Edited script from assignment 7 to generate the blog term matrix without labels for assignment 9, saved output as blogdataNL.txt. The first two rows were manually removed and saved separately from the file, as those two rows were data for the two blogs being compared. Also have edited Euclidean to use cosine matching so that I did not have to find and replace wherever the Euclidean function was used.

**def** euclidean**(**v1**,**v2**):**

d**=**0.0

**for** i **in** range**(**len**(**v1**)):**

# d+=(v1[i]-v2[i])\*\*2

#return math.sqrt(d)

#re-written to use cosine

d **=** d **\*** **((**v1**\***v2**)/((**math**.**sqrt**(**v1**\***v1**))\*(**math**.**sqrt**(**v2**\***v2**))))**

**return** d

I could not figure out what **data** and **vecl** parameters are supposed to be in knnestimate(), though. What follows is my non-functioning best guess.

#estimate.py

**import** numpredict

**import** csv

**import** numpy **as** np

#open file with array data

#infile = open("blogdataNH.txt", "r" )

outfile **=** open**(**"outfile.dat"**,** "w"**)**

#new\_array = np.array((array.float(i) for i in line.split(' ')))

#f-measure vector

blogAs **=** "9 51 2 2 1 43 2 3 2 3 1 3 1 2 5 2 1 1 1 1 2 1 2 2 2 1 1 2 1 1 1 1 3 2 1 1 1 2 1 1 1 2 2 1 8 10 3 5 1 2 1 1 1 1 6 2 1 4 2 9 7 1 14 13 3 2 20 1 1 16 2 1 7 1 2 3 2 3 2 3 2 3 6 2 3 10 2 3 6 1 2 3 3 8 1 3 6 5 7 5 1 5 3 3 1 4 2 4 18 2 1 1 1 1 3 2 1 6 2 3 6 1 2 1 1 2 1 9 7 4 2 3 1 1 5 3 1 1 3 2 1 3 2 1 1 7 2 2 2 1 1 3 7 4 3 12 4 1 1 5 6 8 2 3 2 1 1 1 1 2 1 1 2 1 1 2 4 3 1 1 1 3 1 5 3 2 1 1 3 3 1 2 19 6 1 2 1 1 3 2 4 1 1 1 1 2 2 3 2 2 2 2 1 1 1 1 2 3 4 1 1 2 1 2 1 5 1 1 1 3 2 4 11 3 9 1 1 2 1 3 3 2 9 1 1 1 1 1 3 2 2 1 2 1 1 1 3 6 1 3 2 3 5 3 3 3 1 1 1 2 1 3 1 2 1 1 1 2 1 2 3 1 1 1 5 5 1 1 1 1 1 1 1 1 7 1 1 1 5 1 1 1 3 2 1 6 1 2 2 1 1 4 2 1 3 2 1 2 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 3 1 1 1 2 1 1 3 1 5 3 1 1 2 1 1 3 10 1 1 1 2 1 1 1 2 2 2 1 1 9 9 2 1 1 2 1 3 7 1 3 1 5 5 2 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 7 1 2 1 2 1 1 1 1 1 2 1 7 3 6 1 2 7 13 1 1 2 1 1 1 1 1 1 9 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 4 1 3 1 1 1 1 1 1 1 1 2 2 1 1 2 1 3 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 3 1 1 2 1 1 1 1 1 1 2 7 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 3 1 1 1 1 1 1 1 1 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 5 1 1 1 1 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0"

