 seconds
BoW-ndp=500 p=50 itr=5000 ==> t-SNE done! Time elapsed: 17.470081329345703 seconds
No.Of Data Points - 500, Perplexity - 60, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_60.gif
BoW-ndp=500 p=60 itr=1000 ==> t-SNE done! Time elapsed: 16.32817268371582 seconds
BoW-ndp=500 p=60 itr=2000 ==> t-SNE done! Time elapsed: 18.026220321655273 seconds
BoW-ndp=500 p=60 itr=3000 ==> t-SNE done! Time elapsed: 17.859854459762573 seconds
BoW-ndp=500 p=60 itr=4000 ==> t-SNE done! Time elapsed: 17.868285417556763 seconds
BoW-ndp=500 p=60 itr=5000 ==> t-SNE done! Time elapsed: 17.86010217666626 seconds
No.Of Data Points - 500, Perplexity - 70, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_70.gif
BoW-ndp=500 p=70 itr=1000 ==> t-SNE done! Time elapsed: 16.97932004928589 seconds
BoW-ndp=500 p=70 itr=2000 ==> t-SNE done! Time elapsed: 17.84239959716797 seconds
BoW-ndp=500 p=70 itr=3000 ==> t-SNE done! Time elapsed: 17.681962251663208 seconds
BoW-ndp=500 p=70 itr=4000 ==> t-SNE done! Time elapsed: 17.63396430015564 seconds
BoW-ndp=500 p=70 itr=5000 ==> t-SNE done! Time elapsed: 17.548508644104004 seconds
No.Of Data Points - 500, Perplexity - 80, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_80.gif
BoW-ndp=500 p=80 itr=1000 ==> t-SNE done! Time elapsed: 16.261298418045044 seconds
BoW-ndp=500 p=80 itr=2000 ==> t-SNE done! Time elapsed: 16.36107325553894 seconds
BoW-ndp=500 p=80 itr=3000 ==> t-SNE done! Time elapsed: 16.28835391998291 seconds
BoW-ndp=500 p=80 itr=4000 ==> t-SNE done! Time elapsed: 16.295461893081665 seconds
BoW-ndp=500 p=80 itr=5000 ==> t-SNE done! Time elapsed: 16.354358911514282 seconds
No.Of Data Points - 500, Perplexity - 90, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_90.gif
BoW-ndp=500 p=90 itr=1000 ==> t-SNE done! Time elapsed: 7.81119179725647 seconds
BoW-ndp=500 p=90 itr=2000 ==> t-SNE done! Time elapsed: 9.406161069869995 seconds
BoW-ndp=500 p=90 itr=3000 ==> t-SNE done! Time elapsed: 9.254249334335327 seconds
BoW-ndp=500 p=90 itr=4000 ==> t-SNE done! Time elapsed: 9.14945125579834 seconds
BoW-ndp=500 p=90 itr=5000 ==> t-SNE done! Time elapsed: 9.265161991119385 seconds
No.Of Data Points - 500, Perplexity - 100, Iterations - range(1000, 5001, 1000), ImageName - BoW_tsne_ndp_500_p_100.gif
BoW-ndp=500 p=100 itr=1000 ==> t-SNE done! Time elapsed: 7.864916086196899 seconds
BoW-ndp=500 p=100 itr=2000 ==> t-SNE done! Time elapsed: 15.110626459121704 seconds
BoW-ndp=500 p=100 itr=3000 ==> t-SNE done! Time elapsed: 22.62155318260193 seconds
BoW-ndp=500 p=100 itr=4000 ==> t-SNE done! Time elapsed: 26.734723329544067 seconds
BoW-ndp=500 p=100 itr=5000 ==> t-SNE done! Time elapsed: 27.06672978401184 seconds
```

3 TFIDF

```
In [13]: # Create Vectors

         tf_idf_vec = TfidfVectorizer(ngram_range=(1,2))
         final_counts = tf_idf_vec.fit_transform(final_reviews['CleanedText'].values)

         #.fit_transform(final_reviews['CleanedText'].values)
         print('Shape of BoW Vectorizer: ', final_counts.get_shape())
         print('Total no.of unique words: ', final_counts.get_shape()[1])

         # Standardize the Data
         standardized_data = StandardScaler().fit_transform(final_counts.toarray().astype(np.float64)) #, with_mean=False
         print('Shape of Standardized data', standardized_data.shape)
```

