Assignment 8

Cs432 Web Science Spring 2017

Breon Day

April 14,2017

Question 1:

Part 1:
Create a blog-term matrix.Start by grabbing 100 blogs;include: http://f-measure.blogspot.com/ ,http://ws-dl.blogspot.com/ and grab 98 more as per the method shown in class.

Part 2:
Use the blog title as the identifier for each blog (and row of the matrix). Use the terms from every item/title (RSS) or entry/title (Atom) for the columns of the matrix. The values are the frequency of occurrence.

Part 3:
Limit the number of terms to the most "popular" (i.e., frequent) 1000 terms, this is *after* the criteria on p. 32 (slide 7) has been satisfied.

Answer 1:

Part1

I expanded upon the command line method show in class, curl-I-L'http://www.blogger.com/next-blog?navBar=true\&blogID=3471633091411211117,wrapping it in a for loop and sending the output to the file blogs.txt.

```
bday@sirius:~/cs432$ for((i=1;i<198;i++));do curl -I -L 'http://www.blogger.com/next-blog?navBar=true\
&blogID=34716330914112111117'; done> blogs.txt
```

To account for bad links i increased the initial 98 planned to 198.

The file blogs.py processes the data into uris and rss links through the use of regular expressions and removes duplicates by converting the original list of links into a set of unique links then back into a list creating two files the final product will be the two files uris.txt and rssuris.txt containing the processed links

```
import sys
import re
searchfile = open("blogs.txt", "r")
outFile=open('uris.txt','wb')
outFile2=open('rssuris.txt','wb')
locations=[]
for line in searchfile:
    if "expref=" in line: locations.append(line)
searchfile.close()
uniqueblogs=set(locations)
blogs=list(uniqueblogs)
for blog in blogs:

    link=re.sub('Location: ', '',blog)
    link2=re.sub('\?expref=next-blog', '',link)
    link3 =re.sub('\^M','',link2)
    uri=link3.replace("\r", "").replace("\n", "")
    outFile.write(uri)
    outFile.write('\n')

outFile.write('http://f-measure.blogspot.com/')
outFile.write('\n')
outFile.write('http://ws-dl.blogspot.com/')

for blog in blogs:
    link=re.sub('Location: ','',blog)
    link2 =re.sub('\?expref=next-blog', 'feeds/posts/default?alt=rss',link)
    link3=re.sub('\^M','',link2)
    rss=link3.replace("\r", "").replace("\n", "")
    outFile2.write(rss)
    outFile2.write('\n')

outFile2.write('http://f-measure.blogspot.com/feeds/posts/default?alt=rss')
outFile2.write('\n')
outFile2.write('http://ws-dl.blogspot.com/feeds/posts/default?alt=rss')
```

Part 2 and 3:

The creation of the blog matrix was handled by the class generatefeedvector.py from the PCI book chapter3 github, originally i had errors until another student pointed out the default encoding was the culprit,importing and reloading sys then changing default encoding to utf-8 was all that was required.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import feedparser
import re
import sys

reload(sys)
sys.setdefaultencoding('utf-8')
```

Substituting the original file with my own rssuris.txt allowed me to generate the blog matrixes based on the rss uris i had acquired.

