2. Post_Clean_K-Means_Hier_DBScan

October 31, 2018

1 K-Means, Hierarchical & DBScan on Amazon Reviews (Part II)

1.1 Data Source:

The preprocessing step has produced final.sqlite file after doing the data preparation & cleaning. The review text is now devoid of punctuations, HTML markups and stop words.

1.2 Objective:

To find meaningful clusters using **unsupervised clustering algorithms like K-Means, Hierarchical & DBScan** on the review dataset. The **polarity of the review is removed from the input dataset,** so that the clustering would happen just on the review text given.

4 standard featurizations are used, namely **BoW, tf-idf, W2V and tf-idf weighted W2V featurizations**. Cross validaton or test metrics in supervised algorithm cannot be used as there is no test data. Instead, **random samples from clusters formed are analyzed manually and a conclusion should be arrived at.**

1.3 At a glance:

The **elbow method is used to find the right # of clusters of K-Means. The minPoints for DBScan is set as double the number of dimension of W2V vectors**, as a rule of thumb. The **Eps value is calculated using KNN distance plots.** The point at which the slope of the plot is higher than a set threshold is taken as Eps value.

The hierarchical clustering algorithm is run with different 'k' values & DBScan also is executed with different Eps values. so that the impact of change in hyperparameters can be well understood.

2 Preprocessed Data Loading

```
In [18]: #loading libraries for LR
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         from sklearn.model_selection import train_test_split
         from sklearn.neighbors import KNeighborsClassifier
         # from sklearn.cross_validation import cross_val_score
         from sklearn.model_selection import cross_val_score
```

```
from collections import Counter
from sklearn.metrics import accuracy_score
#from sklearn import cross_validation
from sklearn.cluster import KMeans

#loading libraries for scikit learn, nlp, db, plot and matrix.
import sqlite3
import pdb
import pandas as pd
import numpy as np
import nltk
import string
import collections
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors

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 import tree

from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer


# using the SQLite Table to read data.
con = sqlite3.connect('./final.sqlite')

#filtering only positive and negative reviews i.e.
# not taking into consideration those reviews with Score=3
final = pd.read_sql_query("""
SELECT *
FROM Reviews
""", con)

print(final.head(2))
```

```
    index      Id   ProductId          UserId      ProfileName  \
0  138706  150524  0006641040   ACITT7DI6IDDL  shari zychinski
1  138688  150506  0006641040  A2IW4PEEKO2ROU            Tracy
```

```
   HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
0                     0                       0  positive   939340800
1                     1                       1  positive  1194739200

                                  Summary  \
0                   EVERY book is educational
1  Love the book, miss the hard cover version


0  this witty little book makes my son laugh at loud. i recite it in the car as we're driving a
1  I grew up reading these Sendak books, and watching the Really Rosie movie that incorporates


0  b'witti littl book make son laugh loud recit car drive along alway sing refrain hes learn wh
1                                        b'grew read sendak book watch realli rosi movi inc
```

3 Random Sampling & Time Based Slicing

In [19]:
```
# To randomly sample the data and sort based on time before doing train/ test split.
# The slicing into train & test data is done thereafter.

# hierarchical cannot handle more points coz
# of high time and space complexity
num_points_kmeans = 100000
num_points_hierarchical = 5000

# used to format headings
bold = '\033[1m'
end = '\033[0m'

# ignore yi's for unsupervised learning
d_unsampled = final.drop(['Score'], axis=1)

# dataset for kmeans & DBSCAN clustering is d_kmeans
# you can use random_state for reproducibility
d_kmeans = d_unsampled.sample(n=num_points_kmeans, random_state=2)

# dataset for hierarchical
d_hierarchical = d_unsampled.sample(n=num_points_hierarchical, random_state=5)
```

4 Custom Defined Functions

3 user defined functions are written to

a) Compute Mean Neighbourhood Distance & Distance Plot

b) Elbow Method to find K

c) Analyze the Clusters function.

5 a) Compute Mean Neighbourhood Distance & Distance Plot

In [24]:
```
# For DBSCAN: Methods used to calculate the mean of the neighbors distances &
# to calculate where the slope of the kNNdistPlot is higher than threshold
# Got base src from https://github.com/vincewide/ML_scheduler/blob/master/DBSCAN.py
# Modified to the fit in the requirements.

def Get_distanceMean(points,minPts,previous_distanceMean):

    """
    Method used to calculate the mean of the neighbors distances

    :param points: List containing the training-points you want to use
    :param minPts: Minimum number of points to be considered a cluster
    :param previous_distanceMean: The previous mean of the distances

    :return: Average distance between the points

    """

    if (minPts < len(points)):

        nbrs = NearestNeighbors(n_neighbors=minPts).fit(points)
        distances, indices = nbrs.kneighbors(points)
        d_mean = distances.mean()
        return d_mean

    else:
        return previous_distanceMean




def KNNdist_plot(points,minPts):

    """

    Calculate where the slope of the kNNdistPlot is higher than a user-defined
       value while plotting the K-NN distance
           with respect to the amount of training data

   :param points: List containing the points you want to use
   :param minPts: Minimum number of points to be considered a cluster
```

```
    :return: The most optimal parameter-values i.e Knee point values

    """

    epsPlot = []
    current_distanceMean = previous_distanceMean = 0
    knee_value = knee_found = 0

    for i in range (0,len(points),5):

        current_distanceMean = Get_distanceMean(points[i:],
                                                minPts,previous_distanceMean)
        df = current_distanceMean - previous_distanceMean

        if (df > 0.02 and i > 1 and knee_found == 0):
            knee_value = current_distanceMean
            knee_found = 1
            n_trainingData = i

        epsPlot.append( [i,current_distanceMean] )
        previous_distanceMean = current_distanceMean


    #Plot the kNNdistPlot
    for i in range(0, len(epsPlot)):
                plt.scatter(epsPlot[i][0],epsPlot[i][1],c='r',s=3,marker='o')

    plt.axhline(y=knee_value, color='g', linestyle='-')
    plt.axvline(x=n_trainingData , color='g', linestyle='-')
#     plt.title(object_name)
    plt.show()

    print("Knee value: x=" + str(n_trainingData) + " , y=" + str(knee_value))

    return knee_value
```

6 b) Elbow Method to find K

In [25]:
```
# To find K of K-means using elbow method.
# This fn plots the loss vs k graph to find the elbow point

def findK(d_vect_std):

    sse = {}
    for k in range(2, 20):
        kmeans = KMeans(n_clusters=k, max_iter=300).fit(d_vect_std)
```

```
            print(bold+"\nGroup Counter in Cluster %d is as follows:" % (k) +end)
            print(collections.Counter(kmeans.labels_))

            # Inertia: Sum of distances of samples to their closest cluster center
            sse[k] = kmeans.inertia_
        plt.figure()
        plt.plot(list(sse.keys()), list(sse.values()))
        plt.xlabel("Number of clusters")
        plt.ylabel("Loss Value")
        plt.show()
```

7 c) Analyze the Clusters

In [26]:
```
# Using elbow method, optimal k is found.
# This function analyze the clusters so formed.

def analyzeClusters(d_labels, k, algo='kmeans'):

    count = collections.Counter(d_labels)
    print("\n")
    print(type(count))
    print(count.items())
    print("cluster size = " + str(len(count.items())))
    k = len(count.items())

    if algo == 'kmeans':
        data = d_kmeans
        cluster_index_start = 1
    elif algo == 'hierarchical':
        data = d_hierarchical
        cluster_index_start = 1
    elif algo == 'dbscan':
        data = d_kmeans #change in last run
        cluster_index_start = 0

    print(bold+"\n*** CLUSTERS FORMED BY %s ALGORITHM is as follows: ***" %algo + end)

    for i in range(cluster_index_start, k+cluster_index_start):
        print("CLUSTER = " + str(i))
        # if point is noise then cluster index will be -1. hence exclude.
        if(count.get(i-1) > 1):

            print(bold+"\nThe Review Text in Cluster %d is as follows:" % (i-1) +end)
            print(data[d_labels == i-1].head(5)['Text'])

        else:
            print("Not enough datapoints to display in this cluster!")
```

8 K-Means & Hierarchical Clustering on BoW

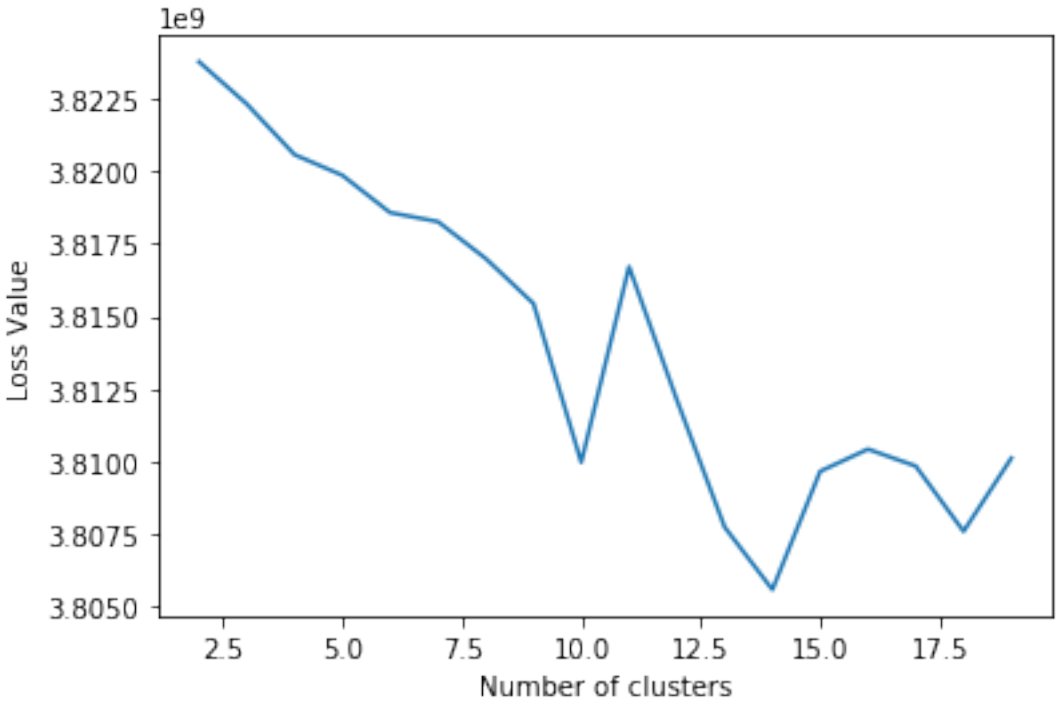
BoW will result in a **sparse matrix with huge number of features** as it creates a feature for each unique word in the review.

For Binary BoW feature representation, CountVectorizer is declared as float, as the values can take non-integer values on further processing.