```
Shape of BoW Vectorizer:  (500, 19908)
Total no.of unique words:  19908
Shape of Standardized data (500, 19908)
```

```
In [16]: genTSNEGif(standardized_data, len(standardized_data), 30, range(1000,6001,1000), 'tfidf-std',closePlt=True)
```

```
No.Of Data Points - 500, Perplexity - 30, Iterations - range(1000, 6001, 1000), ImageName - tfidf-std_tsne_ndp_500_p_30.gif
tfidf-std-ndp=500 p=30 itr=1000 ==> t-SNE done! Time elapsed: 17.061641216278076 seconds
tfidf-std-ndp=500 p=30 itr=2000 ==> t-SNE done! Time elapsed: 22.879101276397705 seconds
tfidf-std-ndp=500 p=30 itr=3000 ==> t-SNE done! Time elapsed: 27.63390612602234 seconds
tfidf-std-ndp=500 p=30 itr=4000 ==> t-SNE done! Time elapsed: 30.708401203155518 seconds
tfidf-std-ndp=500 p=30 itr=5000 ==> t-SNE done! Time elapsed: 30.955078840255737 seconds
tfidf-std-ndp=500 p=30 itr=6000 ==> t-SNE done! Time elapsed: 30.72924494743347 seconds
```

```
In [19]: dense_mat = final_counts.toarray().astype(np.float64)
         for p in range(10, 61, 10):
             genTSNEGif(dense_mat, len(dense_mat), p, range(1000,6001,1000), 'tfidf',closePlt=True)
```

```
No.Of Data Points - 500, Perplexity - 10, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_10.gif
tfidf-ndp=500 p=10 itr=1000 ==> t-SNE done! Time elapsed: 15.436394453048706 seconds
tfidf-ndp=500 p=10 itr=2000 ==> t-SNE done! Time elapsed: 18.523593187332153 seconds
tfidf-ndp=500 p=10 itr=3000 ==> t-SNE done! Time elapsed: 22.647663116455078 seconds
tfidf-ndp=500 p=10 itr=4000 ==> t-SNE done! Time elapsed: 26.716687202453613 seconds
tfidf-ndp=500 p=10 itr=5000 ==> t-SNE done! Time elapsed: 30.898597717285156 seconds
tfidf-ndp=500 p=10 itr=6000 ==> t-SNE done! Time elapsed: 34.807520389556885 seconds
No.Of Data Points - 500, Perplexity - 20, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_20.gif
tfidf-ndp=500 p=20 itr=1000 ==> t-SNE done! Time elapsed: 14.992748975753784 seconds
tfidf-ndp=500 p=20 itr=2000 ==> t-SNE done! Time elapsed: 18.98304295539856 seconds
tfidf-ndp=500 p=20 itr=3000 ==> t-SNE done! Time elapsed: 22.846832513809204 seconds
tfidf-ndp=500 p=20 itr=4000 ==> t-SNE done! Time elapsed: 27.003561973571777 seconds
tfidf-ndp=500 p=20 itr=5000 ==> t-SNE done! Time elapsed: 30.341565370559692 seconds
tfidf-ndp=500 p=20 itr=6000 ==> t-SNE done! Time elapsed: 30.42572259902954 seconds
No.Of Data Points - 500, Perplexity - 30, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_30.gif
tfidf-ndp=500 p=30 itr=1000 ==> t-SNE done! Time elapsed: 15.189840078353882 seconds
tfidf-ndp=500 p=30 itr=2000 ==> t-SNE done! Time elapsed: 19.57518768310547 seconds
tfidf-ndp=500 p=30 itr=3000 ==> t-SNE done! Time elapsed: 24.1771137714386 seconds
tfidf-ndp=500 p=30 itr=4000 ==> t-SNE done! Time elapsed: 25.186442375183105 seconds
tfidf-ndp=500 p=30 itr=5000 ==> t-SNE done! Time elapsed: 25.11744260787964 seconds
tfidf-ndp=500 p=30 itr=6000 ==> t-SNE done! Time elapsed: 25.24114465713501 seconds
No.Of Data Points - 500, Perplexity - 40, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_40.gif
tfidf-ndp=500 p=40 itr=1000 ==> t-SNE done! Time elapsed: 15.80448317527771 seconds
tfidf-ndp=500 p=40 itr=2000 ==> t-SNE done! Time elapsed: 19.980273485183716 seconds
tfidf-ndp=500 p=40 itr=3000 ==> t-SNE done! Time elapsed: 21.159300565719604 seconds
tfidf-ndp=500 p=40 itr=4000 ==> t-SNE done! Time elapsed: 20.981527090072632 seconds
tfidf-ndp=500 p=40 itr=5000 ==> t-SNE done! Time elapsed: 20.87619686126709 seconds
tfidf-ndp=500 p=40 itr=6000 ==> t-SNE done! Time elapsed: 21.330782413482666 seconds
No.Of Data Points - 500, Perplexity - 50, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_50.gif
tfidf-ndp=500 p=50 itr=1000 ==> t-SNE done! Time elapsed: 16.118670225143433 seconds
tfidf-ndp=500 p=50 itr=2000 ==> t-SNE done! Time elapsed: 18.91942524909973 seconds
tfidf-ndp=500 p=50 itr=3000 ==> t-SNE done! Time elapsed: 18.39323902130127 seconds
tfidf-ndp=500 p=50 itr=4000 ==> t-SNE done! Time elapsed: 18.375523567199707 seconds
tfidf-ndp=500 p=50 itr=5000 ==> t-SNE done! Time elapsed: 18.36286687850952 seconds
tfidf-ndp=500 p=50 itr=6000 ==> t-SNE done! Time elapsed: 18.451897859573364 seconds
No.Of Data Points - 500, Perplexity - 60, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_60.gif
tfidf-ndp=500 p=60 itr=1000 ==> t-SNE done! Time elapsed: 16.70004892349243 seconds
tfidf-ndp=500 p=60 itr=2000 ==> t-SNE done! Time elapsed: 20.588300704956055 seconds
tfidf-ndp=500 p=60 itr=3000 ==> t-SNE done! Time elapsed: 23.162419319152832 seconds
tfidf-ndp=500 p=60 itr=4000 ==> t-SNE done! Time elapsed: 23.251046895980835 seconds
tfidf-ndp=500 p=60 itr=5000 ==> t-SNE done! Time elapsed: 23.18180251121521 seconds
tfidf-ndp=500 p=60 itr=6000 ==> t-SNE done! Time elapsed: 23.275164365768433 seconds
No.Of Data Points - 500, Perplexity - 70, Iterations - range(1000, 6001, 1000), ImageName - tfidf_tsne_ndp_500_p_70.gif
tfidf-ndp=500 p=70 itr=1000 ==> t-SNE done! Time elapsed: 16.955317735671997 seconds
```

