ROMANI FLORES ROGELIO

1. Modifique el código de parsear blogs para agrupar entradas individuales en lugar de blogs enteros. ¿Las entradas del mismo blog se agrupan unas con otras? ¿Qué hay de las entradas de la misma fecha?

def getwordcounts(url):

# Analiza lexicamente el feed

d=feedparser.parse(url)

wr={}# variable que almacena un diccionario

# para el conteo de palabras de cada blog

t={} # titulo de cada blog

cont=0 #contador de todas las entradas

# itera sobre todas las entradas

for e in d.entries:

if 'summary' in e: summary=e.summary

else: summary=e.description

# extrae una lista de palabras

words=getwords(e.title+' '+summary)

wc={}

for word in words:

wc.setdefault(word,0)

wc[word]+=1

wr[e.title]=wc

t[cont]=e.title

cont=cont+1

return d.feed.title,wc,t,wr

1. **Pruebe usando la distancia euclidiana para clustering de los blogs. ¿Cómo cambia esto los resultados?**

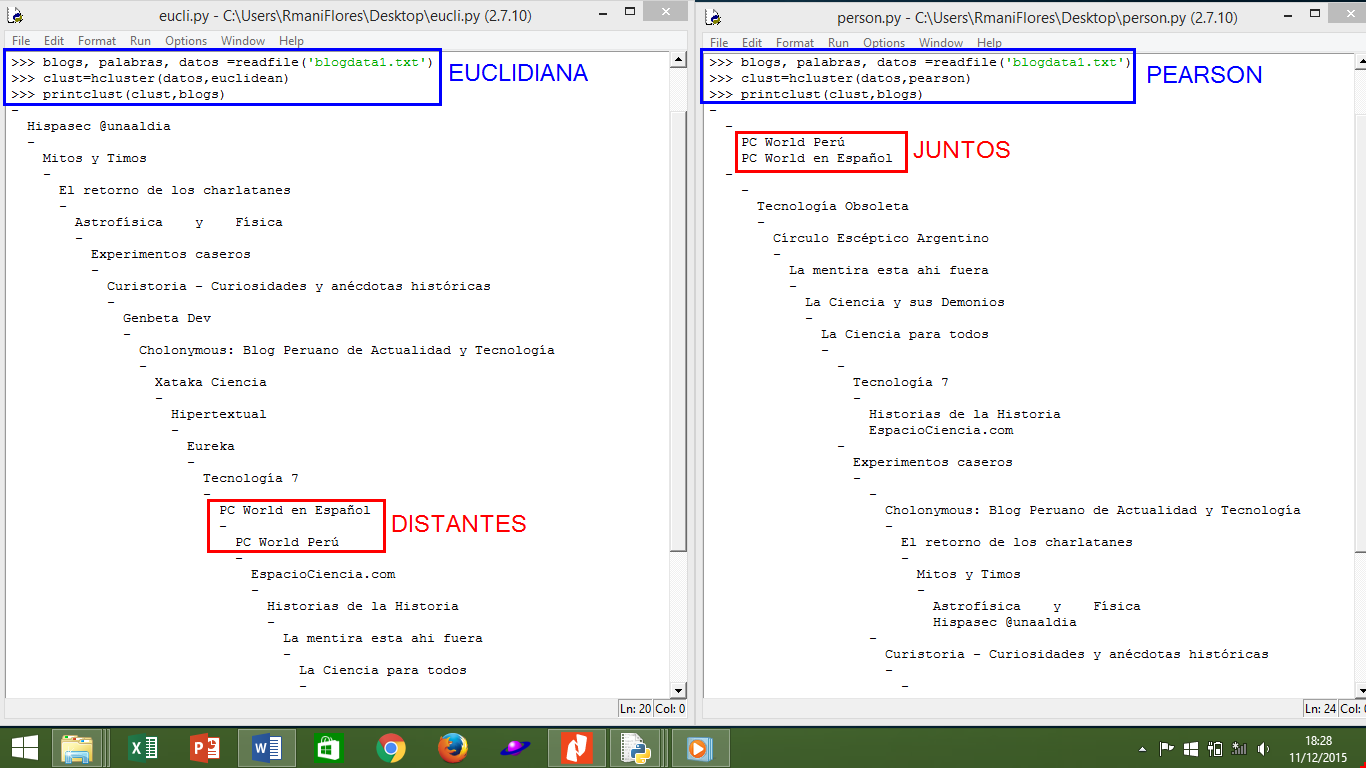
Con las siguientes líneas de código podemos visualizar el resultado

>>> blogs, palabras, datos =readfile('blogdata1.txt')

>>> clust=hcluster(datos,euclidean)

>>> printclust(clust,blogs)

Como vemos en la Figura 1 los resultados de distancia Euclidiana y Pearson son totalmente diferentes.

