Amazon Fine Food Reviews Analysis

Data Source: https://www.kaggle.com/snap/amazon-fine-food-reviews

EDA: https://nycdatascience.com/blog/student-works/amazon-fine-foods-visualization/

The Amazon Fine Food Reviews dataset consists of reviews of fine foods from Amazon.

Number of reviews: 568,454
Number of users: 256,059
Number of products: 74,258
Timespan: Oct 1999 - Oct 2012
Number of Attributes/Columns in data: 10

Attribute Information:

1. Id
2. ProductId - unique identifier for the product
3. UserId - unqiue identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text - text of the review

Objective:

Given a review, determine whether the review is positive (rating of 4 or 5) or negative (rating of 1 or 2).

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[Q] How to determine if a review is positive or negative?

[Ans] We could use Score/Rating. A rating of 4 or 5 can be cosnidered as a positive review. A rating of 1 or 2 can be considered as negative one. A review of rating 3 is considered nuetral and such reviews are ignored from our analysis. This is an approximate and proxy way of determining the polarity (positivity/negativity) of a review.

[1]. Reading Data

[1.1] Loading the data

The dataset is available in two forms

1. .csv file
2. SQLite Database

In order to load the data, We have used the SQLITE dataset as it is easier to query the data and visualise the data efficiently.

Here as we only want to get the global sentiment of the recommendations (positive or negative), we will purposefully ignore all Scores equal to 3. If the score is above 3, then the recommendation wil be set to "positive". Otherwise, it will be set to "negative".

In [17]:
```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")


import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle

from tqdm import tqdm
import os
```

In [2]:

```
# using SQLite Table to read data.
con = sqlite3.connect('database.sqlite')

# filtering only positive and negative reviews i.e.
# not taking into consideration those reviews with Score=3
# SELECT * FROM Reviews WHERE Score != 3 LIMIT 500000, will give top 50
0000 data points
# you can change the number to any other number based on your computing
 power

# filtered_data = pd.read_sql_query(""" SELECT * FROM Reviews WHERE Sco
re != 3 LIMIT 500000""", con)
# for tsne assignment you can take 5k data points
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
filtered_data = pd.read_sql_query("""  SELECT * FROM Reviews WHERE Score
 != 3 LIMIT 5000""", con)

# Give reviews with Score>3 a positive rating(1), and reviews with a sc
ore<3 a negative rating(0).
def partition(x):
    if x < 3:
        return 0
    return 1

#changing reviews with score less than 3 to be positive and vice-versa
actualScore = filtered_data['Score']
positiveNegative = actualScore.map(partition)
filtered_data['Score'] = positiveNegative
print("Number of data points in our data", filtered_data.shape)
filtered_data.head(3)
```

```
Number of data points in our data (5000, 10)
```

Out[2]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	Helpfulnes
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	Helpfulnes
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1

In [3]:
```
display = pd.read_sql_query("""
SELECT UserId, ProductId, ProfileName, Time, Score, Text, COUNT(*)
FROM Reviews
GROUP BY UserId
HAVING COUNT(*)>1
""", con)
```

In [4]:
```
print(display.shape)
display.head()
```

```
(80668, 7)
```

Out[4]:

	UserId	ProductId	ProfileName	Time	Score	Text	COU
0	#oc-R115TNMSPFT9I7	B007Y59HVM	Breyton	1331510400	2	Overall its just OK when considering the price...	2
1	#oc-R11D9D7SHXIJB9	B005HG9ET0	Louis E. Emory "hoppy"	1342396800	5	My wife has recurring extreme muscle spasms, u...	3

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	UserId	ProductId	ProfileName	Time	Score	Text	COU
2	#oc-R11DNU2NBKQ23Z	B007Y59HVM	Kim Cieszykowski	1348531200	1	This coffee is horrible and unfortunately not ...	2
3	#oc-R11O5J5ZVQE25C	B005HG9ET0	Penguin Chick	1346889600	5	This will be the bottle that you grab from the...	3
4	#oc-R12KPBODL2B5ZD	B007OSBE1U	Christopher P. Presta	1348617600	1	I didnt like this coffee. Instead of telling y...	2

In [5]:
```
display[display['UserId']=='AZY10LLTJ71NX']
```

Out[5]:

	UserId	ProductId	ProfileName	Time	Score	Text
80638	AZY10LLTJ71NX	B006P7E5ZI	undertheshrine "undertheshrine"	1334707200	5	I was recommended to try green tea extract to ...

In [6]:
```
display['COUNT(*)'].sum()
```

Out[6]: 393063

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[2] Exploratory Data Analysis

[2.1] Data Cleaning: Deduplication

It is observed (as shown in the table below) that the reviews data had many duplicate entries. Hence it was necessary to remove duplicates in order to get unbiased results for the analysis of the data. Following is an example:

In [7]:
```
display= pd.read_sql_query("""
SELECT *
FROM Reviews
WHERE Score != 3 AND UserId="AR5J8UI46CURR"
ORDER BY ProductID
""", con)
display.head()
```

Out[7]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	Helpfuln
0	78445	B000HDL1RQ	AR5J8UI46CURR	Geetha Krishnan	2	2
1	138317	B000HDOPYC	AR5J8UI46CURR	Geetha Krishnan	2	2

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	Helpfuln
2	138277	B000HDOPYM	AR5J8UI46CURR	Geetha Krishnan	2	2
3	73791	B000HDOPZG	AR5J8UI46CURR	Geetha Krishnan	2	2
4	155049	B000PAQ75C	AR5J8UI46CURR	Geetha Krishnan	2	2

As it can be seen above that same user has multiple reviews with same values for HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary and Text and on doing analysis it was found that

ProductId=B000HDOPZG was Loacker Quadratini Vanilla Wafer Cookies, 8.82-Ounce Packages (Pack of 8)

ProductId=B000HDL1RQ was Loacker Quadratini Lemon Wafer Cookies, 8.82-Ounce Packages (Pack of 8) and so on

It was inferred after analysis that reviews with same parameters other than ProductId belonged to the same product just having different flavour or quantity. Hence in order to reduce redundancy it was decided to eliminate the rows having same parameters.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

The method used for the same was that we first sort the data according to ProductId and then just keep the first similar product review and delelte the others. for eg. in the above just the review for ProductId=B000HDL1RQ remains. This method ensures that there is only one representative for each product and deduplication without sorting would lead to possibility of different representatives still existing for the same product.

In [8]:
```
#Sorting data according to ProductId in ascending order
sorted_data=filtered_data.sort_values('ProductId', axis=0, ascending=Tr
ue, inplace=False, kind='quicksort', na_position='last')
```

In [9]:
```
#Deduplication of entries
final=sorted_data.drop_duplicates(subset={"UserId","ProfileName","Time"
,"Text"}, keep='first', inplace=False)
final.shape
```

Out[9]:
```
(4986, 10)
```

In [10]:
```
#Checking to see how much % of data still remains
(final['Id'].size*1.0)/(filtered_data['Id'].size*1.0)*100
```

Out[10]:
```
99.72
```

Observation:- It was also seen that in two rows given below the value of HelpfulnessNumerator is greater than HelpfulnessDenominator which is not practically possible hence these two rows too are removed from calcualtions

In [11]:
```
display= pd.read_sql_query("""
SELECT *
FROM Reviews
WHERE Score != 3 AND Id=44737 OR Id=64422
ORDER BY ProductID
""", con)

display.head()
```

Out[11]:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	Helpfuln
0	64422	B000MIDROQ	A161DK06JJMCYF	J. E. Stephens "Jeanne"	3	1
1	44737	B001EQ55RW	A2V0I904FH7ABY	Ram	3	2

In [12]:
```
final=final[final.HelpfulnessNumerator<=final.HelpfulnessDenominator]
```

In [13]:
```
#Before starting the next phase of preprocessing lets see the number of
 entries left
print(final.shape)

#How many positive and negative reviews are present in our dataset?
final['Score'].value_counts()
```

```
(4986, 10)
```

Out[13]:
```
1    4178
0     808
Name: Score, dtype: int64
```

[3] Preprocessing

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[3.1]. Preprocessing Review Text

Now that we have finished deduplication our data requires some preprocessing before we go on further with analysis and making the prediction model.

Hence in the Preprocessing phase we do the following in the order below:-

1. Begin by removing the html tags
2. Remove any punctuations or limited set of special characters like , or . or # etc.
3. Check if the word is made up of english letters and is not alpha-numeric
4. Check to see if the length of the word is greater than 2 (as it was researched that there is no adjective in 2-letters)
5. Convert the word to lowercase
6. Remove Stopwords
7. Finally Snowball Stemming the word (it was obsereved to be better than Porter Stemming)

After which we collect the words used to describe positive and negative reviews

In [14]:
```
# printing some random reviews
sent_0 = final['Text'].values[0]
print(sent_0)
print("="*50)

sent_1000 = final['Text'].values[1000]
print(sent_1000)
print("="*50)

sent_1500 = final['Text'].values[1500]
print(sent_1500)
print("="*50)

sent_4900 = final['Text'].values[4900]
print(sent_4900)
print("="*50)
```

```
Why is this $[...] when the same product is available for $[...] here?<
br />http://www.amazon.com/VICTOR-FLY-MAGNET-BAIT-REEL/dp/B00004RBDY<
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
br /><br />The Victor M380 and M502 traps are unreal, of course -- tota
l fly genocide. Pretty stinky, but only right nearby.
==================================================
I recently tried this flavor/brand and was surprised at how delicious t
hese chips are.  The best thing was that there were a lot of "brown" ch
ips in the bsg (my favorite), so I bought some more through amazon and
shared with family and friends.  I am a little disappointed that there
are not, so far, very many brown chips in these bags, but the flavor is
still very good.  I like them better than the yogurt and green onion fl
avor because they do not seem to be as salty, and the onion flavor is b
etter.  If you haven't eaten Kettle chips before, I recommend that you
try a bag before buying bulk.  They are thicker and crunchier than Lays
but just as fresh out of the bag.
==================================================
Wow.  So far, two two-star reviews.  One obviously had no idea what the
y were ordering; the other wants crispy cookies.  Hey, I'm sorry; but t
hese reviews do nobody any good beyond reminding us to look  before ord
ering.<br /><br />These are chocolate-oatmeal cookies.  If you don't li
ke that combination, don't order this type of cookie.  I find the combo
quite nice, really.  The oatmeal sort of "calms" the rich chocolate fla
vor and gives the cookie sort of a coconut-type consistency.  Now let's
also remember that tastes differ; so, I've given my opinion.<br /><br /
>Then, these are soft, chewy cookies -- as advertised.  They are not "c
rispy" cookies, or the blurb would say "crispy," rather than "chewy."
I happen to like raw cookie dough; however, I don't see where these tas
te like raw cookie dough.  Both are soft, however, so is this the confu
sion?  And, yes, they stick together.  Soft cookies tend to do that.  T
hey aren't individually wrapped, which would add to the cost.  Oh yeah,
chocolate chip cookies tend to be somewhat sweet.<br /><br />So, if you
want something hard and crisp, I suggest Nabiso's Ginger Snaps.  If you
want a cookie that's soft, chewy and tastes like a combination of choco
late and oatmeal, give these a try.  I'm here to place my second order.
==================================================
love to order my coffee on amazon.  easy and shows up quickly.<br />Thi
s k cup is great coffee.  dcaf is very good as well
==================================================
```

In [15]:

```
# remove urls from text python: https://stackoverflow.com/a/40823105/40
84039
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
sent_0 = re.sub(r"http\S+", "", sent_0)
sent_1000 = re.sub(r"http\S+", "", sent_1000)
sent_150 = re.sub(r"http\S+", "", sent_1500)
sent_4900 = re.sub(r"http\S+", "", sent_4900)

print(sent_0)
```

```
Why is this $[...] when the same product is available for $[...] here?<
br /> /><br />The Victor M380 and M502 traps are unreal, of course -- t
otal fly genocide. Pretty stinky, but only right nearby.
```

In [16]:

```
# https://stackoverflow.com/questions/16206380/python-beautifulsoup-how
-to-remove-all-tags-from-an-element
from bs4 import BeautifulSoup

soup = BeautifulSoup(sent_0, 'lxml')
text = soup.get_text()
print(text)
print("="*50)

soup = BeautifulSoup(sent_1000, 'lxml')
text = soup.get_text()
print(text)
print("="*50)

soup = BeautifulSoup(sent_1500, 'lxml')
text = soup.get_text()
print(text)
print("="*50)

soup = BeautifulSoup(sent_4900, 'lxml')
text = soup.get_text()
print(text)
```

```
Why is this $[...] when the same product is available for $[...] here?
/>The Victor M380 and M502 traps are unreal, of course -- total fly gen
ocide. Pretty stinky, but only right nearby.
==================================================
I recently tried this flavor/brand and was surprised at how delicious t
hese chips are.  The best thing was that there were a lot of "brown" ch
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ips in the bsg (my favorite), so I bought some more through amazon and
shared with family and friends.  I am a little disappointed that there
are not, so far, very many brown chips in these bags, but the flavor is
still very good.  I like them better than the yogurt and green onion fl
avor because they do not seem to be as salty, and the onion flavor is b
etter.  If you haven't eaten Kettle chips before, I recommend that you
try a bag before buying bulk.  They are thicker and crunchier than Lays
but just as fresh out of the bag.
==================================================
Wow.  So far, two two-star reviews.  One obviously had no idea what the
y were ordering; the other wants crispy cookies.  Hey, I'm sorry; but t
hese reviews do nobody any good beyond reminding us to look  before ord
ering.These are chocolate-oatmeal cookies.  If you don't like that comb
ination, don't order this type of cookie.  I find the combo quite nice,
really.  The oatmeal sort of "calms" the rich chocolate flavor and give
s the cookie sort of a coconut-type consistency.  Now let's also rememb
er that tastes differ; so, I've given my opinion.Then, these are soft,
chewy cookies -- as advertised.  They are not "crispy" cookies, or the
blurb would say "crispy," rather than "chewy."  I happen to like raw co
okie dough; however, I don't see where these taste like raw cookie doug
h.  Both are soft, however, so is this the confusion?  And, yes, they s
tick together.  Soft cookies tend to do that.  They aren't individually
wrapped, which would add to the cost.  Oh yeah, chocolate chip cookies
tend to be somewhat sweet.So, if you want something hard and crisp, I s
uggest Nabiso's Ginger Snaps.  If you want a cookie that's soft, chewy
and tastes like a combination of chocolate and oatmeal, give these a tr
y.  I'm here to place my second order.
==================================================
love to order my coffee on amazon.  easy and shows up quickly.This k cu
p is great coffee.  dcaf is very good as well
```