```
        ---------------------------------------------------------------------------

        KeyboardInterrupt                         Traceback (most recent call last)

        <ipython-input-19-01fa74dd4243> in <module>
          1 dense_mat = final_counts.toarray().astype(np.float64)
          2 for p in range(10, 101, 10):
    ----> 3     genTSNEGif(dense_mat, len(dense_mat), p, range(1000,6001,1000), 'tfidf',closePlt=True)


        <ipython-input-8-0034bdadfe5a> in genTSNEGif(std_data, ndp, p, itr_list, file_prefix, closePlt)
         27
         28         model = TSNE(n_components=2,random_state=0,perplexity=p,n_iter=itr_val) # ,verbose=2
    ---> 29         tsne_data = model.fit_transform(p_data)
         30         time_elapsed = time.time() - time_start
         31         print('{0} ==> t-SNE done! Time elapsed: {1} seconds'.format(img_title, time.time() - time_start))


        ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/manifold/t_sne.py in fit_transform(self, X, y)
        892             Embedding of the training data in low-dimensional space.
        893         """
    --> 894         embedding = self._fit(X)
        895         self.embedding_ = embedding
        896         return self.embedding_


        ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/manifold/t_sne.py in _fit(self, X, skip_num_points)
        759             t0 = time()
        760             distances_nn, neighbors_nn = knn.kneighbors(
--> 761                 None, n_neighbors=k)
    762             duration = time() - t0
    763             if self.verbose:


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py in kneighbors(self, X, n_neighbors, return_distance)
    441                 delayed_query(
    442                     X[s], n_neighbors, return_distance)
--> 443                 for s in gen_even_slices(X.shape[0], n_jobs)
    444             )
    445         else:


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in __call__(self, iterable)
    981             # remaining jobs.
    982             self._iterating = False
--> 983             if self.dispatch_one_batch(iterator):
    984                 self._iterating = self._original_iterator is not None
    985


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in dispatch_one_batch(self, iterator)
    823                 return False
    824             else:
--> 825                 self._dispatch(tasks)
    826                 return True
    827


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in _dispatch(self, batch)
    780         with self._lock:
    781             job_idx = len(self._jobs)
--> 782             job = self._backend.apply_async(batch, callback=cb)
    783             # A job can complete so quickly than its callback is
    784             # called before we get here, causing self._jobs to


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/_parallel_backends.py in apply_async(self, func, callba
    180     def apply_async(self, func, callback=None):
    181         """Schedule a func to be run"""
--> 182         result = ImmediateResult(func)
    183         if callback:
    184             callback(result)


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/_parallel_backends.py in __init__(self, batch)
    543         # Don't delay the application, to avoid keeping the input
    544         # arguments in memory
--> 545         self.results = batch()
    546
    547     def get(self):


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in __call__(self)
    259         with parallel_backend(self._backend):
    260             return [func(*args, **kwargs)
--> 261                     for func, args, kwargs in self.items]
    262
    263     def __len__(self):


    ~/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in <listcomp>(.0)
    259         with parallel_backend(self._backend):
    260             return [func(*args, **kwargs)
--> 261                     for func, args, kwargs in self.items]
    262
    263     def __len__(self):


    KeyboardInterrupt:
```

4 Word2Vec

I am creating vectors having 50 dimensions. Just a random value, not inherent calculation I made on this size decision.

```
In [20]: # Create List arry for creating own W2V
         list_of_sent = []
         for sent in final_reviews['CleanedText'].values:
             list_of_sent.append(sent.decode("utf-8").split())

         print(final_reviews.CleanedText.values[0])
         print(len(list_of_sent), list_of_sent[0])
```

```
b'purchas marin corp husband put care packag deploy afghanistan deliv day couldnt believ thought email made mistak nope even hour later doors
500 ['purchas', 'marin', 'corp', 'husband', 'put', 'care', 'packag', 'deploy', 'afghanistan', 'deliv', 'day', 'couldnt', 'believ', 'thought',
```

```
In [21]: # Required dimension
         w2v_d = 50

         # Considering words that are occured atleast 5 times in the corpus
         w2v_model = Word2Vec(list_of_sent, min_count=5, size=w2v_d, workers=4)

         w2v_words = list(w2v_model.wv.vocab)
         print("number of words that occured minimum 5 times : ",len(w2v_words))
         print("sample words ", w2v_words[0:50])
```

```
number of words that occured minimum 5 times :  884
sample words  ['purchas', 'husband', 'put', 'care', 'packag', 'deliv', 'day', 'couldnt', 'believ', 'thought', 'made', 'even', 'hour', 'later'
```

4.1 Avg-W2V

```
In [22]: # Computing average w2v for each review in selected training dataset
         review_vectors = []
         for sent in tqdm(list_of_sent, ascii=True):
             sent_vec = np.zeros(w2v_d) # array to hold the vectors. Initially assuming no vectors in this review
             no_of_words_in_review = 0 # number of words with valid vector in this review

             # count all the words (that are in w2v model) and take average
             for word in sent:
                 if word in w2v_words:
                     vec = w2v_model.wv[word]
                     sent_vec += vec
                     no_of_words_in_review += 1
             if no_of_words_in_review != 0:
                 sent_vec /= no_of_words_in_review
             review_vectors.append(sent_vec)

         print(len(review_vectors))
         print(len(review_vectors[0]))
```

```
100%|##########| 500/500 [00:00<00:00, 1062.10it/s]

500
50
```

```
In [23]: # t-SNE using Average Word2Vec

         #genTSNEGif(review_vectors, len(review_vectors), 30, range(1000,10001,1000), 'avg-w2v')

         for p in range(10, 101, 10):
             genTSNEGif(review_vectors, len(review_vectors), p, range(1000,5001,1000), 'avg-w2v',closePlt=True)
```