In [ ]:
```
# BoW Featurisation, Standardisation, Grid Search

from sklearn.random_projection import sparse_random_matrix
from sklearn.preprocessing import StandardScaler

############################################################
### BoW for K-means: Vectorization & Standardization ###
count_vect = CountVectorizer(dtype="float") #in scikit-learn
d_kmeans_vect = count_vect.fit_transform(d_kmeans['CleanedText'].values)
d_kmeans_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_kmeans_vect)
d_kmeans_bow_vect_std = scaler.transform(d_kmeans_vect)

############################################################

### BoW for Hierarchical: Vectorization & Standardization ###
count_vect = CountVectorizer(dtype="float") #in scikit-learn
d_hier_vect = count_vect.fit_transform(d_hierarchical['CleanedText'].values)
d_hier_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_hier_vect)
d_hier_bow_vect_std = scaler.transform(d_hier_vect)

############################################################

## To find the best K for K-means
k = findK(d_kmeans_bow_vect_std)

# Hierarchical Clustering
hierarchichal = AgglomerativeClustering(n_clusters=2).fit(
                                        d_hier_bow_vect_std.toarray())
```

```
Group Counter in Cluster 2 is as follows:
Counter({1: 99711, 0: 289})
Group Counter in Cluster 3 is as follows:
Counter({0: 99998, 1: 1, 2: 1})
Group Counter in Cluster 4 is as follows:
Counter({1: 99993, 0: 5, 2: 1, 3: 1})
```

Group Counter in Cluster 5 is as follows:
Counter({0: 99992, 3: 5, 2: 1, 4: 1, 1: 1})
Group Counter in Cluster 6 is as follows:
Counter({0: 99995, 2: 1, 1: 1, 3: 1, 5: 1, 4: 1})
Group Counter in Cluster 7 is as follows:
Counter({4: 99977, 0: 13, 1: 6, 3: 1, 5: 1, 2: 1, 6: 1})
Group Counter in Cluster 8 is as follows:
Counter({0: 99993, 4: 1, 7: 1, 5: 1, 2: 1, 3: 1, 1: 1, 6: 1})
Group Counter in Cluster 9 is as follows:
Counter({0: 99992, 3: 1, 6: 1, 7: 1, 2: 1, 1: 1, 8: 1, 4: 1, 5: 1})
Group Counter in Cluster 10 is as follows:
Counter({6: 99985, 4: 7, 9: 1, 3: 1, 2: 1, 8: 1, 5: 1, 7: 1, 0: 1, 1: 1})
Group Counter in Cluster 11 is as follows:
Counter({5: 99978, 0: 13, 6: 1, 7: 1, 2: 1, 8: 1, 10: 1, 1: 1, 4: 1, 9: 1, 3: 1})
Group Counter in Cluster 12 is as follows:
Counter({1: 99989, 0: 1, 11: 1, 7: 1, 9: 1, 2: 1, 8: 1, 3: 1, 4: 1, 6: 1, 10: 1, 5: 1})
Group Counter in Cluster 13 is as follows:
Counter({9: 99981, 8: 5, 0: 4, 7: 1, 6: 1, 12: 1, 1: 1, 10: 1, 4: 1, 11: 1, 2: 1, 5: 1, 3: 1})
Group Counter in Cluster 14 is as follows:
Counter({3: 99986, 2: 2, 5: 1, 6: 1, 11: 1, 12: 1, 10: 1, 4: 1, 8: 1, 9: 1, 1: 1, 7: 1, 0: 1, 1
Group Counter in Cluster 15 is as follows:
Counter({0: 99986, 8: 1, 14: 1, 12: 1, 2: 1, 6: 1, 13: 1, 9: 1, 3: 1, 7: 1, 4: 1, 10: 1, 1: 1,
Group Counter in Cluster 16 is as follows:
Counter({0: 99985, 9: 1, 15: 1, 10: 1, 14: 1, 1: 1, 8: 1, 3: 1, 4: 1, 12: 1, 11: 1, 5: 1, 13: 1
Group Counter in Cluster 17 is as follows:
Counter({12: 99739, 4: 243, 1: 2, 0: 2, 9: 2, 7: 1, 16: 1, 8: 1, 15: 1, 13: 1, 3: 1, 5: 1, 11:
Group Counter in Cluster 18 is as follows:
Counter({12: 99975, 0: 9, 5: 1, 4: 1, 7: 1, 10: 1, 16: 1, 15: 1, 11: 1, 13: 1, 17: 1, 6: 1, 3:
Group Counter in Cluster 19 is as follows:
Counter({14: 99705, 6: 272, 1: 6, 0: 2, 10: 1, 7: 1, 15: 1, 11: 1, 12: 1, 9: 1, 13: 1, 18: 1, 4



In [ ]:
```
# from the above elbow plot, k = 10 is found to be optimum

# Analyse review in k clusters.
# k is found using elbow method plot above.

pd.options.display.max_colwidth = 200

# Analyze clusters formed by kmeans clustering
kmeans = KMeans(n_clusters=10, max_iter=300).fit(d_kmeans_bow_vect_std)
analyzeClusters(d_labels=kmeans.labels_, k=10, algo='kmeans')

# Analyze clusters formed by hierarchical clustering
analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')
```

```
<class 'collections.Counter'>
dict_items([(2, 99983), (0, 4), (7, 1), (1, 6), (3, 1), (9, 1), (8, 1), (4, 1), (5, 1), (6, 1)]
cluster size = 10
*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
201895    While this honey is not so versatile in its application as a clover, acacia, or other
37529     Revised 4-2-12<br /><br />I love Stash's Green Tea, and their Chamomile is fantastic
```

```
128330    My middle poodle, Tucker, (the almost 4 year old) is a few lbs. overweight.  Tucker l
197357    We discovered this finishing sauce when a friend of ours from St Louis prepared dinne
Name: Text, dtype: object
CLUSTER = 2
The Review Text in Cluster 1 is as follows:
134296    I started buying these for the office in 2009 until Christmas 2010 when they where no
297363    My work colleagues and I have had a couple of bizarrely negative and expensive experi
228057    My two puppies started off on Solid Gold, but one day while I was picking them up fro
115251    OMG!  Why has my vet been keeping these from me?  My vet's office doesn't keep these
364142    I gave these bulbs to the receptionist at my local elder home that I volunteer at, an
Name: Text, dtype: object
CLUSTER = 3
The Review Text in Cluster 2 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is i
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard'
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
Name: Text, dtype: object
CLUSTER = 4
Not enough datapoints to display in this cluster!
CLUSTER = 5
Not enough datapoints to display in this cluster!
CLUSTER = 6
Not enough datapoints to display in this cluster!
CLUSTER = 7
Not enough datapoints to display in this cluster!
CLUSTER = 8
Not enough datapoints to display in this cluster!
CLUSTER = 9
Not enough datapoints to display in this cluster!
CLUSTER = 10
Not enough datapoints to display in this cluster!


<class 'collections.Counter'>
dict_items([(0, 4999), (1, 1)])
cluster size = 2
*** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: ***
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
230993                                This was a gift for a coffee connoisseur friend who was re
197476    Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surprisingly, t
343150                                                                    I love these ba
77376     I have always loved ghee with everything I prepare and eat - be it my daily dose of i
112553    We have a 4 oz stand up electric popcorn popper and find 4 oz bags a little hard to c
Name: Text, dtype: object
CLUSTER = 2
```

```
Not enough datapoints to display in this cluster!
```

9 K-Means & Hierarchical Clustering on tf-IDF

Sparse matrix generated from tf-IDF is fed in to GridSearch GBDT Cross Validator & RF Cross Validator to find the optimal depth value. Performance metrics of optimal GBDT with tf-idf featurization is found.