In [17]:

```python
# https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

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

In [18]:

```
sent_1500 = decontracted(sent_1500)
print(sent_1500)
print("="*50)
```

```
Wow.  So far, two two-star reviews.  One obviously had no idea what the
y were ordering; the other wants crispy cookies.  Hey, I am sorry; but
these reviews do nobody any good beyond reminding us to look  before or
dering.<br /><br />These are chocolate-oatmeal cookies.  If you do not
like that combination, do not order this type of cookie.  I find the co
mbo quite nice, really.  The oatmeal sort of "calms" the rich chocolate
flavor and gives the cookie sort of a coconut-type consistency.  Now le
t is also remember that tastes differ; so, I have given my opinion.<br
/><br />Then, these are soft, chewy cookies -- as advertised.  They are
not "crispy" cookies, or the blurb would say "crispy," rather than "che
wy."  I happen to like raw cookie dough; however, I do not see where th
ese taste like raw cookie dough.  Both are soft, however, so is this th
e confusion?  And, yes, they stick together.  Soft cookies tend to do t
hat.  They are not individually wrapped, which would add to the cost.
Oh yeah, chocolate chip cookies tend to be somewhat sweet.<br /><br />S
o, if you want something hard and crisp, I suggest Nabiso is Ginger Sna
ps.  If you want a cookie that is soft, chewy and tastes like a combina
tion of chocolate and oatmeal, give these a try.  I am here to place my
second order.
==================================================
```

In [19]:

```
#remove words with numbers python: https://stackoverflow.com/a/1808237
0/4084039
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
sent_0 = re.sub("\S*\d\S*", "", sent_0).strip()
print(sent_0)
```

```
Why is this $[...] when the same product is available for $[...] here?<
br /> /><br />The Victor  and  traps are unreal, of course -- total fly
genocide. Pretty stinky, but only right nearby.
```

In [20]:

```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent_1500 = re.sub('[^A-Za-z0-9]+', ' ', sent_1500)
print(sent_1500)
```

```
Wow So far two two star reviews One obviously had no idea what they wer
e ordering the other wants crispy cookies Hey I am sorry but these revi
ews do nobody any good beyond reminding us to look before ordering br b
r These are chocolate oatmeal cookies If you do not like that combinati
on do not order this type of cookie I find the combo quite nice really
The oatmeal sort of calms the rich chocolate flavor and gives the cooki
e sort of a coconut type consistency Now let is also remember that tast
es differ so I have given my opinion br br Then these are soft chewy co
okies as advertised They are not crispy cookies or the blurb would say
crispy rather than chewy I happen to like raw cookie dough however I do
not see where these taste like raw cookie dough Both are soft however s
o is this the confusion And yes they stick together Soft cookies tend t
o do that They are not individually wrapped which would add to the cost
Oh yeah chocolate chip cookies tend to be somewhat sweet br br So if yo
u want something hard and crisp I suggest Nabiso is Ginger Snaps If you
want a cookie that is soft chewy and tastes like a combination of choco
late and oatmeal give these a try I am here to place my second order
```

In [21]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'no
t'
# <br /><br /> ==> after the above steps, we are getting "br br"
# we are including them into stop words list
# instead of <br /> if we have <br/> these tags would have revmoved in
 the 1st step

stopwords= set(['br', 'the', 'i', 'me', 'my', 'myself', 'we', 'our', 'o
urs', 'ourselves', 'you', "you're", "you've",\
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselve
s', 'he', 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'it
s', 'itself', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'th
is', 'that', "that'll", 'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'h
ave', 'has', 'had', 'having', 'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or',
 'because', 'as', 'until', 'while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between',
'into', 'through', 'during', 'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'h
ow', 'all', 'any', 'both', 'each', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 's
o', 'than', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should',
"should've", 'now', 'd', 'll', 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",
'didn', "didn't", 'doesn', "doesn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "is
n't", 'ma', 'mightn', "mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn',
 "shouldn't", 'wasn', "wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"])
```

In [22]:

```
# Combining all the above stundents
from tqdm import tqdm
preprocessed_reviews = []
# tqdm is for printing the status bar
for sentance in tqdm(final['Text'].values):
    sentance = re.sub(r"http\S+", "", sentance)
    sentance = BeautifulSoup(sentance, 'lxml').get_text()
    sentance = decontracted(sentance)
    sentance = re.sub("\S*\d\S*", "", sentance).strip()
    sentance = re.sub('[^A-Za-z]+', ' ', sentance)
    # https://gist.github.com/sebleier/554280
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    sentance = ' '.join(e.lower() for e in sentance.split() if e.lower
() not in stopwords)
    preprocessed_reviews.append(sentance.strip())
```

```
100%|
| 4986/4986 [00:01<00:00, 3137.37it/s]
```

In [23]:
```
preprocessed_reviews[1500]
```

Out[23]:
```
'wow far two two star reviews one obviously no idea ordering wants cris
py cookies hey sorry reviews nobody good beyond reminding us look order
ing chocolate oatmeal cookies not like combination not order type cooki
e find combo quite nice really oatmeal sort calms rich chocolate flavor
gives cookie sort coconut type consistency let also remember tastes dif
fer given opinion soft chewy cookies advertised not crispy cookies blur
b would say crispy rather chewy happen like raw cookie dough however no
t see taste like raw cookie dough soft however confusion yes stick toge
ther soft cookies tend not individually wrapped would add cost oh yeah
chocolate chip cookies tend somewhat sweet want something hard crisp su
ggest nabiso ginger snaps want cookie soft chewy tastes like combinatio
n chocolate oatmeal give try place second order'
```

[3.2] Preprocessing Review Summary

In [6]:
```
## Similartly you can do preprocessing for review summary also.
```

[4] Featurization

[4.1] BAG OF WORDS

In [25]:
```
#BoW
count_vect = CountVectorizer() #in scikit-learn
count_vect.fit(preprocessed_reviews)
print("some feature names ", count_vect.get_feature_names()[:10])
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
print('='*50)

final_counts = count_vect.transform(preprocessed_reviews)
print("the type of count vectorizer ",type(final_counts))
print("the shape of out text BOW vectorizer ",final_counts.get_shape())
print("the number of unique words ", final_counts.get_shape()[1])
```

```
some feature names  ['aa', 'aahhhs', 'aback', 'abandon', 'abates', 'abb
ott', 'abby', 'abdominal', 'abiding', 'ability']
==================================================
the type of count vectorizer  <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text BOW vectorizer  (4986, 12997)
the number of unique words  12997
```

[4.2] Bi-Grams and n-Grams.

In [26]:

```
#bi-gram, tri-gram and n-gram

#removing stop words like "not" should be avoided before building n-gra
ms
# count_vect = CountVectorizer(ngram_range=(1,2))
# please do read the CountVectorizer documentation http://scikit-learn.
org/stable/modules/generated/sklearn.feature_extraction.text.CountVecto
rizer.html

# you can choose these numebrs min_df=10, max_features=5000, of your ch
oice
count_vect = CountVectorizer(ngram_range=(1,2), min_df=10, max_features
=5000)
final_bigram_counts = count_vect.fit_transform(preprocessed_reviews)
print("the type of count vectorizer ",type(final_bigram_counts))
print("the shape of out text BOW vectorizer ",final_bigram_counts.get_s
hape())
print("the number of unique words including both unigrams and bigrams "
, final_bigram_counts.get_shape()[1])
```

```
the type of count vectorizer  <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text BOW vectorizer  (4986, 3144)
the number of unique words including both unigrams and bigrams  3144
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
the number of unique words including both unigrams and bigrams  3144
```

[4.3] TF-IDF

In [27]:
```
tf_idf_vect = TfidfVectorizer(ngram_range=(1,2), min_df=10)
tf_idf_vect.fit(preprocessed_reviews)
print("some sample features(unique words in the corpus)",tf_idf_vect.ge
t_feature_names()[0:10])
print('='*50)

final_tf_idf = tf_idf_vect.transform(preprocessed_reviews)
print("the type of count vectorizer ",type(final_tf_idf))
print("the shape of out text TFIDF vectorizer ",final_tf_idf.get_shape
())
print("the number of unique words including both unigrams and bigrams "
, final_tf_idf.get_shape()[1])
```

```
some sample features(unique words in the corpus) ['ability', 'able', 'a
ble find', 'able get', 'absolute', 'absolutely', 'absolutely deliciou
s', 'absolutely love', 'absolutely no', 'according']
==================================================
the type of count vectorizer  <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text TFIDF vectorizer  (4986, 3144)
the number of unique words including both unigrams and bigrams  3144
```

[4.4] Word2Vec

In [28]:
```
# Train your own Word2Vec model using your own text corpus
i=0
list_of_sentance=[]
for sentance in preprocessed_reviews:
    list_of_sentance.append(sentance.split())
```

In [42]:
```
# Using Google News Word2Vectors

# in this project we are using a pretrained model by google
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# its 3.3G file, once you load this into your memory
# it occupies ~9Gb, so please do this step only if you have >12G of ram
# we will provide a pickle file wich contains a dict ,
# and it contains all our courpus words as keys and  model[word] as val
ues
# To use this code-snippet, download "GoogleNews-vectors-negative300.bi
n"
# from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edi
t
# it's 1.9GB in size.


# http://kavita-ganesan.com/gensim-word2vec-tutorial-starter-code/#.W17
SRFAzZPY
# you can comment this whole cell
# or change these varible according to your need

is_your_ram_gt_16g=False
want_to_use_google_w2v = False
want_to_train_w2v = True

if want_to_train_w2v:
    # min_count = 5 considers only words that occured atleast 5 times
    w2v_model=Word2Vec(list_of_sentance,min_count=5,size=50, workers=4)
    print(w2v_model.wv.most_similar('great'))
    print('='*50)
    print(w2v_model.wv.most_similar('worst'))

elif want_to_use_google_w2v and is_your_ram_gt_16g:
    if os.path.isfile('GoogleNews-vectors-negative300.bin'):
        w2v_model=KeyedVectors.load_word2vec_format('GoogleNews-vectors
-negative300.bin', binary=True)
        print(w2v_model.wv.most_similar('great'))
        print(w2v_model.wv.most_similar('worst'))
    else:
        print("you don't have gogole's word2vec file, keep want_to_trai
n_w2v = True, to train your own w2v ")
```

```
[('snack', 0.9951335191726685), ('calorie', 0.9946465492248535), ('wond
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
erful', 0.9946032166481018), ('excellent', 0.9944332838058472), ('espec
ially', 0.9941144585609436), ('baked', 0.9940600395202637), ('salted',
0.994047224521637), ('alternative', 0.9937226176261902), ('tasty', 0.99
36816692352295), ('healthy', 0.9936649799346924)]
==================================================
[('varieties', 0.9994194507598877), ('become', 0.9992934465408325), ('p
opcorn', 0.9992750883102417), ('de', 0.9992610216140747), ('miss', 0.99
92451071739197), ('melitta', 0.999218761920929), ('choice', 0.999210238
4567261), ('american', 0.9991837739944458), ('beef', 0.999178051948547
4), ('finish', 0.9991567134857178)]
```

In [36]:

```
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  3817
sample words  ['product', 'available', 'course', 'total', 'pretty', 'st
inky', 'right', 'nearby', 'used', 'ca', 'not', 'beat', 'great', 'receiv
ed', 'shipment', 'could', 'hardly', 'wait', 'try', 'love', 'call', 'ins
tead', 'removed', 'easily', 'daughter', 'designed', 'printed', 'use',
'car', 'windows', 'beautifully', 'shop', 'program', 'going', 'lot', 'fu
n', 'everywhere', 'like', 'tv', 'computer', 'really', 'good', 'idea',
'final', 'outstanding', 'window', 'everybody', 'asks', 'bought', 'mad
e']
```

[4.4.1] Converting text into vectors using Avg W2V, TFIDF-W2V

[4.4.1.1] Avg W2v

In [38]:

```
# average Word2Vec
# compute average word2vec for each review.
sent_vectors = []; # the avg-w2v for each sentence/review is stored in
 this list
for sent in tqdm(list_of_sentance): # for each review/sentence
    sent_vec = np.zeros(50) # as word vectors are of zero length 50, yo
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
u might need to change this to 300 if you use google's w2v
    cnt_words =0; # num of words with a valid vector in the sentence/re
view
    for word in sent: # for each word in a review/sentence
        if word in w2v_words:
            vec = w2v_model.wv[word]
            sent_vec += vec
            cnt_words += 1
    if cnt_words != 0:
        sent_vec /= cnt_words
    sent_vectors.append(sent_vec)
print(len(sent_vectors))
print(len(sent_vectors[0]))
```

```
100%|
| 4986/4986 [00:03<00:00, 1330.47it/s]
```

```
4986
50
```

