Web Science Assignment 7 Report

Dr. Nelson

James Pindell

3/27/2018

Content

Problem 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. Note that this method randomly chooses blogs and each student will separately do this process, so it is unlikely that these 98 blogs will be shared among students. In other words, no sharing of blog data. Upload to github your code for grabbing the blogs and provide a list of blog URIs, both in the report and in github.

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. Essentially you are replicating the format of the "blogdata.txt" file included with the PCI book code. Limit the number of terms to the most "popular" (i.e., frequent) 1000 terms, this is *after* the criteria on p. 32 (slide 8) has been satisfied. Remember that blogs are paginated.

Solution

The solution for this problem is outlined by the following steps:

1. **Grabbing Blog URLs:** In order to grab the remaining 98 blogs needed, the website:

   http://f-measure.blogspot.com/

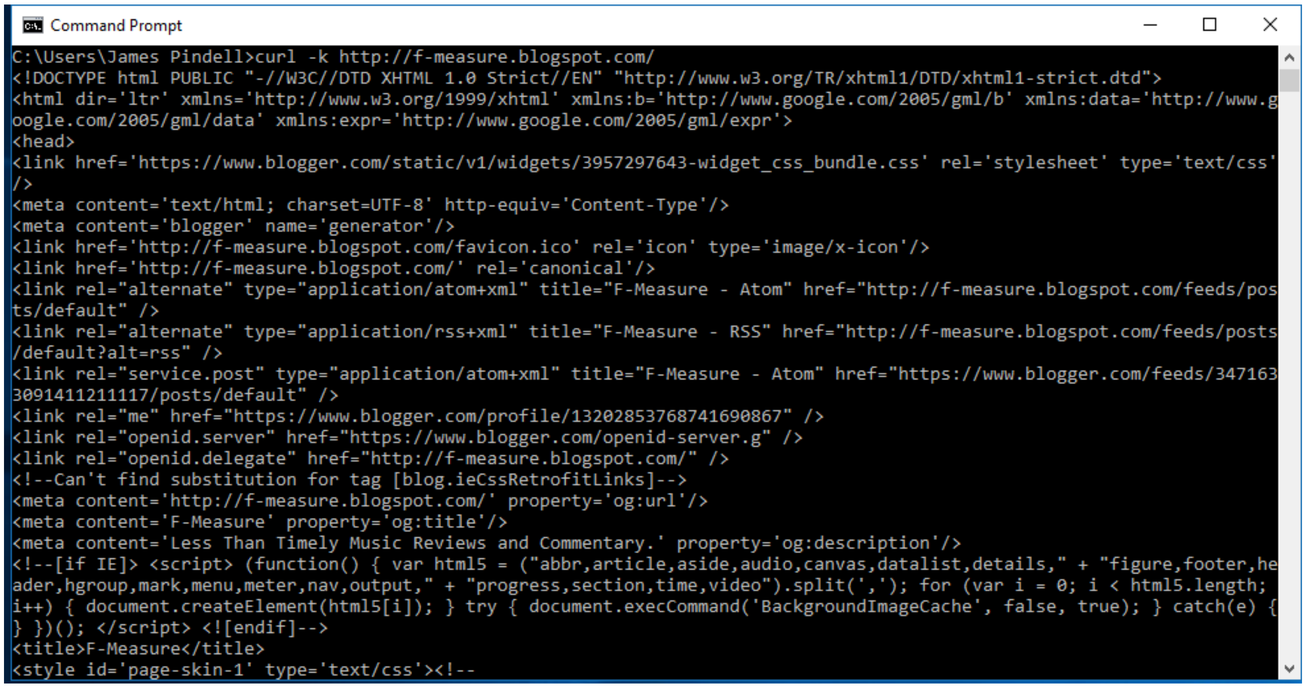
   was accessed, in which the blog has a “Next Blog” button located at the top of the webpage. By clicking that button, a new blog appears on screen. Once the new blog appears, the URL of the blog was copied and pasted into a temporary text file. This process was repeated until there was a total of 100 blogs in the text file.

   Note: This method of grabbing blogs was used over using the “curl” command in the command terminal because “curl” could not recognize the webpages’ SSL certificates, which, unfortunately, would have made the process much more difficult than need be.