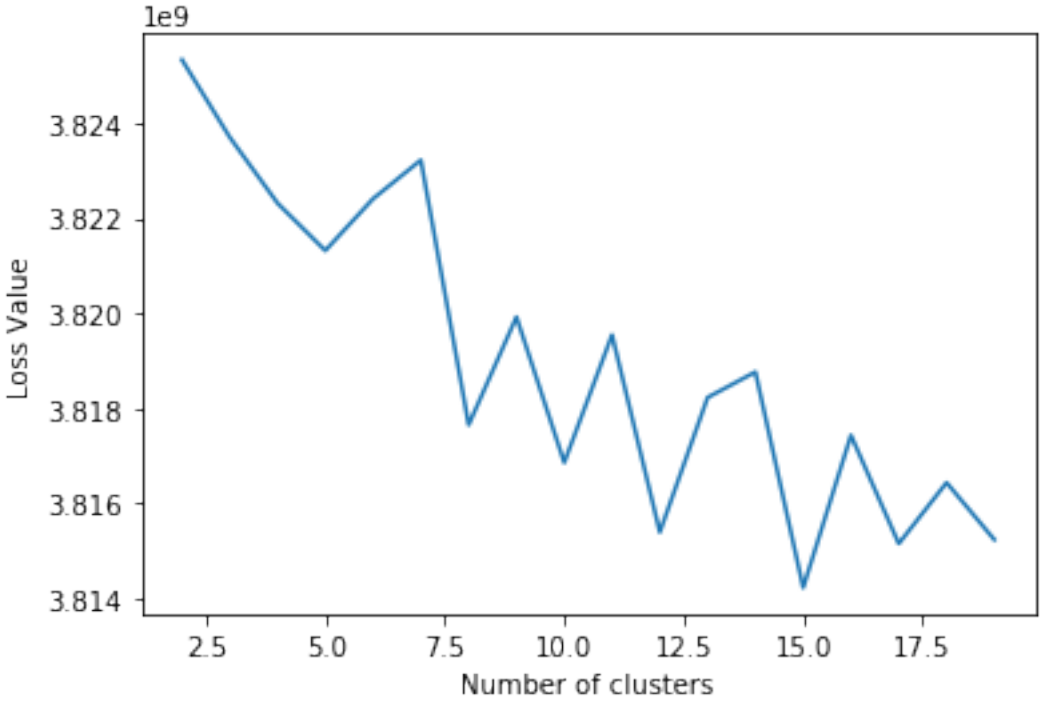
In [ ]:
```
# TFID Featurisation, Standardisation, Grid Search

from sklearn.random_projection import sparse_random_matrix

# # TFID
# count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
# d_vect = count_vect.fit_transform(d['CleanedText'].values)
# d_vect.get_shape()

# # Standardisation. Set "with_mean=False" to preserve sparsity
# scaler = StandardScaler(copy=False, with_mean=False).fit(d_vect)
# d_tfidf_vect_std = scaler.transform(d_vect)

# findK(d_tfidf_vect_std)



############################################################
### TFID for K-means: Vectorization & Standardization ###
count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
d_kmeans_vect = count_vect.fit_transform(d_kmeans['CleanedText'].values)
d_kmeans_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_kmeans_vect)
d_kmeans_bow_vect_std = scaler.transform(d_kmeans_vect)

############################################################

### TFID for Hierarchical: Vectorization & Standardization ###
count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
d_hier_vect = count_vect.fit_transform(d_hierarchical['CleanedText'].values)
d_hier_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_hier_vect)
d_hier_bow_vect_std = scaler.transform(d_hier_vect)
```

```
        ############################################################

        ## To find the best K for K-means
        k = findK(d_kmeans_bow_vect_std)

        # Hierarchical Clustering
        hierarchichal = AgglomerativeClustering(n_clusters=2).fit(
                                                d_hier_bow_vect_std.toarray())
```

```
C:\Users\Anand\Anaconda3\envs\myenv\lib\site-packages\sklearn\feature_extraction\text.py:1547:
  UserWarning)
```

```
Group Counter in Cluster 2 is as follows:
Counter({0: 99999, 1: 1})
Group Counter in Cluster 3 is as follows:
Counter({2: 99968, 1: 27, 0: 5})
Group Counter in Cluster 4 is as follows:
Counter({0: 99997, 3: 1, 2: 1, 1: 1})
Group Counter in Cluster 5 is as follows:
Counter({1: 99996, 2: 1, 0: 1, 3: 1, 4: 1})
Group Counter in Cluster 6 is as follows:
Counter({2: 99994, 0: 2, 3: 1, 5: 1, 1: 1, 4: 1})
Group Counter in Cluster 7 is as follows:
Counter({5: 99948, 4: 25, 3: 14, 0: 10, 2: 1, 6: 1, 1: 1})
Group Counter in Cluster 8 is as follows:
Counter({0: 99993, 2: 1, 6: 1, 7: 1, 1: 1, 4: 1, 3: 1, 5: 1})
Group Counter in Cluster 9 is as follows:
Counter({0: 99992, 2: 1, 4: 1, 8: 1, 7: 1, 1: 1, 5: 1, 6: 1, 3: 1})
Group Counter in Cluster 10 is as follows:
Counter({1: 99987, 0: 4, 8: 2, 9: 1, 6: 1, 2: 1, 4: 1, 3: 1, 7: 1, 5: 1})
Group Counter in Cluster 11 is as follows:
Counter({9: 99816, 0: 104, 2: 53, 1: 19, 3: 2, 6: 1, 7: 1, 8: 1, 4: 1, 10: 1, 5: 1})
Group Counter in Cluster 12 is as follows:
Counter({7: 99351, 0: 536, 1: 43, 6: 37, 5: 20, 3: 7, 2: 1, 11: 1, 9: 1, 10: 1, 4: 1, 8: 1})
Group Counter in Cluster 13 is as follows:
Counter({2: 91326, 3: 8507, 12: 122, 10: 33, 0: 4, 6: 1, 5: 1, 7: 1, 8: 1, 4: 1, 1: 1, 9: 1, 1
Group Counter in Cluster 14 is as follows:
Counter({4: 99844, 2: 143, 0: 2, 5: 1, 3: 1, 11: 1, 12: 1, 9: 1, 6: 1, 7: 1, 8: 1, 13: 1, 1: 1
Group Counter in Cluster 15 is as follows:
Counter({8: 99967, 0: 15, 6: 5, 13: 2, 12: 1, 9: 1, 3: 1, 14: 1, 10: 1, 1: 1, 2: 1, 5: 1, 7: 1
Group Counter in Cluster 16 is as follows:
Counter({2: 99977, 0: 8, 10: 2, 5: 1, 8: 1, 7: 1, 14: 1, 9: 1, 11: 1, 6: 1, 13: 1, 15: 1, 3: 1
Group Counter in Cluster 17 is as follows:
Counter({11: 99874, 7: 64, 6: 18, 4: 16, 5: 15, 0: 2, 10: 1, 12: 1, 8: 1, 9: 1, 16: 1, 14: 1, 3
Group Counter in Cluster 18 is as follows:
Counter({4: 99951, 1: 27, 0: 5, 8: 2, 2: 2, 3: 1, 7: 1, 15: 1, 12: 1, 9: 1, 5: 1, 13: 1, 11: 1
Group Counter in Cluster 19 is as follows:
```

```
Counter({2: 99899, 1: 79, 0: 3, 12: 2, 15: 2, 5: 2, 14: 1, 18: 1, 9: 1, 8: 1, 16: 1, 10: 1, 11
```



In [ ]:
```
# To analyze clusters formed by kmeans & hierarchical clustering

        # Analyse review in k clusters.
        # k is found using elbow method plot above.
        # from the above elbow plot, k = 12 is found to be optimum for K-means

        pd.options.display.max_colwidth = 200

        # Analyze clusters formed by kmeans clustering
        kmeans = KMeans(n_clusters=12, max_iter=300).fit(d_kmeans_bow_vect_std)
        analyzeClusters(d_labels=kmeans.labels_, k=12, algo='kmeans')

        # Analyze clusters formed by hierarchical clustering
        analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')
```

```
<class 'collections.Counter'>
dict_items([(3, 99963), (0, 18), (4, 1), (2, 5), (1, 6), (7, 1), (5, 1), (8, 1), (9, 1), (6, 1
cluster size = 12
*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
```

```
CLUSTER = 1
```
The Review Text in Cluster 0 is as follows:
```
111341    Does ganoderma do all it's said to do?  Heck, I don't know.  I do know that when I dr
66495     Navitas notes that their Maca is slow dried and not irradiated.  The product arrived
206098    In reading the reviews here it is obvious that different tastebuds are affected very
44157     Is it Tea Magic, the placebo effect, or the actual tea that creates such a relaxing r
7975      I've got a room mate that swears by these. I bought a few packs of these, and didn't
Name: Text, dtype: object
CLUSTER = 2
```
The Review Text in Cluster 1 is as follows:
```
256123    I started drinking this coffee while on a trip to Costa Rica about 15 years ago. I li
330104    I have a cat named pounder who is just turning a year old this month. I also have Dal
126461    Just had my 6 month repeat endoscopy. While I probably still have Barrett's (awaiting
117098    I have an elderly dog who gets several meds each day, and these make life so much eas
142105    My dog became very ill (vomiting, diarrhea, excruciating pain) in mid-Oct and spent t
Name: Text, dtype: object
CLUSTER = 3
```
The Review Text in Cluster 2 is as follows:
```
191556    I bought the ICICLE because I have been doing a lot of hip-hop vocals and I was using
191595    I have little doubt that the Blue Icicle will get the job done with a Shure SM-58, or
191581    Though this micpreamp has several weaknesses, I give this 5 stars because it's just p
191618    Let's get this out of the way, first: Yes the case is made of thin flimsy plastic, an
191567    I will admit, the Icicle does allow you to plug any mic with an XLR jack to your comp
Name: Text, dtype: object
CLUSTER = 4
```
The Review Text in Cluster 3 is as follows:
```
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is r
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard"
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
Name: Text, dtype: object
CLUSTER = 5
Not enough datapoints to display in this cluster!
CLUSTER = 6
Not enough datapoints to display in this cluster!
CLUSTER = 7
Not enough datapoints to display in this cluster!
CLUSTER = 8
Not enough datapoints to display in this cluster!
CLUSTER = 9
Not enough datapoints to display in this cluster!
CLUSTER = 10
Not enough datapoints to display in this cluster!
CLUSTER = 11
Not enough datapoints to display in this cluster!
CLUSTER = 12
Not enough datapoints to display in this cluster!
```

```
<class 'collections.Counter'>
dict_items([(0, 4999), (1, 1)])
cluster size = 2
*** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: ***
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
230993                                This was a gift for a coffee connoisseur friend who was re
197476    Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surprisingly, t
343150                                                                          I love these ba
77376     I have always loved ghee with everything I prepare and eat - be it my daily dose of i
112553    We have a 4 oz stand up electric popcorn popper and find 4 oz bags a little hard to c
Name: Text, dtype: object
CLUSTER = 2
Not enough datapoints to display in this cluster!
```

10 K-Means, Hierarchical Clustering & DBScan on Word2Vec

Dense matrix generated from Word2Vec is fed in to GridSearch GBDT Cross Validator & RF Cross Validator to find the optimal depth value. Performance metrics of GBDT and RF with W2V featurization is found.

```
In [27]: # Train your own Word2Vec model using your own text corpus
         import gensim
         import re

         w2v_dim = 100

         def cleanhtml(sentence): #function to clean the word of any html-tags
             cleanr = re.compile('<.*?>')
             cleantext = re.sub(cleanr, ' ', sentence)
             return cleantext

         #function to clean the word of any punctuation or special characters
         def cleanpunc(sentence):
             cleaned = re.sub(r'[?|!|\'|"|#]',r'',sentence)
             cleaned = re.sub(r'[.|,|)|(|\|/]',r' ',cleaned)
             return  cleaned

         def trainW2V_model(reviewText):
             #select subset of points for fast execution
             i=0
             list_of_sent=[]

             for sent in reviewText:
```

```
                sent = str(sent, 'utf-8')
                filtered_sentence=[]
                sent=cleanhtml(sent)
                for w in sent.split():
                    for cleaned_words in cleanpunc(w).split():
                        if(cleaned_words.isalpha()):
                            filtered_sentence.append(cleaned_words.lower())
                        else:
                            continue
                list_of_sent.append(filtered_sentence)

            w2v_model=gensim.models.Word2Vec(list_of_sent,
                                             min_count=5,size=w2v_dim, workers=4)

            return w2v_model
```

In [28]:
```
        # average Word2Vec
        # compute average word2vec for each review.

        def computeAvgW2V(w2vTrained_model, reviewText):
            sent_vectors = []; # the avg-w2v for each sentence/review is stored in this list

            for sent in reviewText: # for each review/sentence
                sent_vec = np.zeros(w2v_dim) # as word vectors are of zero length
                cnt_words =0; # num of words with a valid vector in the sentence/review
                sent = str(sent, 'utf-8')
                sent = re.sub("[^\w]", " ",  sent).split()

                for word in sent: # for each word in a review/sentence
                    try:
                        vec = w2vTrained_model.wv[word]
                        sent_vec += vec
                        cnt_words += 1
                    except:
                        pass
                sent_vec /= cnt_words
                sent_vectors.append(sent_vec)

            return np.nan_to_num(sent_vectors)
```