[4.4.1.2] TFIDF weighted W2v

In [39]:
```
# S = ["abc def pqr", "def def def abc", "pqr pqr def"]
model = TfidfVectorizer()
tf_idf_matrix = model.fit_transform(preprocessed_reviews)
# we are converting a dictionary with word as a key, and the idf as a v
alue
dictionary = dict(zip(model.get_feature_names(), list(model.idf_)))
```

In [41]:
```
# TF-IDF weighted Word2Vec
tfidf_feat = model.get_feature_names() # tfidf words/col-names
# final_tf_idf is the sparse matrix with row= sentence, col=word and ce
ll_val = tfidf

tfidf_sent_vectors = []; # the tfidf-w2v for each sentence/review is st
ored in this list
row=0;
for sent in tqdm(list_of_sentance): # for each review/sentence
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    sent_vec = np.zeros(50) # as word vectors are of zero length
    weight_sum =0; # num of words with a valid vector in the sentence/r
eview
    for word in sent: # for each word in a review/sentence
        if word in w2v_words and word in tfidf_feat:
            vec = w2v_model.wv[word]
#             tf_idf = tf_idf_matrix[row, tfidf_feat.index(word)]
            # to reduce the computation we are
            # dictionary[word] = idf value of word in whole courpus
            # sent.count(word) = tf valeus of word in this review
            tf_idf = dictionary[word]*(sent.count(word)/len(sent))
            sent_vec += (vec * tf_idf)
            weight_sum += tf_idf
    if weight_sum != 0:
        sent_vec /= weight_sum
    tfidf_sent_vectors.append(sent_vec)
    row += 1
```

```
100%|
| 4986/4986 [00:20<00:00, 245.63it/s]
```

In [18]:
```
# using SQLite Table to read data.
con = sqlite3.connect('final.sqlite')
filtered_data = pd.read_sql_query("""  SELECT * FROM Reviews WHERE Score
 != 3 """, con)
```

In [41]:
```
##Sorting data according to Time in ascending order for Time Based Spli
tting
time_sorted_data = filtered_data.sort_values('Time', axis=0, ascending=
True, inplace=False, kind='quicksort', na_position='last')

time_sorted_data=time_sorted_data[:40000]

x = time_sorted_data['CleanedText'].values
```

In [20]:
```
x.shape
```

Out[20]:
```
(40000,)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.1] K-Means Clustering

[5.1.1] Applying K-Means Clustering on BOW, SET 1

In [21]:
```
# Please write all the code with proper documentation
#BoW
count_vect = CountVectorizer(min_df = 1000)
data = count_vect.fit_transform(x)
print("the type of count vectorizer :",type(data))
print("the shape of out text BOW vectorizer : ",data.get_shape())
print("the number of unique words :", data.get_shape()[1])
```

```
the type of count vectorizer : <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text BOW vectorizer :  (40000, 248)
the number of unique words : 248
```

In [22]:
```
from sklearn.cluster import KMeans

k_values = [2,3,4,5,6,7,8,9,10]
loss = []
for i in k_values:
    kmeans = KMeans(n_clusters=i, n_jobs=-1).fit(data)
    loss.append(kmeans.inertia_)
```

elbow method

In [23]:
```
# Draw Loss VS K values plot
plt.plot(k_values, loss)
plt.xlabel('K-values',size=14)
plt.ylabel('Loss',size=14)
plt.title('Loss VS K-values Plot\n',size=18)
plt.grid()
plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- OBSERVATION :- From above we can see that there is inflection at K =3. Befor it loss was decreasing faster as compared to the loss decreasing after it . So , the best value of K is 3

In [24]:
```
optimal_k = 3
# Variable that will be used in the conclusion
bow_means_k = optimal_k

# Implementing K-Means++ using optimal value of K
kmeans = KMeans(n_clusters=optimal_k, n_jobs=-1).fit(data)
```

In [25]:
```
reviews = time_sorted_data['Text'].values
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []

for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
        cluster1.append(reviews[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(reviews[i])
    else :
        cluster3.append(reviews[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
```

```
No. of reviews in Cluster-1 :  5539

No. of reviews in Cluster-2 :  33564

No. of reviews in Cluster-3 :  897
```

[5.1.2] Wordclouds of clusters obtained after applying k-means on BOW SET 1

In [33]:

```
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 The wine saver is great in so many ways. Obviously it's wonderful to b
e able to open a bottle of wine and have it stay good for a day or two,
especially if you just like one glass with dinner. This lets you open u
p and enjoy more expensive wine since you won't waste the whole rest of
the bottle. If you are interested in learning about wines, you'll proba
bly want to open up 2 or more wines at a time to compare them, and this
way you won't waste anything. The wine saver will keep the undrunk port
ion until the next time.  I do recommend putting the bottle in the frid
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

ge if you don't drink it the next day. Just pull the bottle out an hour before you want to drink it if it's red.

Review-2 :
 The spectra paste food colors are simply fantastic. They provide a depth and clarity of color that is impossible to achieve with liquid food colors. And the hues are beautiful and distinctive. The black and red, two colors which are very difficult to get when dying frostings, etc. are especially nice - probably the best of any manufacturer I have tried.<p>For those who had questions, I am not 100% certain about the sizes, but if the set is the same as mine, the bottles are 3/4 ounce. While that might not seem like much, all you need is a dot of this thick gel paste at the end of a toothpick to color large amounts of material. So there is plenty of coloring ability in each bottle - many, many times more than a same sized liquid food color bottle. <p>Highly recommended and at a really good price at the time of this review.

Review-3 :
 Beetlejuice is the story of ghosts (Alec Baldwin and Geena Davis) who can't get along with the family (Jeffrey Jones, Catherine O'Hara, and Winona Ryder) that moves into their house. The ghosts enlist Beetlejuice (Michael Keaton) to help them, and the fun really starts. The movie is directed by Tim Burton. Robert Goulet and Dick Cavett make cameos. The score is by Danny Elfman. This movie has a great cast and a great director. <p>If you're a fan of Burton's other movies, you'll especially like this movie. Like all his movies, it has his distinctive look and feel to it. But anyone can enjoy this far-fetched comedy. Being PG, it should be appropriate for most members of the family. The DVD is lacking some of the special features that I would like to see though - like outtakes, behing-the-scenes, etc...

["The wine saver is great in so many ways. Obviously it's wonderful to be able to open a bottle of wine and have it stay good for a day or two, especially if you just like one glass with dinner. This lets you open up and enjoy more expensive wine since you won't waste the whole rest of the bottle. If you are interested in learning about wines, you'll probably want to open up 2 or more wines at a time to compare them, and this way you won't waste anything. The wine saver will keep the undrunk portion until the next time. I do recommend putting the bottle in the

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
fridge if you don't drink it the next day. Just pull the bottle out an
hour before you want to drink it if it's red."]
```

first cluster first review

- it takes lot of time to plot all reviews in a single cluster.so,i plotted 1st review of the the every cluster

In [34]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

second cluster

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [36]:
```
# Three Reviews of cluster 2
count=1
m=[]
for i in range(3):
    print('Review-%d : \n %s\n'%(count,cluster2[i]))
    count +=1
m.append(cluster2[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [37]:
```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [38]:
```
# Three Reviews of cluster 3
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    print('Review-%d : \n %s\n'%(count,cluster3[i]))
    count +=1
m.append(cluster3[0])
```

```
Review-1 :
 For me, when the days get colder nothing is as rewarding as a simple c
up of hot tea. And for it's claimed immunity benefits, a basic green te
a is a common pick for maintaining a healthy natural balance during the
flu season. From previous experiences in tasting the Tazo brand, both o
f the bottled and boxed products, they have proven to be unsurpassed fo
r quality and flavor. Once I've tried their teas they immediately becam
e my drink of choice. <p>The Zen Green Tea Blend is a wonderful one tha
t has only a few ingredients with no artificial anything. And thankfull
y, doesn't boast the addition of fortified vitamins in some senseless a
mount. It truly is an enlightening blend of green tea, spearmint, lemon
grass and lemon verbena. Thus making it versatile refreshment for anyti
me of the day, whether it's right after meals or between meals, or just
before bedtime. Generally light and mild tasting, but that will depend
upon how long you steep it and if you add a sweetener of some form.<p>I
nteresting too, are the amusing comments and remarks that appear on the
packaging. Reading through this as you drink your tea makes it a distin
ctive experience. I never seen tea so clever! I wonder if consuming Taz
o really does improve a person's outlook on life and affect his or her
well being? I think it just could be the great taste and aroma and prob
ably the reassuring thought of doing your health a favor.<br />Try a co
uple of their teas - they are sure to please!<br />And Tazo Zen Green T
ea makes a good choice.

Review-2 :
 Blend a little apple cider mix into your Tazo Passion Tea and see what
happens. This creates a magical apple cider experience with hints of cr
anberry and hibiscus flowers gathered in an ancient land.<br /><br />Sn
uggle Up Spiced Cider is a good choice because you can then spoon in th
e exact amount you desire, but packets of cider work equally well. This
is the type of tea you would expect to enjoy after skiing in some dista
nt forest where you are also munching on homemade brownies or eggnog co
okies. Apple cider also works very well with the Gypsy Teas, especially
the Lemon Jasmine.<br /><br />The deep purple/red color and hibiscus fr
agrance of Tazo Tea is enough to make anyone passionate about tea. Yet,
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
is this really tea? It looks like tea, but it is really an herbal infus
ion made by steeping hibiscus flowers, orange peel, licorice, cinnamon,
rose hips, lemongrass and red poppy in a swirling of hot water.<br /><b
r />The original flavor is also slightly reminiscent of hot apple cider
or hot cranberry juice which is why it is extra delicious with a cider
mix. Tazo Teas are a mysterious experience into which you can lose your
self with a sense of abandon. So far I've tried more flavors than I can
count and have always been impressed.<br /><br />~The Rebecca Review

Review-3 :
 "There are few hours in life more agreeable than the hour dedicated to
the ceremony known as afternoon tea." -Henry James, Portrait of a Lady<
br /><br />This "English Breakfast" style tea is quite satisfying and h
as a rich flavor that is quite delicious.<br /><br />I'd like to say th
at their teas are best steeped for an extra 5-10 minutes to draw out al
l the delicious flavors. You can use a teapot or a coffee maker to make
tea. Simply put the tea in a coffee filter or the tea bags in the coffe
e pot, then allow the boiling water to drip over the tea leaves.<br /><
br />The sensual blend of exotic spices, teas and herbs used in their t
eas include a mixture of the following: Black tea, green tea, hibiscus
flowers, chamomile flowers, lemongrass, spearmint leaves, blackberry le
aves, cinnamon, Rooibos, Gingerroot, rose hips, orange peel, rose petal
s, licorice root, safflowers, black pepper, lemon verbena leaves, carda
mom, cloves, peppermint leaves, chicory, sarsaparilla root, lemon balm
leaf, star anise and red poppy flowers.<br /><br />This is a blend of b
lack teas, but all their teas are worth trying. Passion and Chai are al
so favorites.<br /><br />~The Rebecca Review
```

In [39]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.1.3] Applying K-Means Clustering on TFIDF, SET 2

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [42]:
```
# Please write all the code with proper documentation
tf_idf_vect = TfidfVectorizer(min_df=1000)
data = tf_idf_vect.fit_transform(x)
print("the type of count vectorizer :",type(data))
print("the shape of out text TFIDF vectorizer : ",data.get_shape())
print("the number of unique words :", data.get_shape()[1])
```

```
the type of count vectorizer : <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text TFIDF vectorizer :  (40000, 248)
the number of unique words : 248
```

In [43]:
```
k_values = [2,3,4,5,6,7,8,9,10]
loss = []
for i in k_values:
    kmeans = KMeans(n_clusters=i, n_jobs=-1).fit(data)
    loss.append(kmeans.inertia_)
```

In [44]:
```
# Draw Loss VS K values plot
plt.plot(k_values, loss)
plt.xlabel('K-values',size=14)
plt.ylabel('Loss',size=14)
plt.title('Loss VS K-values Plot\n',size=18)
plt.grid()
plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- OBSERVATION :- From above we can see that there is inflection at K =3. Befor it loss was decreasing faster as compared to the loss decreasing after it . So , the best value of K is 3

In [45]:
```
optimal_k = 3
# Variable that will be used in the conclusion
tfidf_means_k = optimal_k

# Implementing K-Means++ using optimal value of K
kmeans = KMeans(n_clusters=optimal_k, n_jobs=-1).fit(data)
```

In [46]:
```
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []

for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
        cluster1.append(reviews[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(reviews[i])
    else :
        cluster3.append(reviews[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
```

```
No. of reviews in Cluster-1 :  3613

No. of reviews in Cluster-2 :  33827

No. of reviews in Cluster-3 :  2560
```

[5.1.4] Wordclouds of clusters obtained after applying k-means on TFIDF SET 2

In [47]:

```
# Please write all the code with proper documentation
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 For me, when the days get colder nothing is as rewarding as a simple c
up of hot tea. And for it's claimed immunity benefits, a basic green te
a is a common pick for maintaining a healthy natural balance during the
flu season. From previous experiences in tasting the Tazo brand, both o
f the bottled and boxed products, they have proven to be unsurpassed fo
r quality and flavor. Once I've tried their teas they immediately becam
e my drink of choice. <p>The Zen Green Tea Blend is a wonderful one tha
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

t has only a few ingredients with no artificial anything. And thankfull
y, doesn't boast the addition of fortified vitamins in some senseless a
mount. It truly is an enlightening blend of green tea, spearmint, lemon
grass and lemon verbena. Thus making it versatile refreshment for anyti
me of the day, whether it's right after meals or between meals, or just
before bedtime. Generally light and mild tasting, but that will depend
upon how long you steep it and if you add a sweetener of some form.<p>I
nteresting too, are the amusing comments and remarks that appear on the
packaging. Reading through this as you drink your tea makes it a distin
ctive experience. I never seen tea so clever! I wonder if consuming Taz
o really does improve a person's outlook on life and affect his or her
well being? I think it just could be the great taste and aroma and prob
ably the reassuring thought of doing your health a favor.
Try a co
uple of their teas - they are sure to please!
And Tazo Zen Green T
ea makes a good choice.