2. **Grabbing Blog RSS Feeds**: Once all 100 blog URL’s have been obtained, the “curl” command:

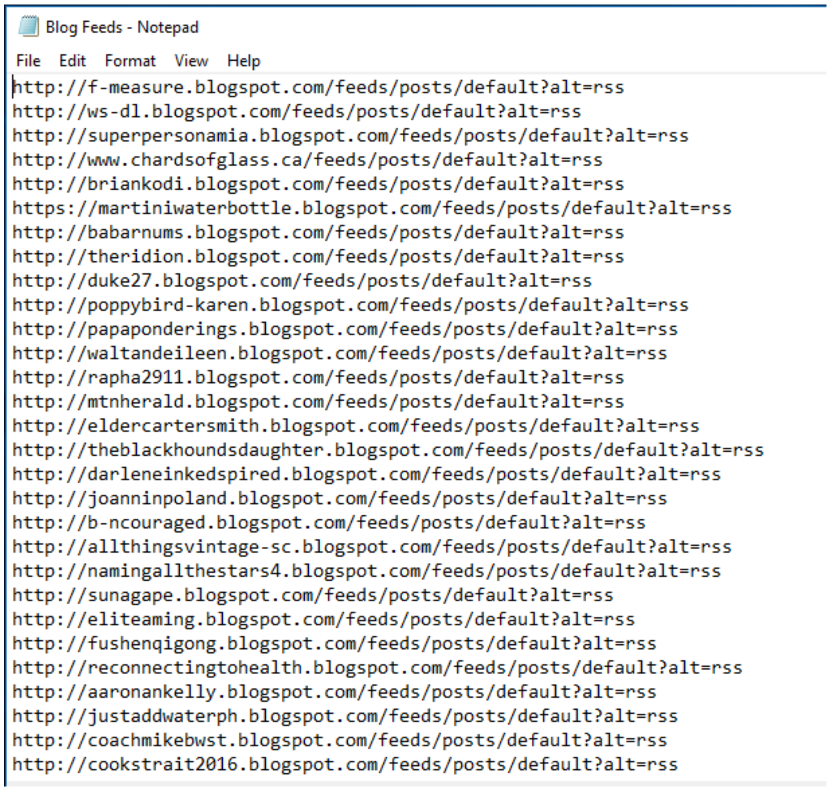
   ```
   curl –k {blog url}
   ```

was entered into the command terminal to access the blog’s RSS feed. An example of this is shown below:


Command Prompt

```
C:\Users\James Pindell>curl -k http://f-measure.blogspot.com/
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html dir='ltr' xmlns='http://www.w3.org/1999/xhtml' xmlns:b='http://www.google.com/2005/gml/b' xmlns:data='http://www.g
oogle.com/2005/gml/data' xmlns:expr='http://www.google.com/2005/gml/expr'>
<head>
<link href='https://www.blogger.com/static/v1/widgets/3957297643-widget_css_bundle.css' rel='stylesheet' type='text/css'
/>
<meta content='text/html; charset=UTF-8' http-equiv='Content-Type'/>
<meta content='blogger' name='generator'/>
<link href='http://f-measure.blogspot.com/favicon.ico' rel='icon' type='image/x-icon'/>
<link href='http://f-measure.blogspot.com/' rel='canonical'/>
<link rel="alternate" type="application/atom+xml" title="F-Measure - Atom" href="http://f-measure.blogspot.com/feeds/pos
ts/default" />
<link rel="alternate" type="application/rss+xml" title="F-Measure - RSS" href="http://f-measure.blogspot.com/feeds/posts
/default?alt=rss" />
<link rel="service.post" type="application/atom+xml" title="F-Measure - Atom" href="https://www.blogger.com/feeds/347163
3091411211117/posts/default" />
<link rel="me" href="https://www.blogger.com/profile/13202853768741690867" />
<link rel="openid.server" href="https://www.blogger.com/openid-server.g" />
<link rel="openid.delegate" href="http://f-measure.blogspot.com/" />
<!--Can't find substitution for tag [blog.ieCssRetrofitLinks]-->
<meta content='http://f-measure.blogspot.com/' property='og:url'/>
<meta content='F-Measure' property='og:title'/>
<meta content='Less Than Timely Music Reviews and Commentary.' property='og:description'/>
<!--[if IE]> <script> (function() { var html5 = ("abbr,article,aside,audio,canvas,datalist,details," + "figure,footer,he
ader,hgroup,mark,menu,meter,nav,output," + "progress,section,time,video").split(','); for (var i = 0; i < html5.length;
i++) { document.createElement(html5[i]); } try { document.execCommand('BackgroundImageCache', false, true); } catch(e) {
} })(); </script> <![endif]-->
<title>F-Measure</title>
<style id='page-skin-1' type='text/css'><!--
```