```
feedlist = [line for line in file('rssuris.txt')]
```

Execution of the line python generatefeedvectors.py>blogtitles.txt
produced a portion of the following blog matrix as well as stored their titles
full matrix: https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/Src/Question1/blogdata1.txt
Partial matrix below

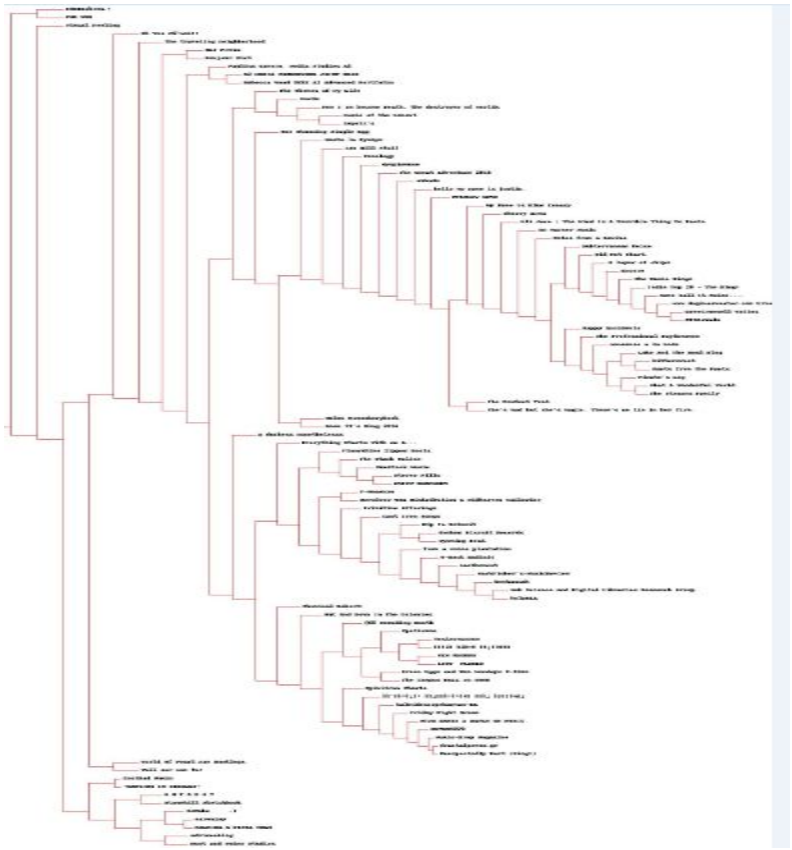
Blog	screaming	kids	golden	catchy	absolute	travel	wrong	fit	songwriter	effects	service	needed
Spotirama	0	2	9	0	0	2	0	3	0	8	3	2	0
U-Rock Radio™	0	1	0	0	1	0	0	1	1	0	3	0	1	3
SEM REGRAS	0	0	4	0	2	1	0	0	0	0	1	0	0	1
Friday Night Dream	0	0	0	0	0	0	1	0	0	0	0	0	0
On Warmer Music	6	7	6	8	1	4	3	2	3	1	2	4	15	1
SEVEN1878	2	2	1	0	2	1	7	7	1	8	0	7	5	7
Spinitron Charts	0	0	0	0	0	0	0	0	0	0	1	0	0
My Name Is Blue Canary	5	10	0	3	4	1	3	3	1	0	0	5	4
Primitive Offerings	1	1	0	0	0	0	0	1	0	0	0	0	0
Web Science and Digital Libraries Research Group	0	2	0	0	1	3	5	2	0
Words	0	0	0	0	0	0	2	0	0	0	0	2	1	0	0
Stereo Pills	0	0	0	7	0	0	0	0	2	0	0	1	0	0
The Stark Online	0	0	1	0	0	0	0	0	1	0	0	0	0	0
Green Eggs and Ham Mondays 8-10am	0	1	3	0	0	0	0	0	0	0	0	0	0
Oh Yes Jónsi!!	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GLI Press	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aubade	0	2	0	1	0	2	3	3	0	1	2	1	2	4	1
from a voice plantation	0	0	0	1	0	1	0	0	0	1	0	0	0
Chemical Robert!	0	1	0	0	0	0	0	0	0	0	1	0	0
A H T A P O T	0	0	0	0	0	0	0	0	0	0	0	0	0	0
holaOLA	0	0	5	2	0	0	0	1	6	2	2	0	0	1	7
World Of Pearl Jam Bootlegs	1	1	0	0	0	0	0	0	0	0	0	0	0
Yestermorrow	0	0	1	0	0	0	0	0	0	0	0	0	3	0
Did Not Chart	0	0	0	0	0	0	0	0	0	1	3	0	2	0
The Great Adventure 2016	0	0	0	0	0	0	2	6	0	0	0	5	2

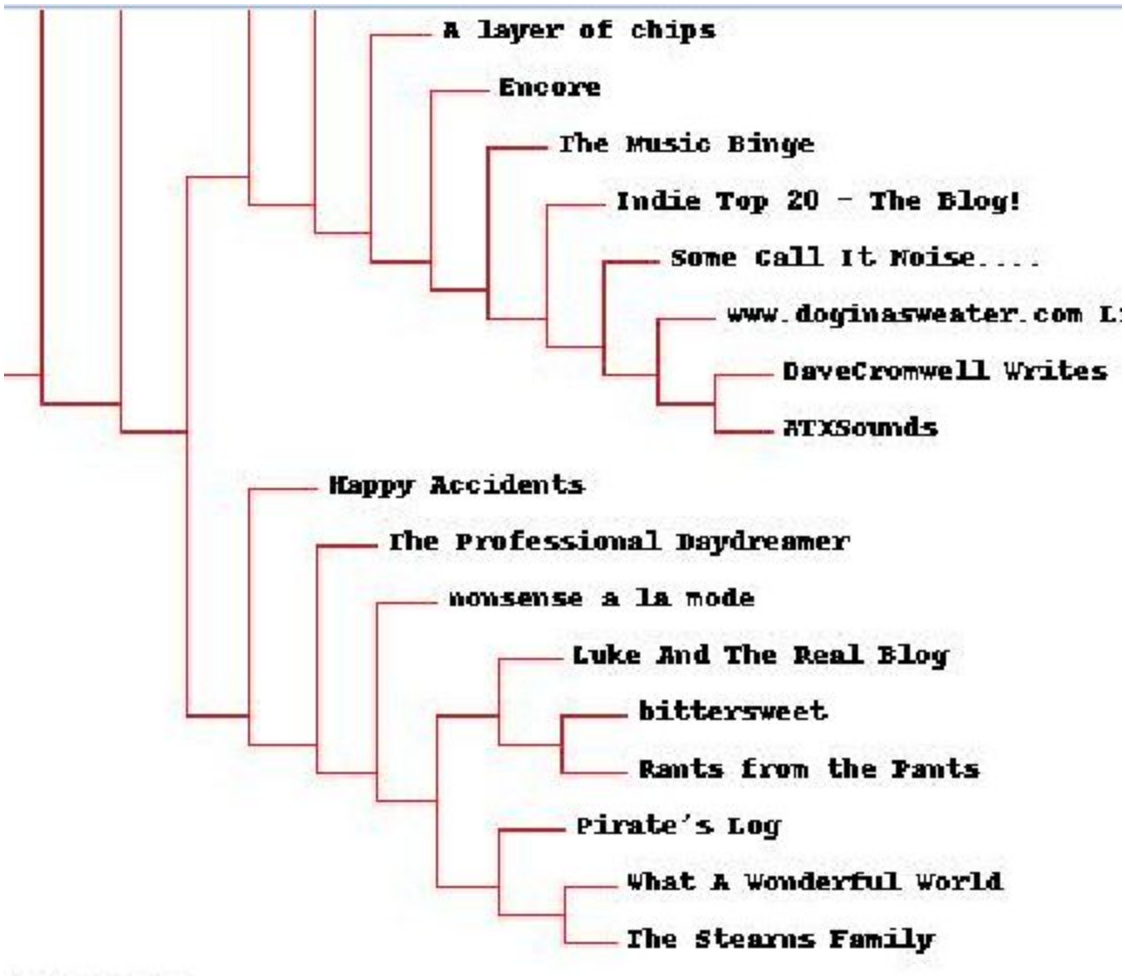
Question 2:

Create an ASCII and JPEG dendrogram that clusters (i.e., HAC) the most similar blogs (see slides 12 & 13).

Answer 2:

First i obtained the prerequisite cluster file from the github repository of the programming collective intelligence chapter 3. Next using the code provided in slides 12 and 13 as well as the previous blog matrix i was able to generate the ascii dendrogram,excluded from this report ,and jpeg dendrogram shown below




A layer of chips
Encore
The Music Binge
Indie Top 20 - The Blog!
Some Call It Noise....
www.doginasweater.com L
DaveCromwell Writes
ATXSounds
Happy Accidents
The Professional Daydreamer
nonsense a la mode
Luke And The Real Blog
bittersweet
Rants from the Pants
Pirate's Log
What A Wonderful World
The Stearns Family

Question 3:

Cluster the blogs using K-Means, using k=5,10,20. (see slide 18). Print the values in each centroid, for each value of k. How many interations were required for each value of k?

Answer 3:

By simply modifying the given code on slide 18 and putting it into a while loop