Review-2 :
 If you like Earl Grey tea, you'll find no better than Harney & So
n's blends. But once you've tried the Earl Grey Supreme you won't be ab
le to live without it. All others will seem washed out by comparison.

Review-3 :
 Blend a little apple cider mix into your Tazo Passion Tea and see what
happens. This creates a magical apple cider experience with hints of cr
anberry and hibiscus flowers gathered in an ancient land.

Sn
uggle Up Spiced Cider is a good choice because you can then spoon in th
e exact amount you desire, but packets of cider work equally well. This
is the type of tea you would expect to enjoy after skiing in some dista
nt forest where you are also munching on homemade brownies or eggnog co
okies. Apple cider also works very well with the Gypsy Teas, especially
the Lemon Jasmine.

The deep purple/red color and hibiscus fr
agrance of Tazo Tea is enough to make anyone passionate about tea. Yet,
is this really tea? It looks like tea, but it is really an herbal infus
ion made by steeping hibiscus flowers, orange peel, licorice, cinnamon,
rose hips, lemongrass and red poppy in a swirling of hot water.
<b
r />The original flavor is also slightly reminiscent of hot apple cider
or hot cranberry juice which is why it is extra delicious with a cider
mix. Tazo Teas are a mysterious experience into which you can lose your
self with a sense of abandon. So far I've tried more flavors than I can

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
count and have always been impressed.<br /><br />~The Rebecca Review
```

In [48]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.show()
```

In [49]:
```
# Three Reviews of cluster 2
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
count=1
m=[]
for i in range(3):
    print('Review-%d : \n %s\n'%(count,cluster2[i]))
    count +=1
m.append(cluster2[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [50]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [51]:
```
# Three Reviews of cluster 3
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    print('Review-%d : \n %s\n'%(count,cluster3[i]))
    count +=1
m.append(cluster3[0])
```

```
Review-1 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).

Review-2 :
 After years of using Starbuck's decaf and regular single serving espre
sso pods, I finally found a substitute that's more economical and MUCH
better tasting.<p>I've used three different brands of pods in my Starbu
ck's Barista machine, and have found that the quality of the pod and th
e flavor of the espresso inside is, by far, the best in the Espressione
pods. The flavor is more full bodied and stronger than in the Starbuc
k's pods. I actually use one pod to make two shots of espresso, and the
flavor is more intense than when I buy my specialty drinks from Starbuc
ks or other coffee houses.<p>Also, the pod itself is of very high quali
ty. Each pod is individually wrapped in an airtight, hard foil wrapper,
and I've never opened a package yet that contained a torn pod with coff
ee spilling out. Such is not the case with other lesser brands.<p>The t
ab on each pod is also handy, making it easier to switch out a hot spen
t pod with a new one when making a second round of espresso.

Review-3 :
 Few things I'd like to point out:<p>1.  &quot;Espresso&quot; is not a
roast.  It's a drink.  In northern Italy they make espresso from Vienna
roast, in southern Italy they use French.  Starbucks appears to be usin
g French roast, so they should just label it French roast like any norm
al person would.<p>2.  The product description says &quot;Taste Espress
o Roast as both a shot of espresso and brewed in a coffee press. This i
s one versatile coffee!&quot;  This is patently absurd.  It's pre-groun
d coffee, and it's either for use in espresso machines or presses, but
not both.  Here's why:<p>Making a shot of espresso requires a very fine
grind.  Making coffee press coffee requires a very course grind.  If yo
u have ground coffee that is appropriate for espresso machines, it will
pass through the filter in a coffee press and you will have a cup of sl
udge.  If you use ground coffee that does not pass through a press's fi
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
lter in an espresso machine, you do not get espresso.  You get cafe cre
ma, which is a good, but entirely different, beverage.<p>Therefore, a &
quot;Universal Grind&quot; is an impossibility.  They should tell you o
n the label what kind of coffee maker it's ground for, like normal peop
le.<p>3.  It's over-roasted.  Starbucks has an excellent bean-sourcing
operation that buys better coffee than any other company of its size.
Then they roast all their beans Vienna or French.  Most beans taste bes
t at a particular level of roasting that varies with the bean, and for
most beans that level is not Vienna or French.  So why roast all your b
eans too dark?  I guess it's just easier that way.  Also, it lets them
rename Vienna Roast as &quot;Starbucks Roast&quot; because the Austrian
s don't seem to care.<p>That said, this is still better than what you g
et in the supermarket - Starbucks specializes in coffee and sources muc
h better beans than any of the companies that also make cheese, hot dog
s, shampoo or cigarettes.<p>Anyway, if you really want some good coffe
e, do this:<p>Buy a grinder.  Nothing makes coffee lose flavor like kee
ping it around after grinding.  If you have a drip coffee machine, get
a grinder with spinning blades; if you have something else like a press
or vacuum pot, get a grinder with burrs.  If you have an espresso machi
ne, get a very nice grinder with burrs.  Amazon has good deals on all t
hree kinds, just do a search for 'coffee grinder'.  It's not that much
money, and if you're somebody who's willing to pay for Starbucks you sh
ould invest in one.<p>Find a local coffee shop that roasts their own co
ffee and gives you the date it's roasted on.  Freshness matters!  Good
beans don't taste nearly as good after they've been sitting around.<p>A
sk the people at your local shop to tell you about the correct amount o
f beans to use.  Most people don't use the right amount.  If you don't
use enough, your coffee comes out bitter, especially if you have a drip
machine.  Not enough coffee means the beans you do have get too much ex
posure to hot water, which makes bitter coffee.
```

In [52]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.1.5] Applying K-Means Clustering on AVG W2V, SET 3

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [53]:
```
# Please write all the code with proper documentation
# List of sentence in X_train text
sent_x = []
for sent in x :
    sent_x.append(sent.split())


# Train your own Word2Vec model using your own train text corpus
# min_count = 5 considers only words that occured atleast 5 times
w2v_model=Word2Vec(sent_x,min_count=5,size=50, workers=4)

w2v_words = list(w2v_model.wv.vocab)
print("number of words that occured minimum 5 times ",len(w2v_words))
```

```
number of words that occured minimum 5 times  8418
```

In [54]:
```
# compute average word2vec for each review for sent_x .
train_vectors = [];
for sent in sent_x:
    sent_vec = np.zeros(50)
    cnt_words =0;
    for word in sent: #
        if word in w2v_words:
            vec = w2v_model.wv[word]
            sent_vec += vec
            cnt_words += 1
    if cnt_words != 0:
        sent_vec /= cnt_words
    train_vectors.append(sent_vec)

data = train_vectors
```

In [55]:
```
k_values = [2,3,4,5,6,7,8,9,10]
loss = []
for i in k_values:
    kmeans = KMeans(n_clusters=i, n_jobs=-1).fit(data)
    loss.append(kmeans.inertia_)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [56]:
```
# Draw Loss VS K values plot
plt.plot(k_values, loss)
plt.xlabel('K-values',size=14)
plt.ylabel('Loss',size=14)
plt.title('Loss VS K-values Plot\n',size=18)
plt.grid()
plt.show()
```

In [57]:
```
optimal_k = 3
# Variable that will be used in the conclusion
avg_w2v_means_k = optimal_k

# Implementing K-Means++ using optimal value of K
kmeans = KMeans(n_clusters=optimal_k, n_jobs=-1).fit(data)

# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
        cluster1.append(reviews[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(reviews[i])
    else :
        cluster3.append(reviews[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
```

```
No. of reviews in Cluster-1 :  6593

No. of reviews in Cluster-2 :  14925

No. of reviews in Cluster-3 :  18482
```

[5.1.6] Wordclouds of clusters obtained after applying k-means on AVG W2V SET 3

In [58]:

```
# Please write all the code with proper documentation
# Please write all the code with proper documentation
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).
```

In [59]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [60]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).
```

In [61]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [62]:
```
# Three Reviews of cluster 3
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster3):
        print('Review-%d : \n %s\n'%(count,cluster3[i]))
        count +=1
m.append(cluster3[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).
```

In [63]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.1.7] Applying K-Means Clustering on TFIDF W2V, SET 4

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [64]:
```
# Please write all the code with proper documentation
# TF-IDF weighted Word2Vec
tf_idf_vect = TfidfVectorizer()

# final_tf_idf1 is the sparse matrix with row= sentence, col=word and c
ell_val = tfidf
final_tf_idf1 = tf_idf_vect.fit_transform(x)

# tfidf words/col-names
tfidf_feat = tf_idf_vect.get_feature_names()

# compute TFIDF Weighted Word2Vec for each review for sent_x .
tfidf_vectors = [];
row=0;
for sent in sent_x:
    sent_vec = np.zeros(50)
    weight_sum =0;
    for word in sent:
        if word in w2v_words:
            vec = w2v_model.wv[word]
            # obtain the tf_idfidf of a word in a sentence/review
            tf_idf = final_tf_idf1[row, tfidf_feat.index(word)]
            sent_vec += (vec * tf_idf)
            weight_sum += tf_idf
    if weight_sum != 0:
        sent_vec /= weight_sum
    tfidf_vectors.append(sent_vec)
    row += 1

data = tfidf_vectors
```

In [65]:
```
k_values = [2,3,4,5,6,7,8,9,10]
loss = []
for i in k_values:
    kmeans = KMeans(n_clusters=i, n_jobs=-1).fit(data)
    loss.append(kmeans.inertia_)
```

In [66]:
```
# Draw Loss VS K values plot
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.plot(k_values, loss)
plt.xlabel('K-values',size=14)
plt.ylabel('Loss',size=14)
plt.title('Loss VS K-values Plot\n',size=18)
plt.grid()
plt.show()
```

- OBSERVATION :- From above we can see that there is inflection at K = 3 . Befor it loss was decreasing faster as compared to the loss decreasing after it . So , the best value of K is 3.

In [67]:
```
optimal_k = 3
# Variable that will be used in the conclusion
tfidf_w2v_means_k = optimal_k

# Implementing K-Means++ using optimal value of K
kmeans = KMeans(n_clusters=optimal_k, n_jobs=-1).fit(data)

# Getting all the reviews in different clusters
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
cluster1 = []
cluster2 = []
cluster3 = []

for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
        cluster1.append(reviews[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(reviews[i])
    else :
        cluster3.append(reviews[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
```

```
No. of reviews in Cluster-1 :  5682

No. of reviews in Cluster-2 :  18145

No. of reviews in Cluster-3 :  16173
```

[5.1.8] Wordclouds of clusters obtained after applying k-means on TFIDF W2V SET 4

In [68]:

```
# Please write all the code with proper documentation
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).
```

In [69]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [70]:
```
# Please write all the code with proper documentation
# Three Reviews of cluster 2
count=1
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
m=[]
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).
```

In [71]:
```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [72]:
```
# Please write all the code with proper documentation
# Three Reviews of cluster 3
count=1
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
m=[]
for i in range(3):
    if i < len(cluster3):
        print('Review-%d : \n %s\n'%(count,cluster3[i]))
        count +=1
m.append(cluster3[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).
```

In [73]:
```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.2] Agglomerative Clustering

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [77]:
```
##Sorting data according to Time in ascending order for Time Based Spli
tting
time_sorted_data = filtered_data.sort_values('Time', axis=0, ascending=
True, inplace=False, kind='quicksort', na_position='last')

# We will collect different 4K rows without repetition from time_sorted
_data dataframe
my_final = time_sorted_data[:4000]

x = my_final['CleanedText'].values
```

In [78]:
```
x.shape
```

Out[78]:
```
(4000,)
```

[5.2.1] Applying Agglomerative Clustering on AVG W2V, SET 3

In [122]:
```
# Please write all the code with proper documentation
# List of sentence in X_train text
sent_x = []
for sent in x :
    sent_x.append(sent.split())


# Train your own Word2Vec model using your own train text corpus
# min_count = 5 considers only words that occured atleast 5 times
w2v_model=Word2Vec(sent_x,min_count=5,size=50, workers=4)

w2v_words = list(w2v_model.wv.vocab)
print("number of words that occured minimum 5 times ",len(w2v_words))
```

```
number of words that occured minimum 5 times  3498
```

In [123]:
```
# compute average word2vec for each review for sent_x .
train_vectors = [];
for sent in sent_x:
    sent_vec = np.zeros(50)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    cnt_words =0;
    for word in sent: #
        if word in w2v_words:
            vec = w2v_model.wv[word]
            sent_vec += vec
            cnt_words += 1
    if cnt_words != 0:
        sent_vec /= cnt_words
    train_vectors.append(sent_vec)

data = train_vectors
```

In [124]:

```
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=2).fit(data)

reviews = my_final['Text'].values
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    else :
        cluster2.append(reviews[i])


# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
```

```
No. of reviews in Cluster-1 :  2406

No. of reviews in Cluster-2 :  1594
```

In [125]:

```
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    print('Review-%d : \n %s\n'%(count,cluster1[i]))
    count +=1
m.append(cluster1[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [126]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [127]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 These are easy to use, they do not make a mess, and offer vibrant colo
rs.  They do not taint what you are decorting as some colors can.  I wo
uld  highly recommend these to anyone to likes to decorate.