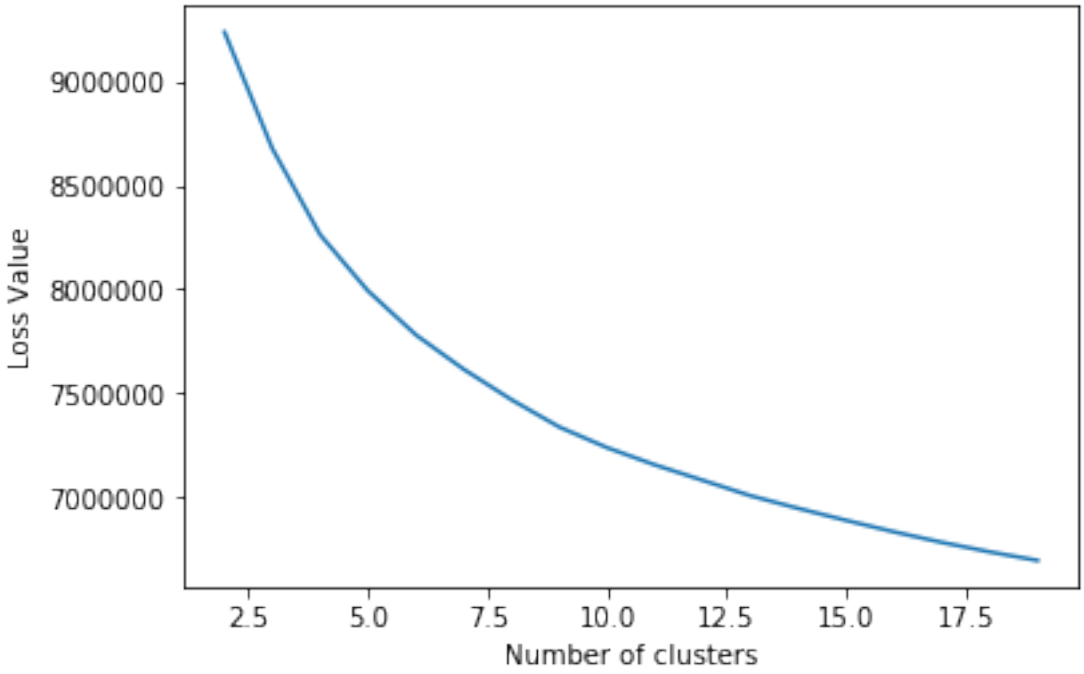
In [29]:
```
        # W2V Main Function
        # W2V Featurisation, Standardisation, Grid Search and Random Search,
        # Perturbation test to remove multicollinear features

        from sklearn.preprocessing import StandardScaler

        ##############################################################
        ### W2V for K-means: Vectorization & Standardization ###
```

```
        # W2V Train
        w2v_kmeans_Model = trainW2V_model(d_kmeans['CleanedText'].values)
        d_kmeans_vect = computeAvgW2V(w2v_kmeans_Model, d_kmeans['CleanedText'].values)

        # Standardisation.
        scaler = StandardScaler(copy=False).fit(d_kmeans_vect)
        d_w2v_kmeans_vect_std = scaler.transform(d_kmeans_vect)

        ############################################################
        ### W2V for Hierarchical: Vectorization & Standardization ###
        w2v_hier_Model = trainW2V_model(d_hierarchical['CleanedText'].values)
        d_hier_vect = computeAvgW2V(w2v_hier_Model, d_hierarchical['CleanedText'].values)

        # Standardisation.
        scaler = StandardScaler(copy=False).fit(d_hier_vect)
        d_w2v_heir_vect_std = scaler.transform(d_hier_vect)

        ############################################################
        print("***before finding k***")
        ## To find the best K for K-means
        findK(d_w2v_kmeans_vect_std)

        print("***before clustering***")
        # Hierarchical Clustering
        hierarchichal = AgglomerativeClustering(n_clusters=2).fit(d_w2v_heir_vect_std)

        print("***after 1st clustering***")
        # Hierarchical Clustering - different K
        hierarchichal_test = AgglomerativeClustering(n_clusters=5).fit(d_w2v_heir_vect_std)

        print("***after 2nd clustering***")
        # by rule of thumb, min_samples should be 2*dimensionality = 200
        # we need to estimate eps value by doing an elbow plot.
        kneeValue = KNNdist_plot(d_w2v_heir_vect_std,200)
        print("***ended clustering***")
```

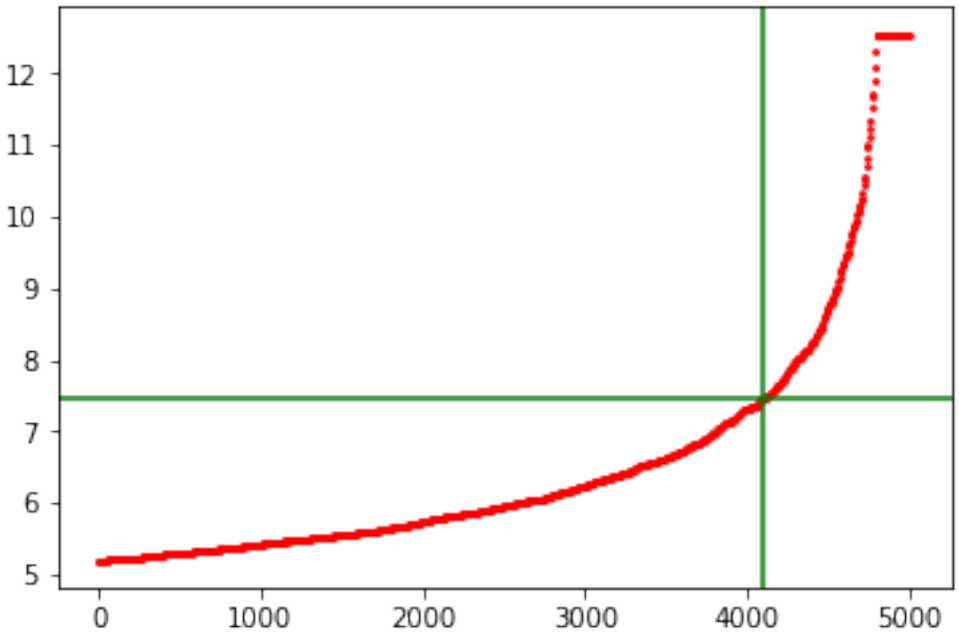
```
C:\Users\Anand\Anaconda3\envs\myenv\lib\site-packages\ipykernel_launcher.py:20: RuntimeWarning
```

```
***before finding k***
Group Counter in Cluster 2 is as follows:
Counter({0: 72190, 1: 27810})
Group Counter in Cluster 3 is as follows:
Counter({2: 41502, 1: 40864, 0: 17634})
Group Counter in Cluster 4 is as follows:
Counter({1: 41374, 0: 25604, 2: 17063, 3: 15959})
Group Counter in Cluster 5 is as follows:
Counter({1: 25283, 0: 23856, 4: 19574, 2: 16064, 3: 15223})
```

Group Counter in Cluster 6 is as follows:
Counter({3: 23529, 5: 21573, 4: 16838, 2: 15234, 1: 12577, 0: 10249})
Group Counter in Cluster 7 is as follows:
Counter({6: 18659, 3: 16382, 2: 15693, 4: 14491, 1: 12558, 5: 11754, 0: 10463})
Group Counter in Cluster 8 is as follows:
Counter({0: 18249, 5: 16077, 2: 15194, 6: 14860, 3: 11542, 4: 10381, 7: 7991, 1: 5706})
Group Counter in Cluster 9 is as follows:
Counter({3: 18001, 0: 14262, 6: 13702, 4: 11393, 2: 11026, 1: 9300, 8: 8796, 7: 7862, 5: 5658})
Group Counter in Cluster 10 is as follows:
Counter({9: 15201, 8: 12310, 6: 11530, 1: 10788, 3: 10319, 7: 9518, 0: 8718, 4: 8361, 5: 7707,
Group Counter in Cluster 11 is as follows:
Counter({2: 11941, 0: 11764, 5: 11006, 4: 10311, 3: 8978, 6: 8593, 10: 8374, 7: 8025, 1: 7838,
Group Counter in Cluster 12 is as follows:
Counter({3: 11893, 2: 10700, 9: 8799, 0: 8433, 11: 8428, 7: 8352, 10: 7737, 8: 7724, 1: 7654, 6
Group Counter in Cluster 13 is as follows:
Counter({7: 11713, 6: 8728, 8: 8541, 5: 8401, 9: 8339, 12: 7649, 11: 7622, 2: 7538, 0: 7324, 4
Group Counter in Cluster 14 is as follows:
Counter({10: 10753, 13: 8590, 3: 8210, 9: 7987, 8: 7509, 5: 7337, 11: 7212, 1: 6834, 2: 6809, 0
Group Counter in Cluster 15 is as follows:
Counter({0: 10508, 7: 8618, 3: 8082, 4: 7470, 9: 7442, 12: 7115, 6: 6918, 1: 6863, 8: 6718, 10
Group Counter in Cluster 16 is as follows:
Counter({11: 8273, 2: 8058, 9: 7786, 6: 7451, 8: 7252, 10: 6725, 3: 6699, 14: 6541, 12: 6305, 1
Group Counter in Cluster 17 is as follows:
Counter({7: 8206, 4: 7988, 3: 7370, 5: 6810, 10: 6692, 14: 6567, 16: 6276, 13: 6250, 9: 5794, 1
Group Counter in Cluster 18 is as follows:
Counter({16: 8057, 11: 7977, 15: 6612, 14: 6527, 8: 6497, 0: 6248, 2: 6195, 4: 5778, 1: 5279, 9
Group Counter in Cluster 19 is as follows:
Counter({0: 7799, 11: 7714, 13: 7613, 18: 6367, 3: 5998, 8: 5607, 9: 5154, 1: 5145, 14: 5120, 1




```
***before clustering***
***after 1st clustering***
***after 2nd clustering***
```



```
Knee value: x=4095 , y=7.465998466888701
***ended clustering***
```