```
import clusters
centroids=[]
blognames, words, data = clusters.readfile('blogdata1.txt')

kclust=clusters.kcluster(data,k=5)


print ('k=5')
n=0
while(n<5):
    print('[blognames[r] for r in kclust['+str(n)+ ']]')
    s=[blognames[r] for r in kclust[n]]
    n=n+1
    print str(s) + '\n'
```

I was able to achieve the desired output

Full view:

```
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
k=5
[blognames[r] for r in kclust[0]]
['Friday Night Dream', 'SEVEN1878', 'Spinitron Charts', 'GLI Press', 'A H T A P O T', 'kaleidoscopekanvas-KK', 'fractalpress.gr', 'Music-Drop Magazine', 'MT\
JR RANTS & RAVES ON MUSIC', 'Unexpectedly Bart (King!)', '\xce\x94\xce\xaf\xcf\x83\xce\xba\xce\xbf\xce\xb9 \xce\x9c\xce\xbf\xcf\x85\xcf\x83\xce\xb9\xce\xba\\
xce\xae\xcf\x82 \xcf\x83\xcf\x84\xce\xbf \xce\xa7\xcf\x81\xcf\x8c\xce\xbd\xce\xbf', 'Out And Down In The Colonies']

[blognames[r] for r in kclust[1]]
['Spotirama', 'SEM REGRAS', 'Green Eggs and Ham Mondays 8-10am', 'Chemical Robert!', 'Yestermorrow', 'IoTube     :)', '@65 Sounding Booth', 'The Campus Buzz\
 on WSOU', 'LOST  PLACES', 'THE HUB', 'earenjoy', '\xce\x9c\xce\x95\xce\xa3\xce\x91 \xce\xa3\xce\xa4\xce\x97 \xce\x92\xce\xa1\xce\xa9\xce\x9c\xce\x99\xce\x9\
1', 'Vull ser com tu!', 'Ian Hill Stuff']

[blognames[r] for r in kclust[2]]
['U-Rock Radio\xe2\x84\xa2', 'Web Science and Digital Libraries Research Group', 'Oh Yes J\xc3\xb3nsi!!', 'holaOLA', 'World Of Pearl Jam Bootlegs', 'Floorsh\
ime Zipper Boots', 'A2 MEDIA COURSEWORK JOINT BLOG', 'Paulina Gamero. Media Studies A2', 'INDIEohren.!', 'Rebecca Wood 9282 A2 Advanced Portfolio', 'the tra\
veling neighborhood', 'macthemost', 'hmmhannah', "MarkFisher's-MusicReview", 'PALMIRA A PISTA TRES']

[blognames[r] for r in kclust[3]]
['On Warmer Music', 'My Name Is Blue Canary', 'Primitive Offerings', 'Stereo Pills', 'The Stark Online', 'Did Not Chart', 'DaveCromwell Writes', 'Bonjour Gi\
rl', 'GYPSY RHAPSODY', 'Myopiamuse', 'Notes from a Genius', 'www.doginasweater.com Live Show Review Archive', 'Eli Jace | The Mind Is A Terrible Thing To Pa\
ste', 'Subterranean Noise', 'Everything Starts With an A...', 'SunStock Music', 'A layer of chips', 'Some Call It Noise....', 'ATXSounds', 'Dust and Water S\
tudios', 'Visual Feeling', 'Broken Biscuit Records', 'The Music Binge', 'Indie Top 20 - The Blog!', 'F-Measure', 'Hip In Detroit', 'Wyoming Beat', 'Encore',\
 'Revolver USA Distribution & Midheaven mailorder', 'Cast Iron Songs']

[blognames[r] for r in kclust[4]]
['Words', 'aubade', 'from a voice plantation', 'The Great Adventure 2016', 'adrianoblog', 'STANLEY SAYS', 'Stonehill Sketchbook', 'hello my name is justin.'\
, 'a duchess nonethelesss', 'nonsense a la mode', 'Happy Accidents', 'music of the moment', 'Luke And The Real Blog', "Pirate's Log", 'What A Wonderful Worl\
d', 'Hasta la Byebye', 'The Perfect Vent', 'The Themes of My Life', 'Now I am become Death, the destroyer of worlds', 'Helen McCookerybook', 'The Stearns Fa\
mily', 'Sonology', 'CoolDad Music', 'bittersweet', "Room 19's Blog 2016", 'One Stunning Single Egg', 'Rants from the Pants', 'Cherry Area', 'The Professiona\
l Daydreamer', "She's mad but she's magic. There's no lie in her fire.", "isyeli's", '"DANCING IN CIRCLES"']
```

Total iterations were 6,6 and 4 respectively

Question 4:

Use MDS to create a JPEG of the blogs similar to slide 29 of the week 12 lecture. How many iterations were required?

Answer 4:

Modified slide 28 code to produce the jpeg and fed the data into a file mds.py>mds.txt that had the iteration count

jpeg:

Primitive Offerings

Cast Iron Songs

Notes from a Genius

Lul

from a voice plantation

Pirate's Log

Group

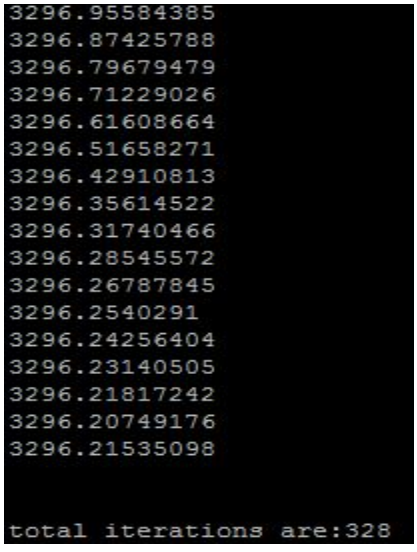
Bonjour Girl

The Great Adventure 2016

aubade

Helen McCookeryBook

Iteration count:328*



* note this count will not represent the one in the github repo as the screen capture from the file was not coming out the way i wanted

Uris