Review-2 :
 I'm sick of scads of nasty toothpicks all over my counter when I'm tin
ting icings;  the squeeze-bottle concept is one that is long overdue an
d exceptionally welcome.  The colors are vibrant, do not have an offens
ive taste, and blend, in my opinion, more smoothly than the Wilton past
es.  You'll need to experiment a bit to get hues right at first, but i
t's well worth it.

Review-3 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.
```

In [128]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Hierarchical Clustering with 5 clusters

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [129]:
```
model = AgglomerativeClustering(n_clusters=5).fit(data)

# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []
cluster4 = []
cluster5 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    elif model.labels_[i] == 1:
        cluster2.append(reviews[i])
    elif model.labels_[i] == 2:
        cluster3.append(reviews[i])
    elif model.labels_[i] == 3:
        cluster4.append(reviews[i])
    else :
        cluster5.append(reviews[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
print("\nNo. of reviews in Cluster-4 : ",len(cluster4))
print("\nNo. of reviews in Cluster-5 : ",len(cluster5))
```

```
No. of reviews in Cluster-1 :  1342

No. of reviews in Cluster-2 :  930

No. of reviews in Cluster-3 :  134

No. of reviews in Cluster-4 :  521

No. of reviews in Cluster-5 :  1073
```

In [130]:
```
# Three Reviews of cluster 1
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.

Review-3 :
 A twist of rumplestiskin captured on film, starring michael keaton and
geena davis in their prime.  Tim Burton's masterpiece, rumbles with abs
urdity, and is wonderfully paced to the point where there is not a dull
  moment.
```

In [131]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [132]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-2 :
 I bought a few of these after my apartment was infested with fruit fli
es. After only a few hours, the trap had &quot;attracted&quot; many fli
es and within a few days they were practically gone. This may not be a
long term  solution, but if flies are driving you crazy, consider buyin
g this. One  caution- the surface is very sticky, so try to avoid touch
ing it.

Review-3 :
 What happens when you say his name three times? Michael Keaten stars i
n this comedy about two couples that live in an old two story house.  W
hile coming back from a supply store, the couple suddenly get caught in
side of a  &quot;broken-up&quot; bridge and then just before they start
to tumble down  into the lake, a board catches them.  But just when the
y've got their hopes  up, and small dog steps on the board and the car
starts to slide off the  bridge and into the lake waters.  A few minute
s later...<p>They find  themselves back into their home, they find that
somehow somehad light the  fireplace, as if done by magic.  From then o
n, they find a weird-looking  dead guy known as Bettlejuice.  The only
way they can get him for help is  to call him by his name three times a
nd he will appear at their service.  But they soon wish that they have
never called his name, because  Bettlejuice was once a troublemaker but
he is the only one who can save  them, on the account that they said hi
s name three times.  They can't leave  their houses or else they will f
ind theirselves in another world with giant  sandworms.  This is a stel
lar comedy that you should see! Michael Keaton is  awesome as he plays
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
the leading role of Bettlejuice.
```

In [133]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.show()
```

In [134]:
```
# Three Reviews of cluster 3
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
count=1
m=[]
for i in range(3):
    if i < len(cluster3):
        print('Review-%d : \n %s\n'%(count,cluster3[i]))
        count +=1
m.append(cluster3[0])
```

```
Review-1 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-2 :
 After years of using Starbuck's decaf and regular single serving espre
sso pods, I finally found a substitute that's more economical and MUCH
better tasting.<p>I've used three different brands of pods in my Starbu
ck's Barista machine, and have found that the quality of the pod and th
e flavor of the espresso inside is, by far, the best in the Espressione
pods. The flavor is more full bodied and stronger than in the Starbuc
k's pods. I actually use one pod to make two shots of espresso, and the
flavor is more intense than when I buy my specialty drinks from Starbuc
ks or other coffee houses.<p>Also, the pod itself is of very high quali
ty. Each pod is individually wrapped in an airtight, hard foil wrapper,
and I've never opened a package yet that contained a torn pod with coff
ee spilling out. Such is not the case with other lesser brands.<p>The t
ab on each pod is also handy, making it easier to switch out a hot spen
t pod with a new one when making a second round of espresso.

Review-3 :
 If you've never tried Kona coffee, Aloha Island is definitely the one
to start with.  <p>Kona coffee is grown in the Kona district of the Big
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
Island and regarded by some to be the best coffee in the world.  I'm no
t a coffee aficionado by any means, but you can taste the difference wi
th Kona coffee - it's full and rich and almost mocha-ish.  I always res
erve Kona for lazy weekend brunches or to serve with dessert after a lo
vely dinner.  <p>Aloha Island operates the only quality management prog
ram in Kona, so their coffees are certified as 100% Kona.  Which just m
eans that you're guaranteed to get quality Kona coffee if you order fro
m them. <p>Keep in mind that Aloha Island has three grades of Kona coff
ee that they sell - gold, platinum, and diamond (which is supposed to b
e their private reserve).  Obviously, it gets more expensive as you go
up the ladder.  The gold is very good and a great place to start sampli
ng their fare.
```

In [135]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [136]:
```
# Three Reviews of cluster 4
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster4):
        print('Review-%d : \n %s\n'%(count,cluster4[i]))
        count +=1
m.append(cluster4[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).

Review-3 :
 For me, when the days get colder nothing is as rewarding as a simple c
up of hot tea. And for it's claimed immunity benefits, a basic green te
a is a common pick for maintaining a healthy natural balance during the
flu season. From previous experiences in tasting the Tazo brand, both o
f the bottled and boxed products, they have proven to be unsurpassed fo
r quality and flavor. Once I've tried their teas they immediately becam
e my drink of choice. <p>The Zen Green Tea Blend is a wonderful one tha
t has only a few ingredients with no artificial anything. And thankfull
y, doesn't boast the addition of fortified vitamins in some senseless a
mount. It truly is an enlightening blend of green tea, spearmint, lemon
grass and lemon verbena. Thus making it versatile refreshment for anyti
me of the day, whether it's right after meals or between meals, or just
before bedtime. Generally light and mild tasting, but that will depend
upon how long you steep it and if you add a sweetener of some form.<p>I
nteresting too, are the amusing comments and remarks that appear on the
packaging. Reading through this as you drink your tea makes it a distin
ctive experience. I never seen tea so clever! I wonder if consuming Taz
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
o really does improve a person's outlook on life and affect his or her
well being? I think it just could be the great taste and aroma and prob
ably the reassuring thought of doing your health a favor.<br />Try a co
uple of their teas - they are sure to please!<br />And Tazo Zen Green T
ea makes a good choice.
```

In [137]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [138]:
```
# Three Reviews of cluster 5
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if i < len(cluster5):
        print('Review-%d : \n %s\n'%(count,cluster5[i]))
        count +=1
m.append(cluster5[0])
```

```
Review-1 :
 These are easy to use, they do not make a mess, and offer vibrant colo
rs.  They do not taint what you are decorting as some colors can.  I wo
uld  highly recommend these to anyone to likes to decorate.

Review-2 :
 I'm sick of scads of nasty toothpicks all over my counter when I'm tin
ting icings;  the squeeze-bottle concept is one that is long overdue an
d exceptionally welcome.  The colors are vibrant, do not have an offens
ive taste, and blend, in my opinion, more smoothly than the Wilton past
es.  You'll need to experiment a bit to get hues right at first, but i
t's well worth it.

Review-3 :
 Fresh limes are underappreciated, but a joy to have in the kitchen.  A
few squirts of lime juice into water makes a wonderful and healthy alte
rnative to carbonated beverages (and without sugar!). <p>Additionally,
they're essential for doing salsa right.  <p>1/4 C Lime juice (from 4 l
imes)<br />10 small tomatillos, husks removed and washed (ASIN B0000TL6
DG)<br />2-3 Chipotles, seeds removed (ASIN B0000EIDSR)<br />1 C cilant
ro, roughly chopped<br />8 small Tomatoes<br />1 small onion (optional)
<br />Salt to taste<p>Cook tomatillos, tomatoes and onion: place on coo
kie sheet and broil at 500 degrees. The tomatoes will come out in about
10 minutes, the tomatillos and onion 15-20 minutes. Let cool, then pure
e the tomatillos in a blender with the lime juice, boullion garlic and
the chipotles. Add half the cilantro, and whirl just a little bit. Add
remaining ingredients. Let stand for about 30 minutes then serve at roo
m temperature.
```

In [139]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Hierarchical Clustering with 10 clusters

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [99]:
```
model = AgglomerativeClustering(n_clusters=10).fit(data)
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []
cluster4 = []
cluster5 = []
cluster6 = []
cluster7 = []
cluster8 = []
cluster9 = []
cluster10 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    elif model.labels_[i] == 1:
        cluster2.append(reviews[i])
    elif model.labels_[i] == 2:
        cluster3.append(reviews[i])
    elif model.labels_[i] == 3:
        cluster4.append(reviews[i])
    elif model.labels_[i] == 4:
        cluster5.append(reviews[i])
    elif model.labels_[i] == 5:
        cluster6.append(reviews[i])
    elif model.labels_[i] == 6:
        cluster7.append(reviews[i])
    elif model.labels_[i] == 7:
        cluster8.append(reviews[i])
    elif model.labels_[i] == 8:
        cluster9.append(reviews[i])
    else :
        cluster10.append(reviews[i])
```

In [100]:
```
# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
print("\nNo. of reviews in Cluster-4 : ",len(cluster4))
print("\nNo. of reviews in Cluster-5 : ",len(cluster5))
print("\nNo. of reviews in Cluster-6 : ",len(cluster6))
print("\nNo. of reviews in Cluster-7 : ",len(cluster7))
print("\nNo. of reviews in Cluster-8 : ",len(cluster8))
print("\nNo. of reviews in Cluster-9 : ",len(cluster9))
print("\nNo. of reviews in Cluster-10 : ",len(cluster10))
```

```
No. of reviews in Cluster-1 :  540

No. of reviews in Cluster-2 :  729

No. of reviews in Cluster-3 :  182

No. of reviews in Cluster-4 :  701

No. of reviews in Cluster-5 :  647

No. of reviews in Cluster-6 :  217

No. of reviews in Cluster-7 :  230

No. of reviews in Cluster-8 :  64

No. of reviews in Cluster-9 :  559

No. of reviews in Cluster-10 :  131
```

In [102]:

```
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 As a kid this ice cream was a favorite treat.  I've been told it has t
he highest milk fat content of any ice cream around and man can you tas
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
te it. This stuff is amazing. If you are a fan of premium ice cream thi
s is the best stuff.

Review-2 :
 I just love this sauce. I put in on everything.

Review-3 :
 A piquant, sweet, colorful treat.  Don't be put off by the Hot designa
tion - no hotter than a pepperoncini.  Really great for adding a little
color and sweetness to lots of dishes
```

In [103]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [104]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [105]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [106]:
```
# Three Reviews of cluster 3
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster3):
        print('Review-%d : \n %s\n'%(count,cluster3[i]))
        count +=1
m.append(cluster3[0])
```

```
Review-1 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-2 :
 After years of using Starbuck's decaf and regular single serving espre
sso pods, I finally found a substitute that's more economical and MUCH
better tasting.<p>I've used three different brands of pods in my Starbu
ck's Barista machine, and have found that the quality of the pod and th
e flavor of the espresso inside is, by far, the best in the Espressione
pods. The flavor is more full bodied and stronger than in the Starbuc
k's pods. I actually use one pod to make two shots of espresso, and the
flavor is more intense than when I buy my specialty drinks from Starbuc
ks or other coffee houses.<p>Also, the pod itself is of very high quali
ty. Each pod is individually wrapped in an airtight, hard foil wrapper,
and I've never opened a package yet that contained a torn pod with coff
ee spilling out. Such is not the case with other lesser brands.<p>The t
ab on each pod is also handy, making it easier to switch out a hot spen
t pod with a new one when making a second round of espresso.

Review-3 :
 If you've never tried Kona coffee, Aloha Island is definitely the one
to start with.  <p>Kona coffee is grown in the Kona district of the Big
Island and regarded by some to be the best coffee in the world.  I'm no
t a coffee aficionado by any means, but you can taste the difference wi
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
th Kona coffee - it's full and rich and almost mocha-ish.  I always res
erve Kona for lazy weekend brunches or to serve with dessert after a lo
vely dinner.  <p>Aloha Island operates the only quality management prog
ram in Kona, so their coffees are certified as 100% Kona.  Which just m
eans that you're guaranteed to get quality Kona coffee if you order fro
m them. <p>Keep in mind that Aloha Island has three grades of Kona coff
ee that they sell - gold, platinum, and diamond (which is supposed to b
e their private reserve).  Obviously, it gets more expensive as you go
up the ladder.  The gold is very good and a great place to start sampli
ng their fare.
```

In [107]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [108]:
```
# Three Reviews of cluster 4
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if i < len(cluster4):
        print('Review-%d : \n %s\n'%(count,cluster4[i]))
        count +=1
m.append(cluster4[0])
```

```
Review-1 :
 This are so much easier to use than the Wilson paste colors.  Colors a
re  vibrant, and do not taint the frosting like some colors can.  These
are  simple to use, and do not make a mess.  My only complaint is that
I did not  find these years ago.  This is a must have if you decorate o
ften!

Review-2 :
 I don't know why anyone would ever use those little liquid food colors
again after trying these pastes. The colors are more intense, and you c
an easily blend lovely shades for your icings. <p>Martha Stewart shows
wonderful seasonal cookies from time to time on her TV show. A simple e
ggwhite-sugar icing on cookies in exciting colors makes such a wonderfu
l table decoration as well as great dessert for parties, teas and other
events. These paste colors are a must if you like to make frosted cooki
es or party cakes. <p>And these are especially fun if you have kids. A
cookie-icing contest is a fun event for a birthday party, Christmas par
ty or just a rainy afternoon.

