# Assignment #9

Tuesday, May 1st, 2018

RANDY DO

# Contents

# Problem 1

```
1.  Support your answer: include all relevant discussion, assumptions,
examples, etc.

(10 points)
1.  Using the data from A7:

- Consider each row in the blog-term matrix as a 1000 dimension vector,
corresponding to a blog.

- Use knnestimate() to compute the nearest neighbors for both:
        http://f-measure.blogspot.com/
        http://ws-dl.blogspot.com/

        for k={1,2,5,10,20}.

        Use cosine distance metric (chapter 8) not euclidean distance.
        So you have to implement numpredict.cosine() instead of using
        numpredict.euclidean() in:
        https://github.com/arthur-e/Programming-Collective-
Intelligence/blob/master/chapter8/numpredict.py
```

# Answer:

**To obtain the nearest neighbors results, we need to use both logs (F-Measure and WSDL). Afterward, we used knnestimate() using the cosine between the vectors to compute the distance. The program I used is A9.py. This script grabs the blogdata.txt from assignment 7. Most of the script was taken from "PCI" book. I edited some of the function. Mainly, using the "Scipy" module to calculate the cosine distance.**

```
import sys
import math
from scipy import spatial

def readfile(filename):
    lines = [line for line in open(filename)]
    colnames = lines[0].strip().split('\t')[1:]
    rownames = []
    data = []
    for line in lines[1:]:
        p = line.strip().split('\t')
        rownames.append(p[0])
        data.append([float(x) for x in p[1:]])
    return (rownames, colnames, data)

def cosine(v1, v2):
    return spatial.distance.cosine(v1, v2)

def getdistances(data,vec1):
    distancelist=[]
    for i in range(len(data)):
        vec2=data[i]
        distancelist.append((cosine(vec1,vec2),i))
    distancelist.sort()
    return distancelist

def knnestimate(blogs, data,vec1,k):
    dlist=getdistances(data,vec1)
    for i in range(k):
        print i+1, '. ', blogs[dlist[i][1]], ': ', dlist[i][0]
    return dlist

def main():
    (blogs,words,data) = readfile('blogdata.txt')
    ks = [1,2,5,10,20]

    counter = 0
    for blog in blogs:
        if (blog == "F-Measure"):
            fm = blog
            blogdata = data[counter]
            blogs.pop(counter)
            data.pop(counter)
            counter = counter + 1

    print fm
    print "=============="

    for k in ks:
        print 'For K = ', k, ' :'
        print '******************'
        knnestimate(blogs, data, blogdata, k)
```

```
def knnestimate(blogs, data,vecl,k):
    dlist=getdistances(data,vecl)
    for i in range(k):
        print i+1, '. ', blogs[dlist[i][1]], ': ', dlist[i][0]
    return dlist

def main():
    (blogs,words,data) = readfile('blogdata.txt')
    ks = [1,2,5,10,20]

    counter = 0
    for blog in blogs:
        if (blog == "F-Measure"):
            fm = blog
            blogdata = data[counter]
            blogs.pop(counter)
            data.pop(counter)
            counter = counter + 1

    print fm
    print "=============="

    for k in ks:
        print 'For K = ', k, ' :'
        print '******************'
        knnestimate(blogs, data, blogdata, k)

    blogs.append(fm)
    data.append(blogdata)

    print "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"

    counter = 0
    for blog in blogs:
        if (blog == "Web Science and Digital Libraries Research Group"):
            wsdl = blog
            blogdata = data[counter]
            blogs.pop(counter)
            data.pop(counter)
            counter = counter + 1

    print wsdl
    print "=============="
        for k in ks:
            print 'For K = ', k, ' :'
            print '******************'
            knnestimate(blogs, data, blogdata, k)


if __name__ == "__main__":
    main()
```