Figura 1. Comparando distancias de Pearson y Euclidiana

1. **Investigue acerca de la distancia de Manhattan. Cree una función para esta y vea cómo cambia los resultados.**

Función que devuelve la distancia entre dos vectores

def manhattan(v1, v2):

suma=0

for i in range(len(v1)):

suma+=pow(pow(v1[i]-v2[i], 2), 0.5)

return suma

Líneas de código para visualizar los resultados

>>> blogs, palabras, datos =readfile('blogdata1.txt')

>>> clust=hcluster(datos, manhattan)

>>> printclust(clust,blogs)

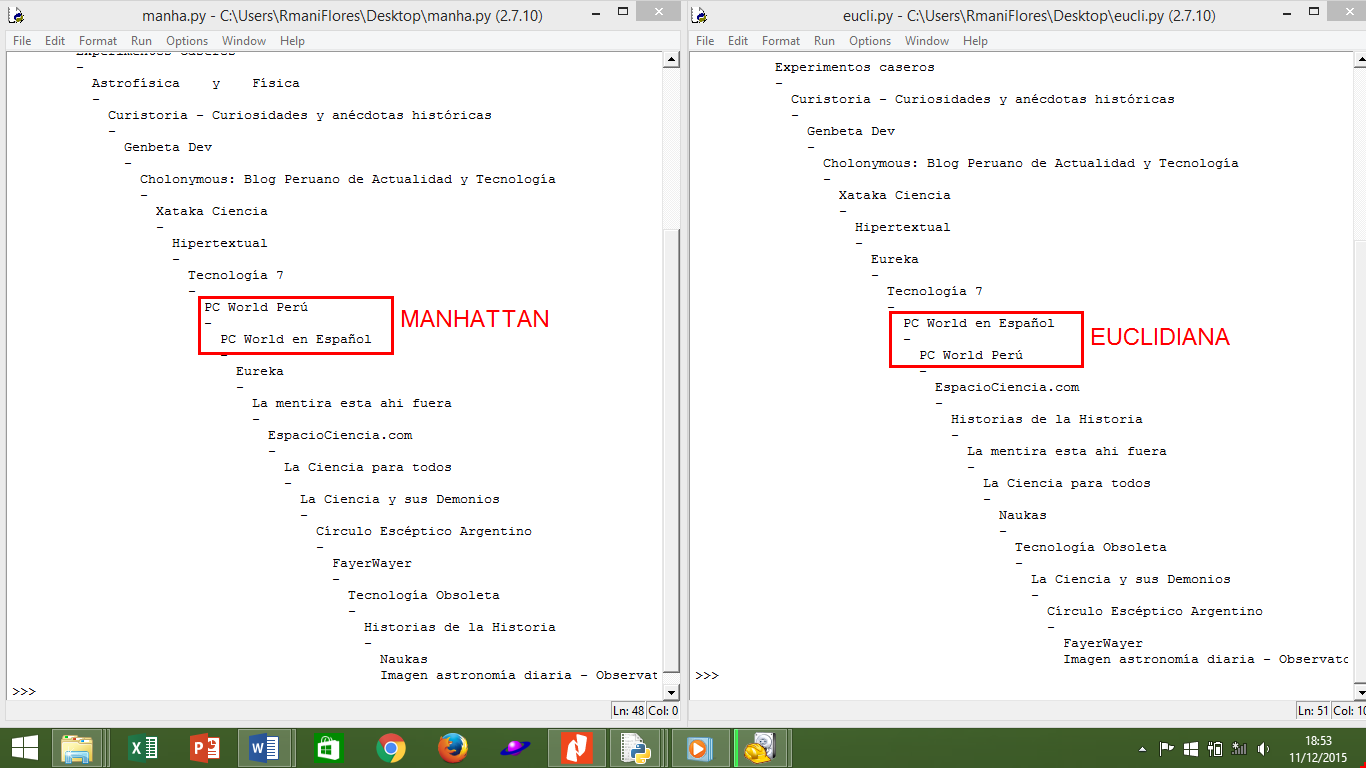
Como se observa en la Figura 2 no hay mucha diferencia entre las distancias Manhattan y Euclidiana.