#ws-dl vector

blogBs **=** "3 1 9 2 6 0 4 0 0 0 0 0 0 0 0 8 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 2 0 0 0 0 0 0 1 0 0 24 4 13 3 1 3 11 68 122 257 10 447 121 21 1 0 9 3 0 12 6 6 0 7 3 0 6 11 0 1 16 0 10 1 5 3 5 1 2 7 26 0 0 1 5 1 5 2 4 0 6 0 8 0 1 0 2 1 0 4 0 3 3 2 5 1 1 2 4 0 1 0 11 7 6 13 0 0 12 5 6 0 0 0 1 1 1 3 1 1 3 0 1 1 7 3 1 0 9 1 8 6 2 0 1 1 0 0 4 0 2 1 3 1 9 1 2 4 12 2 15 1 17 0 0 2 7 2 1 1 1 2 9 3 5 3 1 1 11 7 3 2 6 0 0 2 6 3 1 2 9 0 2 2 0 0 19 0 6 2 0 7 1 5 2 1 7 1 16 17 0 5 2 0 0 1 1 1 3 1 12 1 7 7 9 2 1 11 0 7 1 1 1 2 1 3 9 0 14 40 0 0 3 11 0 3 11 1 0 9 0 0 0 2 0 0 1 0 0 4 6 6 0 7 4 1 2 1 15 8 0 4 1 8 2 1 1 3 2 17 3 0 2 0 0 8 0 0 0 1 3 6 1 0 1 0 2 0 26 9 1 3 0 0 0 2 14 1 0 1 0 6 0 1 0 0 4 4 3 1 0 1 2 3 5 19 8 4 2 0 12 4 76 2 4 3 65 0 2 1 2 1 0 0 9 0 5 1 2 1 5 0 4 11 3 0 2 0 20 0 318 15 2 7 1 59 4 0 6 8 0 1 0 1 0 0 1 0 0 0 2 10 0 0 1 0 2 0 2 3 0 1 0 0 2 0 1 2 7 4 6 0 0 4 14 0 0 1 0 2 7 3 2 3 0 3 2 6 2 0 0 9 1 0 15 5 0 2 1 2 2 6 2 3 11 1 1 1 0 2 0 3 2 1 18 2 1 0 18 0 11 5 13 5 12 4 57 27 0 1 2 0 2 4 6 1 18 1 0 5 11 1 2 2 1 15 2 1 0 0 0 7 2 0 2 0 7 13 1 0 2 2 0 0 3 10 0 0 0 1 0 1 23 0 1 0 0 0 0 3 0 2 0 2 0 2 0 0 2 2 5 0 0 18 1 1 1 5 2 4 2 2 0 2 0 3 5 1 0 0 1 6 1 21 1 13 2 7 0 4 2 2 2 0 4 1 3 6 6 0 9 6 0 1 0 2 1 12 14 0 3 0 2 3 36 2 2 0 9 0 3 1 4 3 0 1 0 0 0 6 1 1 16 13 17 11 3 23 2 2 4 3 6 38 6 7 2 6 2 7 41 1 9 2 5 1 21 16 102 7 1 27 1 6 4 129 47 12 4 5 54 3 18 4 6 7 3 1 3 6 1 6 4 3 4 11 3 2 142 2 4 4 1 3 14 1 3 5 2 1 28 9 4 6 5 10 14 2 2 15 11 65 28 25 21 16 19 35 10 8 14 3 10 11 63 9 2 7 2 1 2 2 1 8 3 1 22 8 4 2 29 33 10 52 16 9 8 2 8 35 2 6 1 1 4 8 9 5 7 4 4 3 2 1 12 2 7 27 11 1 13 1 3 29 10 28 5 4 1 4 5 50 10 8 19 13 10 54 1 1 35 6 6 5 1 7 2 4 13 2 30 2 10 4 5 3 1 30 13 3 47 3 9 8 10 2 11 7 17 11 13 4 4 1 1 10 7 13 8 4 6 7 1 3 3 2 7 13 11 18 2 9 3 2 2 3 4 4 37 8 2 7 14 2 5 11 2 1 1 7 4 4 27 5 4 25 8 2 11 3 3 5 5 5 3 4 4 3 9 17 9 2 1 19 4 6 42 2 2 2 11 3 3 4 3 6 6 4 4 15 25 3 2 1 1 5 11 4 3 5 1 2 4 3 6 2 2 2 2 1 2 11 11 3 2 2 2 5 15 1 5 1 1 4 2 2 11 7 1 1 5 1 8 19 4 5 1 1 1 4 1 13 5 13 2 3 2 14 10 20 3 1 3 7 4 1 1 1 8 3 1 9 7 2 21 11 3 3 40 57 4 32 12 8 2 3 6 7 2 1 3 10 1 3 1 3 2 24 1 4 5 54 7 3 3 6 4 6 5 3 4 7 4 6 3 3 4 4 3 16 4 3 5 4 4 1 1 1 2 5 4 3 18 7 7 5 7 3 4"

blogA **=** np**.**array**((**array**.**int**(**i**)** **for** i **in** blogAs**.**split**(**' '**)))**

blogB **=** np**.**array**((**array**.**int**(**i**)** **for** i **in** blogBs**.**split**(**' '**)))**

data **=** **[]**

parser **=** csv**.**reader**(**open**(**"blogdataNL.txt"**),** delimiter**=**' '**)**

**for** l **in** parser**:**

data**.**append**(**np**.**array**((**array**.**int**(**i**)** **for** i **in** l**)))**

#compare f-measure to array

outfile**.**write**(**"f-measure \n"**)**

#for i in range(data.size):

outfile**.**write**(**"k=1: "**+** numpredict**.**knnestimate**(**data**,**blogA**,**k**=**1**)** **+** "\n"**)**

#outfile.write("k=2: "+ numpredict.knnestimate(data,blogA,2)+ "\n")

#outfile.write("k=5: "+ numpredict.knnestimate(data,blogA,5)+ "\n")

#outfile.write("k=10: "+ numpredict.knnestimate(data,blogA,10)+ "\n")

#outfile.write("k=20: "+ numpredict.knnestimate(data,blogA,20)+ "\n")

#compare sw-dl to array

#outfile.write("sw-dl \n")

#outfile.write("k=1: "+ numpredict.knnestimate(data,blogB,1)+ "\n")

#outfile.write("k=2: "+ numpredict.knnestimate(data,blogB,2)+ "\n")

#outfile.write("k=5: "+ numpredict.knnestimate(data,blogB,5)+ "\n")

#outfile.write("k=10: "+ numpredict.knnestimate(data,blogB,10)+ "\n")

#outfile.write("k=20: "+ numpredict.knnestimate(data,blogB,20)+ "\n")