Review-3 :
 I bought these to decorate some dia de los muertos skulls with icing.
The Ateco food coloring gave nice bright, vibrant colors and mixed with
the icing very well. I was totally satisfied with the purchase.
```

In [109]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [110]:
```
# Three Reviews of cluster 5
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster5):
        print('Review-%d : \n %s\n'%(count,cluster5[i]))
        count +=1
m.append(cluster6[0])
```

```
Review-1 :
 I'm getting crazy.<p>Is it really impossible today not to find the Fre
nch VHS version of this film ?<p>Could U please tell me something about
it ? Tks

Review-2 :
 This was a favorite book of mine when I was a little girl and I would
read it all the time and my Mom would read it to me as a bedtime book,
I have had two copies of this book the first one had been either lost o
r given away but I found another copy at a garage sale and it was in ve
ry good condition and I just had to buy it, I don't know where it is bu
t it's around somewhere probably up in the attic and I will have to loo
k for it so I can read it to my niece. I recommend this book!

Review-3 :
 Although this was a gift to use for keeping wine, I have also found it
invaluable for keeping olive oils and other items fresh for longer peri
ods.  It is easy to use and easy to store.
```

In [111]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [112]:
```
# Three Reviews of cluster 6
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster6):
        print('Review-%d : \n %s\n'%(count,cluster6[i]))
        count +=1
m.append(cluster6[0])
```

```
Review-1 :
 For me, when the days get colder nothing is as rewarding as a simple c
up of hot tea. And for it's claimed immunity benefits, a basic green te
a is a common pick for maintaining a healthy natural balance during the
flu season. From previous experiences in tasting the Tazo brand, both o
f the bottled and boxed products, they have proven to be unsurpassed fo
r quality and flavor. Once I've tried their teas they immediately becam
e my drink of choice. <p>The Zen Green Tea Blend is a wonderful one tha
t has only a few ingredients with no artificial anything. And thankfull
y, doesn't boast the addition of fortified vitamins in some senseless a
mount. It truly is an enlightening blend of green tea, spearmint, lemon
grass and lemon verbena. Thus making it versatile refreshment for anyti
me of the day, whether it's right after meals or between meals, or just
before bedtime. Generally light and mild tasting, but that will depend
upon how long you steep it and if you add a sweetener of some form.<p>I
nteresting too, are the amusing comments and remarks that appear on the
packaging. Reading through this as you drink your tea makes it a distin
ctive experience. I never seen tea so clever! I wonder if consuming Taz
o really does improve a person's outlook on life and affect his or her
well being? I think it just could be the great taste and aroma and prob
ably the reassuring thought of doing your health a favor.<br />Try a co
uple of their teas - they are sure to please!<br />And Tazo Zen Green T
ea makes a good choice.

Review-2 :
 I think that this is a great Chai! Far from sweet.. I found it very un
sweet (and that is exactly why I like it). The Tiger Spice is a great b
lend of Black Tea, Cinnamon, Ginger, herbs, and more. I was not a big t
ea drinker (until now) but this is a very smooth and light Chai. All yo
u do is add water and it comes out creamy and perfect (some people add
milk but this Chai mix already has a creamer in it). This is my new hea
lthy breakfast wake me up. It is great. I have many Indian friends with
whom I drink freshly made Chai with and this is as close as you can com
e with an instant blend. I highly recommend this.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
Review-3 :
 For a bag tea--which generally are not as good quality, this brews up
like a top quality loose tea. It's a delicious snappy black blend, with
a lovely scent and taste. One of my favorites.
```

In [113]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [114]:
```
# Three Reviews of cluster 7
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if i < len(cluster7):
        print('Review-%d : \n %s\n'%(count,cluster7[i]))
        count +=1
m.append(cluster7[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 I've purchased both the Espressione Espresso (classic) and the 100% Ar
abica.  My vote is definitely with the 100% Arabica.  The flavor has mo
re bite and flavor (much more like European coffee than American).

Review-3 :
 Few things I'd like to point out:<p>1.  &quot;Espresso&quot; is not a
roast.  It's a drink.  In northern Italy they make espresso from Vienna
roast, in southern Italy they use French.  Starbucks appears to be usin
g French roast, so they should just label it French roast like any norm
al person would.<p>2.  The product description says &quot;Taste Espress
o Roast as both a shot of espresso and brewed in a coffee press. This i
s one versatile coffee!&quot;  This is patently absurd.  It's pre-groun
d coffee, and it's either for use in espresso machines or presses, but
not both.  Here's why:<p>Making a shot of espresso requires a very fine
grind.  Making coffee press coffee requires a very course grind.  If yo
u have ground coffee that is appropriate for espresso machines, it will
pass through the filter in a coffee press and you will have a cup of sl
udge.  If you use ground coffee that does not pass through a press's fi
lter in an espresso machine, you do not get espresso.  You get cafe cre
ma, which is a good, but entirely different, beverage.<p>Therefore, a &
quot;Universal Grind&quot; is an impossibility.  They should tell you o
n the label what kind of coffee maker it's ground for, like normal peop
le.<p>3.  It's over-roasted.  Starbucks has an excellent bean-sourcing
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
operation that buys better coffee than any other company of its size.
Then they roast all their beans Vienna or French.  Most beans taste bes
t at a particular level of roasting that varies with the bean, and for
most beans that level is not Vienna or French.  So why roast all your b
eans too dark?  I guess it's just easier that way.  Also, it lets them
rename Vienna Roast as &quot;Starbucks Roast&quot; because the Austrian
s don't seem to care.<p>That said, this is still better than what you g
et in the supermarket - Starbucks specializes in coffee and sources muc
h better beans than any of the companies that also make cheese, hot dog
s, shampoo or cigarettes.<p>Anyway, if you really want some good coffe
e, do this:<p>Buy a grinder.  Nothing makes coffee lose flavor like kee
ping it around after grinding.  If you have a drip coffee machine, get
a grinder with spinning blades; if you have something else like a press
or vacuum pot, get a grinder with burrs.  If you have an espresso machi
ne, get a very nice grinder with burrs.  Amazon has good deals on all t
hree kinds, just do a search for 'coffee grinder'.  It's not that much
money, and if you're somebody who's willing to pay for Starbucks you sh
ould invest in one.<p>Find a local coffee shop that roasts their own co
ffee and gives you the date it's roasted on.  Freshness matters!  Good
beans don't taste nearly as good after they've been sitting around.<p>A
sk the people at your local shop to tell you about the correct amount o
f beans to use.  Most people don't use the right amount.  If you don't
use enough, your coffee comes out bitter, especially if you have a drip
machine.  Not enough coffee means the beans you do have get too much ex
posure to hot water, which makes bitter coffee.
```

In [115]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [116]:
```
# Three Reviews of cluster 8
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if i < len(cluster8):
        print('Review-%d : \n %s\n'%(count,cluster8[i]))
        count +=1
m.append(cluster8[0])
```

```
Review-1 :
 love the snack.  wanted to buy a bunch.<p>shipping at amazon is more t
han the price of the snack.  try ordering a dozen and then look at the
shipping price.<p>every link i've tried to this snack takes you back to
amazon's exhorbitant shipping rates.<p>another reason to buy your books
at overstock.com.  too back they don't carry food as well.

Review-2 :
 I think it is Very Expensive  !!<br />I can get it at local Indian or
Chinese store for $20 in California.

Review-3 :
 Just got my parm and WOW! SO much better than what you can get in the
store here. I can see how this will go fast. I will buy from these guys
again.
```

In [117]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [118]:
```
# Three Reviews of cluster 9
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster9[i]))
        count +=1
m.append(cluster9[0])
```

```
Review-1 :
 Beetlejuice is an excellent and funny movie. Keaton is hilarious as th
e wacky beetlejuice. The great special effects help the film. I think t
his is one of the best movies ever made and I'm sure you'll agree. For
a good  time, watch beetlejuice!

Review-2 :
 These are easy to use, they do not make a mess, and offer vibrant colo
rs.  They do not taint what you are decorting as some colors can.  I wo
uld  highly recommend these to anyone to likes to decorate.

Review-3 :
 This is such a great film, I don't even know how to sum it up. First o
f  all, it is completely original and it is unlike any film I have ever
seen  before. Second of all, it's a great comedy with kind of a spooky,
weird  feel to it, which is something all of Tim Burton's films have. T
he look of  the film is probably what I like the best. Art Director Bo
Welch and Tim  Burton show us a world unlike anything seen in a movie.
This is a great  film, and I would recommend it to anyone looking for a
n enjoyable,  entertaining film that is original and inventive.
```

In [119]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [120]:
```
# Three Reviews of cluster 10
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster10[i]))
        count +=1
m.append(cluster10[0])
```

```
Review-1 :
 Winona Ryder is HOT! HOT! HOT! as the gothic death-rock princess of Do
om (see &quot;Lucas&quot; for more W.R. action) The best performance fr
om that eldest Baldwin brother that I've ever seen from any of them.His
zenith I'm sure. BEST PERFORMANCE - goes to - Miss Ryder!,of course,
I'm a BIG fan, you see? FAV PART - when Beetlejuice was chumming up to
Baldwin's character while wearing the same type clothes. No big deal, j
ust witty and clever charm with spit. and when Beetlejuice hits up on G
ina Davis and/or Winona Ryder

Review-2 :
 Embarrassing comedy that comes over like a child with 'attention defic
it disorder' begging for your approval. Even the film's own internal lo
gic is thrown away towards the end to no extra purpose. The girl floati
ng in the air at the end is rather sweet though.

Review-3 :
 Fresh limes are underappreciated, but a joy to have in the kitchen.  A
few squirts of lime juice into water makes a wonderful and healthy alte
rnative to carbonated beverages (and without sugar!). <p>Additionally,
they're essential for doing salsa right.  <p>1/4 C Lime juice (from 4 l
imes)<br />10 small tomatillos, husks removed and washed (ASIN B0000TL6
DG)<br />2-3 Chipotles, seeds removed (ASIN B0000EIDSR)<br />1 C cilant
ro, roughly chopped<br />8 small Tomatoes<br />1 small onion (optional)
<br />Salt to taste<p>Cook tomatillos, tomatoes and onion: place on coo
kie sheet and broil at 500 degrees. The tomatoes will come out in about
10 minutes, the tomatillos and onion 15-20 minutes. Let cool, then pure
e the tomatillos in a blender with the lime juice, boullion garlic and
the chipotles. Add half the cilantro, and whirl just a little bit. Add
remaining ingredients. Let stand for about 30 minutes then serve at roo
m temperature.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [121]:
```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.2.3] Applying Agglomerative Clustering on TFIDF W2V, SET 4

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [140]:
```
# Please write all the code with proper documentation
# TF-IDF weighted Word2Vec
tf_idf_vect = TfidfVectorizer()

# final_tf_idf1 is the sparse matrix with row= sentence, col=word and c
ell_val = tfidf
final_tf_idf1 = tf_idf_vect.fit_transform(x)

# tfidf words/col-names
tfidf_feat = tf_idf_vect.get_feature_names()

# compute TFIDF Weighted Word2Vec for each review for sent_x .
tfidf_vectors = [];
row=0;
for sent in sent_x:
    sent_vec = np.zeros(50)
    weight_sum =0;
    for word in sent:
        if word in w2v_words:
            vec = w2v_model.wv[word]
            # obtain the tf_idfidf of a word in a sentence/review
            tf_idf = final_tf_idf1[row, tfidf_feat.index(word)]
            sent_vec += (vec * tf_idf)
            weight_sum += tf_idf
    if weight_sum != 0:
        sent_vec /= weight_sum
    tfidf_vectors.append(sent_vec)
    row += 1

data = tfidf_vectors
```

In [141]:
```
model = AgglomerativeClustering(n_clusters=2).fit(data)

reviews = my_final['Text'].values
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []

for i in range(model.labels_.shape[0]):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    else :
        cluster2.append(reviews[i])


# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
```

```
No. of reviews in Cluster-1 :  2156

No. of reviews in Cluster-2 :  1844
```

In [142]:

```
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 This are so much easier to use than the Wilson paste colors.  Colors a
re  vibrant, and do not taint the frosting like some colors can.  These
are  simple to use, and do not make a mess.  My only complaint is that
I did not  find these years ago.  This is a must have if you decorate o
ften!

Review-2 :
 These are easy to use, they do not make a mess, and offer vibrant colo
rs.  They do not taint what you are decorting as some colors can.  I wo
uld  highly recommend these to anyone to likes to decorate.

Review-3 :
 This is such a great film, I don't even know how to sum it up. First o
f  all, it is completely original and it is unlike any film I have ever
seen  before. Second of all, it's a great comedy with kind of a spooky,
weird  feel to it, which is something all of Tim Burton's films have. T
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
he look of  the film is probably what I like the best. Art Director Bo
Welch and Tim  Burton show us a world unlike anything seen in a movie.
This is a great  film, and I would recommend it to anyone looking for a
n enjoyable,  entertaining film that is original and inventive.
```

In [143]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [145]:
```
# Three Reviews of cluster 2
count=1
m=[]
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [146]:
```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Hierarchical Clustering with 5 clusters

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [147]:
```
model = AgglomerativeClustering(n_clusters=5).fit(data)

# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []
cluster4 = []
cluster5 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    elif model.labels_[i] == 1:
        cluster2.append(reviews[i])
    elif model.labels_[i] == 2:
        cluster3.append(reviews[i])
    elif model.labels_[i] == 3:
        cluster4.append(reviews[i])
    else :
        cluster5.append(reviews[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
print("\nNo. of reviews in Cluster-4 : ",len(cluster4))
print("\nNo. of reviews in Cluster-5 : ",len(cluster5))
```

```
No. of reviews in Cluster-1 :  1608

No. of reviews in Cluster-2 :  1100

No. of reviews in Cluster-3 :  276

No. of reviews in Cluster-4 :  272

No. of reviews in Cluster-5 :  744
```

In [148]:
```
# Three Reviews of cluster 1
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 This are so much easier to use than the Wilson paste colors.  Colors a
re  vibrant, and do not taint the frosting like some colors can.  These
are  simple to use, and do not make a mess.  My only complaint is that
I did not  find these years ago.  This is a must have if you decorate o
ften!