```
No.Of Data Points - 500, Perplexity - 10, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_10.gif
avg-w2v-ndp=500 p=10 itr=1000 ==> t-SNE done! Time elapsed: 2.695676803588867 seconds
avg-w2v-ndp=500 p=10 itr=2000 ==> t-SNE done! Time elapsed: 4.945517301559448 seconds
avg-w2v-ndp=500 p=10 itr=3000 ==> t-SNE done! Time elapsed: 7.169031143188477 seconds
avg-w2v-ndp=500 p=10 itr=4000 ==> t-SNE done! Time elapsed: 9.788629531860352 seconds
avg-w2v-ndp=500 p=10 itr=5000 ==> t-SNE done! Time elapsed: 11.720644235610962 seconds
No.Of Data Points - 500, Perplexity - 20, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_20.gif
avg-w2v-ndp=500 p=20 itr=1000 ==> t-SNE done! Time elapsed: 2.78783917427063 seconds
avg-w2v-ndp=500 p=20 itr=2000 ==> t-SNE done! Time elapsed: 5.255230665206909 seconds
avg-w2v-ndp=500 p=20 itr=3000 ==> t-SNE done! Time elapsed: 7.647207021713257 seconds
avg-w2v-ndp=500 p=20 itr=4000 ==> t-SNE done! Time elapsed: 8.59337592124939 seconds
avg-w2v-ndp=500 p=20 itr=5000 ==> t-SNE done! Time elapsed: 8.381384134292603 seconds
No.Of Data Points - 500, Perplexity - 30, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_30.gif
avg-w2v-ndp=500 p=30 itr=1000 ==> t-SNE done! Time elapsed: 3.1816389560699463 seconds
```

```
avg-w2v-ndp=500 p=30 itr=2000 ==> t-SNE done! Time elapsed: 6.156204700469971 seconds
avg-w2v-ndp=500 p=30 itr=3000 ==> t-SNE done! Time elapsed: 9.112237215042114 seconds
avg-w2v-ndp=500 p=30 itr=4000 ==> t-SNE done! Time elapsed: 11.321281671524048 seconds
avg-w2v-ndp=500 p=30 itr=5000 ==> t-SNE done! Time elapsed: 11.349352598190308 seconds
No.Of Data Points - 500, Perplexity - 40, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_40.gif
avg-w2v-ndp=500 p=40 itr=1000 ==> t-SNE done! Time elapsed: 3.2230210304260254 seconds
avg-w2v-ndp=500 p=40 itr=2000 ==> t-SNE done! Time elapsed: 6.201868057250977 seconds
avg-w2v-ndp=500 p=40 itr=3000 ==> t-SNE done! Time elapsed: 9.366081237792969 seconds
avg-w2v-ndp=500 p=40 itr=4000 ==> t-SNE done! Time elapsed: 12.70339322090149 seconds
avg-w2v-ndp=500 p=40 itr=5000 ==> t-SNE done! Time elapsed: 16.11518144607544 seconds
No.Of Data Points - 500, Perplexity - 50, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_50.gif
avg-w2v-ndp=500 p=50 itr=1000 ==> t-SNE done! Time elapsed: 3.585568428039551 seconds
avg-w2v-ndp=500 p=50 itr=2000 ==> t-SNE done! Time elapsed: 6.977082014083862 seconds
avg-w2v-ndp=500 p=50 itr=3000 ==> t-SNE done! Time elapsed: 10.204594135284424 seconds
avg-w2v-ndp=500 p=50 itr=4000 ==> t-SNE done! Time elapsed: 11.217078447341919 seconds
avg-w2v-ndp=500 p=50 itr=5000 ==> t-SNE done! Time elapsed: 11.249882936477661 seconds
No.Of Data Points - 500, Perplexity - 60, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_60.gif
avg-w2v-ndp=500 p=60 itr=1000 ==> t-SNE done! Time elapsed: 4.143369913101196 seconds
avg-w2v-ndp=500 p=60 itr=2000 ==> t-SNE done! Time elapsed: 6.533319473266602 seconds
avg-w2v-ndp=500 p=60 itr=3000 ==> t-SNE done! Time elapsed: 6.431298732757568 seconds
avg-w2v-ndp=500 p=60 itr=4000 ==> t-SNE done! Time elapsed: 6.622186183929443 seconds
avg-w2v-ndp=500 p=60 itr=5000 ==> t-SNE done! Time elapsed: 6.493930101394653 seconds
No.Of Data Points - 500, Perplexity - 70, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_70.gif
avg-w2v-ndp=500 p=70 itr=1000 ==> t-SNE done! Time elapsed: 4.291093111038208 seconds
avg-w2v-ndp=500 p=70 itr=2000 ==> t-SNE done! Time elapsed: 5.5432984828948975 seconds
avg-w2v-ndp=500 p=70 itr=3000 ==> t-SNE done! Time elapsed: 5.562952518463135 seconds
avg-w2v-ndp=500 p=70 itr=4000 ==> t-SNE done! Time elapsed: 5.520572900772095 seconds
avg-w2v-ndp=500 p=70 itr=5000 ==> t-SNE done! Time elapsed: 5.519697666168213 seconds
No.Of Data Points - 500, Perplexity - 80, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_80.gif
avg-w2v-ndp=500 p=80 itr=1000 ==> t-SNE done! Time elapsed: 4.289063215255737 seconds
avg-w2v-ndp=500 p=80 itr=2000 ==> t-SNE done! Time elapsed: 5.180687189102173 seconds
avg-w2v-ndp=500 p=80 itr=3000 ==> t-SNE done! Time elapsed: 5.191259860992432 seconds
avg-w2v-ndp=500 p=80 itr=4000 ==> t-SNE done! Time elapsed: 5.221221446990967 seconds
avg-w2v-ndp=500 p=80 itr=5000 ==> t-SNE done! Time elapsed: 5.2324018478393555 seconds
No.Of Data Points - 500, Perplexity - 90, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_90.gif