Figura 2. Comparando distancias Euclidiana y Manhattan

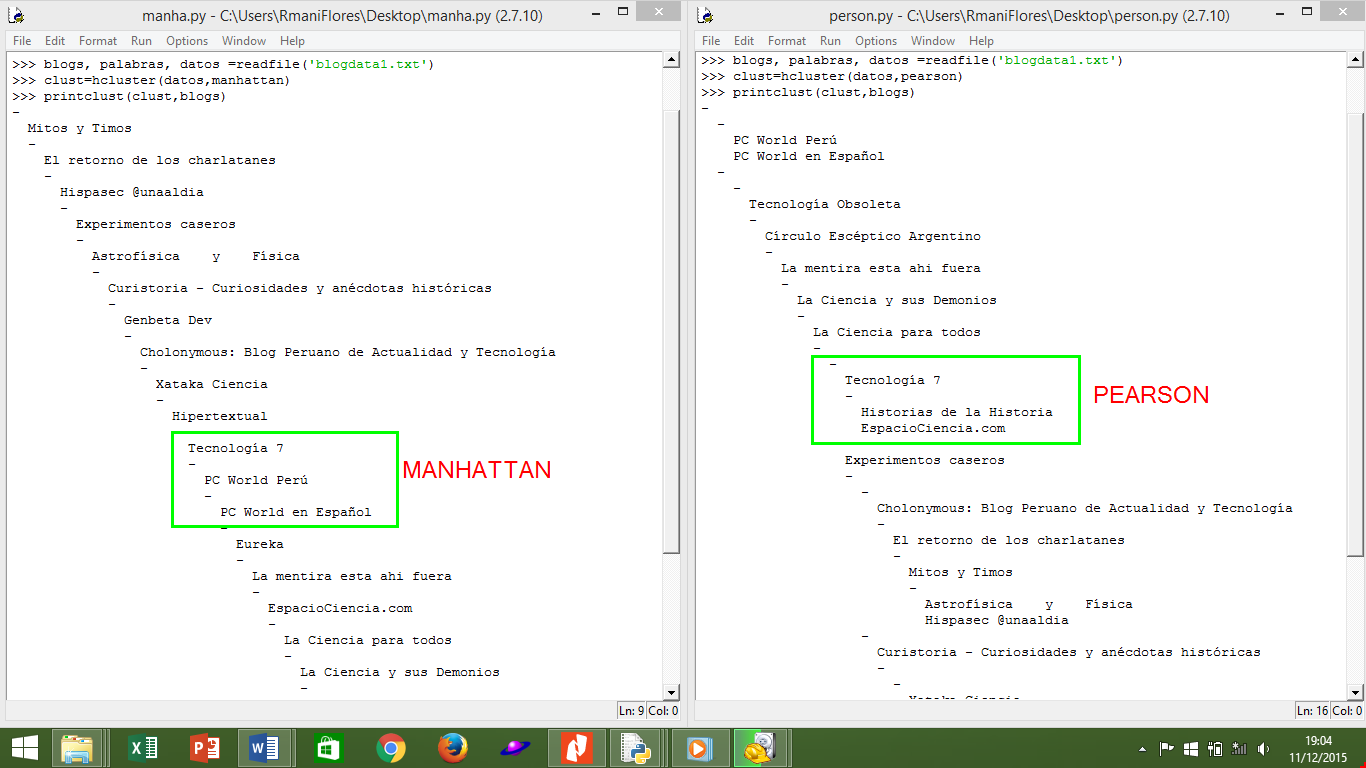
También en la figura 3 vemos la diferencia entre distancias de Manhattan y Pearson.

Figura 3. Comparando distancias entre Pearson y Manhattan

1. Modifique la función del clustering K-means para que devuelva la distancia total ente todos los ítems de sus respectivos centroides.