```
http://bartkings.blogspot.com/
http://paradoxical-era.blogspot.com/
http://mondaywakeup.blogspot.com/
http://johnandmaureensanto.blogspot.com/
http://kaleidoscopekanvas-kk.blogspot.com/
http://primitiveofferings.blogspot.com/
http://ngaio1619.blogspot.com/
http://ianthill.blogspot.com/
http://thehubkxci.blogspot.com/
http://globalgoon.blogspot.com/
http://urockradio.blogspot.com/
http://stanleysaystanley.blogspot.com/
http://wyomusic.blogspot.com/
http://mobbie2.blogspot.com/
http://ahtapotunbahcesi.blogspot.com/
http://momentarilymusical.blogspot.com/
http://norecordshopsleft.blogspot.com/
http://glipress.blogspot.com/
http://spotirama.blogspot.com/
http://mts-dailythemes.blogspot.com/
http://outanddowninthecolonies.blogspot.com/
http://somecallitnoise.blogspot.com/
http://lost-places-hamburg.blogspot.com/
http://londynsky.blogspot.com/
http://mccookerybook.blogspot.com/
```

Full links:
https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/rssuris.txt

Question 5:

Re-run question 2, but this time with proper TFIDF calculations instead of the hack discussed on slide 7 (p. 32). Use the same 1000 words, but this time replace their frequency count with TFIDF scores as computed in assignment #3. Document the code, techniques, methods, etc. used to generate these TFIDF values. Upload the new data file to github. Compare and contrast the resulting dendrogram with the dendrogram from question #2.

Answer 5:

The mention of the hack gave me a clue where to begin, starting in generate feed vector i made a copy of the code and renamed it tfidf.py
I then defined the 3 functions below to handle the tfidf calculations

```
def tf(wordcount,totalwords):

    return (float((wordcount))/(totalwords))



def idf(totalDocuments,numDocsWithWord):

    return(math.log(float((totalDocuments))/float(1+numDocsWithWord),2))

def tfidf(wordcount,totalwords,totalDocuments,numDocsWithWord):
    return (float(tf(wordcount,totalwords)*idf(totalDocuments,numDocsWithWord)))
```

Then in the blog matrix generation section i applied the defined functions

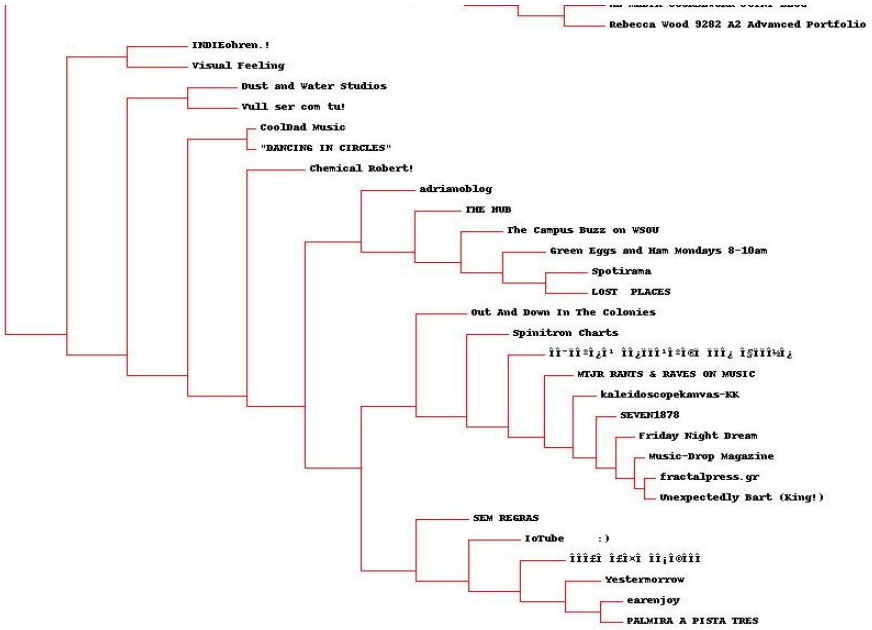
```
tf2=tf(wc[word],totalwords[blog])
idf2=idf(totaldocs,apcount[word])
print(str(blog)+" has "+ str(totalwords[blog])+"checking word"+str(word)+"which has a word count of"+str(wc[word]))

print("tf ="+str(tf2))
print("idf2="+str(idf2))
tfidf2=float(tf2*idf2)
print(tfidf2)