Once the link to the blog’s RSS feed has been found, the link was copied and pasted into a text file. You can find this text file, called Blog Feeds.txt, from within the package, but a snippet of the text file is shown below:


Blog Feeds - Notepad

File Edit Format View Help

```
http://f-measure.blogspot.com/feeds/posts/default?alt=rss
http://ws-dl.blogspot.com/feeds/posts/default?alt=rss
http://superpersonamia.blogspot.com/feeds/posts/default?alt=rss
http://www.chardsofglass.ca/feeds/posts/default?alt=rss
http://briankodi.blogspot.com/feeds/posts/default?alt=rss
https://martiniwaterbottle.blogspot.com/feeds/posts/default?alt=rss
http://babarnums.blogspot.com/feeds/posts/default?alt=rss
http://theridion.blogspot.com/feeds/posts/default?alt=rss
http://duke27.blogspot.com/feeds/posts/default?alt=rss
http://poppybird-karen.blogspot.com/feeds/posts/default?alt=rss
http://papaponderings.blogspot.com/feeds/posts/default?alt=rss
http://waltandeileen.blogspot.com/feeds/posts/default?alt=rss
http://rapha2911.blogspot.com/feeds/posts/default?alt=rss
http://mtnherald.blogspot.com/feeds/posts/default?alt=rss
http://eldercartersmith.blogspot.com/feeds/posts/default?alt=rss
http://theblackhoundsdaughter.blogspot.com/feeds/posts/default?alt=rss
http://darleneinkedspired.blogspot.com/feeds/posts/default?alt=rss
http://joanninpoland.blogspot.com/feeds/posts/default?alt=rss
http://b-ncouraged.blogspot.com/feeds/posts/default?alt=rss
http://allthingsvintage-sc.blogspot.com/feeds/posts/default?alt=rss
http://namingallthestars4.blogspot.com/feeds/posts/default?alt=rss
http://sunagape.blogspot.com/feeds/posts/default?alt=rss
http://eliteaming.blogspot.com/feeds/posts/default?alt=rss
http://fushenqigong.blogspot.com/feeds/posts/default?alt=rss
http://reconnectingtohealth.blogspot.com/feeds/posts/default?alt=rss
http://aaronankelly.blogspot.com/feeds/posts/default?alt=rss
http://justaddwaterph.blogspot.com/feeds/posts/default?alt=rss
http://coachmikebwst.blogspot.com/feeds/posts/default?alt=rss
http://cookstrait2016.blogspot.com/feeds/posts/default?alt=rss
```

3. **Create Blog-Term Matrix**: Once all 100 blog feeds had been obtained, the code provided from the link:

   https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter3/generatefeedvector.py

   was slightly modified in order to create the blog-term matrix. Unfortunately, both the code used and the blog-term matrix are too large to show as snippets, but you can find both the code used and the blog-term matrix, called BlogParser.py and Blog Data.txt respectively, from within the package.

Problem 2

Create an ASCII and JPEG dendrogram that clusters (i.e., HAC) the most similar blogs (see slides 13 & 14). Include the JPEG in your report and upload the ascii file to github (it will be too unwieldy for inclusion in the report).

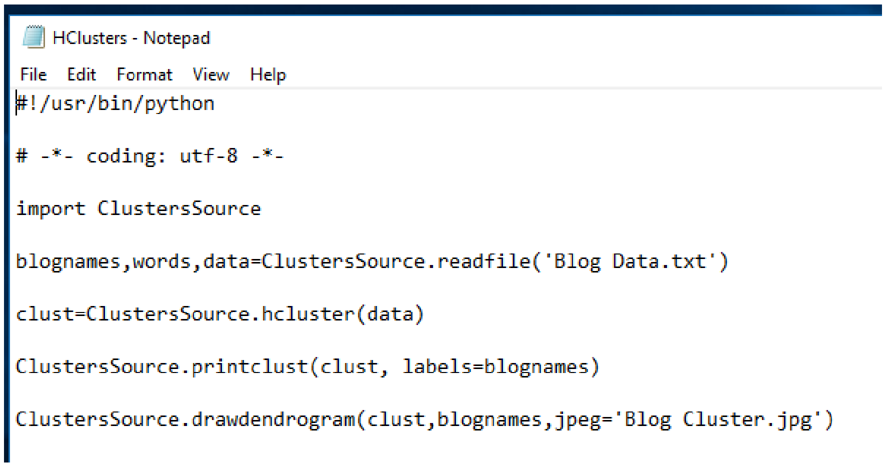
Solution

The solution for this problem is outlined by the following steps:

1. **Create ASCII Blog Cluster Dendrogram**: In order to create the ACSII blog cluster dendogram, the code provided from both the Week 11 Slides, Slide 13 and the link:

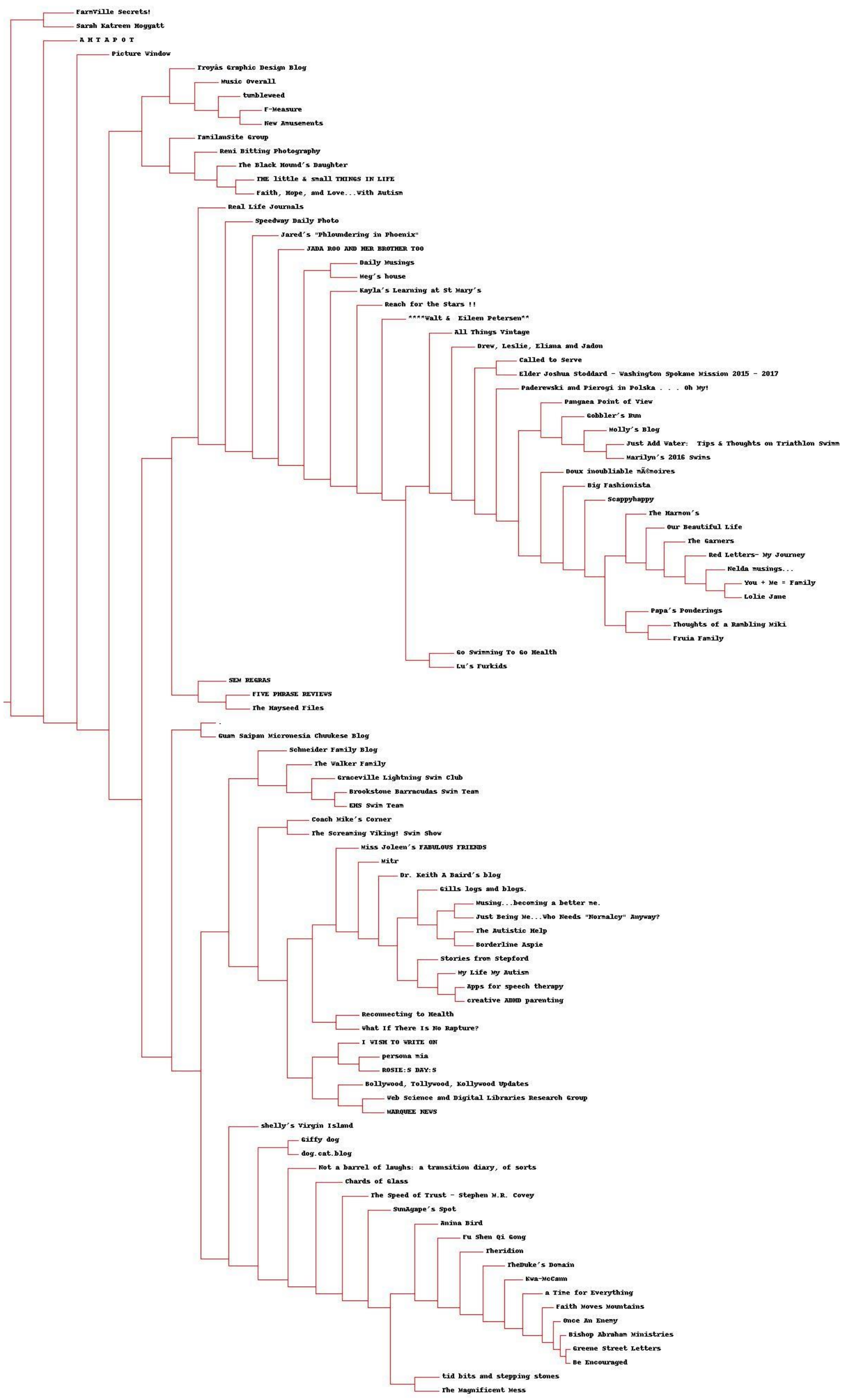
   https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter3/clusters.py

   was slightly modified and run. Unfortunately, the code from the link is too large to show as a snippet, but you can find the code in a file, called ClustersSource.py, from within the package. You can also find the other code in a file, called HClusters.py, from within the package, but a snippet of the code is shown below:



```
#!/usr/bin/python

# -*- coding: utf-8 -*-

import ClustersSource

blognames,words,data=ClustersSource.readfile('Blog Data.txt')

clust=ClustersSource.hcluster(data)

ClustersSource.printclust(clust, labels=blognames)

ClustersSource.drawdendrogram(clust,blognames,jpeg='Blog Cluster.jpg')
```