Review-2 :
 These are easy to use, they do not make a mess, and offer vibrant colo
rs.  They do not taint what you are decorting as some colors can.  I wo
uld  highly recommend these to anyone to likes to decorate.

Review-3 :
 This is such a great film, I don't even know how to sum it up. First o
f  all, it is completely original and it is unlike any film I have ever
seen  before. Second of all, it's a great comedy with kind of a spooky,
weird  feel to it, which is something all of Tim Burton's films have. T
he look of  the film is probably what I like the best. Art Director Bo
Welch and Tim  Burton show us a world unlike anything seen in a movie.
This is a great  film, and I would recommend it to anyone looking for a
n enjoyable,  entertaining film that is original and inventive.
```

In [149]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [151]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [152]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [153]:
```
# Three Reviews of cluster 3
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster3):
        print('Review-%d : \n %s\n'%(count,cluster3[i]))
        count +=1
m.append(cluster3[0])
```

```
Review-1 :
 For years I have been trying to simulate a truly Italian espresso or c
appuccino without success.  Several packages arrived [...] just before
Christmas.  My son had sent a Briel Cadiz Espresso machine, two lovely
Russian china cups and a case of Espressione Classic Espresso Pods.  It
was the kindest, most wonderful Christmas present I can remember.  The
espresso machine excels in quality and ease of use.  The espresso is ex
quisite.  Each time I prepare an espresso or cappuccino I experience my
son's loving kindness and a small part of the many gifts of Italy.  Hea
ven on Earth.

Review-2 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-3 :
 After years of using Starbuck's decaf and regular single serving espre
sso pods, I finally found a substitute that's more economical and MUCH
better tasting.<p>I've used three different brands of pods in my Starbu
ck's Barista machine, and have found that the quality of the pod and th
e flavor of the espresso inside is, by far, the best in the Espressione
pods. The flavor is more full bodied and stronger than in the Starbuc
k's pods. I actually use one pod to make two shots of espresso, and the
flavor is more intense than when I buy my specialty drinks from Starbuc
ks or other coffee houses.<p>Also, the pod itself is of very high quali
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ty. Each pod is individually wrapped in an airtight, hard foil wrapper,
and I've never opened a package yet that contained a torn pod with coff
ee spilling out. Such is not the case with other lesser brands.<p>The t
ab on each pod is also handy, making it easier to switch out a hot spen
t pod with a new one when making a second round of espresso.
```

In [154]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [155]:
```
# Three Reviews of cluster 4
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if i < len(cluster4):
        print('Review-%d : \n %s\n'%(count,cluster4[i]))
        count +=1
m.append(cluster4[0])
```

```
Review-1 :
 For me, when the days get colder nothing is as rewarding as a simple c
up of hot tea. And for it's claimed immunity benefits, a basic green te
a is a common pick for maintaining a healthy natural balance during the
flu season. From previous experiences in tasting the Tazo brand, both o
f the bottled and boxed products, they have proven to be unsurpassed fo
r quality and flavor. Once I've tried their teas they immediately becam
e my drink of choice. <p>The Zen Green Tea Blend is a wonderful one tha
t has only a few ingredients with no artificial anything. And thankfull
y, doesn't boast the addition of fortified vitamins in some senseless a
mount. It truly is an enlightening blend of green tea, spearmint, lemon
grass and lemon verbena. Thus making it versatile refreshment for anyti
me of the day, whether it's right after meals or between meals, or just
before bedtime. Generally light and mild tasting, but that will depend
upon how long you steep it and if you add a sweetener of some form.<p>I
nteresting too, are the amusing comments and remarks that appear on the
packaging. Reading through this as you drink your tea makes it a distin
ctive experience. I never seen tea so clever! I wonder if consuming Taz
o really does improve a person's outlook on life and affect his or her
well being? I think it just could be the great taste and aroma and prob
ably the reassuring thought of doing your health a favor.<br />Try a co
uple of their teas - they are sure to please!<br />And Tazo Zen Green T
ea makes a good choice.

Review-2 :
 Blend a little apple cider mix into your Tazo Passion Tea and see what
happens. This creates a magical apple cider experience with hints of cr
anberry and hibiscus flowers gathered in an ancient land.<br /><br />Sn
uggle Up Spiced Cider is a good choice because you can then spoon in th
e exact amount you desire, but packets of cider work equally well. This
is the type of tea you would expect to enjoy after skiing in some dista
nt forest where you are also munching on homemade brownies or eggnog co
okies. Apple cider also works very well with the Gypsy Teas, especially
the Lemon Jasmine.<br /><br />The deep purple/red color and hibiscus fr
agrance of Tazo Tea is enough to make anyone passionate about tea. Yet,
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
is this really tea? It looks like tea, but it is really an herbal infus
ion made by steeping hibiscus flowers, orange peel, licorice, cinnamon,
rose hips, lemongrass and red poppy in a swirling of hot water.<br /><b
r />The original flavor is also slightly reminiscent of hot apple cider
or hot cranberry juice which is why it is extra delicious with a cider
mix. Tazo Teas are a mysterious experience into which you can lose your
self with a sense of abandon. So far I've tried more flavors than I can
count and have always been impressed.<br /><br />~The Rebecca Review

Review-3 :
 For a bag tea--which generally are not as good quality, this brews up
like a top quality loose tea. It's a delicious snappy black blend, with
a lovely scent and taste. One of my favorites.
```

In [156]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [158]:
```
# Three Reviews of cluster 5
count=1
m=[]
for i in range(3):
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if i < len(cluster5):
        print('Review-%d : \n %s\n'%(count,cluster5[i]))
        count +=1
m.append(cluster5[0])
```

```
Review-1 :
 Beetlejuice is an excellent and funny movie. Keaton is hilarious as th
e wacky beetlejuice. The great special effects help the film. I think t
his is one of the best movies ever made and I'm sure you'll agree. For
a good  time, watch beetlejuice!

Review-2 :
 What happens when you say his name three times? Michael Keaten stars i
n this comedy about two couples that live in an old two story house.  W
hile coming back from a supply store, the couple suddenly get caught in
side of a  &quot;broken-up&quot; bridge and then just before they start
to tumble down  into the lake, a board catches them.  But just when the
y've got their hopes  up, and small dog steps on the board and the car
starts to slide off the  bridge and into the lake waters.  A few minute
s later...<p>They find  themselves back into their home, they find that
somehow somehad light the  fireplace, as if done by magic.  From then o
n, they find a weird-looking  dead guy known as Bettlejuice.  The only
way they can get him for help is  to call him by his name three times a
nd he will appear at their survice.  But they soon wish that they have
never called his name, because  Bettlejuice was once a troublemaker but
he is the only one who can save  them, on the account that they said hi
s name three times.  They can't leave  their houses or else they will f
ind theirselves in another world with giant  sandworms.  This is a stel
lar comedy that you should see! Michael Keaton is  awesome as he plays
the leading role of Bettlejuice.

Review-3 :
 I THOUGHT THIS MOVIE WAS SO FUNNY, MICHAEL KEATON AS BEETLEJUICE WAS R
IGHT ON.  AND ALEC BALDWIN PLAYING A GEEKY HUSBAND IS PRICELESS.  I LOV
E THE DA-OO SCENCE.  THE KIDS AND I REALLY ENJOY THIS ONE.
```

In [159]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.3] DBSCAN Clustering

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.3.1] Applying DBSCAN on AVG W2V, SET 3

In [160]:
```
# Please write all the code with proper documentation
# List of sentence in X_train text
sent_x = []
for sent in x :
    sent_x.append(sent.split())


# Train your own Word2Vec model using your own train text corpus
# min_count = 5 considers only words that occured atleast 5 times
w2v_model=Word2Vec(sent_x,min_count=5,size=100, workers=4)

w2v_words = list(w2v_model.wv.vocab)
print("number of words that occured minimum 5 times ",len(w2v_words))
```

```
number of words that occured minimum 5 times  3498
```

In [164]:
```
# compute average word2vec for each review for sent_x .
from sklearn.preprocessing import StandardScaler
train_vectors = [];
for sent in sent_x:
    sent_vec = np.zeros(100)
    cnt_words =0;
    for word in sent: #
        if word in w2v_words:
            vec = w2v_model.wv[word]
            sent_vec += vec
            cnt_words += 1
    if cnt_words != 0:
        sent_vec /= cnt_words
    train_vectors.append(sent_vec)

#Standardising the data
data = StandardScaler().fit_transform(train_vectors)
```

- ELBOW METHOD TO FIND RIGHT EPSILON :

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [166]:
```
# function to determinethe distance of nth-nearest neighbour to all poi
nts in a multi-dimensional array
def n_neighbour(vectors , n):
    distance = []
    for point in vectors:
        temp = np.sort(np.sum((vectors-point)**2,axis=1),axis=None)
        distance.append(temp[n])
    return np.sqrt(np.array(distance))
```

In [167]:
```
min_points = 2*data.shape[1]

# Computing distances of nth-nearest neighbours
distances = n_neighbour(data,min_points)
sorted_distance = np.sort(distances)
points = [i for i in range(data.shape[0])]

# Draw distances(d_i) VS points(x_i) plot
plt.plot(points, sorted_distance)
plt.xlabel('Points (x_i)',size=14)
plt.ylabel('Distances (d_i)',size=14)
plt.title('Distances VS Points Plot\n',size=18)
plt.grid()
plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- OBSERVATION :- From above we can see that the right value of Epsilon is 10 because after that there is sharp rise in the value of distances (d_i's)

dbscan

In [171]:
```
# Function definition for implementing DBSCAN
from sklearn.cluster import DBSCAN

def dbscan(epsilon, samples, Data):
    from sklearn.cluster import DBSCAN
    db = DBSCAN(eps=epsilon, min_samples=samples, n_jobs=-1).fit(Data)

    # Number of clusters in labels, ignoring noise(-1) if present.
    n_clusters = len(set(db.labels_))
    print("Number of clusters for MinPts = %d and Epsilon = %f is : %d
 "%(samples,epsilon,n_clusters))
    print("Labels(-1 is for Noise) : ",set(db.labels_))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    print()
    return db
```

In [172]:
```
optimal_eps = 10
# Clustering with right epsilon
db1 = dbscan(optimal_eps, min_points, data)


# Clustering with  epsilon = 11
db2 = dbscan(11, min_points, data)


# Clustering with epsilon = 12
db3 = dbscan(12, min_points, data)


# Clustering with epsilon = 13
db4 = dbscan(13, min_points, data)
```

```
Number of clusters for MinPts = 200 and Epsilon = 10.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 200 and Epsilon = 11.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 200 and Epsilon = 12.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 200 and Epsilon = 13.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}
```

[5.3.2] Wordclouds of clusters obtained after applying DBSCAN on AVG W2V SET 3

In [176]:
```
from sklearn.cluster import DBSCAN
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
model = DBSCAN(eps=10, min_samples=min_points, n_jobs=-1).fit(data)

reviews = my_final['Text'].values
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    else :
        cluster2.append(reviews[i])


# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
```

```
No. of reviews in Cluster-1 :  3954

No. of reviews in Cluster-2 :  46
```

In [177]:

```
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

In [178]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
                 background_color ='white',
                 stopwords = stopwords,
                 min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [179]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 Starbucks coffee is in my opinon second to none. Best coffee I have ev
er had.

Review-2 :
 La bourbena e autunno in australia foglio dimmi dimmi dimi, voglio sup
re dutto dimme non mi pice la turbon, mi pice la bourbon, le occhi azzu
ri, e gli nuvoloso e rosso e tu?<p>P.S. All hail the mightly .... SAFRA
BOURBON!!!!!!!

Review-3 :
 The Dark Roast is the best of the coffee pods available for the Senseo
Coffee Machine.  Full body coffee and as with all the coffee made with
the Senseo Coffee Machine, never burnt.  Fantastic!
```

In [180]:

```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Visualizing The Clusters :

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [175]:
```
from sklearn.decomposition import PCA
pca_2d = PCA(n_components=2).fit_transform(data)

# Scatter plot for DBSCAN with Eps = 10
plt.figure(figsize=(18,9))
plt.subplot(221)
for i in range(0, pca_2d.shape[0]):
    if db1.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db1.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 10')
plt.ylabel('Dim_2')

# Scatter plot for DBSCAN with Eps = 11
plt.subplot(222)
for i in range(0, pca_2d.shape[0]):
    if db2.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db2.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 11')

# Scatter plot for DBSCAN with Eps = 12
plt.subplot(223)
for i in range(0, pca_2d.shape[0]):
    if db3.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db3.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 12')
plt.ylabel('Dim_2')
plt.xlabel('Dim_1')

# Scatter plot for DBSCAN with Eps = 13
plt.subplot(224)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(0, pca_2d.shape[0]):
    if db4.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db4.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1,c2], ['Cluster 1','Noise'])
plt.title('DBSCAN With Eps = 13')
plt.xlabel('Dim_1')

plt.show()
```

[5.3.3] Applying DBSCAN on TFIDF W2V, SET 4

In [182]:
```
# Please write all the code with proper documentation
# TF-IDF weighted Word2Vec
tf_idf_vect = TfidfVectorizer()