# Output:

```
F-Measure
==========
For K =  1  :
******************
C:\Python27\lib\site-packages\scipy\spatial\distance.py:644: RuntimeWarning: invalid value encountered in double_scalars
  dist = 1.0 - uv / np.sqrt(uu * vv)
1 .  music of the moment :  0.747502008981
For K =  2  :
******************
1 .  music of the moment :  0.747502008981
2 .  She May Be Naked :  0.800373694827
For K =  5  :
******************
1 .  music of the moment :  0.747502008981
2 .  She May Be Naked :  0.800373694827
3 .  Pithy Title Here :  0.81329033906
4 .  Cuz Music Rocks :  0.842757274492
5 .  Bonjour Girl :  0.856777025192
For K =  10  :
******************
1 .  music of the moment :  0.747502008981
2 .  She May Be Naked :  0.800373694827
3 .  Pithy Title Here :  0.81329033906
4 .  Cuz Music Rocks :  0.842757274492
5 .  Bonjour Girl :  0.856777025192
6 .  Angie Dynamo :  0.857346502496
7 .  Playing Favorites :  0.858534428862
8 .  Steel City Rust :  0.866412775217
9 .  Pirate's Log :  0.877225518148
10 .  a duchess nonethelesss :  0.877852632654
For K =  20  :
******************
1 .  music of the moment :  0.747502008981
2 .  She May Be Naked :  0.800373694827
3 .  Pithy Title Here :  0.81329033906
4 .  Cuz Music Rocks :  0.842757274492
5 .  Bonjour Girl :  0.856777025192
6 .  Angie Dynamo :  0.857346502496
7 .  Playing Favorites :  0.858534428862
8 .  Steel City Rust :  0.866412775217
9 .  Pirate's Log :  0.877225518148
10 .  a duchess nonethelesss :  0.877852632654
11 .  jaaackie. :  0.878778473733
12 .  Myopiamuse :  0.886772296586
13 .  Did Not Chart :  0.890792040846
14 .  Eli Jace :  0.893973065738
15 .  Bleak Bliss :  0.896026862623
16 .  Stories From the City, Stories From the Sea :  0.898620770659
17 .  DaveCromwell Writes :  0.901894855021
18 .  ORGANMYTH :  0.907549967296
19 .  Web Science and Digital Libraries Research Group :  0.908590553493
20 .  Music-Drop Magazine :  0.910729154577
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Web Science and Digital Libraries Research Group
==========
For K =  1  :
******************
1 .  Pithy Title Here :  0.761460270285
For K =  2  :
******************
1 .  Pithy Title Here :  0.761460270285
2 .  She May Be Naked :  0.772901734394
For K =  5  :
```

```
For K =  5  :
******************
1 .  Pithy Title Here :  0.761460270285
2 .  She May Be Naked :  0.772901734394
3 .  *Sixeyes: by Alan Williamson :  0.781782109764
4 .  jaaackie. :  0.789156340257
5 .  a duchess nonethelesss :  0.789478572747
For K =  10  :
******************
1 .  Pithy Title Here :  0.761460270285
2 .  She May Be Naked :  0.772901734394
3 .  *Sixeyes: by Alan Williamson :  0.781782109764
4 .  jaaackie. :  0.789156340257
5 .  a duchess nonethelesss :  0.789478572747
6 .  Pirate's Log :  0.792850056431
7 .  i'm in too truthful a mood :  0.795643663428
8 .  Steel City Rust :  0.798163396988
9 .  Tremagazine :  0.799013583396
10 .  The Ideal Copy :  0.815073860692
For K =  20  :
******************
1 .  Pithy Title Here :  0.761460270285
2 .  She May Be Naked :  0.772901734394
3 .  *Sixeyes: by Alan Williamson :  0.781782109764
4 .  jaaackie. :  0.789156340257
5 .  a duchess nonethelesss :  0.789478572747
6 .  Pirate's Log :  0.792850056431
7 .  i'm in too truthful a mood :  0.795643663428
8 .  Steel City Rust :  0.798163396988
9 .  Tremagazine :  0.799013583396
10 .  The Ideal Copy :  0.815073860692
11 .  music of the moment :  0.819240309687
12 .  Make Up, Music & Fashion :  0.823002098342
13 .  Eli Jace :  0.839152245527
14 .  Stonehill Sketchbook :  0.83999841973
15 .  F-Measure :  0.842757274492
16 .  KiDCHAIR :  0.844045585341
17 .  The Great Adventure 2016 :  0.846202532845
18 .  A Day in the Life of...Me!! :  0.846330342038
19 .  Did Not Chart :  0.846797421553
20 .  My Name Is Blue Canary :  0.84760146366
```