2. **Create JPEG Blog Cluster Dendrogram**: In addition to the code above creating an ASCII blog cluster dendrogram, the code also creates a JPEG blog cluster dendrogram. The code isn't shown here, since it was shown in the previous step, but you can find both the code and the JPEG, called HClusters.py and Blog Cluster.jpeg respectively, from within the package, and the JPEG is also shown below:



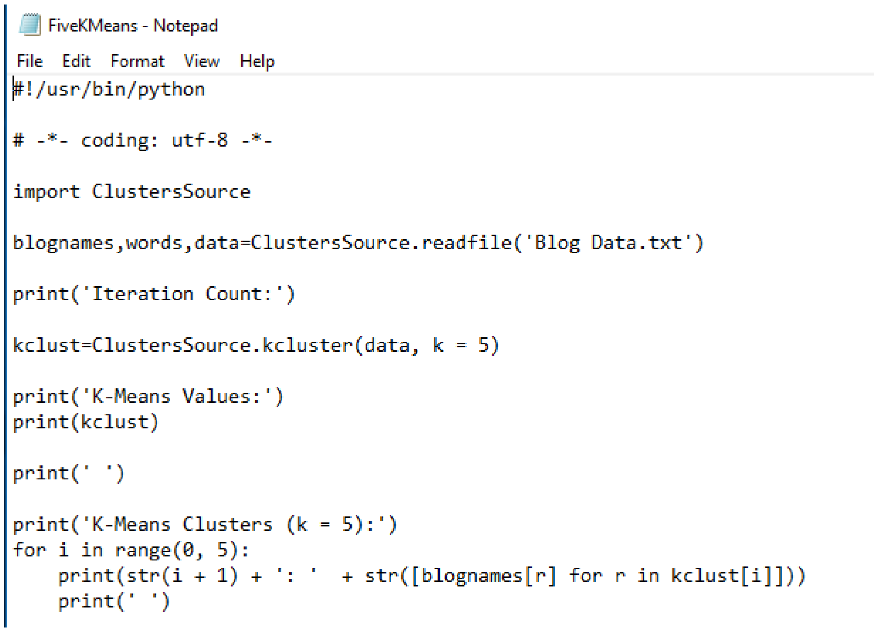
Problem 3

Cluster the blogs using K-Means, using k=5,10,20. (see slide 25). Print the values in each centroid, for each value of k. How many iterations were required for each value of k?

Solution

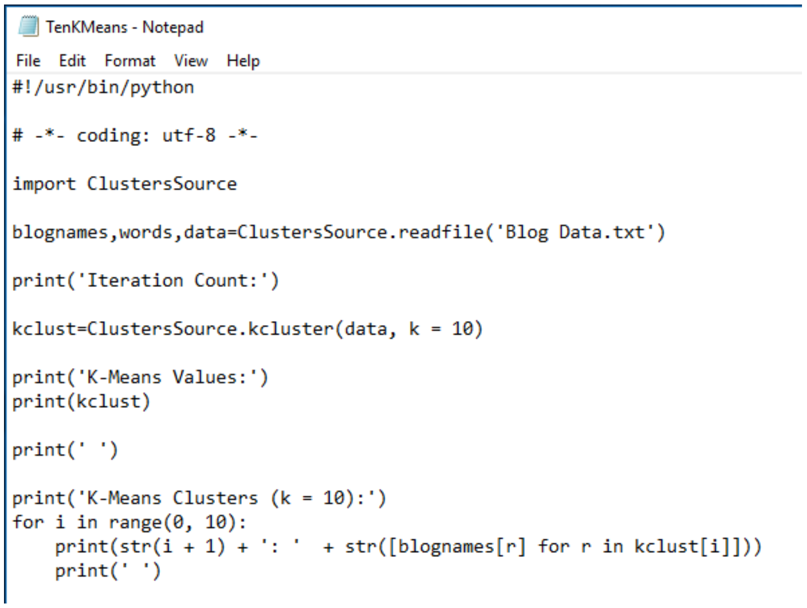
The solution for this problem is outlined by the following steps:

1. **Cluster Blogs Using K-Means of 5:** In order to cluster the blogs using K-Means of 5, the code provided from the Week 11 Slides, Slide 19 was used. The number of iterations required for each value of k was 5. You can find the code in a file, called FiveKMeans.py, from within the package, but a snippet of the code is shown below:



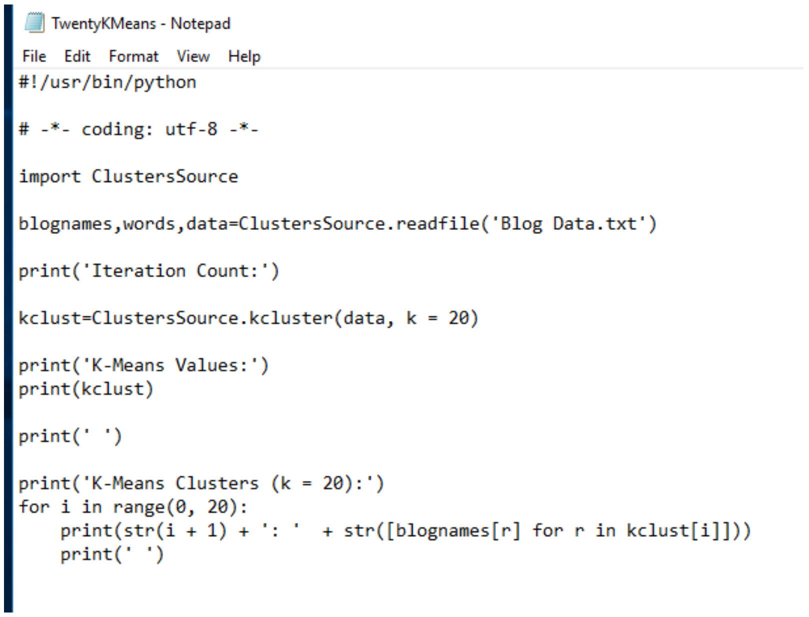
```
#!/usr/bin/python

# -*- coding: utf-8 -*-

import ClustersSource

blognames,words,data=ClustersSource.readfile('Blog Data.txt')

print('Iteration Count:')

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

print('K-Means Values:')
print(kclust)

print(' ')

print('K-Means Clusters (k = 5):')
for i in range(0, 5):
    print(str(i + 1) + ': '  + str([blognames[r] for r in kclust[i]]))
    print(' ')
```

2. **Cluster Blogs Using K-Means of 10**: In order to cluster the blogs using K-Means of 10, the code provided from the Week 11 Slides, Slide 19 was used. The number of iterations required for each value of k was 6. You can find the code in a file, called TenKMeans.py, from within the package, but a snippet of the code is shown below:



```
#!/usr/bin/python

# -*- coding: utf-8 -*-

import ClustersSource

blognames,words,data=ClustersSource.readfile('Blog Data.txt')

print('Iteration Count:')

kclust=ClustersSource.kcluster(data, k = 10)

print('K-Means Values:')
print(kclust)

print(' ')

print('K-Means Clusters (k = 10):')
for i in range(0, 10):
    print(str(i + 1) + ': '  + str([blognames[r] for r in kclust[i]]))
    print(' ')
```

3. **Cluster Blogs Using K-Means of 20:** In order to cluster the blogs using K-Means of 20, the code provided from the Week 11 Slides, Slide 19 was slightly modified and run. The number of iterations required for each value of k was 7. You can find the code in a file, called TwentyKMeans.py, from within the package, but a snippet of the code is shown below:



```
#!/usr/bin/python

# -*- coding: utf-8 -*-

import ClustersSource

blognames,words,data=ClustersSource.readfile('Blog Data.txt')

print('Iteration Count:')

kclust=ClustersSource.kcluster(data, k = 20)

print('K-Means Values:')
print(kclust)

print(' ')

print('K-Means Clusters (k = 20):')
for i in range(0, 20):
    print(str(i + 1) + ': '  + str([blognames[r] for r in kclust[i]]))
    print(' ')
```