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=90 itr=1000 ==> t-SNE done! Time elapsed: 3.7089908123016357 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=90 itr=2000 ==> t-SNE done! Time elapsed: 3.550213098526001 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=90 itr=3000 ==> t-SNE done! Time elapsed: 3.5614142417907715 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=90 itr=4000 ==> t-SNE done! Time elapsed: 3.559083938598633 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=90 itr=5000 ==> t-SNE done! Time elapsed: 3.5471158027648926 seconds
No.Of Data Points - 500, Perplexity - 100, Iterations - range(1000, 5001, 1000), ImageName - avg-w2v_tsne_ndp_500_p_100.gif


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=100 itr=1000 ==> t-SNE done! Time elapsed: 3.695096015930176 seconds
```

```
/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=100 itr=2000 ==> t-SNE done! Time elapsed: 3.6859006881713867 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=100 itr=3000 ==> t-SNE done! Time elapsed: 3.707569122314453 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=100 itr=4000 ==> t-SNE done! Time elapsed: 3.692902088165283 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


avg-w2v-ndp=500 p=100 itr=5000 ==> t-SNE done! Time elapsed: 3.7627687454223633 seconds
```

4.2 TFIDF Weighted W2V

Computing tfidf weighted w2v over the selected training dataset

```
In [24]: # Create tf-idf vector matrix
         tf_idf_model = TfidfVectorizer(ngram_range=(1,2))
         tf_idf_matrix = tf_idf_model.fit_transform(final_reviews['CleanedText'].values)

         # Create dictionary having words (features) as keys, its tf-idf values as values
         tf_idf_dict = dict(zip(tf_idf_model.get_feature_names(), list(tf_idf_model.idf_)))
         len(tf_idf_dict)
```

```
Out[24]: 19908
```

```
In [25]: tf_idf_feat = tf_idf_model.get_feature_names()

         # Computing tf-idf weighted w2v for each review in selected training dataset
         review_vectors = []
         for sent in tqdm(list_of_sent, ascii=True):
             sent_vec = np.zeros(w2v_d) # array to hold the vectors
             no_of_words_in_review = 0 # number of words with valid vector in this review

             # count all the words (that are in w2v model) and take average
             for word in sent:
                 if word in w2v_words:
                     vec = w2v_model.wv[word]
                     # calculate tf-idf weighted w2v value for this word
                     tf_idf = tf_idf_dict[word] * (sent.count(word)/len(sent))
                     sent_vec += (vec * tf_idf)
                     no_of_words_in_review += 1
             if no_of_words_in_review != 0:
                 sent_vec /= no_of_words_in_review
             review_vectors.append(sent_vec)

         print(len(review_vectors))
         print(len(review_vectors[0]))
```

```
100%|##########| 500/500 [00:00<00:00, 831.26it/s]

500
50
```

```
In [26]: # t-SNE using tf-idf weighted s2v

         for p in range(10, 101, 10):
             genTSNEGif(review_vectors, len(review_vectors), p, range(1000,5001,1000), 'tfidf-weighted-w2v',closePlt=True)
```