In [30]:
```
# from the above elbow plot, k = 15 is found to be optimum

# Analyse review in k clusters.
# k is found using elbow method plot above.

pd.options.display.max_colwidth = 200
# analyzeClusters(d_vect_std=d_w2v_vect_std, k=15)

print("***before kmeans clustering***")
# Analyze clusters formed by kmeans clustering
kmeans = KMeans(n_clusters=15, max_iter=300).fit(d_w2v_kmeans_vect_std)
analyzeClusters(d_labels=kmeans.labels_, k=15, algo='kmeans')

print("***before heir clustering***")
# Analyze clusters formed by hierarchical clustering
analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')

print("***before 2nd heir clustering***")
# Analyze clusters formed by hierarchical clustering
print(bold + "%%% Hierarchical Clustering with 5 Clusters %%%" + end)
analyzeClusters(d_labels=hierarchichal_test.labels_, k=5, algo='hierarchical')

print("***ended heir clustering***")

#### eps = kneeValue from the above plot
# Analyze clusters formed by DBSCAN clustering
# by rule of thumb, min_samples should be 2*dimensionality = 200
# set function is used to determine unique values.
dbscan = DBSCAN(eps=kneeValue, min_samples=200).fit(d_w2v_kmeans_vect_std)
analyzeClusters(d_labels=dbscan.labels_, k=len(set(dbscan.labels_)), algo='dbscan')
print("***after dbscan clustering***")

# To try out other Eps values
# by rule of thumb, min_samples should be 2*dimensionality = 200
print(bold + "%%% DBSCAN Clustering with Eps = 1 %%%" + end)
dbscan = DBSCAN(eps=1, min_samples=200).fit(d_w2v_kmeans_vect_std)
print(len(set(dbscan.labels_)))
analyzeClusters(d_labels=dbscan.labels_, k=len(set(dbscan.labels_)), algo='dbscan')

print("***after 2nd dbscan clustering***")
```

```
        print(bold + "%%% DBSCAN Clustering with Eps = 50 %%%" + end)
        dbscan = DBSCAN(eps=10, min_samples=200).fit(d_w2v_kmeans_vect_std)
        print(len(set(dbscan.labels_)))
        analyzeClusters(d_labels=dbscan.labels_, k=len(set(dbscan.labels_)), algo='dbscan')

        print("***after 3rd dbscan clustering***")
```

```
***before kmeans clustering***


<class 'collections.Counter'>
dict_items([(0, 8632), (5, 7130), (1, 10642), (6, 8074), (12, 4014), (2, 7476), (4, 5995), (14
cluster size = 15
*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is
37085     I got this pasta for my 14 month old and she loves it. I hated giving her the Kraft
73433     This is a real gourmet product. You can add them to melon cut in dices and it makes
347980    The ingredients are listed below (taken from the product package and website):<br />
156014    So I guess I'm the first to try this on this site. So I shall tell you my experience
Name: Text, dtype: object
CLUSTER = 2
The Review Text in Cluster 1 is as follows:
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard
180735    I had been thinking for sometime to look for Graisse de Canard and where else to loo
70177     I got these at Sprouts because I thought they were croutons.  When I actually looked
163735    Not what I expected, nothing like the illustration on the package.  I have enjoyed R
77958     I dont have a severe allergy to peanuts, but they do cause me break out badly, and i
Name: Text, dtype: object
CLUSTER = 3
The Review Text in Cluster 2 is as follows:
311384                                                                                  Cafe
240280    I ordered Dark Sumatra Gayoland coffee bean from Coffee Bean Direct via Amazon. It ca
346142    SF Bay Coffee's Fog Chaser was a nice "bold" surprise.  I am a big fan of Amazon's Su
242252                                                                 Received quickly a
236062    If you're trying to avoid sweets over the holidays, but drink coffee and like chocola
Name: Text, dtype: object
CLUSTER = 4
The Review Text in Cluster 3 is as follows:
32904     These are the most addictive cookies on the market.  They are Very rich, Ultimately
49234     I've always loved Thai yellow curries, but never had much luck trying to duplicate wh
55966     I used to eat these as a kid growing up in Frankfurt, Germany.  When I recieved my or
12582     These are the best tasting pretzels I have found.  Since they are not available in th
104868                                                                  For my favorite
Name: Text, dtype: object
CLUSTER = 5
The Review Text in Cluster 4 is as follows:
```

190346
18115 When I read a customer review which said that QUICK GRITS were sent when OLD FASHION
190726
104158 Received can with large dent. The box was in good shape, so can was damaged before sh
172556
Name: Text, dtype: object
CLUSTER = 6
The Review Text in Cluster 5 is as follows:
23280 I think this is probably as good as it gets for sugar free chocolate syrup. It's sti
222236 I was turned on to Rooibos tea about 6 months ago and have enjoyed the taste. Lookin
204057 The Black Cherry Switch is an interesting beverage, a cross between the traditional c
175821 I have been hooked on crystal sugar like this ever since I used it when I worked for
228686 Arctic Zero ice cream is AMAZING. It's SO nice to finally have a fantastic tasting "I
Name: Text, dtype: object
CLUSTER = 7
The Review Text in Cluster 6 is as follows:
63408
363410 This is my favorite coconut milk. I use
168038 This is definitely THE cereal to get. It's good for you and actually tastes good. I
306512 The water is great but the price has me at a loss. I have been paying anywhere from $
24827 These are a staple in my house. Great to have them shipped in rather than trying to
Name: Text, dtype: object
CLUSTER = 8
The Review Text in Cluster 7 is as follows:
198671 I received my Keurig for Christmas and since then have tried as many black teas that
37311 I just tried orange spice tea for the first time on a recent trip to Hawaii, and they
284144 This Numi tea i
19187 The original flavor of Red Rose tea is one that I have enjoyed almost all of my life
35444 This tea is amazing!

The first thing I noticed was the strong scent of cit
Name: Text, dtype: object
CLUSTER = 9
The Review Text in Cluster 8 is as follows:
77439 Started buying this product 4 months ago after the whole "Canidae" issue (Im sure tha
115393 Started feeding my two older dogs (12 and 13) this food. My lab has always had dry it
331471 My dog, like all dogs, will eat anything. But my pooch would NOT eat this crap. Smart
132203 Great food
339096 My cats aren't very fond of the taste. It took forever to get them to switch to this
Name: Text, dtype: object
CLUSTER = 10
The Review Text in Cluster 9 is as follows:
256827 Our labrador has an iron constitution. She has eaten a broad variety of food (and oth
218656 With the first swipe, you will see it work. The fur just comes off!
While brushi
357438 I was very please to have been introduced and found this wonderful, refreshing healthl
257587 I used this product on my 9 week old kittens. Instead of using 5 drops for kittens a
218707 I've tried 5-hour energy, red rain, NOS, and other energy shots. This is my favorite
Name: Text, dtype: object
CLUSTER = 11
The Review Text in Cluster 10 is as follows:

92619 Gluten-free or not, this is the best cornbread mix I have ever had - period. Even my
232993 I was expecting for this cake to turn out as dense and as hard as a rock, but boy, wh
352746 I had been using PB2, but saw this at the store and scooped up a jar. Came home, toa
304073 I bought this to try as a replacement for a combination of whole wheat flour and mult
3220 Okay, I know it sugar free, but does that mean it also has to be flavor free? A ging
Name: Text, dtype: object
CLUSTER = 12
The Review Text in Cluster 11 is as follows:
133845 I have tried another rice shell and liked it, but this one is closer to a "real" pizz
255523 I can recognize bad popcorn but either I've never had great popcorn or I'm just not t
317914
18220 I don't know what others got, but I was shipped a box of teeny tiny little pig ears i
246216 The tea has a very good vanilla flavor, i
Name: Text, dtype: object
CLUSTER = 13
The Review Text in Cluster 12 is as follows:
245004 Never mind that this dog food is kind of gross looking - my dogs just love it! While
92406 I bought these as a treat for my dog and he started vomiting about an hour later. I
110069
352610 We got this for our senior kitty, who is a picky eater but loves wet food (which make
113364 I also bought a bag of ZiwiPeak dog food and it's the same size and texture as the ti
Name: Text, dtype: object
CLUSTER = 14
The Review Text in Cluster 13 is as follows:
358834 I've been eating energy and protein bars for years. Most are like flavored cardboard
90125 My kids like Pirate Booty a lot. It is a great snack for lunches but I find that you
39914 I got this pack of cookies for my mom she cant have gluten she loves them so much she
37071 Th
58773 This cereal is an excellent way to gain healthy weight. Each box has 1470 calories,
Name: Text, dtype: object
CLUSTER = 15
The Review Text in Cluster 14 is as follows:
311440 I was very cautious and cut off a small corner of the pepper and set it on my tongue
71765 This is my new favorite flavor of Lundberg Rice Chips. They are so tasty and addictiv
318265 I am a big fan of scharffen berger and the
249289 These are delicious, especially for something that is sugar-free. They are light, wi
348665 I just love the nips they are a hard candy an
Name: Text, dtype: object
before heir clustering

<class 'collections.Counter'>
dict_items([(0, 3556), (1, 1444)])
cluster size = 2
***** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: *****
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
230993 This was a gift for a coffee connoisseur friend who was re

```
197476    Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surprisingly, t
77376     I have always loved ghee with everything I prepare and eat - be it my daily dose of r
112553    We have a 4 oz stand up electric popcorn popper and find 4 oz bags a little hard to c
121608    I go through alot of these bags and they are great! I love the colors and the size is
Name: Text, dtype: object
CLUSTER = 2
```
The Review Text in Cluster 1 is as follows:
```
343150                                                                    I love these ba
192872    I ordered these when they went on sale. At the sale price, they were about the same p
45809     I'm trying to reduce the amount of sugary snacks in my life but when I get the 'jones
166163    I live in the USA and I love the original HP sauce from England. I had found an Ameri
343053      I have tried several bags including Lanisoh and Medela and these are the only bags
Name: Text, dtype: object
***before 2nd heir clustering***
```
%%% Hierarchical Clustering with 5 Clusters %%%

```
<class 'collections.Counter'>
dict_items([(1, 1031), (0, 1444), (3, 1246), (2, 923), (4, 356)])
cluster size = 5
```
***** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: *****
```
CLUSTER = 1
```
The Review Text in Cluster 0 is as follows:
```
343150                                                                    I love these ba
192872    I ordered these when they went on sale. At the sale price, they were about the same p
45809     I'm trying to reduce the amount of sugary snacks in my life but when I get the 'jones
166163    I live in the USA and I love the original HP sauce from England. I had found an Ameri
343053      I have tried several bags including Lanisoh and Medela and these are the only bags
Name: Text, dtype: object
CLUSTER = 2
```
The Review Text in Cluster 1 is as follows:
```
230993                                This was a gift for a coffee connoisseur friend who was re
197476    Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surprisingly, t
77376     I have always loved ghee with everything I prepare and eat - be it my daily dose of r
112553    We have a 4 oz stand up electric popcorn popper and find 4 oz bags a little hard to c
161429    These flowers arrived to my door on time, and as expected they were in bud form. So,
Name: Text, dtype: object
CLUSTER = 3
```
The Review Text in Cluster 2 is as follows:
```
308605                                                                      Great tasti
99863           Best tasting Crystalized ginger on the market which is all natural with no sul
184356    I can't believe that I didn't see the light printing on the enlarged picture, but it
246682    first off, the size of the box this bar of chocolate arrived in was ridiculous. it co
188834    This is a delicious oil. The walnut flavor really stands out. We keep this oil in the
Name: Text, dtype: object
CLUSTER = 4
```
The Review Text in Cluster 3 is as follows:
```
121608    I go through alot of these bags and they are great! I love the colors and the size is
```

```
11279                                                                I used canola oil inst
65095     When you are on a strict diet like the hcg diet, you look for variety, and these are
273188    I like my coffee, but I'm no coffee-ophile or whatever the correct term might be. I l
299461                                                    This is as good canned kale as I
Name: Text, dtype: object
CLUSTER = 5
The Review Text in Cluster 4 is as follows:
48532     Both of my large breed dogs love Canidae.  I actually have to limit how much they can
218825    i was skeptical at first, i had been a loyal science diet customer for years.  One of
132266                It is too soon to tell if it is going to help her allergies, but my cocke
219740    . . . I'd have her write the review!  But since she can't speak, I can only speculate
150914
Name: Text, dtype: object
***ended heir clustering***


<class 'collections.Counter'>
dict_items([(0, 73431), (-1, 26569)])
cluster size = 2
*** CLUSTERS FORMED BY dbscan ALGORITHM is as follows: ***
CLUSTER = 0
The Review Text in Cluster -1 is as follows:
63408                                                          my wife and I enjoy cooking ar
311384                                      Cafe' Escapes Chai Latte K Cups-it taste so smooth
190346                                               I'm happy with the Tahini that I purchased
363410    This is my favorite coconut milk. I used to buy it at my local safeway, but they have
110069                                                 These are low calorie treats for you
Name: Text, dtype: object
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is i
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard'
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
92406     I bought these as a treat for my dog and he started vomiting about an hour later.  I
Name: Text, dtype: object
***after dbscan clustering***
%%% DBSCAN Clustering with Eps = 1 %%%
1


<class 'collections.Counter'>
dict_items([(-1, 100000)])
cluster size = 1
*** CLUSTERS FORMED BY dbscan ALGORITHM is as follows: ***
CLUSTER = 0
The Review Text in Cluster -1 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is i
```

```
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
Name: Text, dtype: object
***after 2nd dbscan clustering***
%%% DBSCAN Clustering with Eps = 50 %%%
2


<class 'collections.Counter'>
dict_items([(0, 98299), (-1, 1701)])
cluster size = 2
*** CLUSTERS FORMED BY dbscan ALGORITHM is as follows: ***
CLUSTER = 0
The Review Text in Cluster -1 is as follows:
37071     The kids eat these up quickly.  I pack bunnies instead of chips in their lunch withou
190726                         The dried berries arrived promptly and in good condition. We tho
253538               Just the fact of having the ingredient "High Fructose Corn syrup" Deters m
104631                                I buy these for my baby (17months). you can melt them in
75000                               Love Coconut oil! Ordered this because of the great value.
Name: Text, dtype: object
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is i
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
Name: Text, dtype: object
***after 3rd dbscan clustering***
```