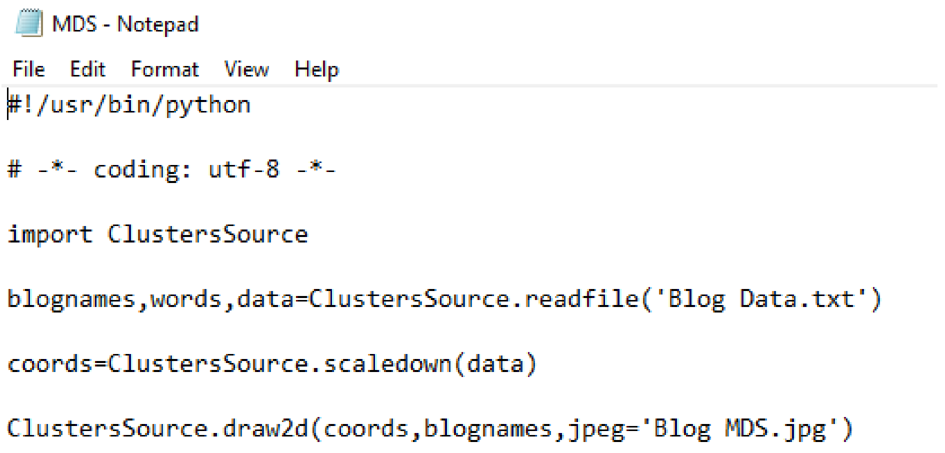
Problem 4

Use MDS to create a JPEG of the blogs similar to slide 29 of the week 11 lecture. How many iterations were required?