```
No.Of Data Points - 500, Perplexity - 10, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_10.gif
tfidf-weighted-w2v-ndp=500 p=10 itr=1000 ==> t-SNE done! Time elapsed: 2.5075390338897705 seconds
tfidf-weighted-w2v-ndp=500 p=10 itr=2000 ==> t-SNE done! Time elapsed: 4.713393211364746 seconds
tfidf-weighted-w2v-ndp=500 p=10 itr=3000 ==> t-SNE done! Time elapsed: 6.844162702560425 seconds
tfidf-weighted-w2v-ndp=500 p=10 itr=4000 ==> t-SNE done! Time elapsed: 9.025461196899414 seconds
tfidf-weighted-w2v-ndp=500 p=10 itr=5000 ==> t-SNE done! Time elapsed: 11.266832113265991 seconds
No.Of Data Points - 500, Perplexity - 20, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_20.gif
tfidf-weighted-w2v-ndp=500 p=20 itr=1000 ==> t-SNE done! Time elapsed: 2.715935468673706 seconds
tfidf-weighted-w2v-ndp=500 p=20 itr=2000 ==> t-SNE done! Time elapsed: 4.954932928085327 seconds
tfidf-weighted-w2v-ndp=500 p=20 itr=3000 ==> t-SNE done! Time elapsed: 7.181711435317993 seconds
tfidf-weighted-w2v-ndp=500 p=20 itr=4000 ==> t-SNE done! Time elapsed: 9.274171590805054 seconds
tfidf-weighted-w2v-ndp=500 p=20 itr=5000 ==> t-SNE done! Time elapsed: 9.258382320404053 seconds
No.Of Data Points - 500, Perplexity - 30, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_30.gif
tfidf-weighted-w2v-ndp=500 p=30 itr=1000 ==> t-SNE done! Time elapsed: 3.1726553440093994 seconds
tfidf-weighted-w2v-ndp=500 p=30 itr=2000 ==> t-SNE done! Time elapsed: 6.342355728149414 seconds
tfidf-weighted-w2v-ndp=500 p=30 itr=3000 ==> t-SNE done! Time elapsed: 9.66136884689331 seconds
tfidf-weighted-w2v-ndp=500 p=30 itr=4000 ==> t-SNE done! Time elapsed: 12.984240293502808 seconds
tfidf-weighted-w2v-ndp=500 p=30 itr=5000 ==> t-SNE done! Time elapsed: 13.600743770599365 seconds
No.Of Data Points - 500, Perplexity - 40, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_40.gif
tfidf-weighted-w2v-ndp=500 p=40 itr=1000 ==> t-SNE done! Time elapsed: 3.2050914764404297 seconds
tfidf-weighted-w2v-ndp=500 p=40 itr=2000 ==> t-SNE done! Time elapsed: 6.1780688762664795 seconds
tfidf-weighted-w2v-ndp=500 p=40 itr=3000 ==> t-SNE done! Time elapsed: 8.903522253036499 seconds
tfidf-weighted-w2v-ndp=500 p=40 itr=4000 ==> t-SNE done! Time elapsed: 9.247819900512695 seconds
tfidf-weighted-w2v-ndp=500 p=40 itr=5000 ==> t-SNE done! Time elapsed: 9.498867511749268 seconds
No.Of Data Points - 500, Perplexity - 50, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_50.gif
tfidf-weighted-w2v-ndp=500 p=50 itr=1000 ==> t-SNE done! Time elapsed: 3.792397975921631 seconds
tfidf-weighted-w2v-ndp=500 p=50 itr=2000 ==> t-SNE done! Time elapsed: 7.169574737548828 seconds
tfidf-weighted-w2v-ndp=500 p=50 itr=3000 ==> t-SNE done! Time elapsed: 10.625329494476318 seconds
tfidf-weighted-w2v-ndp=500 p=50 itr=4000 ==> t-SNE done! Time elapsed: 12.24459195137024 seconds
tfidf-weighted-w2v-ndp=500 p=50 itr=5000 ==> t-SNE done! Time elapsed: 11.952564239501953 seconds
No.Of Data Points - 500, Perplexity - 60, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_60.gif
tfidf-weighted-w2v-ndp=500 p=60 itr=1000 ==> t-SNE done! Time elapsed: 3.899441719055176 seconds
tfidf-weighted-w2v-ndp=500 p=60 itr=2000 ==> t-SNE done! Time elapsed: 7.253939151763916 seconds
tfidf-weighted-w2v-ndp=500 p=60 itr=3000 ==> t-SNE done! Time elapsed: 7.49826192855835 seconds
tfidf-weighted-w2v-ndp=500 p=60 itr=4000 ==> t-SNE done! Time elapsed: 7.470948934555054 seconds
tfidf-weighted-w2v-ndp=500 p=60 itr=5000 ==> t-SNE done! Time elapsed: 7.301303386688232 seconds
No.Of Data Points - 500, Perplexity - 70, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_70.gif
tfidf-weighted-w2v-ndp=500 p=70 itr=1000 ==> t-SNE done! Time elapsed: 4.374642372131348 seconds
tfidf-weighted-w2v-ndp=500 p=70 itr=2000 ==> t-SNE done! Time elapsed: 5.161747455596924 seconds
tfidf-weighted-w2v-ndp=500 p=70 itr=3000 ==> t-SNE done! Time elapsed: 4.979130029678345 seconds
tfidf-weighted-w2v-ndp=500 p=70 itr=4000 ==> t-SNE done! Time elapsed: 4.946432590484619 seconds
tfidf-weighted-w2v-ndp=500 p=70 itr=5000 ==> t-SNE done! Time elapsed: 5.052804946899414 seconds
No.Of Data Points - 500, Perplexity - 80, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_80.gif
tfidf-weighted-w2v-ndp=500 p=80 itr=1000 ==> t-SNE done! Time elapsed: 4.954975128173828 seconds
tfidf-weighted-w2v-ndp=500 p=80 itr=2000 ==> t-SNE done! Time elapsed: 9.083971738815308 seconds
tfidf-weighted-w2v-ndp=500 p=80 itr=3000 ==> t-SNE done! Time elapsed: 8.615956544876099 seconds
tfidf-weighted-w2v-ndp=500 p=80 itr=4000 ==> t-SNE done! Time elapsed: 8.18886113166809 seconds
tfidf-weighted-w2v-ndp=500 p=80 itr=5000 ==> t-SNE done! Time elapsed: 8.209148645401001 seconds
No.Of Data Points - 500, Perplexity - 90, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_90.gif
```