out.write('\t%f' %tfidf( wc[word],totalwords[blog],totaldocs ,apcount[word]))
```

Due to the number of terms, there appearances in the documents ,and number of documents the values were incredibly small i had an issue with getting outwrite to express the fractions
Values increased by 300 below* the normal was just showing up as zeros

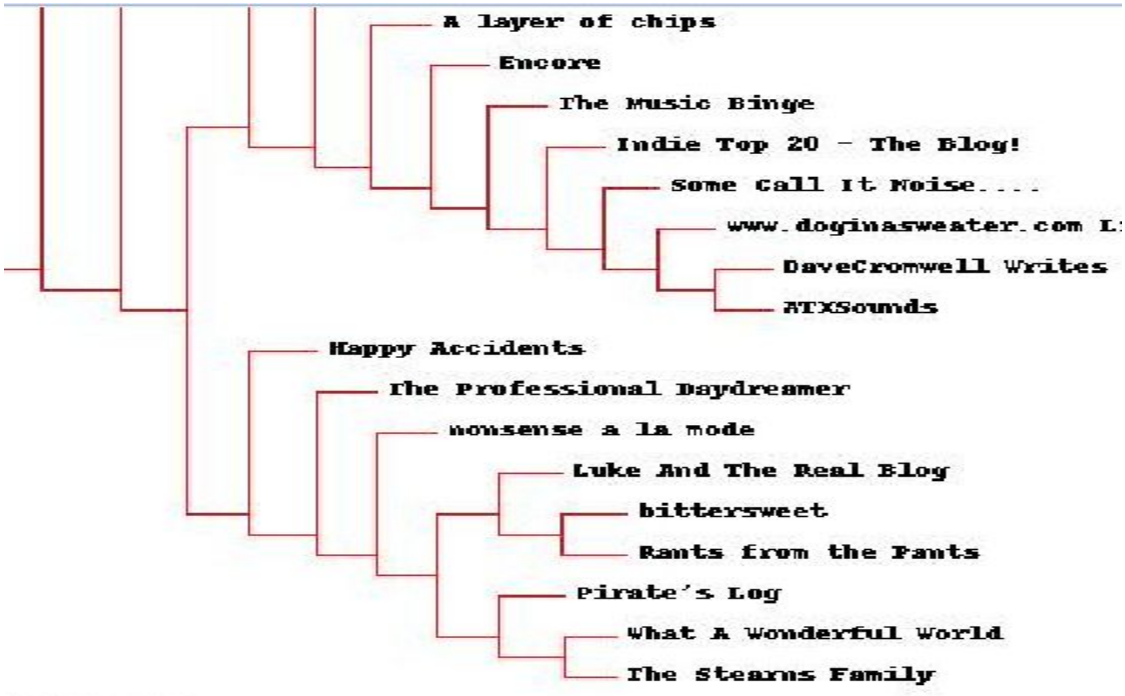
Blog	screaming	kids	golden	catchy	absolute	travel	wrong	fit	s
Spotirama	0	0.000250		0.001959	0	0	0	0.000263	
U-Rock Radio™	0	0.000193		0	0	0.000361	0	0	0
SEM REGRAS	0	0	0.005676		0	0.003037	0.001467		0
Friday Night Dream	0	0	0	0	0	0	0.002663		0
On Warmer Music	0.000647		0.000390		0.000584	0.000708		0.000104	
SEVEN1878	0.000148		0.000077		0.000067	0	0.000143		0
Spinitron Charts	0	0	0	0	0	0	0	0	0
My Name Is Blue Canary	0.000951		0.000983		0	0.000468		0.000734	
Primitive Offerings	0.001552		0.000802		0	0	0	0	0
Web Science and Digital Libraries Research Group					0	0.000089		0	0
Words	0	0	0	0	0	0.001445	0	0	0
Stereo Pills	0	0	0	0.011305	0	0	0	0	0
The Stark Online	0	0	0.002167		0	0	0	0	0
Green Eggs and Ham Mondays 8-10am			0	0.000532		0.002786		0	0
Oh Yes Jónsi!!	0	0	0	0	0	0	0	0	0
GLI Press	0	0	0	0	0	0	0	0	0
aubade	0	0.000274	0	0.000217	0	0.000494		0.000432	
from a voice plantation	0	0	0	0.000385	0	0.000438			0

However the normal still produced a dendrogram so i classified simply as a bug and carried on
Full view: tfidf
https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/Src/Question5/tfidfblogclust.jpg
Full view:original
https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/Src/Question2/blogclust.jpg

tfidf



Original



Comparison notes:
The tfidf being a more accurate means of grouping in comparison to the hack produced much more defined subgroups in being in stark contrast to the originals general over arching staircase of semi-similar blogs

Question 6:

Re-run questions 1-4, but this time instead of using the 98 "random" blogs, use 98 blogs that should be "similar" to:
http://f-measure.blogspot.com/
http://ws-dl.blogspot.com/

Answer 6:

Believing you would want this done programmatically i started by the creation of the similiarblogs.py program
Using requests i grabbed the blogs and made a soup variable with BeautifulSoup so that i could grab the necessary sections of the html

```
r= requests.get('http://www.blogger.com/next-blog?navBar=true&blogID=3471633091411211117')
soup=BeautifulSoup(r.text)
```

Using the soup variable i grabbed the blogs titles.

```
print soup.title.string
title2=soup.title.string
title=str(title2)
```

I limited my search to the title,subtitles, and if the blog had one the description If the document contained keywords i would increase a counter for the type of blog depending on the keyword.

```
for csKeyword in csKeywords:
   if csKeyword in str.lower(title):
      csCounter=+1

for musicKeyword in musicKeywords:
   if musicKeyword in str.lower(title):
      musicCounter+=1
  # search for keywords in subtitles
for subTitle in soup(parseOnlyThese=SoupStrainer('title')):
   for csKeyword in csKeywords:
      if csKeyWord in str.lower(subTitle):
         csCounter=+1
   for musicKeyword in musicKeywords:
      if musicKeyword in str.lower(subTitle):
         musicCounter+=1
  #if they have a description  search for keywords here as well
try:
   metadescription=soup.find('meta',property="og:description")
   desc=metadescription['content']
   description=str(desc)
   print(description)
   for csKeyword in csKeywords:
      if csKeyword in str.lower(description):
         csCounter+=1
   for musicKeyword in musicKeywords:
      if musicKeyword in str.lower(description):
         musicCounter+=1
except:
   continue
```

Then depending on the number of keywords it would listed and written to file as either a music or computer science blog, skipping if tied.

```
if ((musicCounter or csCounter)>0):
    global computerscienceurls,musicblogrssurls
    if ((csCounter>musicCounter) and (len(computerscienceurls)<39)):
        global rss
        print("added csblog")
        musicCounter=0
        csCounter=0
        computerscienceurls.append(rss)
        outFile2.write(rss)

    elif (musicCounter==csCounter):
        print("tied so skipping")
        musicCounter=0
        csCounter=0
    elif ((musicCounter>csCounter) and (len(musicblogrssurls)<59)):
        global rss
        print("added musicblog")
        musicCounter=0
        csCounter=0
        musicblogrssurls.append(rss)
        outFile2.write(rss)

    elif():
        print("doing nothing")
```

Keywords and variables below

```
outFile=open('similiaruris.txt','wb')
outFile2=open('similiarrss.txt','wb')
rss=""
musicblogrssurls= []
computerscienceurls= []
musicKeywords=["music","techno","rock","rap","dj","song","radio","dancing","dance","pop"]
csKeywords= ["data","science","research","python","java","tech","technology","web","tech","code","coding","c++","program","programming","university","teach"\
,"teaching"]

musicCounter=0
csCounter=0
```

The entire code was bound in a while loop and would run until the number of blogs needed were acquired