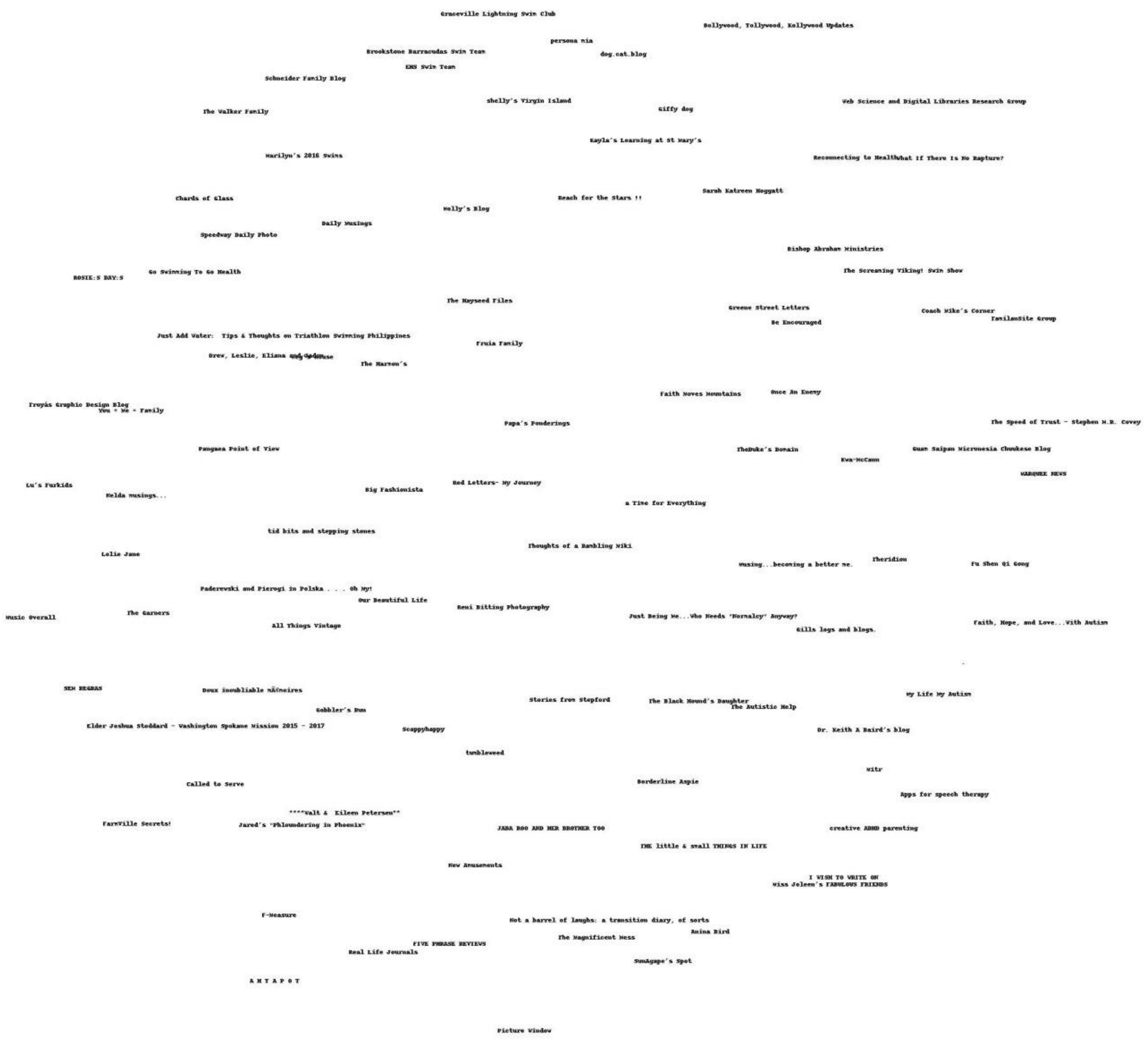
Solution

The solution for this problem is outlined by the following steps:

1. **Create JPEG of Blog MDS**: In order to create a JPEG of the blog MDS, the code provided from the Week 11 Slides, Slide 29 was slightly modified and run. Unfortunately, the number of iterations required is unknown. However, you can find both the code and the JPEG, called MDS.py and Blog MDS.jpeg respectively, from within the package, but a snippet of the code and the JPEG is shown below:



```
#!/usr/bin/python

# -*- coding: utf-8 -*-

import ClustersSource

blognames,words,data=ClustersSource.readfile('Blog Data.txt')

coords=ClustersSource.scaledown(data)

ClustersSource.draw2d(coords,blognames,jpeg='Blog MDS.jpg')
```


Graceville Lightning Swim Club
Bollywood, Tollywood, Kollywood Updates
persona mia
Brookstone Barracudas Swim Team
dog.cat.blog
EMS Swim Team
Schneider Family Blog
shelly's Virgin Island
Giffy dog
Web Science and Digital Libraries Research Group
The Walker Family
Kayla's Learning at St Mary's
Marilyn's 2016 Swims
Reconnecting to HealthWhat If There Is No Rapture?
Sarah Katreen Hoggatt
Chards of Glass
Reach for the Stars !!
Molly's Blog
Daily Musings
Speedway Daily Photo
Bishop Abraham Ministries
Go Swimming To Go Health
The Screaming Viking! Swim Show
ROSIE:S DAY:S
The Hayseed Files
Greene Street Letters
Coach Mike's Corner
TamilanSite Group
Be Encouraged
Just Add Water: Tips & Thoughts on Triathlon Swimming Philippines
Fruia Family
Drew, Leslie, Eliana and Gedon
The Marmon's
Faith Moves Mountains
Once An Enemy
Froyás Graphic Design Blog
You + Me = Family
Papa's Ponderings
The Speed of Trust – Stephen M.R. Covey
Pangaea Point of View
TheDuke's Domain
Guam Saipan Micronesia Chuukese Blog
Kwa-McCann
MARQUEE NEWS
Lu's Furkids
Red Letters- My Journey
Big Fashionista
Nelda musings...
a Time for Everything
tid bits and stepping stones
Thoughts of a Rambling Niki
Lolie Jane
Musing...becoming a better me.
Theridion
Fu Shen Qi Gong
Paderewski and Pierogi in Polska . . . Oh My!
Our Beautiful Life
Remi Bitting Photography
Music Overall
The Garners
Just Being Me...Who Needs 'Normalcy' Anyway?
All Things Vintage
Gills logs and blogs.
Faith, Hope, and Love...With Autism
SEM REGRAS
Doux inoubliable mÃ©moires
Stories from Stepford
The Black Hound's Daughter
The Autistic Help
My Life My Autism
Gobbler's Bun
Elder Joshua Stoddard – Washington Spokane Mission 2015 – 2017
Scappyhappy
Dr. Keith A Baird's blog
tumbleweed
witr
Called to Serve
Borderline Aspie
Apps for speech therapy
****Walt & Eileen Petersen**
FarmVille Secrets!
Jared's "Phloundering in Phoenix"
JABA ROO AND HER BROTHER TOO
creative ADHD parenting
THE little & small THINGS IN LIFE
New Amusements
I WISH TO WRITE ON
Miss Joleen's FABULOUS FRIENDS
F-Measure
Not a barrel of laughs: a transition diary, of sorts
Anina Bird
The Magnificent Mess
FIVE PHRASE REVIEWS
Real Life Journals
SunAgape's Spot
A N T A P O T
Picture Window