39.15.Amazon_food_review_truncated_SVD

July 6, 2018

1 Amazon food review dataset apply truncated SVD

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

2 Objective

1. Take 2000 words by TFIDF importance
2. Calculate cooccurance matrix with neighbourhood of size 5 and count how many times wi occur in context of wj
3. Then do truncated SVD
4. try multiple value of k(find optimal k by amount of variance explained)[use singular value]
5. cluster(kmeans k=50) word vector for top 2000
6. word cluster together should be related

3 Import data and libraries

In [1]:
```
from sklearn.manifold import TSNE
import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
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
from sklearn.cross_validation import train_test_split,KFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.cross_validation import cross_val_score
from collections import Counter
```

```
from sklearn.metrics import accuracy_score
from sklearn import cross_validation
from sklearn.grid_search import GridSearchCV
from sklearn.linear_model import LogisticRegression

con = sqlite3.connect('database.sqlite')

#get only +ve and -ve review
raw_data = pd.read_sql_query("""SELECT * FROM Reviews WHERE Score != 3""", con)
```

```
C:\Users\suman\Anaconda3\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning:
  "This module will be removed in 0.20.", DeprecationWarning)
C:\Users\suman\Anaconda3\lib\site-packages\sklearn\grid_search.py:42: DeprecationWarning: This
  DeprecationWarning)
```

4 Data preprocessing

In [2]:
```
filtered_data=raw_data
# Score>3 a positive rating, and score<3 a negative rating.
def partition(x):
    if x < 3:
        return 'negative'
    return 'positive'

#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

filtered_data.sample(5)
filtered_data['Score'].value_counts()

#Sorting data according to ProductId in ascending order
sorted_data=filtered_data.sort_values('ProductId', axis=0, ascending=True, inplace=Fals

#Deduplication of entries for same profilename,userid, time, text and take first eleme
sorted_data=sorted_data.drop_duplicates(subset={"UserId","ProfileName","Time","Text"},
```

In [43]:
```
#take only 50000 data
print('total data \n',sorted_data['Score'].value_counts())
#clean_data=sorted_data.sample(frac=1).groupby('Score').head(10000)
#take stratified sampling i.e. positive and negative reviews are proportionate to raw
#testing
_ , clean_data = train_test_split(sorted_data, test_size = 50000, random_state=0,stra
clean_data['Score'].value_counts()
```

```
total data
 positive    307063
```

```
negative     57110
Name: Score, dtype: int64
```

```
Out[43]: positive    42159
         negative     7841
         Name: Score, dtype: int64
```

```
In [68]: # Clean html tag and punctuation
         import re
         import string
         from nltk.corpus import stopwords
         from nltk.stem import PorterStemmer
         from nltk.stem.wordnet import WordNetLemmatizer

         stop = set(stopwords.words('english')) #set of stopwords
         sno = nltk.stem.SnowballStemmer('english') #initialising the snowball stemmer

         #substitute html tag and punctuation
         def cleanhtml(sentence): #function to clean the word of any html-tags
             cleanr = re.compile('<.*?>')
             cleantext = re.sub(cleanr, ' ', sentence)
             return cleantext
         def cleanpunc(sentence): #function to clean the word of any punctuation or special ch
             cleaned = re.sub(r'[?|!|\'|"|#]',r'',sentence)
             cleaned = re.sub(r'[.|,|)|(|\|/]',r' ',cleaned)
             return  cleaned
         #print(sno.stem('tasty'))

         i=0
         str1=' '
         mystop={'of','four','one','would'}
         final_string=[]
         all_positive_words=[] # store words from +ve reviews here
         all_negative_words=[] # store words from -ve reviews here.
         s=''
         #Create new catagory as Cleanedtext after removing htmltag and punctuation and upperc
         for sent in clean_data['Text'].values:
             #change later
             #testing
             filtered_sentence=[]
             #print(sent);
             sent=cleanhtml(sent) # remove HTMl tags
             for w in sent.split():
                 for cleaned_words in cleanpunc(w).split():
                     if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                         if((cleaned_words.lower() not in stop) & (cleaned_words.lower() not i
                             s=(sno.stem(cleaned_words.lower())).encode('utf8')
```

```
                    filtered_sentence.append(s)
                    if (clean_data['Score'].values)[i] == 'positive':
                        all_positive_words.append(s) #list of all words used to descr
                    if(clean_data['Score'].values)[i] == 'negative':
                        all_negative_words.append(s) #list of all words used to descr
                else:
                    continue
            else:
                continue
        str1 = b" ".join(filtered_sentence) #final string of cleaned words

        final_string.append(str1)
        i+=1

clean_data['CleanedText']=final_string
print(clean_data.shape)
#Sort data on timestamp
clean_data=clean_data.sort_values(by=['Time'],ascending=False)
#clean_data
clean_data['CleanedText'].sample(2)
clean_data['CleanedText'].iloc[0]
```

```
(50000, 11)
```

Out[68]: b'plum sweet juici aroma like perfum doesnt hurt good'

5 Get top 2000 words by TFIDF score and create co-occurence matrix by window 5

In [69]:
```
x=clean_data['CleanedText'].values
y = clean_data['Score']
#n=x.shape[0]
#n1=int(n*.3)
#X_test_raw = x[0:n1]
#X_train_raw= x[n1:n+1]
#y_test=y[0:n1]
#y_train=y[n1:n+1]
# Create BOW and try grid search for logistic regreession with penalty l1 and l2
tf_idf_vect = TfidfVectorizer()
final_counts = tf_idf_vect.fit_transform(x)
#use the same vectors to convert test data
#X_test=count_vect.transform(X_test_raw)
indices = np.argsort(tf_idf_vect.idf_)[::-1]
features = tf_idf_vect.get_feature_names()
#testing
top_n = 2000
```

```
top_features = [features[i] for i in indices[:top_n]]
print (top_features[0:20])
print('len of top feature',len(top_features))

#remove other words from review
final_string=[]
all_string=[]

i=0
for sent in clean_data['CleanedText'].values:
    i=i+1
    filtered_sentence=[]
    for w in sent.decode('utf8').split():
                #print(w)
                if(w in top_features):
                    filtered_sentence.append(w.encode('utf8'))
                    #if (i==1):
                    #print('print')
                else:
                    continue
    str1 = b" ".join(filtered_sentence) #final string of cleaned words
    if (str1.decode('utf8') !=''):
      final_string.append(str1)
    all_string.append(str1)

clean_data['CleanedText']=all_string
#Now final_string is ready to work with
#print(clean_data['CleanedText'].shape)
final_string[0]
```

```
['île', 'foetida', 'filo', 'plait', 'filteer', 'filterbag', 'filtr', 'plactic', 'finagl', 'pla
len of top feature 2000
```

Out[69]: b'disform'

In [80]:
```
#Convert to cooccurance mat
#type(final_string)
#print(final_string)
window=5
len1=len(top_features)
#print(len)
m=np.zeros([len1,len1])
columns=top_features
rows=top_features
df=pd.DataFrame(m,columns=columns,index=rows)
#print(df)
```

```
def cal_occ(sentence,df):
    sen=sentence.split()
    l=len(sen)-1
    for i,word in enumerate(sen):
        for j in range(max(i-window,0),min(i+window+1,l+1)):
            if word!=sen[j]:
                #print('printing',word,sen[j])
                df[word][sen[j]]+=1

for sentence in final_string:
    #print('call',sentence)
    cal_occ(sentence.decode('utf8'),df)

print(df.shape)
```

```
(2000, 2000)
```

```
In [81]: #top_features
         #final_string[0]
         print(len)
         len(top_features)
         #df
         from sklearn.preprocessing import StandardScaler
         #scaler = StandardScaler(with_mean=False).fit(df)
         #df = scaler.transform(df)
```

```
<built-in function len>
```

```
In [90]: #df
```

```
In [74]: type(U)
```

```
Out[74]: numpy.ndarray
```

6 Create countvectorizer using cooccurence matrix

```
In [85]: #count_vect = CountVectorizer(vocabulary=top_features) #in scikit-learn
         #X = count_vect.fit_transform(final_string)

         #print(X.shape)

         #Cooccurance matrix
         #X = (X.T * X) # this is co-occurrence matrix in sparse csr format
         #X.setdiag(0) # sometimes you want to fill same word cooccurence to 0
         #print(X.todense())
```

```
#print(count_vect.vocabulary_)

#Create truncated SVD
from sklearn.decomposition import TruncatedSVD

#Try different component
l=[20,50,100,150,200,250]
for i in l:
  svd = TruncatedSVD(n_components=i, n_iter=7, random_state=0)
  svd.fit(df.values)
  #print(svd.explained_variance_ratio_)
  l1=svd.explained_variance_ratio_
  print('% variance explained with component ',i,svd.explained_variance_ratio_.sum())
  #print('singular values',svd.singular_values_)

#So looks like with 25 component 96% variance is explained
#print(svd)
```

```
% variance explained with component  20 0.596657737216
% variance explained with component  50 0.716447280124
% variance explained with component  100 0.825836500755
% variance explained with component  150 0.869494648993
% variance explained with component  200 0.913161825629
% variance explained with component  250 0.956826688872
```

SO by 250 component 96% variance is explained # Use SVD

In [86]:
```
#VT = svd.components_
#TruncatedSVD is basically a wrapper around sklearn.utils.extmath.randomized_svd; you

from sklearn.utils.extmath import randomized_svd

U, Sigma, VT = randomized_svd(df.values,
                              n_components=250,
                              n_iter=50,
                              random_state=0)

print('U value\n')
#print(U)
print('sigma value\n')
#print(Sigma)
print('VT value\n')
#print(VT)
print(U.shape,Sigma.shape,VT.shape)
```

```
U value

sigma value
```

```
VT value

(2000, 250) (250,) (250, 2000)
```

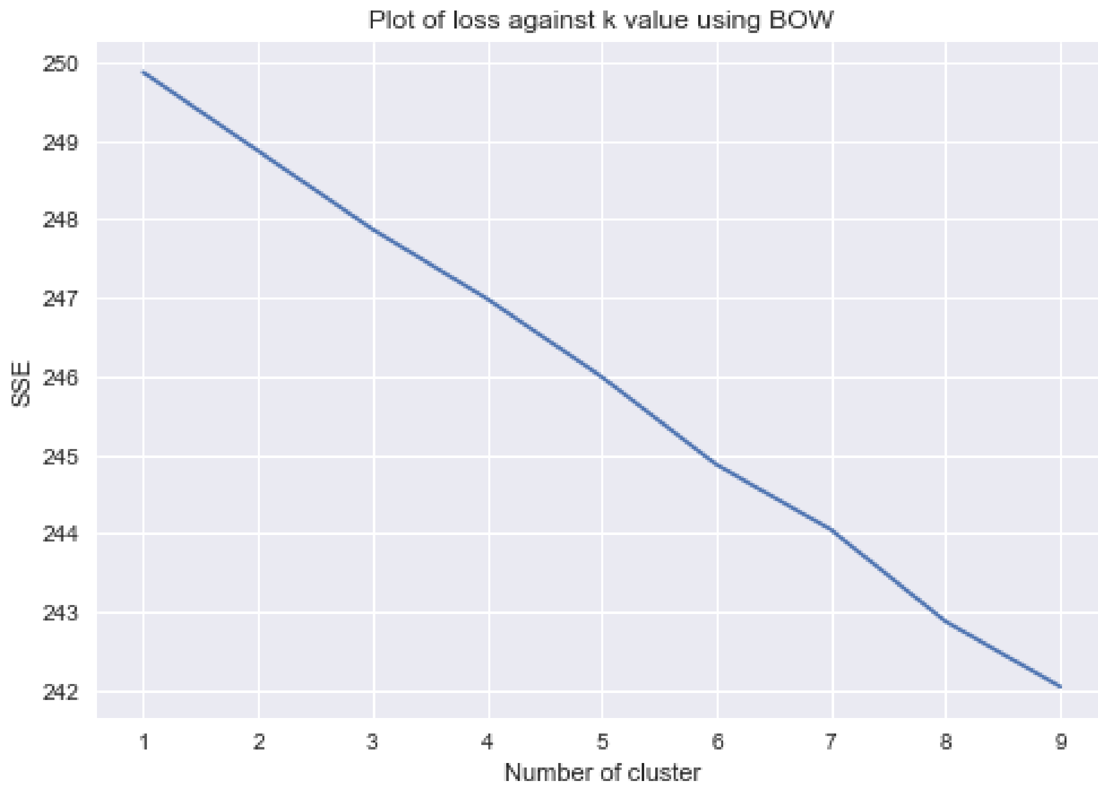
7 Form cluster of 10 using those important words SVD value

```
In [89]: from sklearn.cluster import KMeans
         # Now U is vec presentation of words
         n_clusters=10
         kmeans=KMeans(n_clusters=10, random_state=0).fit(U)
         kmeans.cluster_centers_

         sse = {}
         for k in range(1, 10):
             kmeans = KMeans(init='k-means++',n_clusters=k, max_iter=100).fit(U)
             sse[k] = kmeans.inertia_ # Inertia: Sum of distances of samples to their closest
         plt.figure()
         plt.plot(list(sse.keys()), list(sse.values()))
         plt.title("Plot of loss against k value using BOW")
         plt.xlabel("Number of cluster")
         plt.ylabel("SSE")
         plt.show()

         #a=np.where(kmeans.labels_ == 1)[0]
         #b=np.where(kmeans.labels_ == 0)[0]
         #check 5 text for cluster 1

         kmeans = KMeans(init='k-means++',n_clusters=50, max_iter=100).fit(U)
         n_clusters=50
         print(a.shape)
         for i in range(n_clusters):
           a=np.where(kmeans.labels_ == i)[0]
           print('in cluster \n',i)
           print(a[0:])
           for j in a:
             print(top_features[j])
```




```
(1,)
in cluster
 0
[   0    1    2 ..., 1997 1998 1999]
île
foetida
filo
plait
filteer
filterbag
filtr
plactic
finagl
placei
finali
financ
findabl
pkt
findaspr
finefar
pkging
finess
```

fing
fingerfood
pizzell
finiski
finki
pizzaria
pizzaiola
filmsi
filltrat
plani
figh
plavor
plave
fica
platypus
ficus
fidd
fiddlestick
fieldss
fifficult
figgi
figish
planifolia
fike
plaqueoff
planti
planthard
plantfus
fila
filful
filibert
filippo
fillo
finlandia
pizett
finley
pipa
piroulin
piripiri
firepot
firest
firewood
firmiliar
firsat
piperin
firshand
firsli
pip

pioki
firstit
firtst
fishermen
fishnet
fishstick
fissur
pinhol
fiter
pingshui
pirtl
pistacho
finncrisp
pitr
finnecki
piut
piulp
pitur
finnicki
pittmix
finniki
finnish
pittbul
pitiabl
pith
pite
fiocchetti
firecrack
firefight
pitango
firend
pistol
pistil
pistchio
playabl
fiberi
playdoh
fentimin
podner
podmerch
podg
podd
feng
pocuch
pocono
fennugreek
fenstermach
fentiman

pobabl
podunk
feodorovich
poa
pnw
pnfs
fer
pnb
pnam
ferdi
plzd
poduct
poe
fermi
feistier
polarpack
feeel
feell
feh
poivr
feild
poisin
feint
feisti
pointsplus
fel
femor
feldcamp
feldman
poictur
poi
poha
pogo
felix
felll
poetic
feloni
ferel
fern
playdooh
pleasnt
pli
feul
plesent
plese
feverish
fevertre
pleni

few
fexex
ffamili
ffor
pliant
ffortless
fhe
pleaas
plazma
playmat
playground
fiat
fiberglass
fettucin
plier
ferquent
fescu
ferrera
plumros
plumpynut
ferrerro
ferric
plummit
fert
plummer
plumber
ferul
plugra
plight
fesh
fess
fetaccini
ploughman
plot
fetish
plocki
fettuccini
pllllleas
plink
piney
pinewood
pinenut
flouresc
phallic
phal
phad
pgtip
pgpr

pgh
pfrespak
pfpc
flounder
pfffft
peychaud
floorsweep
pewter
pew
pevent
petzlif
petunia
petticoat
flourid
petso
flourless
floz
phantom
pharmacut
petsaf
pheramon
flexse
philistin
philanthropi
philanthrop
philadelphian
flimsey
flimsier
phew
pheronet
phenylalin
floati
flinti
phenylalani
phenylalamin
phenomon
flipper
phenomenen
flippin
phenobarbit
phelan
phd
petshop
petruska
flexatarian
focd
pestil
flybuy

flypap
pestal
foamer
fob
pessim
pesci
pescetarian
pescatarian
pervious
flux
pervert
pervers
foco
fodmap
perugia
foertun
perturb
pertin
pertfect
foetid
fluxuat
flutter
flub
fluey
fluditi
petrol
petrodex
petrifi
petri
petrefi
petra
petperk
fluentli
petm
petguy
peta
fluffer
petey
fluffiest
petersburg
fluor
flur
flushabl
petcar
petbot
petag
phillipp
fitst

piecrust
pilao
flamenco
pik
pignoli
flandr
flann
pigg
flapjack
flashcard
flathead
flatish
pilat
flatt
picutr
pictu
flattop
pickypicki
flavanol
pickmeup
flavarcol
flavir
pilau
flavocol
fizzer
fitsugar
fitt
fitter
fitz
fivehundr
fiver
fixat
fixi
fizzel
pimb
flambeau
fji
pillpocket
flacker
flagil
flagstaff
pilliow
pilgrimag
pilfer
pileup
flakier
flavius
pickest

phillippin
phospat
flavourt
flavr
photoshoot
flaxmilk
photograh
flaxusa
phosporus
flee
fleec
phos
phyllo
fleishmann
phonetag
flem
fleshi
phoeb
flex
phlem
flexabl
philosoph
philmont
phylli
flavourless
flavorcol
flavori
pickemup
pickel
pickard
flavord
flavordo
picdog
flavorfil
picaridin
flavorfukl
flavorgan
pib
physican
piano
flavoric
piac
phytonutri
flavorsw
phytocyt
phytochem
physiqu
flavorul

febreez
featurett
featherweight
experci
exorcis
premak
expat
expatri
prelud
prelin
expediti
expen
prehistori
experiec
exlaim
pregneson
expierenc
expl
preggo
prefrenc
prefix
explainatori
explaiun
explanatori
exploit
premi
preminum
expound
prepara
excut
prescott
prescient
prescib
execept
prerequisit
preprint
preposter
preponder
exerienc
exersic
existenti
preorder
preoocupi
preoccupi
preoccup
prenat
premuim
premonit

premius
premiun
exil
preferenti
expr
preselect
prayish
precio
extclus
precancer
exterm
preasant
extinguish
extort
extractd
extractish
prduce
extractor
ext
extraordinair
extravaganzo
prato
pratfal
prasi
pranw
extravergin
extreem
pramesan
extri
extant
exstract
prefab
predicat
expreesso
expreso
predominatey
expressio
expressivo
prednisolon
predisposit
expung
predilect
exract
exrem
exstat
prediabet
exsist
predawn

exspeci
precut
exspect
exspeic
precondit
exspens
preconceiv
exctract
excpet
extroadinari
princeton
everythin
everythong
everyweek
everyyh
prioirti
prioduct
evewn
eveyon
eveyth
evinc
everyht
evironment
primula
evn
evoc
primia
primer
evolutionali
evri
evrywher
evvvveeerr
everythign
everyflavor
ewe
probalbi
problema
everal
problabl
probioticsmart
probiotc
everclear
probioit
probelem
probar
evergrow
probal
priviledg

everida
everon
probabali
everorgan
everremind
everri
prmote
everycup
everydayhealth
everyewher
evwn
prik
presentaion
excellet
pretz
pretyt
prettti
excelencia
excelent
pretoria
pretezel
preterm
excellect
excellentdeliveri
excelnt
pretzelli
excema
exceptthat
excers
exciti
exclusif
exclusivley
preshampoo
preset
preservt
exclusivli
excedrin
pretzl
ewwww
exager
priciest
ewwwww
ewwwwwwwwww
exac
priceto
priceth
pricer
pricepoint

exacto
exactuli
prferenc
prev
prfere
exc
excact
previouli
previo
preview
preventit
excalibur
prevar
excat
extrins
featheri
fastfood
popcornwhatels
farmingtonhil
faro
popcicl
farrrr
fasatchi
pooti
poorest
fashin
fasta
pooper
fark
fastpitch
fatcat
fatcream
poofi
fatdog
fatdogmustard
fatertast
fathi
fatkitten
ponytail
poper
popocorn
poni
popup
familiy
familyit
fanatast
porchini
porchin

fanboi
porcelin
porc
porag
fancyfeast
fanfar
popp
fanfreakintast
fantastc
fantastica
fantastico
fantasticveri
popppi
fantstic
farder
farewel
poppabl
pont
fatoush
familey
polpo
favort
favorti
favortit
favotit
polynesian
polym
polyethylen
polydextros
polycycl
polycarbon
polo
favorito
fax
fay
fazer
fbd
fbi
polka
politician
fearsom
polino
fearur
polyscia
favoritesth
pomtast
fatto
pomston

pompous
fattest
pomodoro
pommeri
pomm
fattiest
pomgran
pomganit
pometta
pomerainian
favoritebreakfast
pomengrant
pomengran
pomelo
fatwallet
fauchon
pomegranet
fauna
pomad
faverit
favoit
familiari
porphyra
extrovert
poulsbo
faaaaaantast
faaaar
faboo
poundcak
poundbag
poundag
faborit
pounch
fabulos
poultic
pouf
pourgouri
poudr
pouchkin
fabulousthey
pottl
fabuluo
facecream
facin
pottasium
potstick
potroast
faa

pourin
facp
exxtrem
extrud
practico
practicament
exuber
ppor
ppm
exxcel
exxit
powerlift
exxxtra
pourov
powerf
powercaf
powerberri
eyerol
powederi
ezcema
ezek
pow
poverti
ezin
facori
facter
porport
fallot
falafil
faleev
posion
posh
falili
fallaci
fallback
fallon
portrait
fallout
possiabl
falsest
falsifi
porter
portal
fam
famer
porselain
porrig
familar

posol
possibilitiesavail
faction
faeriesfinest
potencey
faculti
potatoey
facundo
potat
faddl
potassim
potasium
faecium
postur
postprandi
fairchild
fago
postpartum
postmeal
postiv
postassium
faillac
failsaf
fairacr
fairbank
possit
pert
fogchas
problemno
fogger
pachag
galic
pacakg
pacaket
gallari
pacak
gallat
pablum
paack
ozzi
galleri
ozarka
gallet
gallolea
gallston
galvan
galveston
galz

oxford
gam
gamier
gamze
gan
pacifica
galaxey
packa
gaia
padfilt
paddywack
paddlefish
gaget
paddington
gagia
padano
gah
paco
packsg
packmat
galanga
gaiser
gait
gala
galacia
galactica
packagign
galang
packadg
packackag
packack
owi
gander
garni
garicin
oversel
overseason
garish
garlex
garlicpepp
garm
overrul
overroast
garnet
overreat
oversit
overreact
garnigh

overproduct
overproduc
garrido
overpowering
garvey
overpour
overpopul
overplay
garic
oversold
ovul
gaood
overwork
overwis
overwir
gangster
overwhelem
overwelm
ganja
overus
ganoush
overthink
oversp
overtaken
garbanza
garcinia
oversuppli
overstew
garf
gargl
overspil
overspic
gaffigan
paesano
gad
panchetta
fumi
fumig
pane
pandoro
pander
pandem
funfresh
fungicid
funish
pancit
pancakk
pani

funitur
panca
panayoti
funkier
panarello
funnier
funniest
panacea
funyun
pamplona
fumar
fulsom
pamona
pantheon
fuelbelt
papagalo
papadum
fufu
fuggedaboutit
paolo
fughedaboudit
fuhgedabowdit
panti
fuhrman
panicki
pantainorasingh
pantai
fulfillmet
panorama
panoest
pano
pannini
pannela
fullbodi
fulln
furgirl
pamida
pagkag
fwoot
palek
futurist
futz
fuur
palac
fuze
fuzzboy
fuzziwigg
pake

pakc
futhermor
fy
fye
painstak
gab
gabanzo
painfre
gabiel
gabixlerreview
palenqu
futher
furit
palmeri
pamelasproduct
furkid
furnish
furrow
furter
furthest
palo
palmolein
palmit
palmetto
fush
futaba
palma
pallit
pallett
fussbudget
pallat
paleybar
fustrat
palet
gasm
gassey
overnit
osoba
gevailia
otakus
otaku
oswego
osu
ostrim
gfaf
osteoporosi
gfbc
osso

gfcfsf
getwellfeelwel
osmophil
osmanthus
gfg
osem
osectomi
gfi
gfic
gfma
oryza
otder
oth
orthophosph
ouc
ousid
ourself
ouncn
geta
ouma
oui
ouhui
oughta
geterdon
gett
ottoman
gettu
gettin
otreat
otr
gettinng
oti
othewis
otherworld
otherss
otherhand
ortiz
gger
gerolstein
organcvill
orgini
ghostbust
giadia
gianni
giardinera
giardiniera
organic
organg

organc
orgnan
gibraltar
orgainix
gice
giftabl
orfarm
orem
gigiant
gil
gilbert
oregahno
orgnaic
orgonit
ggs
orijin
orrigion
orrder
orphanag
ghanouj
gharardelli
ornish
ornament
orlistat
orleanean
gherkin
ghey
ghiridelli
origon
origion
original
ghiradhelli
ghirardel
ghirardell
origami
orig
ghirardella
oridnari
gest
outa
gastrectomi
gebhardt
gbs
overcam
overbrown
overbright
gci
overbold

gday
gdiaper
gdl
geht
gazzetta
overadvertis
overacid
gelli
gello
ovenproof
gemani
gemini
ovat
genach
gbj
overcoook
genardi
gaur
gastriti
overlay
overland
gastronomi
overhyp
overhwhelm
gastronomiqu
overful
overflavor
overfish
gav
gaylord
gavalia
overextend
overexagger
overestim
gawd
overdri
overdramat
overdraft
overdr
gay
ovalin
generali
germless
gerard
outlaw
genus
georgetown
georgous

geraldin
outgrew
outgo
geranium
outerwear
outermost
outdoorsman
outliv
outdoorsi
gerat
gere
gereatr
geriatr
outclass
outch
geriatrx
germaphob
outand
gentlest
gentian
ouuuuuch
genit
outweight
outward
outther
outter
outtak
genger
outstandingal
geni
genious
outsmart
genki
gennaro
genoa
genocid
genrat
gens
genseng
outpati
outo
outmost
outmod
papal
papau
fudgier
pemberton
penelop

pendleton
forrest
forsak
penchant
penc
forst
forsur
forta
pelt
peni
peloponnes
pelligrino
fortiflora
fortnam
pell
fortnight
pelicanbay
pel
fortnum
fortuna
formular
penis
pekin
peop
pepin
pepermint
pepercorn
forend
pepeer
pepe
pepar
pepa
peoplefood
forgiven
peoe
penj
peoblem
forkful
pentobarbit
pentaphyllum
pent
penquin
forlif
formallow
penna
formaula
pekines
peki

foreleg
peckish
fpos
pedestian
peddler
peddl
pedal
peda
ped
fps
frabjous
frackin
peck
pedialyt
pecanish
pecanflavor
fraganc
peber
fragmentat
peatmoss
peati
frakenfood
peasant
framer
pedi
foyer
peke
foulest
peiod
fortunat
fortunit
pegasuss
forword
fos
fosomax
fossil
fotr
foud
peelu
foxi
peeler
foundland
foundri
fourpound
fourteenth
pedro
fourti
pedicur

pediatrictian
pediatrian
pepit
peporoni
francais
foodborn
fomula
permuat
fondor
fontain
permanec
fontina
permalink
foodbank
perlit
perkin
foodgawk
perpendicular
foodist
periwinkl
peristalsi
foodland
periperi
periot
periodont
foodlock
perimit
foodmak
fom
folow
foodmil
foligno
personnali
foggi
foist
persnick
perski
foldger
persimmon
folicin
pershap
pershabl
persev
perrfect
folksi
perscrit
perscript
perscrib

persay
follicl
perrrfect
folliclli
followthrough
perri
perillo
perigord
forefing
pepporoni
foothil
footlong
footnot
perahp
footrac
peqin
footstep
peptid
pepsin
fop
peppod
perciev
peppier
forag
pepperwood
forastero
peppermintmint
pepperk
foreclos
peppep
peppar
perch
foosh
perier
foodwork
foodrenegad
foodservic
pergola
perginotti
foodstor
foodth
foodthi
foodwis
perfom
perfict
foofoo
percul
perfectionist

perfection
foood
perfct
perfact
perez
foor
perel
perefect
perdu
peari
francesco
papdi
parley
fromt
fromturn
frond
fronm
parmalat
parmagiana
parmacotto
frontlin
frontyard
froom
from
froos
froyo
pariti
frozt
paricular
frr
parfum
frsenergi
parenthood
frsh
frolicki
parenthes
partialiali
partygo
fright
frigid
frilli
partida
friojol
particuar
particluar
frisco
partiallli
frist

parmmesan
fritata
fritatta
partaken
fritolay
frivol
froci
frofti
froggi
parotid
parodi
frst
fructiuss
pas
fruttato
parachut
paraben
frutal
fruti
papya
fruticosa
frutini
paprica
papper
pappardella
pappardall
fruster
papous
paperwork
frypan
ftm
paperpl
fuction
papercup
fuddi
fudgesicl
fruta
fructo
paraphanalia
pardner
fructor
frugl
fruictos
parbroil
fruitast
paratha
fruitea
fruiter

paraphernalia
fruiteria
paraffin
paranoia
parampara
paralyz
fruitloop
paraleg
fruitsnack
paraiso
fruitti
fruitylicoius
frustat
friger
pasadena
frangelico
frappichino
pawsit
frappuchino
frascati
frat
pavlovian
fray
pavilion
pavesini
frazier
pavarotti
frapachino
freaken
paunchi
frederick
patton
fredericksburg
patterson
fredricksburg
patsi
freedent
frappacino
payong
patrick
pdq
peagl
franken
peacework
peacelili
peaceful
frankenstein
peabutt

frankensweeten
pdt
pdk
frapaccino
pdf
pckged
pckg
pckage
pci
pch
franni
fransico
pbjs
freegan
freel
pasar
freshpak
pastir
pastina
frescobaldi
freshdirect
freshman
pasteri
freshmix
pastariso
pastamia
pastachio
freshroast
frere
passiontini
freudian
passionflow
passionberri
frey
fricken
frickin
friel
paskesz
paskag
pastiso
frengl
pato
patern
freeland
freeload
freemont
patial
pathogen

patho
freesom
freestand
freestyl
paterson
freeway
frend
freexer
freher
patchwork
patchouli
patchi
freightcahrg
patassium
fremch
frenchfri
eventuli
eventho
deliciou
dumbbel
dns
dnt
dobe
reshteh
dobey
dobottom
resevoir
docil
docksid
reservatol
dodger
dodo
dodoni
doesnd
doeuvr
doevr
rese
rescur
rescuer
dogfind
dogfoodadvis
reschedul
dnoir
resini
dms
dlanz
restauraunt
restat

divina
restar
respritori
respray
respos
respoond
divvi
responisbl
responc
dmrs
dle
dleiver
respectful
dlim
dlish
dmae
dmd
dmdm
dmenth
resaur
rerun
doggiemunchi
domenica
reportag
dolumbian
repons
repond
replug
domanc
domata
dome
repleat
replay
dominick
dolor
dominion
repetoir
dominiqu
repetet
repet
dominizion
domperidon
don
repalc
repairman
reposit
dolli
rerol