```
while((   (len(musicblogrssurls)<59) or (len(computerscienceurls)<39)   )):
```

When removing similiar blogs i noticed that there were a lot of repeats bringing my total down from 100 to roughly 60. Which is a enough to make a note of,
I then ran the rss urls generated and compared them to the output of the original questions 1-4

Original matrix:

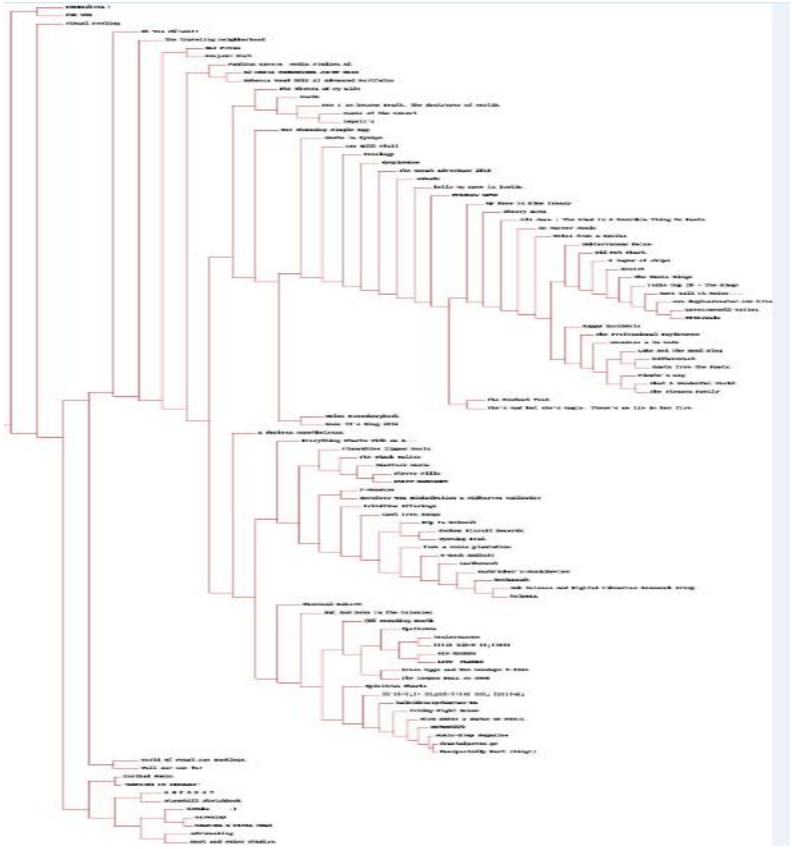
Blog	screaming	kids	golden	catchy	absolute		travel	wrong	fit	songwriter		effects	service	needed
Spotirama	0	2	9	0	0	0	2	0	3	0	8	3	2	0
U-Rock Radio™	0	1	0	0	1	0	0	1	1	0	3	0	1	3
SEM REGRAS	0	0	4	0	2	1	0	0	0	0	1	0	0	1
Friday Night Dream	0	0	0	0	0	0	0	1	0	0	0	0	0	0
On Warmer Music	6	7	6	8	1	4	3	2	3	1	2	4	15	1
SEVEN1878	2	2	1	0	2	1	7	7	1	8	0	7	5	7
Spinitron Charts	0	0	0	0	0	0	0	0	0	0	0	1	0	0
My Name Is Blue Canary	5	10	0	3	4	1	3	3	1	0	0	5	4	
Primitive Offerings	1	1	0	0	0	0	0	1	0	0	0	0	0	
Web Science and Digital Libraries Research Group						0	2	0	0	1	3	5	2	0
Words	0	0	0	0	0	0	2	0	0	0	0	2	1	0	0
Stereo Pills	0	0	0	7	0	0	0	0	2	0	0	1	0	0
The Stark Online		0	0	1	0	0	0	0	0	1	0	0	0	0
Green Eggs and Ham Mondays 8-10am			0	1	3	0	0	0	0	0	0	0	0	0
Oh Yes Jónsi!!	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GLI Press	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aubade	0	2	0	1	0	2	3	3	0	1	2	1	2	4	1
from a voice plantation	0	0	0	1	0	1	0	0	0	1	0	0	0	
Chemical Robert!	0	1	0	0	0	0	0	0	0	0	1	0	0	
A H T A P O T	0	0	0	0	0	0	0	0	0	0	0	0	0	0
holaOLA	0	0	5	2	0	0	0	1	6	2	2	0	0	1	7
World Of Pearl Jam Bootlegs		1	1	0	0	0	0	0	0	0	0	0	0	0
Yestermorrow	0	0	1	0	0	0	0	0	0	0	0	0	3	0
Did Not Chart	0	0	0	0	0	0	0	0	0	1	3	0	2	0
The Great Adventure 2016		0	0	0	0	0	0	2	6	0	0	0	5	2

Similar matrix:

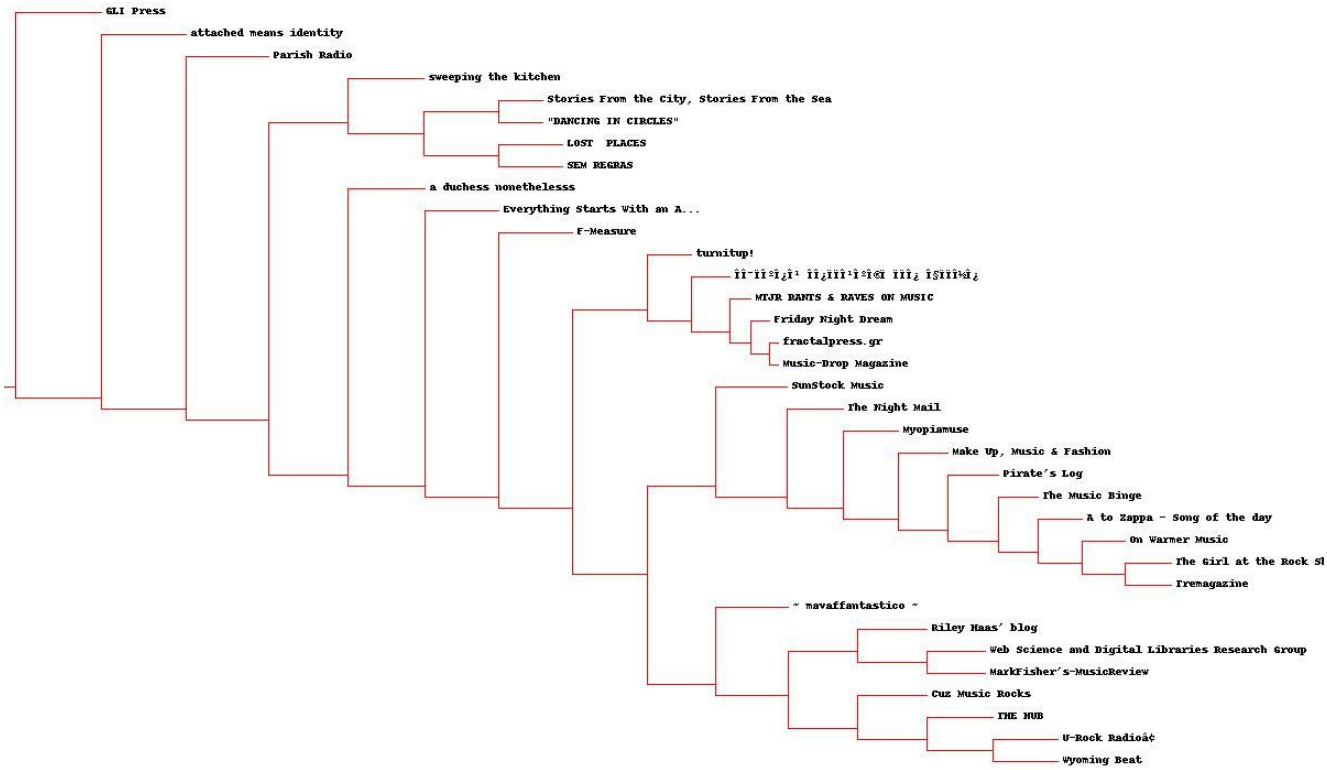
Blog	woods	opener	kids	golden	catchy	absolute		travel	wrong	fit	screaming		fix	songwriter		effects	ser
LOST PLACES	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	
MTJR RANTS & RAVES ON MUSIC	3	9	5	10	7	1	3	5	2	5	0	22	3	1			
sweeping the kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Cuz Music Rocks	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	
U-Rock Radio™	0	0	1	0	0	1	0	0	1	0	0	1	0	3	0	0	
SEM REGRAS	0	0	0	4	0	2	1	0	0	0	0	0	0	1	0	0	
Friday Night Dream	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
On Warmer Music	2	0	7	6	8	1	4	3	2	6	1	3	1	2	4	0	
attached means identity	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
The Music Binge	0	2	1	0	1	0	1	1	2	0	0	2	1	0	0	0	
A to Zappa - Song of the day		0	0	0	0	0	2	4	1	1	2	1	0	0	1	0	
a duchess nonethelesss	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
THE HUB	0	0	2	1	0	0	0	0	0	0	0	0	1	0	0	1	
The Girl at the Rock Show		0	1	1	0	2	0	0	1	1	0	0	0	0	0	0	
Web Science and Digital Libraries Research Group						0	0	2	0	0	1	3	6	1	0	1	
SunStock Music	0	0	0	0	1	0	0	0	0	0	0	1	2	0	0	0	
Parish Radio	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
The Night Mail	0	3	1	2	0	30	4	4	13	4	4	8	20	4	5	14	

In certain blogs the similar words were much higher

Original dendro:



Similar dendro:



In comparison to basic structure i was not able to make any major notes on the difference though it could be a fault of the reduced sample size

Original Ascii:
https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/Src/Question2/asciidendogram

Similar Ascii:
https://github.com/BreonDay/cs532-s17/blob/master/Submissions/A8/Src/Question6/simasciidendogram

Original kmeans:

```
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
k=5
[blognames[r] for r in kclust[0]]
['Friday Night Dream', 'SEVEN1878', 'Spinitron Charts', 'GLI Press', 'A H T A P O T', 'kaleidoscopekanvas-KK', 'fractalpress.gr', 'Music-Drop Magazine', 'MT\
JR RANTS & RAVES ON MUSIC', 'Unexpectedly Bart (King!)', '\xce\x94\xce\xaf\xcf\x83\xce\xba\xce\xbf\xce\xb9 \xce\x9c\xce\xbf\xcf\x85\xcf\x83\xce\xb9\xce\xba\\
xce\xae\xcf\x82 \xcf\x83\xcf\x84\xce\xbf \xce\xa7\xcf\x81\xcf\x8c\xce\xbd\xce\xbf', 'Out And Down In The Colonies']

[blognames[r] for r in kclust[1]]
['Spotirama', 'SEM REGRAS', 'Green Eggs and Ham Mondays 8-10am', 'Chemical Robert!', 'Yestermorrow', 'IoTube     :)', '@65 Sounding Booth', 'The Campus Buzz\
 on WSOU', 'LOST  PLACES', 'THE HUB', 'earenjoy', '\xce\x9c\xce\x95\xce\xa3\xce\x91 \xce\xa3\xce\xa4\xce\x97 \xce\x92\xce\xa1\xce\xa9\xce\x9c\xce\x99\xce\x9\
1', 'Vull ser com tu!', 'Ian Hill Stuff']

[blognames[r] for r in kclust[2]]
['U-Rock Radio\xe2\x84\xa2', 'Web Science and Digital Libraries Research Group', 'Oh Yes J\xc3\xb3nsi!!', 'holaOLA', 'World Of Pearl Jam Bootlegs', 'Floorsh\
ime Zipper Boots', 'A2 MEDIA COURSEWORK JOINT BLOG', 'Paulina Gamero. Media Studies A2', 'INDIEohren.!', 'Rebecca Wood 9282 A2 Advanced Portfolio', 'the tra\
veling neighborhood', 'macthemost', 'hmmhannah', "MarkFisher's-MusicReview", 'PALMIRA A PISTA TRES']