11 K-Means, Hierarchical Clustering & DBScan on TF-ID Weighted W2V

```
In [31]: # average Word2Vec
         # compute average word2vec for each review.

         def compute_tfidW2V(w2v_model, model_tf_idf, count_vect, reviewText):

             # the tfidf-w2v for each sentence/review is stored in this list
             tfidf_sent_vectors = [];
             row=0;

             # TF-IDF weighted Word2Vec
             tfidf_feats = count_vect.get_feature_names() # tfidf words/col-names
```

```
        # iterate for each review/sentence
        for sent in reviewText:
            sent_vec = np.zeros(w2v_dim) # as word vectors are of zero length
            weight_sum =0; # num of words with a valid vector in the sentence/review
            sent = str(sent, 'utf-8')
            sent = re.sub("[^\w]", " ",  sent).split()

            for word in sent: # for each word in a review/sentence
                try:
                    vec = w2v_model.wv[word]
                    # obtain the tf_idfidf of a word in a sentence/review
                    tfidf = model_tf_idf[row, tfidf_feats.index(word)]
                    sent_vec += (vec * tfidf)
                    weight_sum += tfidf
                except:
                    pass
            sent_vec /= weight_sum

            tfidf_sent_vectors.append(sent_vec)
            row += 1

        return np.nan_to_num(tfidf_sent_vectors)
```

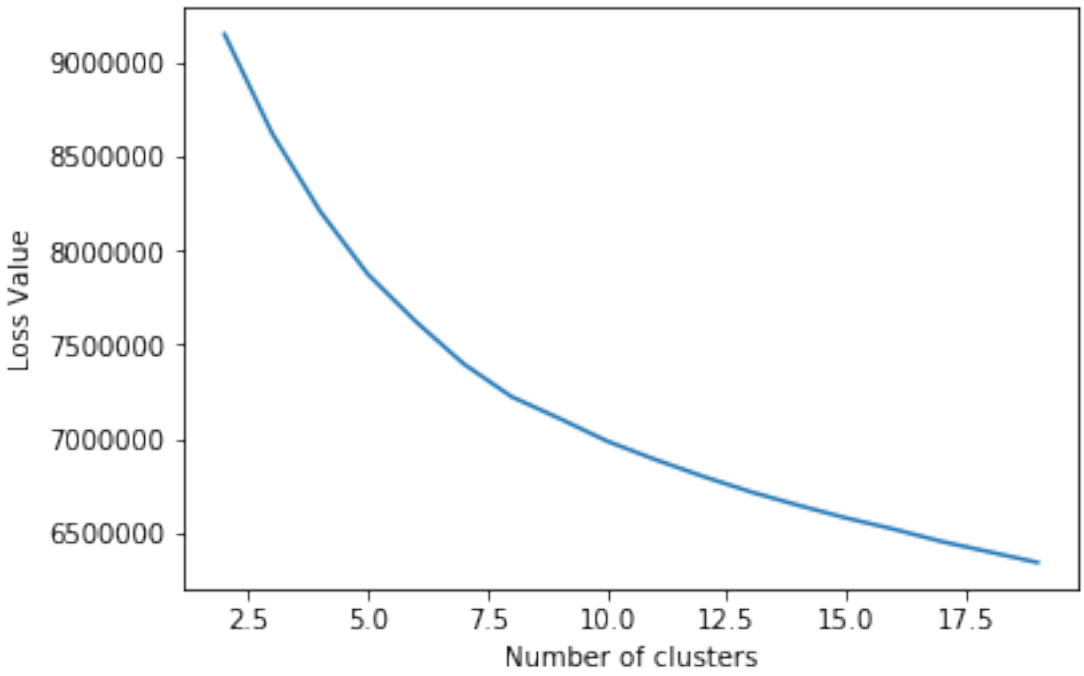
In [32]:
```
# tf-df weighted W2V Main Function
# tfidf and W2V Featurisation, Standardisation, Grid Search
# Perturbation test to remove multicollinear features

from sklearn.preprocessing import StandardScaler

############################################################
### TFIDW2V for K-means: Vectorization & Standardization ###
# TFID W2V Train
count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
d_kmeans_tfidW2v_vect = count_vect.fit_transform(d_kmeans['CleanedText'].values)

d_kmeans_avg_vect = compute_tfidW2V(w2v_kmeans_Model, d_kmeans_tfidW2v_vect,
                                    count_vect, d_kmeans['CleanedText'].values)

# Standardisation.
scaler = StandardScaler(copy=False).fit(d_kmeans_avg_vect)
d_tfidf_w2v_kmeans_vect_std = scaler.transform(d_kmeans_avg_vect)

############################################################
### TFIDW2V for Hierarchical: Vectorization & Standardization ###

count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
d_hier_tfidW2v_vect = count_vect.fit_transform(d_hierarchical['CleanedText'].values)
```

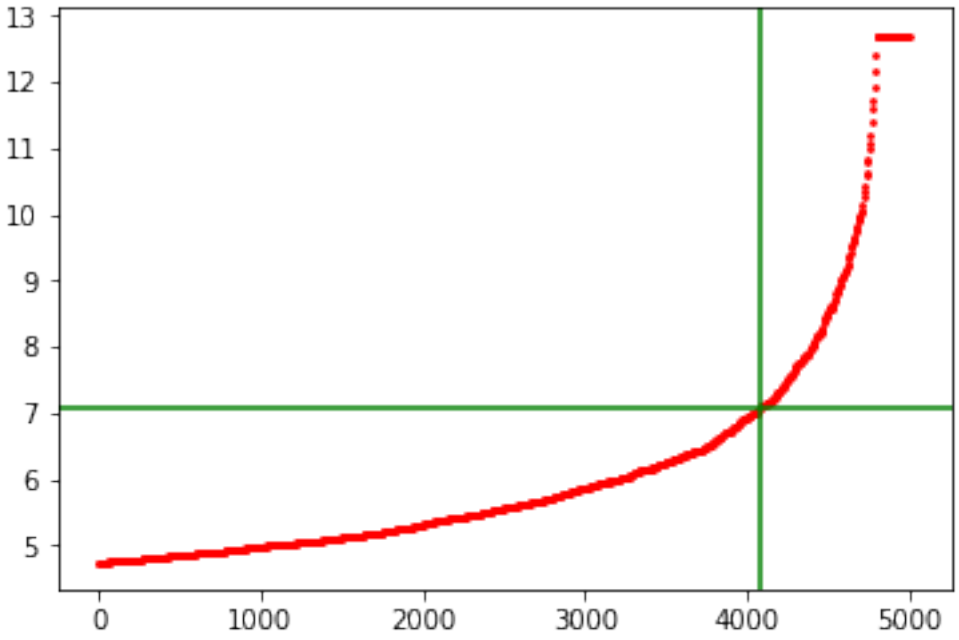
```
        d_hier_avg_vect = compute_tfidW2V(w2v_hier_Model, d_hier_tfidW2v_vect,
                                          count_vect, d_hierarchical['CleanedText'].values)

        # Standardisation.
        scaler = StandardScaler(copy=False).fit(d_hier_avg_vect)
        d_tfidf_w2v_hier_vect_std = scaler.transform(d_hier_avg_vect)
        ###############################################################

        ## To find the best K for K-means
        findK(d_tfidf_w2v_kmeans_vect_std)

        # Hierarchical Clustering
        hierarchichal = AgglomerativeClustering(n_clusters=2).fit(d_tfidf_w2v_hier_vect_std)

        # Hierarchical Clustering - different K
        hierarchichal_test = AgglomerativeClustering(n_clusters=5).fit(d_tfidf_w2v_hier_vect_s

        # by rule of thumb, min_samples should be 2*dimensionality = 200
        # we need to estimate eps value by doing an elbow plot.
        kneeValue = KNNdist_plot(d_tfidf_w2v_hier_vect_std,200)
```

```
C:\Users\Anand\Anaconda3\envs\myenv\lib\site-packages\sklearn\feature_extraction\text.py:1547:
  UserWarning)
C:\Users\Anand\Anaconda3\envs\myenv\lib\site-packages\ipykernel_launcher.py:29: RuntimeWarning
C:\Users\Anand\Anaconda3\envs\myenv\lib\site-packages\sklearn\feature_extraction\text.py:1547:
  UserWarning)
```

```
Group Counter in Cluster 2 is as follows:
Counter({0: 79235, 1: 20765})
Group Counter in Cluster 3 is as follows:
Counter({0: 43088, 1: 41398, 2: 15514})
Group Counter in Cluster 4 is as follows:
Counter({0: 42088, 3: 32424, 2: 14478, 1: 11010})
Group Counter in Cluster 5 is as follows:
Counter({3: 31167, 4: 27921, 2: 17134, 0: 13092, 1: 10686})
Group Counter in Cluster 6 is as follows:
Counter({4: 30965, 0: 28201, 1: 17170, 3: 10677, 2: 7925, 5: 5062})
Group Counter in Cluster 7 is as follows:
Counter({3: 27834, 2: 19639, 1: 16084, 0: 14591, 5: 10277, 4: 7342, 6: 4233})
Group Counter in Cluster 8 is as follows:
Counter({5: 24761, 7: 14315, 4: 14258, 0: 13673, 3: 13345, 1: 8256, 2: 7209, 6: 4183})
Group Counter in Cluster 9 is as follows:
Counter({3: 23171, 0: 13468, 8: 12756, 1: 12495, 6: 9891, 7: 8906, 5: 7993, 4: 7154, 2: 4166})
Group Counter in Cluster 10 is as follows:
Counter({0: 22837, 9: 12310, 5: 11389, 4: 11386, 7: 9753, 8: 8766, 1: 7936, 6: 7155, 2: 4304, 3
Group Counter in Cluster 11 is as follows:
```

```
Counter({10: 20422, 6: 11365, 8: 10631, 0: 9035, 3: 8578, 5: 8553, 4: 8475, 1: 7419, 7: 7195, 1
Group Counter in Cluster 12 is as follows:
Counter({10: 16878, 8: 10681, 0: 10606, 7: 10132, 11: 8863, 5: 8074, 2: 7768, 1: 7345, 9: 7024
Group Counter in Cluster 13 is as follows:
Counter({3: 16701, 8: 10469, 1: 10339, 11: 8748, 9: 8703, 7: 8080, 5: 7732, 2: 7302, 6: 7012, 0
Group Counter in Cluster 14 is as follows:
Counter({8: 15136, 10: 9584, 7: 8483, 11: 8106, 4: 7891, 0: 7668, 1: 7441, 6: 7208, 9: 7058, 2
Group Counter in Cluster 15 is as follows:
Counter({0: 15134, 9: 9443, 5: 8459, 6: 8143, 12: 7633, 10: 7611, 14: 7399, 4: 6957, 13: 6884,
Group Counter in Cluster 16 is as follows:
Counter({9: 14864, 1: 9076, 15: 8393, 11: 7445, 8: 7275, 12: 6956, 14: 6936, 10: 6924, 6: 6347
Group Counter in Cluster 17 is as follows:
Counter({16: 13396, 11: 8949, 1: 8925, 5: 8302, 7: 7303, 0: 7067, 2: 7037, 6: 6756, 12: 6270, 3
Group Counter in Cluster 18 is as follows:
Counter({10: 12587, 16: 9558, 15: 8219, 2: 7610, 4: 7341, 14: 7126, 0: 6943, 12: 6918, 9: 6704
Group Counter in Cluster 19 is as follows:
Counter({17: 9478, 6: 8931, 9: 7872, 11: 7814, 3: 7537, 16: 7288, 1: 6556, 13: 6539, 7: 5344, 5
```






```
Knee value: x=4080 , y=7.083716272480987
```

In [33]:
```
# from sklearn.neighbors import NearestNeighbors
```

In [34]:
```
# Analyse review in k clusters.
# k is found using elbow method plot above.

pd.options.display.max_colwidth = 200

# KMEANS
# Analyze clusters formed by kmeans clustering
kmeans = KMeans(n_clusters=15, max_iter=300).fit(d_tfidf_w2v_kmeans_vect_std)
analyzeClusters(d_labels=kmeans.labels_, k=15, algo='kmeans')

## HIERARCHICAL
# Analyze clusters formed by hierarchical clustering
analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')

# Analyze clusters formed by hierarchical clustering
print(bold + "%%% Hierarchical Clustering with 5 Clusters %%%" + end)
analyzeClusters(d_labels=hierarchichal_test.labels_, k=5, algo='hierarchical')


## DBSCAN
# by rule of thumb, min_samples should be 2*dimensionality = 200
```

```
        dbscan = DBSCAN(eps=kneeValue, min_samples=200).fit(d_tfidf_w2v_kmeans_vect_std)
        analyzeClusters(d_labels=dbscan.labels_, k=2, algo='dbscan')

        # To try out other Eps values
        # by rule of thumb, min_samples should be 2*dimensionality = 200
        print(bold + "%%% DBSCAN Clustering with Eps = 1 %%%" + end)
        dbscan = DBSCAN(eps=1, min_samples=200).fit(d_tfidf_w2v_kmeans_vect_std)
        analyzeClusters(d_labels=dbscan.labels_, k=2, algo='dbscan')

        print(bold + "%%% DBSCAN Clustering with Eps = 50 %%%" + end)
        dbscan = DBSCAN(eps=10, min_samples=200).fit(d_tfidf_w2v_kmeans_vect_std)
        analyzeClusters(d_labels=dbscan.labels_, k=2, algo='dbscan')
```

```
<class 'collections.Counter'>
dict_items([(14, 8390), (4, 6765), (7, 14818), (9, 7438), (5, 7182), (2, 6989), (10, 7369), (1
cluster size = 15
*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
306512    The water is great but the price has me at a loss. I have been paying anywhere from $
204057    The Black Cherry Switch is an interesting beverage, a cross between the traditional c
175821    I have been hooked on crystal sugar like this ever since I used it when I worked for
228686    Arctic Zero ice cream is AMAZING. It's SO nice to finally have a fantastic tasting "I
273820    King's Cupboard Sugar Free Dark Chocolate Chunk Cocoa is the best I've ever had, incl
Name: Text, dtype: object
CLUSTER = 2
The Review Text in Cluster 1 is as follows:
358834    I've been eating energy and protein bars for years.  Most are like flavored cardboard
32904     These are the most addictive cookies on the market.  They are Very rich, Ultimately c
115655    I tried these bars yesterday for the first time. They're delicious and I don't get a
85223     Not a creamy chocolate - I like the edge - almost bitter taste in this chocolate - an
207493                 This is my favorite adult cereal - low in sugar and fats, high on crunc
Name: Text, dtype: object
CLUSTER = 3
The Review Text in Cluster 2 is as follows:
311384                                                                              Cafe
318265                                                I am a big fan of scharffen berger and the
222236    I was turned on to Rooibos tea about 6 months ago and have enjoyed the taste.  Lookin
242252                                                                  Received quickly a
236062    If you're trying to avoid sweets over the holidays, but drink coffee and like chocola
Name: Text, dtype: object
CLUSTER = 4
The Review Text in Cluster 3 is as follows:
255523    I can recognize bad popcorn but either I've never had great popcorn or I'm just not t
333694    I have a popcorn machine rental business and everyone who rents from me all tell me I
75000
```

276235   If you find yourself spending tons on those little spice-bottles of sesame seeds at t
8065   I love this oil.

My first attraction to it was based upon my research whic
Name: Text, dtype: object
CLUSTER = 5
The Review Text in Cluster 4 is as follows:
23280   I think this is probably as good as it gets for sugar free chocolate syrup. It's sti
311440   I was very cautious and cut off a small corner of the pepper and set it on my tongue
71765   This is my new favorite flavor of Lundberg Rice Chips. They are so tasty and addictiv
246216   The tea has a very good vanilla flavor, i
249289   These are delicious, especially for something that is sugar-free. They are light, wi
Name: Text, dtype: object
CLUSTER = 6
The Review Text in Cluster 5 is as follows:
245004   Never mind that this dog food is kind of gross looking - my dogs just love it! While
92406   I bought these as a treat for my dog and he started vomiting about an hour later. I
110069
115393   Started feeding my two older dogs (12 and 13) this food. My lab has always had dry it
331471   My dog, like all dogs, will eat anything. But my pooch would NOT eat this crap. Smart
Name: Text, dtype: object
CLUSTER = 7
The Review Text in Cluster 6 is as follows:
90125   My kids like Pirate Booty a lot. It is a great snack for lunches but I find that you l
37085   I got this pasta for my 14 month old and she loves it. I hated giving her the Kraft wi
37071   The
58773   This cereal is an excellent way to gain healthy weight. Each box has 1470 calories, a
12582   These are the best tasting pretzels I have found. Since they are not available in the
Name: Text, dtype: object
CLUSTER = 8
The Review Text in Cluster 7 is as follows:
171368   I love these cornflakes. I can't believe anyone would say they taste like cardboard'
18115   When I read a customer review which said that QUICK GRITS were sent when OLD FASHIONE
180735   I had been thinking for sometime to look for Graisse de Canard and where else to look
133845   I have tried another rice shell and liked it, but this one is closer to a "real" pizz
49234   I've always loved Thai yellow curries, but never had much luck trying to duplicate wh
Name: Text, dtype: object
CLUSTER = 9
The Review Text in Cluster 8 is as follows:
198671   I received my Keurig for Christmas and since then have tried as many black teas that
37311   I just tried orange spice tea for the first time on a recent trip to Hawaii, and they
284144   This Numi tea i
19187   The original flavor of Red Rose tea is one that I have enjoyed almost all of my life
35444   This tea is amazing!