dogsbar
dogi
requuir
requri
reqular
dogladi
doglet
requi
dogma
reqir
dogo
dogsbutt
dollah
dogsv
republican
doh
reptil
repsond
doinki
repris
reprint
reprim
represnt
diversi
diverg
div
disgut
disentegr
rewrit
disfigur
disform
rewash
disfrutar
disfunct
disgest
disguist
revolv
dishwateri
disect
disillusion
disinclin
reviuew
disinfect
revisit
disinigr
revil
revier
disip

dislk
disenfranch
disea
reverber
rhode
discound
riba
discounttommi
rialto
discoverd
discredit
rhwnj
rhodlia
rhodiola
discretionari
discrib
rfri
rhis
discriminatori
rhino
rhineston
discus
rheumat
rheum
rheeboot
rhas
rhan
dislki
disoovl
ditract
distear
rethought
dissappear
dissent
dissert
dissovlv
distain
resutl
resurrect
resurg
resuppli
distemp
diss
resuli
resubscrib
resuabl
distinquish
distort

distr
restrant
distrib
ditalia
ditalini
dissap
retitl
disorgan
disposa
disori
disorient
reveiv
disparag
reva
dispit
reuseabl
displasea
displaysia
reup
reunif
disreput
reult
reuben
disposalhowev
retsin
disposebal
retriv
disproportion
retrevi
retreiv
retrain
donchel
donckel
doniut
refurbish
regetta
reget
regener
regaurdless
regatta
regarldess
drooler
droopi
drope
drudg
drump
drogheria
drumrol

refrigerat
refrigerant
drunkin
refrigar
drv
dryest
refridg
drysdal
drom
drizzli
refreez
regrind
dremel
dresden
reguardless
reguard
reguar
regualr
regualar
regrown
regrow
regrip
regriger
regiman
dressingnow
driest
reglar
registr
drinkin
drinkwel
dripless
drivebi
regimem
refres
refreash
dreg
redvin
ducksmandi
reencourag
reel
duct
reef
reeeeaalli
reece
reeaallli
redwood
redvelvetless
reenlist

redund
reductas
duffi
duhhh
dulci
redros
redonk
dulcosid
duller
dullest
reenforc
ducal
refram
duan
dscale
dsl
refold
refocus
dsvr
reflujo
refluff
dualsport
refl
refirger
dubai
reevalu
refiil
refiger
reffer
referr
referiger
referesh
dubonnet
refere
dubuqu
refelct
dregg
regur
donnabahama
rema
dork
remenb
remememb
rememeb
rememberd
dorota
dortito
doseag

```
doser
douchey
rem
remington
relxar
relunct
doughbal
relooc
doughey
douglass
rellur
in cluster
 1
[328]
piddl
in cluster
 2
[81]
pix
in cluster
 3
[1678]
reserva
in cluster
 4
[311]
phillip
in cluster
 5
[795]
falnurum
in cluster
 6
[406]
pianist
in cluster
 7
[929]
gardenlab
in cluster
 8
[1935]
duffel
in cluster
 9
[1039]
palestinian
in cluster
```

```
 10
[1100]
organiz
in cluster
 11
[369]
flawse
in cluster
 12
[122]
pnut
in cluster
 13
[1890]
drunken
in cluster
 14
[337]
pilari
in cluster
 15
[1149]
overcast
in cluster
 16
[1714]
resond
in cluster
 17
[1035 1525]
palin
frw
in cluster
 18
[1563]
frazzl
in cluster
 19
[1004]
fwrs
in cluster
 20
[1011]
gabbi
in cluster
 21
[351]
piloncillo
```

```
in cluster
 22
[173]
fho
in cluster
 23
[1103]
gibberish
in cluster
 24
[66]
firepit
in cluster
 25
[246]
flinger
in cluster
 26
[921]
overwat
in cluster
 27
[1161]
geier
in cluster
 28
[1002]
fwr
in cluster
 29
[1237]
outmeal
in cluster
 30
[1061]
oscuro
in cluster
 31
[624]
pragma
in cluster
 32
[176]
plaza
in cluster
 33
[886]
ganeden
```

in cluster
34
[765]
ppad
in cluster
35
[524]
prise
in cluster
36
[1662]
fren
in cluster
37
[787]
falernum
in cluster
38
[1590]
frango
in cluster
39
[1603]
pcp
in cluster
40
[1582]
fredrick
in cluster
41
[92]
pistacio
in cluster
42
[428]
prehistor
in cluster
43
[1482]
parmesiano
in cluster
44
[1089]
gettng
in cluster
45
[1470]
parlanc

```
in cluster
 46
[ 976 1922]
panthenol
regimin
in cluster
 47
[1535]
paradox
in cluster
 48
[1261]
fortiflor
in cluster
 49
[1427]
forefath
```

8 Observation

Most of the cluster contains 1-2 words and most of the words in one cluster

```
In [ ]:
```

Ignore the above 2 plots those plots are plotted below again