```
/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind
```

```
tfidf-weighted-w2v-ndp=500 p=90 itr=1000 ==> t-SNE done! Time elapsed: 4.156220197677612 seconds
```

```
/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind
```

```
tfidf-weighted-w2v-ndp=500 p=90 itr=2000 ==> t-SNE done! Time elapsed: 4.141205072402954 seconds
```

```
/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind
```

```
tfidf-weighted-w2v-ndp=500 p=90 itr=3000 ==> t-SNE done! Time elapsed: 4.069101095199585 seconds
```

```
/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind
```

```
tfidf-weighted-w2v-ndp=500 p=90 itr=4000 ==> t-SNE done! Time elapsed: 4.1565775871276855 seconds
```

```
/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


tfidf-weighted-w2v-ndp=500 p=90 itr=5000 ==> t-SNE done! Time elapsed: 4.461912155151367 seconds
No.Of Data Points - 500, Perplexity - 100, Iterations - range(1000, 5001, 1000), ImageName - tfidf-weighted-w2v_tsne_ndp_500_p_100.gif


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


tfidf-weighted-w2v-ndp=500 p=100 itr=1000 ==> t-SNE done! Time elapsed: 5.4860756397247314 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


tfidf-weighted-w2v-ndp=500 p=100 itr=2000 ==> t-SNE done! Time elapsed: 5.561805248260498 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


tfidf-weighted-w2v-ndp=500 p=100 itr=3000 ==> t-SNE done! Time elapsed: 5.460094928741455 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


tfidf-weighted-w2v-ndp=500 p=100 itr=4000 ==> t-SNE done! Time elapsed: 5.468733072280884 seconds


/home/shin/anaconda3/envs/ml_study/lib/python3.6/site-packages/sklearn/neighbors/base.py:316: RuntimeWarning: invalid value encountered in sq
  result = np.sqrt(dist[sample_range, neigh_ind]), neigh_ind


tfidf-weighted-w2v-ndp=500 p=100 itr=5000 ==> t-SNE done! Time elapsed: 5.332413673400879 seconds
```