The first thing I noticed was the strong scent of cin
Name: Text, dtype: object
CLUSTER = 10
The Review Text in Cluster 9 is as follows:
63408
190346
363410   This is my favorite coconut milk. I use

```
168038    This is definitely THE cereal to get.  It's good for you and actually tastes good.  I
24827     These are a staple in my house.  Great to have them shipped in rather than trying to
Name: Text, dtype: object
CLUSTER = 11
The Review Text in Cluster 10 is as follows:
77439     Started buying this product 4 months ago after the whole "Canidae" issue (Im sure tha
256827    Our labrador has an iron constitution. She has eaten a broad variety of food (and otl
317914
218656    With the first swipe, you will see it work. The fur just comes off!<br />While brushi
357438    I was very please to have been introduced and found this wonderful, refreshing healtl
Name: Text, dtype: object
CLUSTER = 12
The Review Text in Cluster 11 is as follows:
39914     I got this pack of cookies for my mom she cant have gluten she loves them so much she
92619     Gluten-free or not, this is the best cornbread mix I have ever had - period.  Even my
232993    I was expecting for this cake to turn out as dense and as hard as a rock, but boy, wh
304073    I bought this to try as a replacement for a combination of whole wheat flour and mult
3220      Okay, I know it sugar free, but does that mean it also has to be flavor free?  A ging
Name: Text, dtype: object
CLUSTER = 13
The Review Text in Cluster 12 is as follows:
55966     I used to eat these as a kid growing up in Frankfurt, Germany.  When I recieved my or
190726
18220     I don't know what others got, but I was shipped a box of teeny tiny little pig ears i
104158    Received can with large dent. The box was in good shape, so can was damaged before sh
172556
Name: Text, dtype: object
CLUSTER = 14
The Review Text in Cluster 13 is as follows:
240280    I ordered Dark Sumatra Gayoland coffee bean from Coffee Bean Direct via Amazon. It ca
346142    SF Bay Coffee's Fog Chaser was a nice "bold" surprise.  I am a big fan of Amazon's Su
193386    This is the greatest coffee but where did it go?  Why did Folgers change their planog
340715    I love this French Vanilla cappuccino!  It is exactly what I was looking for to make
242934    I got this Illy Cappuccino drink for my coffee-fanatic husband, but only had a taste
Name: Text, dtype: object
CLUSTER = 15
The Review Text in Cluster 14 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is i
347980    The ingredients are listed below (taken from the product package and website):<br /><
156014    So I guess I'm the first to try this on this site. So I shall tell you my experience
258023                                                                                  Wow,
101418    After reading the rave reviews about these noodles for months & months.. I finally to
Name: Text, dtype: object


<class 'collections.Counter'>
dict_items([(0, 4002), (1, 998)])
cluster size = 2
```

***** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: *****

CLUSTER = 1

The Review Text in Cluster 0 is as follows:

230993 This was a gift for a coffee connoisseur friend who was re
197476 Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surprisingly, t
77376 I have always loved ghee with everything I prepare and eat - be it my daily dose of i
112553 We have a 4 oz stand up electric popcorn popper and find 4 oz bags a little hard to
192872 I ordered these when they went on sale. At the sale price, they were about the same p
Name: Text, dtype: object

CLUSTER = 2

The Review Text in Cluster 1 is as follows:

343150 I love these ba
121608 I go through alot of these bags and they are great! I love the colors and the size is
166163 I live in the USA and I love the original HP sauce from England. I had found an Ameri
343053 I have tried several bags including Lanisoh and Medela and these are the only bags
48532 Both of my large breed dogs love Canidae. I actually have to limit how much they can
Name: Text, dtype: object

%%% Hierarchical Clustering with 5 Clusters %%%

<class 'collections.Counter'>

dict_items([(0, 2447), (1, 802), (2, 1224), (4, 331), (3, 196)])

cluster size = 5

***** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: *****

CLUSTER = 1

The Review Text in Cluster 0 is as follows:

230993 This was a gift for a coffee connoisseur friend who was re
197476 Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surprisingly, t
77376 I have always loved ghee with everything I prepare and eat - be it my daily dose of i
112553 We have a 4 oz stand up electric popcorn popper and find 4 oz bags a little hard to
161429 These flowers arrived to my door on time, and as expected they were in bud form. So,
Name: Text, dtype: object

CLUSTER = 2

The Review Text in Cluster 1 is as follows:

343150 I love these ba
121608 I go through alot of these bags and they are great! I love the colors and the size is
166163 I live in the USA and I love the original HP sauce from England. I had found an Ameri
343053 I have tried several bags including Lanisoh and Medela and these are the only bags
55255 I made this bread with my bread maker - except I made it into rolls. It is a gre
Name: Text, dtype: object

CLUSTER = 3

The Review Text in Cluster 2 is as follows:

192872 I ordered these when they went on sale. At the sale price, they were about the same p
11279 I used canola oil inst
45809 I'm trying to reduce the amount of sugary snacks in my life but when I get the 'jones
65095 When you are on a strict diet like the hcg diet, you look for variety, and these are
273188 I like my coffee, but I'm no coffee-ophile or whatever the correct term might be. I l
Name: Text, dtype: object

```
CLUSTER = 4
The Review Text in Cluster 3 is as follows:
48532     Both of my large breed dogs love Canidae.  I actually have to limit how much they ca
218825    i was skeptical at first, i had been a loyal science diet customer for years.  One o
219740    . . . I'd have her write the review!  But since she can't speak, I can only speculat
363521                    My dog absolutely loves these treats and he goes nuts every time I
253397    I received a free bag of these when I went down for the Westminster Dog Show this pa
Name: Text, dtype: object
CLUSTER = 5
The Review Text in Cluster 4 is as follows:
308605                                                                    Great tast
193231    I really enjoy black tea. This is from England and I think they make some of the bes
317816    If your green tea isn't green, it's overly oxidized and probably doesn't taste great
298943    Simply put...I love this coffee!!! It's my favorite!!! Hard to find sometimes, so I
250442    I got tired of overpaying for tea at Teavana so decided to try this brand of Oolong
Name: Text, dtype: object


<class 'collections.Counter'>
dict_items([(0, 84671), (-1, 15329)])
cluster size = 2
*** CLUSTERS FORMED BY dbscan ALGORITHM is as follows: ***
CLUSTER = 0
The Review Text in Cluster -1 is as follows:
63408
363410                                                This is my favorite coconut milk. I us
110069
306512    The water is great but the price has me at a loss. I have been paying anywhere from
37071                                                                          T
Name: Text, dtype: object
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's st
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  Whil
92406     I bought these as a treat for my dog and he started vomiting about an hour later.  I
Name: Text, dtype: object
%%% DBSCAN Clustering with Eps = 1 %%%


<class 'collections.Counter'>
dict_items([(-1, 100000)])
cluster size = 1
*** CLUSTERS FORMED BY dbscan ALGORITHM is as follows: ***
CLUSTER = 0
The Review Text in Cluster -1 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is
```

```
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard'
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
Name: Text, dtype: object
%%% DBSCAN Clustering with Eps = 50 %%%


<class 'collections.Counter'>
dict_items([(0, 99829), (-1, 171)])
cluster size = 2
*** CLUSTERS FORMED BY dbscan ALGORITHM is as follows: ***
CLUSTER = 0
The Review Text in Cluster -1 is as follows:
104631                                I buy these for my baby (17months). you can melt them in a
247948    I was happy with purchase and recommend it to any one who is thinking about it.  Try
36799                                              I bought these for my 14-month old, but my
250450              Good stuff.  Have purchased twice now. Can't use real half & half and most
7419                                                                        this seller was
Name: Text, dtype: object
CLUSTER = 1
The Review Text in Cluster 0 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend it, it is i
23280     I think this is probably as good as it gets for sugar free chocolate syrup.  It's sti
171368    I love these cornflakes.  I can't believe anyone would say they taste like cardboard'
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love it!  While
Name: Text, dtype: object
```

12 Observations

1) All the **clusters are formed based on word (or contextual similarities) and NOT on +ve or -ve review rating** as they are not given 'y' values as input, while clustering.

2) The analysis of clusters formed using 4 featurizations are done.

A) BoW

K-Means:

Cluster 0: most reviews about taste of food derived from flavour.
Cluster 1: reviews focussed on 'work' environment products. Eg: office, work, colleagues, receptionist etc are repeated.
Cluster 2: groups reviews related to food. The repeated words are food, sugar, flours, oil etc.

Hierarchical:

The clustering is not meaningful as all points except 1, is grouped into 1 single cluster.

B) tf-idf

K-Means:

Cluster 0: customers are in dilemma. Whether the effectiveness is +ve or -ve or just placebo.
Cluster 1: talks about illness and effectiveness of medicines. Many medical terminologies.
Cluster 2: all reviews are about sound and equipments related to sound. Eg: mic, icicle, jack etc.
Cluster 3: groups reviews related to food. The repeated words are food, sugar, flours, oil etc.

Hierarchical:

The clustering is not meaningful as all points except 1, is grouped into 1 single cluster.

C) Word2Vec:

K-Means:

From the cluster groups, it can be seen that the **reviews obtained from kmeans clustering are more distributed.**

Cluster 0: Reviews are going in-depth about using the purchased product for cooking. Eg: noodies, oil, chicken, ice cream, melons etc.
Cluster 1: There are some negative words (reviews) repeated, in this group. Some breakfast products are clubbed in this group. Eg: corn-flakes, peanut butter etc.
Cluster 2: This group is all about drinks. Eg: coffee, chai, latte, cups, drink, chocolate etc are repeated.
Cluster 3: Extremely positive reviews. Most of the reviews are about products which are rarely available in the market, but only in Amazon. Logically, as customers were able to find such 'hard-gets', they are extremely happy.
Cluster 4: This group focus on delivery and damage caused for the shipment.
Cluster 5: Most of the reviews are about bakery food items. Product reviews are about chocolate syrup, scones, shortbread, ice cream, carbonated drinks and fruit juice etc.
Cluster 6: Groups products available in amazon, vis-a-vis offline stores.
Cluster 7: This group is all about tea and tea products.
Cluster 8: This group is all about pets, mostly dog and cats.

Cluster 9: Group focus on energy drinks and health drinks.
Cluster 10: Groups reviews about Bread and associated combinations.
Cluster 11: Very personal opinion about the products are shared.
Cluster 12: This group is all about dog food and cat food.
Cluster 13: Groups reviews about healthy related products and meal replacements. Eg: protein bars, energy bars, pirate booty, healthy cereal etc.
Clusters 14: Snacks & toffees are grouped. Eg: rice chips, toffee, pepper, berger, chocolate etc.

Hierarchical:

The first 2 clusters formed when K=2 & K=5 are similar.

Cluster 0: Groups reviews about snacks and sauces.
Cluster 1: Groups cuisines of different cultures.
Cluster 2: Groups tea and coffee reviews.
Cluster 3: Nothing found in common.
Clsuter 4: Groups dog and cat foods.

DBSCAN:

The Eps value is found out using Min Points value. 2 Clusters are formed while using the computed Eps value. One cluster has id of -1, which means they are identified as noise. When Eps value is reduced all the points are identified as noise, whereas, when the Eps value is increased, then number of noise points drastically reduced.

D) tf-idf W2V:

K-Means:

The 15 clusters formed via tf-idf weighted W2V vectors have **similar grouping pattern compared to W2V vectors. The cluster separation may be slightly more meaningful** than using W2V alone, but they are not significantly better.

Hierarchical:

The 5 clusters formed are **similar to the groups obtained from W2V method, but they are more meaningfully separated**. For instance, Cluster 3 talks only about dogs and dog foods, whil Cluster 4 contains reviews about different variants of tea, such as black tea, green tea etc.

DBSCAN:

The DBSCAN results shows similar results as with W2V vector. When Eps value is reduced all the points are identified as noise, whereas, when the Eps value is increased, noise points are reduced.

3) From the above analysis, it can deduced that **K-Means algorithm on TF-ID Weighted W2V or Word2Vec is the clustering algorithm of choice.**