[blognames[r] for r in kclust[3]]
['On Warmer Music', 'My Name Is Blue Canary', 'Primitive Offerings', 'Stereo Pills', 'The Stark Online', 'Did Not Chart', 'DaveCromwell Writes', 'Bonjour Gi\
rl', 'GYPSY RHAPSODY', 'Myopiamuse', 'Notes from a Genius', 'www.doginasweater.com Live Show Review Archive', 'Eli Jace | The Mind Is A Terrible Thing To Pa\
ste', 'Subterranean Noise', 'Everything Starts With an A...', 'SunStock Music', 'A layer of chips', 'Some Call It Noise....', 'ATXSounds', 'Dust and Water S\
tudios', 'Visual Feeling', 'Broken Biscuit Records', 'The Music Binge', 'Indie Top 20 - The Blog!', 'F-Measure', 'Hip In Detroit', 'Wyoming Beat', 'Encore',\
 'Revolver USA Distribution & Midheaven mailorder', 'Cast Iron Songs']

[blognames[r] for r in kclust[4]]
['Words', 'aubade', 'from a voice plantation', 'The Great Adventure 2016', 'adrianoblog', 'STANLEY SAYS', 'Stonehill Sketchbook', 'hello my name is justin.'\
, 'a duchess nonethelesss', 'nonsense a la mode', 'Happy Accidents', 'music of the moment', 'Luke And The Real Blog', "Pirate's Log", 'What A Wonderful Worl\
d', 'Hasta la Byebye', 'The Perfect Vent', 'The Themes of My Life', 'Now I am become Death, the destroyer of worlds', 'Helen McCookerybook', 'The Stearns Fa\
mily', 'Sonology', 'CoolDad Music', 'bittersweet', "Room 19's Blog 2016", 'One Stunning Single Egg', 'Rants from the Pants', 'Cherry Area', 'The Professiona\
l Daydreamer', "She's mad but she's magic. There's no lie in her fire.", "isyeli's", '"DANCING IN CIRCLES"']
```

Similiar kmeans:

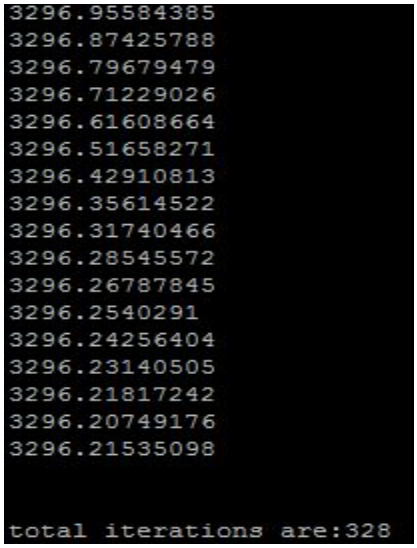
```
Iteration 0
Iteration 1
Iteration 2
Iteration 3
k=5
[blognames[r] for r in kclust[0]]
[]

[blognames[r] for r in kclust[1]]
['Cuz Music Rocks', 'U-Rock Radio\xe2\x84\xa2', 'On Warmer Music', 'The Music Bi
nge', 'A to Zappa - Song of the day', 'a duchess nonethelesss', 'THE HUB', 'The
Girl at the Rock Show', 'SunStock Music', 'F-Measure', "Riley Haas' blog", 'GLI
Press', 'Wyoming Beat', 'Make Up, Music & Fashion', '~ mavaffantastico ~', 'Trem
agazine', 'Everything Starts With an A...', 'Myopiamuse', "Pirate's Log"]

[blognames[r] for r in kclust[2]]
['MTJR RANTS & RAVES ON MUSIC', 'Friday Night Dream', 'attached means identity',
 'Web Science and Digital Libraries Research Group', '\xce\x94\xce\xaf\xcf\x83\x
ce\xba\xce\xbf\xce\xb9 \xce\x9c\xce\xbf\xcf\x85\xcf\x83\xce\xb9\xce\xba\xce\xae\
xcf\x82 \xcf\x83\xcf\x84\xce\xbf \xce\xa7\xcf\x81\xcf\x8c\xce\xbd\xce\xbf', 'tur
nitup!', 'fractalpress.gr', 'Music-Drop Magazine', "MarkFisher's-MusicReview"]

[blognames[r] for r in kclust[3]]
['The Night Mail']

[blognames[r] for r in kclust[4]]
['LOST  PLACES', 'sweeping the kitchen', 'SEM REGRAS', 'Parish Radio', 'Stories
From the City, Stories From the Sea', '"DANCING IN CIRCLES"']
```

Done in 4,4 and 4 respectively

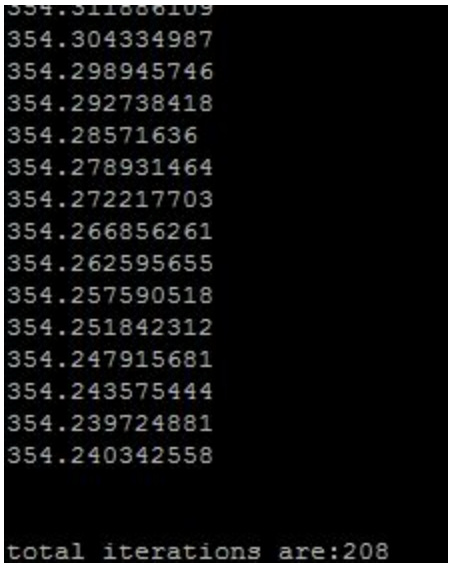
As would be expected the similiar blog had much tighter clusters all throughout

For mds section of the comparison i decided to focus on the total iterations instead of the jpgs

Original mds:



Similar mds:



Total iterations and all numbers in general were much smaller in the similar section with the iterations being exactly 120 less and the list of numbers being almost one tenth the size of the original.