# final_tf_idf1 is the sparse matrix with row= sentence, col=word and c
ell_val = tfidf
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
final_tf_idf1 = tf_idf_vect.fit_transform(x)

# tfidf words/col-names
tfidf_feat = tf_idf_vect.get_feature_names()

# compute TFIDF Weighted Word2Vec for each review for sent_x .
tfidf_vectors = [];
row=0;
for sent in sent_x:
    sent_vec = np.zeros(100)
    weight_sum =0;
    for word in sent:
        if word in w2v_words:
            vec = w2v_model.wv[word]
            # obtain the tf_idfidf of a word in a sentence/review
            tf_idf = final_tf_idf1[row, tfidf_feat.index(word)]
            sent_vec += (vec * tf_idf)
            weight_sum += tf_idf
    if weight_sum != 0:
        sent_vec /= weight_sum
    tfidf_vectors.append(sent_vec)
    row += 1

#Standardising the data
data = StandardScaler().fit_transform(tfidf_vectors)
```

In [183]:

```
min_points = 2*data.shape[1]

# Computing distances of nth-nearest neighbours
distances = n_neighbour(data,min_points)
sorted_distance = np.sort(distances)
points = [i for i in range(data.shape[0])]

# Draw distances(d_i) VS points(x_i) plot
plt.plot(points, sorted_distance)
plt.xlabel('Points (x_i)',size=14)
plt.ylabel('Distances (d_i)',size=14)
plt.title('Distances VS Points Plot\n',size=18)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
plt.grid()
plt.show()
```

OBSERVATION :- From above we can see that the right value of Epsilon is 10 because after that there is sharp rise in the value of distances (d_i's)

In [184]:
```
optimal_eps = 10
# Clustering with right epsilon
db1 = dbscan(optimal_eps, min_points, data)


# Clustering with  epsilon = 11
db2 = dbscan(11, min_points, data)


# Clustering with epsilon = 12
db3 = dbscan(12, min_points, data)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Clustering with epsilon = 13
db4 = dbscan(13, min_points, data)
```

```
Number of clusters for MinPts = 200 and Epsilon = 10.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 200 and Epsilon = 11.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 200 and Epsilon = 12.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 200 and Epsilon = 13.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}
```

In [185]:

```
pca_2d = PCA(n_components=2).fit_transform(data)

# Scatter plot for DBSCAN with Eps = 10
plt.figure(figsize=(18,9))
plt.subplot(221)
for i in range(0, pca_2d.shape[0]):
    if db1.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db1.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 10')
plt.ylabel('Dim_2')

# Scatter plot for DBSCAN with Eps = 11
plt.subplot(222)
for i in range(0, pca_2d.shape[0]):
    if db2.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db2.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 11')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Scatter plot for DBSCAN with Eps = 12
plt.subplot(223)
for i in range(0, pca_2d.shape[0]):
    if db3.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db3.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 12')
plt.ylabel('Dim_2')
plt.xlabel('Dim_1')

# Scatter plot for DBSCAN with Eps = 13
plt.subplot(224)
for i in range(0, pca_2d.shape[0]):
    if db4.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker='o')
    elif db4.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='+')
plt.legend([c1,c2], ['Cluster 1','Noise'])
plt.title('DBSCAN With Eps = 13')
plt.xlabel('Dim_1')

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

[5.3.4] Wordclouds of clusters obtained after applying DBSCAN on TFIDF W2V SET 4

In [186]:
```
# Please write all the code with proper documentation
from sklearn.cluster import DBSCAN

model = DBSCAN(eps=10, min_samples=min_points, n_jobs=-1).fit(data)

reviews = my_final['Text'].values
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1.append(reviews[i])
    else :
        cluster2.append(reviews[i])
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
```

```
No. of reviews in Cluster-1 :  3921

No. of reviews in Cluster-2 :  79
```

In [187]:

```
# Three Reviews of cluster 1
count=1
m=[]
for i in range(3):
    if i < len(cluster1):
        print('Review-%d : \n %s\n'%(count,cluster1[i]))
        count +=1
m.append(cluster1[0])
```

```
Review-1 :
 this witty little book makes my son laugh at loud. i recite it in the
car as we're driving along and he always can sing the refrain. he's lea
rned about whales, India, drooping roses:  i love all the new words thi
s book  introduces and the silliness of it all.  this is a classic book
i am  willing to bet my son will STILL be able to recite from memory wh
en he is  in college

Review-2 :
 I can remember seeing the show when it aired on television years ago,
when I was a child.  My sister later bought me the LP (which I have to
this day,  I'm thirty something).I used this series of books &amp; song
s when I did my  student teaching for preschoolers &amp; turned the who
le school on to it.  I am now purchasing it on CD, along with the books
for my children 5 &amp;  2.  The tradition lives on!

Review-3 :
 Beetlejuice is a well written movie ..... everything about it is excel
lent! From the acting to the special effects you will be delighted you
chose to view this movie.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [188]:
```
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [189]:
```
# Three Reviews of cluster 2
count=1
m=[]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
for i in range(3):
    if i < len(cluster2):
        print('Review-%d : \n %s\n'%(count,cluster2[i]))
        count +=1
m.append(cluster2[0])
```

```
Review-1 :
 Using the Briel Chamonix machine I purchased from Amazon, I have produ
ced delicious lattes using these pods. The smell of the Espressione pod
s (regular and decaf) when opened is incredible - dark, rich roasts.  F
YI - the pods are produced in Italy, which is a good sign.  Also, each
pod has a handy paper "tab" so that you can remove it from your machine
without burning your fingers.  I compared these pods to Starbucks espre
sso pods (regular and decaf), and the Espressione are my favorite.  Onl
y one warning:  150 pods (individually wrapped) take up a lot of space,
so be sure you know where you're going to store them.  More good news:
each box of 150 is stamped with a "best if used by" date, which in my c
ase was 16 months after the date of purchase.

Review-2 :
 After years of using Starbuck's decaf and regular single serving espre
sso pods, I finally found a substitute that's more economical and MUCH
better tasting.<p>I've used three different brands of pods in my Starbu
ck's Barista machine, and have found that the quality of the pod and th
e flavor of the espresso inside is, by far, the best in the Espressione
pods. The flavor is more full bodied and stronger than in the Starbuc
k's pods. I actually use one pod to make two shots of espresso, and the
flavor is more intense than when I buy my specialty drinks from Starbuc
ks or other coffee houses.<p>Also, the pod itself is of very high quali
ty. Each pod is individually wrapped in an airtight, hard foil wrapper,
and I've never opened a package yet that contained a torn pod with coff
ee spilling out. Such is not the case with other lesser brands.<p>The t
ab on each pod is also handy, making it easier to switch out a hot spen
t pod with a new one when making a second round of espresso.

Review-3 :
 "Sitting on the porch of a bungalow on a tea plantation in Darjeeling,
you can see the peaks of Everest and Kanchenjunga as you sip a cup of t
his exquisite brew. Life is good." ~tea packet<br /><br />If you want a
subtle tea with a smooth, soothing flavor, then Darjeeling is an excell
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ent choice. The slightly floral flavor is much milder than an Earl Gre
y. I especially love this blend when eating a Caramel Apple Cobbler Cli
f Bar. For some reason, the flavors mingle nicely. I think it is the sl
ightly nutty flavor of the Darjeeling.<br /><br />While many of the Taz
o teas are blends of a variety of teas, this tea is purely a blend of O
rganic Black Darjeeling teas. The tea is grown near India's border with
Bhutan. There are only "true Darjeeling" tea leaves in this tea.<br /><
br />This tea is just perfect when blended with an Organic Sucanat: Nat
ural Sugar Cane. You might also want to try using "milk powder" instead
of cold milk with your tea. I find it helps to keep the tea strength an
d adds a creamy element to the smoothness of this tea.<br /><br />Darje
eling is a very satisfying blend for morning tea. You might also enjoy
AWAKE and Earl Grey if you enjoy this tea.<br /><br />I just love the p
ackaging of the Tazo Teas. The individual wrappings have a nice crinkly
sound and the colors are all so soothing and quite beautiful. These tea
s are definitely for the aesthetically aware. Drinking Tazo teas litera
lly becomes a magical experience. There is something mysterious going o
n here. The teas even seem to have a magical glint or luster to them. W
hen you drink Tazo teas the world looks more beautiful and you feel muc
h more aware of life itself. I find that my creativity soars after drin
king Tazo teas.<br /><br />For more information on the mysteries of Taz
o Tea, read the review for the Tazo Assorted Teas.<br /><br />For a cal
ming experience try the Tazo Calm Herbal Infusion which is perfect for
quiet evenings, after completing a yoga workout or just to sip on while
you write in<br />your journal.<br /><br />~The Rebecca Review
```

In [190]:

```python
# Please write all the code with proper documentation
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

comment_words = ' '
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in m:

    # typecaste each val to string
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

procedure for k means

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- STEP 1 :- Text Preprocessing
- STEP 2 :- Taking all text data and ignoring class variable .
- STEP 3:- Training the vectorizer on text_data and later applying same vectorizer on text_data to transform it into vectors bow,tfidf,avgw2v,tfidfw2v
- STEP 4:- Applying Elbow Method using K-means++ in order to find optimal value of K(i.e. number of clusters)
- STEP 5:- Draw loss VS K-values plot
- STEP 6:- Once we find optimal value of K then again train K-Means++ and K-medoids for clustering text_data into K clusters
- STEP 7:- Reading reviews manually for each cluster
- STEP 8:- word cloud for cluster

procedure for agglomerative

- STEP 1 :- Text Preprocessing
- STEP 2 :- Taking all text data and ignoring class variable .
- STEP 3:- Training the vectorizer on text_data and later applying same vectorizer on text_data to transform it into vectors avgw2v,tfidfw2v
- STEP 4:- Implementing Hierarchical Clustering using multiple values of clusters .
- STEP 7:- Reading reviews manually for each cluster
- STEP 8:- word cloud for cluster

procedure for dbscan

- STEP 1 :- Text Preprocessing
- STEP 2 :- Taking all text data and ignoring class variable .
- STEP 3:- Training the vectorizer on text_data and later applying same vectorizer on text_data to transform it into vectors avgw2v,tfidfw2v
- STEP 4:- Applying the Elbow Method in order to find the right value of Epsilon
- STEP 5:- Draw distances VS points plot

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- STEP 6:- Implementing DBSCAN with various values of Epsilon including the optimal value of Epsilon
- STEP 7:- Reading reviews manually for each cluster
- STEP 8:- word cloud for cluster
- STEP 9:- Draw Scatter plots for DBSCAN with various Epsilon values

[6] Conclusions

In [193]:
```
# Please compare all your models using Prettytable library.
# You can have 3 tables, one each for kmeans, agllomerative and dbscan

# Creating table using PrettyTable library
from prettytable import PrettyTable

names = ['K-means++ for BoW','K-means++ for TFIDF','K-means++ for Avg_W
ord2Vec','K-means++ for tfidf_Word2Vec']

optimal_k = [bow_means_k,tfidf_means_k,avg_w2v_means_k,tfidf_w2v_means_
k]

numbering = [1,2,3,4]

# Initializing prettytable
table = PrettyTable()

# Adding columns
table.add_column("S.NO.",numbering)
table.add_column("MODEL",names)
table.add_column("Number of Clusters ",optimal_k)

# Printing the Table
print(table)
```

```
+-------+------------------------------+---------------------+
| S.NO. |             MODEL            | Number of Clusters  |
+-------+------------------------------+---------------------+
|   1   |      K-means++ for BoW       |          3          |
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
|   2   |     K-means++ for TFIDF      |          3          |
|   3   |  K-means++ for Avg_Word2Vec  |          3          |
|   4   | K-means++ for tfidf_Word2Vec |          3          |
+-------+------------------------------+---------------------+
```

In [195]:

```python
# Please compare all your models using Prettytable library.
# You can have 3 tables, one each for kmeans, agllomerative and dbscan

# Creating table using PrettyTable library
from prettytable import PrettyTable

names = ['hierarchichal for Avg_Word2Vec','hierarchichal for tfidf_Word
2Vec']

optimal_k = [{2,5,10},{2,5,10}]

numbering = [1,2]

# Initializing prettytable
table = PrettyTable()

# Adding columns
table.add_column("S.NO.",numbering)
table.add_column("MODEL",names)
table.add_column("Number of Clusters ",optimal_k)

# Printing the Table
print(table)
```

```
+-------+---------------------------------+---------------------+
| S.NO. |              MODEL              | Number of Clusters  |
+-------+---------------------------------+---------------------+
|   1   |  hierarchichal for Avg_Word2Vec |      {2, 10, 5}     |
|   2   | hierarchichal for tfidf_Word2Vec |      {2, 10, 5}     |
+-------+---------------------------------+---------------------+
```

In [194]:

```python
# Please compare all your models using Prettytable library.
# You can have 3 tables, one each for kmeans, agllomerative and dbscan
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
# Creating table using PrettyTable library
from prettytable import PrettyTable

names = ['dbscan for Avg_Word2Vec','dbscan for tfidf_Word2Vec']

optimal_k = [2,2]

numbering = [1,2]

# Initializing prettytable
table = PrettyTable()

# Adding columns
table.add_column("S.NO.",numbering)
table.add_column("MODEL",names)
table.add_column("Number of Clusters ",optimal_k)

# Printing the Table
print(table)
```

```
+-------+---------------------------+---------------------+
| S.NO. |           MODEL           | Number of Clusters  |
+-------+---------------------------+---------------------+
|   1   |  dbscan for Avg_Word2Vec  |          2          |
|   2   | dbscan for tfidf_Word2Vec |          2          |
+-------+---------------------------+---------------------+
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD