Τρίτη Σειρά Ασκήσεων – Εξόρυξη Δεδομένων

Όνομα: Χρύσα Τεριζή
ΑΜ: 2553
Ημερομηνία: 27/5/2017
Free passes: Δεν χρησιμοποίησα κανένα

Ερώτηση 1(Maximum Likelihood Estimation)

Αρχικά έχουμε, L(a) = $\prod_{i=1}^{n}(a-1)Xi^{-a}$,
ξέρουμε ότι για να πάμε από γινόμενο σε άθροισμα πρέπει να χρησιμοποιήσουμε τον λογάριθμο οπότε,

LL(a) = $\sum_{i=1}^{n}\log(a-1)Xi^{-a} = \sum_{i=1}^{n}\log(a-1)+\log(Xi^{-a}) = \sum_{i=1}^{n}\log(a-1)-a\log(Xi)$.

Έπειτα πρέπει να παραγωγίσω το LL(a) ως προς a και έχουμε,

$\frac{LL(a)}{\partial a} = \sum_{i=1}^{n}(a-1)'\frac{1}{a-1}-(a)'\log(Xi)-a(\log(Xi))' = \sum_{i=1}^{n}1\frac{1}{a-1}-\log(Xi)$,

τώρα πρέπει να εξισώσω το $\frac{LL(a)}{\partial a}=0$ => $\sum_{i=1}^{n}1\frac{1}{a-1}-\log(Xi)=0$ =>
$\sum_{i=1}^{n}1\frac{1}{a-1}-\sum_{i=1}^{n}\log(Xi)=0$ => $\sum_{i=1}^{n}1\frac{1}{a-1}=\sum_{i=1}^{n}\log(Xi)$ => $\frac{1}{a-1}\sum_{i=1}^{n}1 = \sum_{i=1}^{n}\log(Xi)$ =>
$\frac{1}{a-1}n=\sum_{i=1}^{n}\log(Xi)$ => $a-1=\frac{n}{\sum_{i=1}^{n}\log(Xi)}$ => $a=1+\frac{n}{\sum_{i=1}^{n}\log(Xi)}$.

Άρα, βρήκα ότι $a=1+\frac{n}{\sum_{i=1}^{n}\log(Xi)}$.

Ερώτηση 2

Μπορούμε να το υλοποιήσουμε με έναν ευρετικό αλγόριθμο τον top-down greedy αλγόριθμο ο οποίος υλοποιείται σε γραμμικό χρόνο O(NK). Η ιδέα είναι να εισάγουμε κάθε φορά ένα όριο με το οποίο θα ελαχιστοποιήσουμε το άθροισμα των διαμέτρων μέχρι να δημιουργηθούν οι k ομάδες που θέλουμε.

numbers = ***{X1, X2, … , Xn}*** οι οποίοι είναι ταξινομημένοι και έχουν ένα index, δηλαδή ο αριθμός X1 έχει index 0, ο Xn έχει σαν index το n-1
k = ***το πλήθος των ομάδων που θέλουμε να φτιάξουμε***

bounds* = *{[είναι τα index στα οποία θα προσθέσουμε ένα όριο δεξιά από αυτά]}

Για ***i από το 0 έως όσες είναι ο ομάδες που θέλουμε να φτιάξουμε – 1***:

minIndex = 0, κάθε φορά θα βάζουμε το όριο δεξιά από τον πρώτο αριθμό όπου είναι και ο μικρότερος και θα μετακινούμε αυτό το όριο προς τα δεξιά κατά μία θέση μέχρι θα βρούμε σε ποιο index το error θα γίνει το μικρότερο

minError = Θα βρούμε το λάθος με βάση τα index που έχουμε βάλει στο bounds

Για ***number όλους τους αριθμούς(για τα index τους) από το number εκτός από τον πρώτο μέχρι τον τελευταίο***:

Εάν αυτό το index ανήκει στα bounds τότε να συνεχίσω

Διαφορετικά, προσθέτω το index αυτό στα bounds, δηλαδή ότι εκεί υπάρχει ένα όριο που χωρίζονται οι αριθμοί. Υπολογίζω το error και ***αν είναι μικρότερο από το minError*** τότε κρατάω το index του.

Αφαιρώ το index αυτό από τα bounds

Όταν τελειώσουν όλες οι επαναλήψεις για τις πιθανές θέσεις που μπορώ να βάλω το ένα όριο ***τότε έχω βρει ένα όριο για τους αριθμούς μου***. Οπότε προσθέτω αυτό το index στο bounds.

Συνεχίζω για τις υπόλοιπες ομάδες με την for που υπάρχει πιο πάνω και μετά θα βρω το 2ο όριο που θα πρέπει να μπει ώστε το error να είναι το ελάχιστο.

Όταν τελειώσουν όλες οι επαναλήψεις θα έχω βρει τα όριο στα οποία χωρίζονται οι γραμμές.

Με την εξωτερική επανάληψη θα κάνω k επαναλήψεις, με την εσωτερική θα κάνω N επαναλήψεις, άρα ο χρόνος θα είναι O(kN). Σε κάθε βήμα του αλγορίθμου βρίσκω την καλύτερη λύση και πάνω σε αυτήν την λύση βρίσκω την επόμενη καλύτερη. Ένας greedy αλγόριθμος επικεντρώνεται στην επίτευξη της τοπικής βέλτιστης λύσης.

Ερώτηση 3

Αρχικά, έχω την ίδια συνάρτηση όπως και στις προηγούμενες σειρές ασκήσεων την "***def getTagsHandles(x)***" η οποία παίρνει σαν όρισμα μία γραμμή(το text του tweet) και επιστρέφει τα hashtags/handles που περιέχει. Έπειτα διαβάζω το αρχείο και κρατάω στο λεξικό "***userName = {}***" τα ονόματα των χρηστών σαν κλειδιά και σαν τιμή τους μία λίστα που περιέχει τα hashtags/handles που έχουν χρησιμοποιήσει. Στο λεξικό "***dictionaryForHashtagsHandles = {}***" κρατάω σαν κλειδί το hashtag/handle και σαν τιμή του το πλήθος που έχει εμφανιστεί. Σε αυτήν την φάση αν ένα hashtag/handle έχει χρησιμοποιηθεί περισσότερες από μία φορά από τον ίδιο χρήστη αυξάνω το πλήθος του αλλά πιο μετά το χειρίζομαι. Ξεκινάω να διαβάζω μία μία τις γραμμές τους αρχείου, στην μεταβλητή "***splittedTweet***" κρατάω το όλο tweet σαν λίστα ενώ το έχω κάνει split με βάση το tab. Ελέγχω αν είναι retweet, δηλαδή αν το "***splittedTweet[-1]***" που είναι το κύριο μέρος του tweet που έχει κάνει ο χρήστης περιέχει το "***RT***" τότε δεν βρίσκω για αυτόν τον χρήστη τα hashtag/handle που υπάρχουν στο υπόλοιπο μέρος του tweet, ουσιαστικά τα αγνοώ. Ξέρω ότι την θέση "***splittedTweet[1]***" θα βρίσκεται το userName του χρήστη. Στην "***hhList***" κρατάω τα hashtag/handle που έχει χρησιμοποιήσει. Βρίσκω ότι έχω συνολικά,

Number of total users: 164712
Number of total hashtags/handles: 639961

Κατόπιν, κάνω το πρώτο κλάδεμα για τους χρήστες, δηλαδή αν το πλήθος από διαφορετικά hashtags/handles του κάθε χρήστη είναι λιγότερα ή ίσα από 20 τους πετάω και μέσα στο "***firstUserPruning = {}***" κρατάω τους χρήστες που θέλω μόνο. Στο "***realCrowdOfHashtagsHandles***" βρίσκω το πραγματικό πλήθος του κάθε hashtag/handle δηλαδή από πόσους διαφορετικούς χρήστες έχει χρησιμοποιηθεί. Κάνω και εδώ το δικό του κλάδεμα και τα καινούρια αποτελέσματα τα κρατάω μέσα στο λεξικό "***firstHashtagHandlePruning = {}***". Βρίσκω ότι υπάρχουν,

Number of total users after first pruning: 16179
Number of total hashtag/handle after first pruning: 5998

Δηλαδή, έχω κρατήσει μόνο το *9.82% των αρχικών χρηστών* και το *0.93% των αρχικών hashtags/handles*. Ακολουθεί το επαναληπτικό κλάδεμα, με την ίδια λογική που είχα κάνει και στην 2η σειρά ασκήσεων. Τα αποτελέσματα που βρίσκω είναι τα ακόλουθα,

1η επανάληψη
After users: 4370

2η επανάληψη
After HH: 2104
After users: 3303

3η επανάληψη
After HH: 1836
After users: 3156

4η επανάληψη
After HH: 1783
After users: 3118

5η επανάληψη
After HH: 1768
After users: 3111

6η επανάληψη
After HH: 1765
After users: 3108

7η επανάληψη
After HH: 1765
After users: 3108

Δηλαδή, έχουμε κρατήσει μόνο το *1.88% των αρχικών χρηστών* και το *0.27% των αρχικών hashtags/handles*.

Μετά κάνω import κάποιες βιβλιοθήκες που ίσως μου χρειαστούν.

1ο πρόβλημα

Θέλω να κρατήσω για κάθε hashtag/handle ένα διάνυσμα για αυτό που να έχει σαν πληροφορία από τον κάθε χρήστη πόσες φορές χρησιμοποιήθηκε. Τα διανύσματα αυτά θα είναι αρκετά αραιά αλλά επειδή τα δεδομένα μου είναι λίγα σε πλήθος θα τα κρατήσω και με τα μηδενικά για να τα εισάγω μετά τους αλγορίθμους clustering. Για να μπορέσω να φτιάξω αυτά τα διανύσματα έχω φτιάξει μία δομή λεξικού το "***userDictHHDict = {}***" το οποίο έχει την μορφή ***{user:{hashtag:συχνότητα}}***. Για παράδειγμα για κάθε χρήστη φαίνεται έτσι,

robert2266 : {'#healthcare': 8, '#Italy': 2, '#Halloween': 1, '@': 1, '#Putin': 1, '#Obamacare': 2, '#Periscope': 1, '#Trumps': 1, '#WakeUpAmerica': 7, '#DAPL': 6, '#FBI': 5, '#Americans': 4, '#Republicans': 1, '#HillaryClinton': 3, '#PodestaEmails': 1, '#America': 1, '#Clinton': 5, '#ClimateChange': 2, '#Trump': 2, '#tcot': 12, '#Syria': 8, '#truth': 1, '#Florida': 2, '#Israel': 2, '#US': 2, '#BREAKING': 1, '#Iran': 1, '#corruption': 2, '#DonaldTrump': 2, '#China': 3, '#DNC': 1, '#halloween': 3, '#NoDAPL': 5, '#GOP': 1, '#election2016': 4, '#Russia': 1, '#politics': 2, '#Nevada': 1, '#TrumpPence': 1, '#Wikileaks': 2, '@TIME': 1, '#Turkey': 1, '#Election2016': 1, '#economy': 1, '#Haiti': 1, '#Clintons': 1, '#JamesComey': 2, '#ISIS': 3, '#Aleppo': 5}

Έχω φτιάξει 2 λίστες με ονόματα "***hhInSeries = []***" η μία και η άλλη με "***namesInSeries = []***", ουσιαστικά έχω αποθηκεύσει μέσα όλα τα hashtags/handles μέσα που θα χρησιμοποιήσω ώστε να τα έχω σε μία σειρά, το ίδιο κάνω και με τους χρήστες. Οι πρώτες 5 θέσεις των λιστών αυτών είναι κάπως

έτσι,
['#healthcare', '#Italy', '#Halloween', '@', '#Putin']
['robert2266', 'CaptainNormal', 'UnBitterEnd2013', 'Jodyjtaylor', 'DonKeehotey']

Ξεκινάω να φτιάχνω τα διανύσματα για το κάθε hashtag/handle. Θεωρώ ότι στον κάθετο άξονα είναι όλα τα hashtag/handle και στον οριζόντιο άξονα είναι όλοι οι χρήστες. Τα διανύσματα αυτά τα κρατάω μέσα στο "***hashtagHandleVector = []***", αρχικά γεμίζω την λίστα με κενές λίστες γεμάτες με 0 μεγέθους όσοι είναι και οι χρήστες δηλαδή ***[[0 0 0 ... #χρηστών], [0 0 0 ... #χρηστών], [0 0 0 ... #χρηστών], ... #hashtag/handle]***. Για κάθε έναν χρήστη(κλειδί από το "***userDictHHDict***") παίρνω το index της θέσης του από την λίστα "***namesInSeries***" ώστε να ξέρω σε ποια στήλη θα μπουν τα δεδομένα του. Στο "***listOfHHKeys***" κρατάω τα κλειδιά από το λεξικό που είναι σαν τιμή του κάθε χρήστη. Στην "***listOfHHValues***" κρατάω τις τιμές των κλειδιών αυτών. Μετά για κάθε ένα στοιχείο της λίστας "***listOfHHKeys***" βρίσκω το index του hashtag/handle ώστε να ξέρω σε ποια γραμμή του "***hashtagHandleVector***" να πάω να συμπληρώσω δεδομένα. Το "***hashtagHandleVector***" είναι της μορφής,

```
[[8 0 0 ..., 0 0 0]
 [2 0 0 ..., 0 0 0]
 [1 0 0 ..., 0 0 0]
 ...,
 [0 0 0 ..., 0 0 0]
 [0 0 0 ..., 0 0 0]
 [0 0 0 ..., 0 0 0]]
```

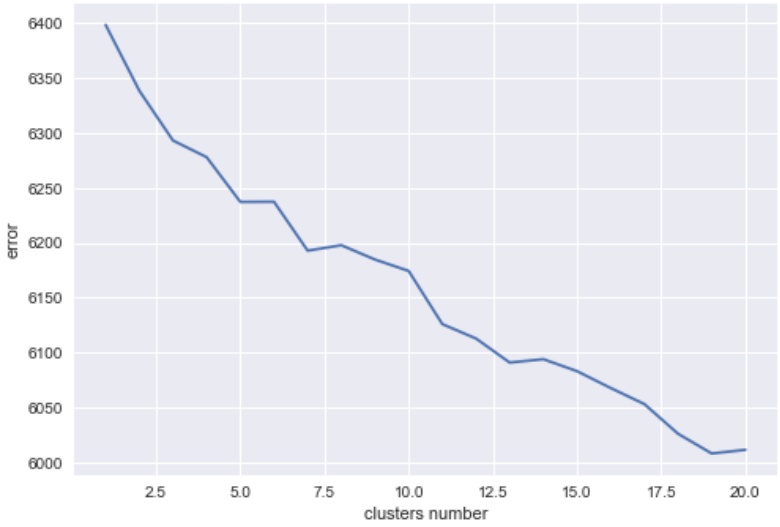
Η κανονικοποιημένη τους μορφή είναι η ακόλουθη,

```
[[ 0.25        0.        0.        ...,  0.        0.        0.        ]
 [ 0.01818182  0.        0.        ...,  0.        0.        0.        ]
 [ 0.16666667  0.        0.        ...,  0.        0.        0.        ]
 ...,
 [ 0.          0.        0.        ...,  0.        0.        0.        ]
 [ 0.          0.        0.        ...,  0.        0.        0.        ]
 [ 0.          0.        0.        ...,  0.        0.        0.        ]]
```

Αλγόριθμοι clustering

K-means

Πρέπει να τον τρέξω για διάφορες τιμές([1, 20]) πλήθους cluster. Έχω χρησιμοποιήσει την κλάση *sklearn.cluster* **Kmeans**(*n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1, algorithm='auto'*) αυτή είναι η γενική της μορφή για να βρίσκω τα αποτελέσματα του k-means αλγορίθμου. Το γράφημα για το SSE error για διάφορες τιμές clusters είναι το παρακάτω,



Συμπέρασμα από το γράφημα: Βλέπω ότι όταν ο αριθμός των clusters είναι από [10, 20] το error πέφτει αρκετά. Όμως δεν είναι καλό να διαλέξω 20 clusters γιατί έχω λίγα hashtags/handles και θα διασκορπιστούν σε διάφορα clusters και δεν θα έχω ένα καλό αποτέλεσμα. Οπότε θα διαλέξω να τρέχω τον k-means αλγόριθμο για αριθμό clusters ίσο με το 12.

Τρέχω τον k-means για 12 clusters και έχω τα εξής αποτελέσματα,

```
The total error of the clustering is:  6123.25005415
Cluster labels: [ 1  6 11 ...,  6  6  6]
Cluster Centroids:
[[ -8.67361738e-19   2.16840434e-19  -1.73472348e-18 ...,  -4.33680869e-19
    4.33680869e-19  -2.16840434e-19]
 [  1.64373167e-02   1.40946282e-18   2.57254579e-05 ...,   1.73472348e-18
    7.23975806e-03   2.62690908e-03]
 [  0.00000000e+00   0.00000000e+00   0.00000000e+00 ...,   0.00000000e+00
    0.00000000e+00   2.50000000e-01]
 ...,
 [  5.20417043e-18   2.58965199e-03   7.98425167e-02 ...,   2.16840434e-19
    6.72246057e-04   6.58472344e-04]
 [  0.00000000e+00   0.00000000e+00   0.00000000e+00 ...,   0.00000000e+00
    0.00000000e+00   0.00000000e+00]
 [  4.62962963e-03   3.56125356e-04   3.46944695e-18 ...,   2.08083803e-02
    3.56125356e-04   4.33680869e-19]]
```

Θα φτιάξω τόσα αρχεία όσα είναι τα clusters και μέσα θα βάλω τα hashtags/handles που ανήκουν σε κάθε ένα cluster,

Label 1 :

@AriMelber	@sarahkendzior	#StrongerTogether	@IMPL0RABLE	@shannoncoulter
@kylegriffin1	@coton_luver	@damonbethea1	@Maggyw519	@PoliticusSarah
@bannerite	@WeNeedHillary	@activist360	#GrabYourWallet	
@mmpadellan	@igorbobic	#LoveTrumpsHate	@kharyp	

Label 2:

#Haiti	#Wikileaks	#DonaldTrump	#FBIReopensCase	#ClintonEmails	#TrumpTrain
#Election2016	#JamesComey	#FBI	#election	#voting	#Facebook
#HillaryClinton	#America	#Israel	#DOJ	#USA	#racism
#Obamacare	#WakeUpAmerica	#Trump	#Election	#ImWithHer	
#truth	#Americans	@YouTube	#Hillary	#POTUS	
#DNC	#Clintons	#Rigged	#Catholic	#MSM	

Label 3: Έχω μόνο ένα handle οπότε δεν μπορώ να πω ότι το cluster αυτό έχει κάποιο θέμα

@CLewandowski_

Label 4: Υπάρχουν διάφορα ονόματα εφημερίδων, σταθμών ειδήσεων

@NBCNews	@ABC	@FoxNews	@HillaryClinton	@GeraldoRivera	@CNN

@WSJ	@MichaelCohen212	@DiamondandSilk	@LouDobbs	@ABCPolitics	@wikileaks
@donnabrazile	@megynkelly	@larryelder	@DanScavino	@MMFlint	@foxnewspolitics
@DanaPerino	@realDonaldTrump	@newtgingrich	@seanhannity	@nytimes	@oreillyfactor
@LindaSuhler	@foxandfriends	@GovMikeHuckabee	@marthamaccallum	@greggutfeld	@ThisWeekABC

Label 5: Δεν υπάρχει κάποιο συγκεκριμένο θέμα, σε αυτήν την ομάδα υπάρχουν αρκετά handles από τα οποία δεν μπορούμε να βγάλουμε κάποιο συμπέρασμα

@LifeZette	@Freedom_Daily	#8221	#8216	@amlookout	@Doug_Giles
@AllenWest	@worldnetdaily	@dailycaller	@DCExaminer	@KevinJacksonTBS	@youngcons
@wordpressdotcom	#8220	#TTT16	@100percFEDUP	@BarracudaMama	
@ConstitutionNat	#8230	@https	@DailySignal	@thelastrefuge2	
@realalexjones	#8217	@OnlinePatriots	@AllenWestRepub	@Patriotic_Folks	

Label 6: Και εδώ το ίδιο υπάρχουν μόνο handles όπου δεν βγαίνει κάποιο θέμα

@washingtonpost	@wpjenna	@davidaxelrod	@mckaycoppins	@DailyNewsBin	@NormOrnstein
@thehill	@mcuban	@rubycramer	@realDenaldTrump	@ThePlumLineGS	@pattonoswalt
@FBI	@jonfavs	@KevinDarryl	@RadioFreeTom	@ZekeJMiller	@JRubinBlogger
@politico	@JeffersonObama	@Lee_in_Iowa	@HeerJeet	@kairyssdal	@Olivianuzzi
@guardian	@coopah	@KeithOlbermann	@WalshFreedom	@ktumulty	@Variety

Label 7: Αυτή η ομάδα φαίνεται να περιέχει χώρες και διάφορες περιοχές

#ClimateChange	#corruption	#Aleppo	#Periscope	#TrumpPence	#halloween
#Russia	#Syria	#Iran	#Florida	#Halloween	#Turkey
#Republicans	#Italy	#NoDAPL	#PodestaEmails	#ISIS	#politics
#healthcare	#China	#GOP	#Putin	#DAPL	
#Trumps	#Nevada	#US	#election2016	#economy	

Label 8:

@Reuters	@crooksandliars	@politicususa	@law_newz	@newyorker	@LibAmericaOrg
@thedailybeast	@MotherJones	@Salon	@nbcnews	@alternet	@FortuneMagazine
@HuffPostPol	@christinawilkie	@UpshotNYT	@forbes	@buzzfeednews	@kira_lerner
@intelligencer	@TPM	@milesjreed	@BillMoyersHQ	@Ireland0828	@Bipartisan
@CNNMoney	@voxdotcom	@HuffPostBlog	@dailynewsbin	@HuffPostMedia	
@BorowitzReport	@politicalwire	@Shareaholic	@slate	#imwithher2016	

Label 9:

@HuffingtonPost	@TomCottonAR	@katiepack	@samsteinhp	@DavidShuster
@rickklein	@KatyTurNBC	@WSJPolitics	@JoshuaGreen	@jeffpeguescbs
@dailykos	@paulkrugman	@brianefallon	@aterkel	@NickMerrill
@PostRoz	@chucktodd	@KGBVeteran	@kasie	@KassyDillon

Label 10:

@ThePatriot143	@PatVPeters	@PatriotByGod	@LeahR77	@Girls4urtrump	@RedNationRising
@Lrihendry	@ChristieC733	@hrtablaze	@JrcheneyJohn	@CarmineZozzora	@10thAmendment
@mitchellvii	@surfermom77	@bfraser747	@TruthFeedNews	@DrMartyFox	@Mathiasian
@slone	@magnifier661	@ConstanceQueen8	@PolitixGal	@RealJamesWoods	
@tamaraleighllc	@GaetaSusan	@ObamaMalik	@WDFx2EU7	@SandraTXAS	

Label 11:

@11thHour

Label 12:

@TIME	@Lawrence	@SopanDeb	@hardball	@ddale8	@MarkHalperin
@MSNBC	@Fahrenthold	@howardfineman	@HardballChris	@mitchellreports	@BuzzFeedBen
@jasoninthehouse	@chrislhayes	@alivitali	@amjoyshow	@maddow	@CandaceSmith_
@kurteichenwald	@KFILE	@NYTnickc	@Morning_Joe	@jimsciutto	@allinwithchris
@JoyAnnReid	@maggieNYT	@DavidCornDC	@OutFrontCNN	@TheLastWord	@tonyschwartz

Agglomerative

Χρησιμοποίησα την κλάση sklearn.cluster **AgglomerativeClustering**(*n_clusters=2, affinity='euclidean', memory=Memory(cachedir=None), connectivity=None, compute_full_tree='auto', linkage='ward', pooling_func=<function mean>*) αυτή είναι η γενική της μορφή, για linkage έβαλα το complete και χρησιμοποίησα την cosine απόσταση.

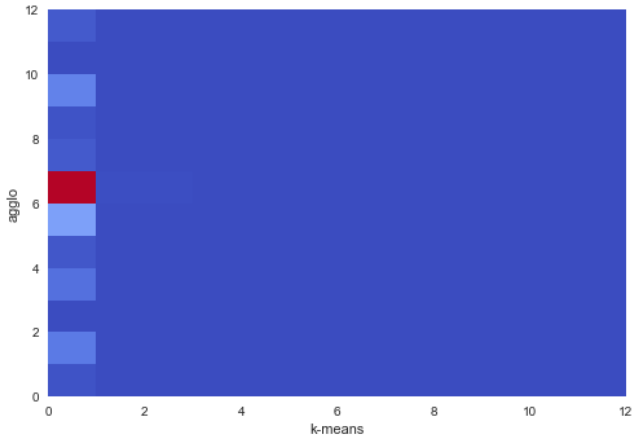
DBSCAN

Χρησιμοποίησα ξανά την κλάση sklearn.cluster και τον DBSCAN(*eps=0.5, min_samples=5, metric='euclidean', algorithm='auto', leaf_size=30, p=None, n_jobs=1*) αυτή είναι η γενική του μορφή. Τον έτρεξα για “eps=0.3” δηλαδή η maximum απόσταση μεταξύ 2 δειγμάτων ώστε να θεωρηθούν γειτονιά τα υπόλοιπα πεδία δεν τα συμπεριέλαβα και .

Confusion matrix

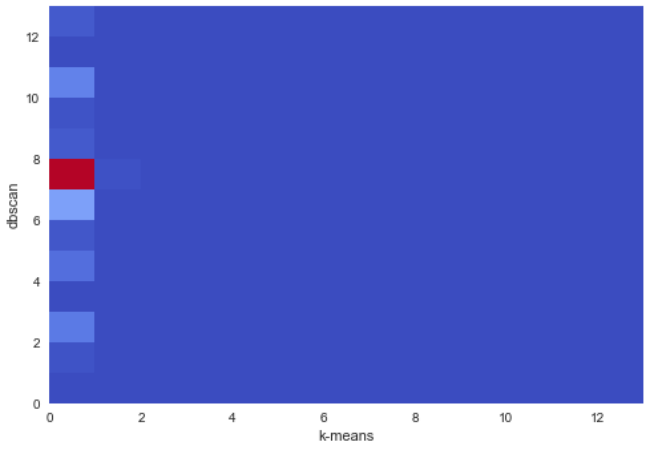
k-means vs agglomerative

```
[[  17    0    1    0    0    0    0    0    0    0    0    0]
 [ 112    0    0    0    1    0    0    0    0    0    0    0]
 [   0    0    1    0    0    0    0    0    0    0    0    0]
 [  87    0    0    0    0    0    0    0    0    0    0    0]
 [  26    0    0    1    0    0    0    0    0    0    0    0]
 [ 215    3    0    0    0    0    0    0    0    0    0    0]
 [1053    5    5    3    4    4    2    0    0    0    0    0]
 [  34    0    0    0    0    0    0    0    0    0    0    0]
 [  20    0    0    0    0    0    0    0    0    0    0    0]
 [ 134    0    0    0    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0    0    0    0]
 [  36    0    0    0    0    0    0    0    0    0    0    0]]
```



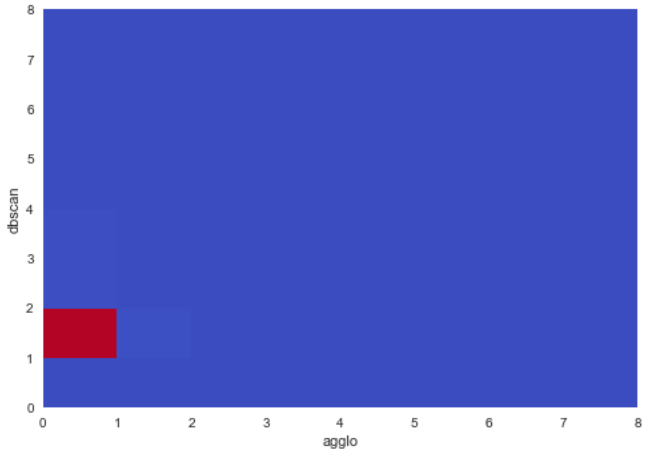
k-means vs dbscan

```
[[   0    0    0    0    0    0    0    0    0    0    0    0    0]
 [  18    0    0    0    0    0    0    0    0    0    0    0    0]
 [ 113    0    0    0    0    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0    0    0    0    0]
 [  87    0    0    0    0    0    0    0    0    0    0    0    0]
 [  27    0    0    0    0    0    0    0    0    0    0    0    0]
 [ 218    0    0    0    0    0    0    0    0    0    0    0    0]
 [1062   14    0    0    0    0    0    0    0    0    0    0    0]
 [  34    0    0    0    0    0    0    0    0    0    0    0    0]
 [  20    0    0    0    0    0    0    0    0    0    0    0    0]
 [ 134    0    0    0    0    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0    0    0    0    0]
 [  36    0    0    0    0    0    0    0    0    0    0    0    0]]
```



agglomerative vs dbscan

```
[[   0    0    0    0    0    0    0    0]
 [1721   14    0    0    0    0    0    0]
 [   8    0    0    0    0    0    0    0]
 [   7    0    0    0    0    0    0    0]
 [   4    0    0    0    0    0    0    0]
 [   5    0    0    0    0    0    0    0]
 [   4    0    0    0    0    0    0    0]
 [   2    0    0    0    0    0    0    0]]
```



2ο πρόβλημα

Θέλω να αναπαραστήσω τώρα σαν διανύσματα τους χρήστες, οπότε είναι ο ανάστροφος πίνακα από αυτόν των hashtags/handles. Στο "***userVector***" κρατάω τα διανύσματα των χρηστών.

```
[[ 0.25        0.01818182  0.16666667 ...,  0.          0.          0.        ]
 [ 0.          0.          0.         ...,  0.          0.          0.        ]
 [ 0.          0.          0.         ...,  0.          0.          0.        ]
 ...,
 [ 0.          0.          0.         ...,  0.          0.          0.        ]
 [ 0.          0.          0.         ...,  0.          0.          0.        ]
 [ 0.          0.          0.         ...,  0.          0.          0.        ]]
```

Στο αρχείο "***clinton_trump_user_classes.txt***" είναι αποθηκευμένοι οι χρήστες με τα id τους. Εγώ είχα χρησιμοποιήσει τους χρήστες με τα username τους, οπότε πρέπει να κάνω μία αντιστοιχία τα username με τα id τους. Αυτήν την αντιστοιχία την κρατάω στο λεξικό με όνομα "***ids = {}***" και η μορφή του είναι,

robert2266 : 17101060
judypgpd : 1247276484
CaptainNormal : 2447279666
gregster212 : 24215368
AnnCali : 172469352
UnBitterEnd2013 : 1567006406
watersurf58 : 154236193
Jodyjtaylor : 1480641168

Διαβάζω το αρχείο "***clinton_trump_user_classes.txt***" γραμμή γραμμή και κρατάω μέσα στις λίστες με ονόματα "***trumpFollowers = []***" και "***clintonFollowers = []***" τους ακόλουθους του trump και clinton

αντίστοιχα με τα username τους. Βρίσκω, ότι από τους χρήστες που έχω κρατήσει όλοι είναι ακόλουθοι τις clinton,
Trump followers: 0
Clinton followers: 3108

Αλγόριθμοι clustering

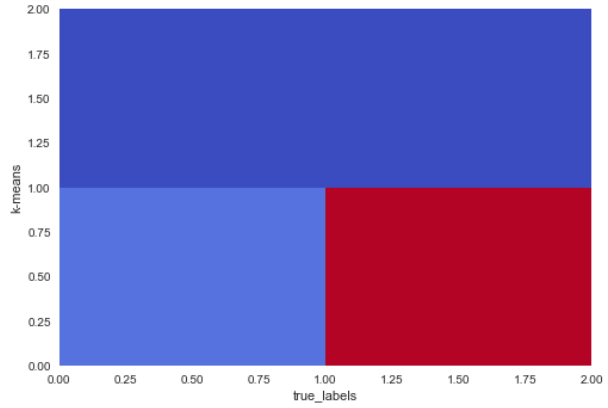
k-means

Τρέχω τον k-means αλγόριθμο για 2 clusters και βρίσκω,

The total error of the clustering is: 6358.29231655
Cluster labels: [1 1 1 ..., 1 1 1]
Cluster Centroids
[[0.00100806 0.00703812 0.03293011 ..., 0.00705645 0.00117066
0.00201613]
[0.00332168 0.00398601 0.00681818 ..., 0.00091783 0.00074442
0.00262238]]

Σύγκριση με τα πραγματικά δεδομένα

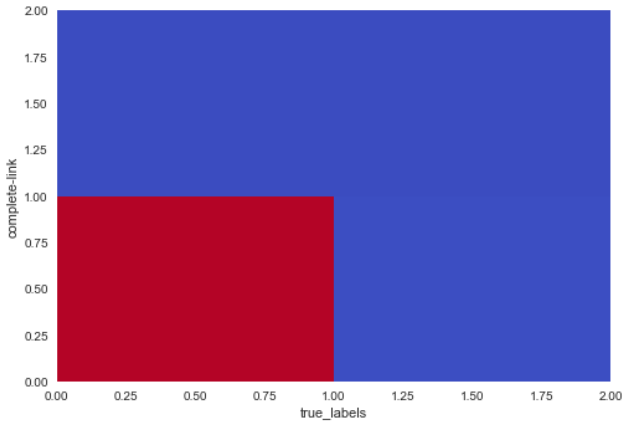
```
[[ 248 2860]
 [   0    0]]

PRECISION: [ 1.  0.]
RECALL: [ 0.07979408  0.        ]
F_MEASURE: [ 0.14779499  0.        ]
```



Agglomerative → complete-link

Σύγκριση με τα πραγματικά δεδομένα

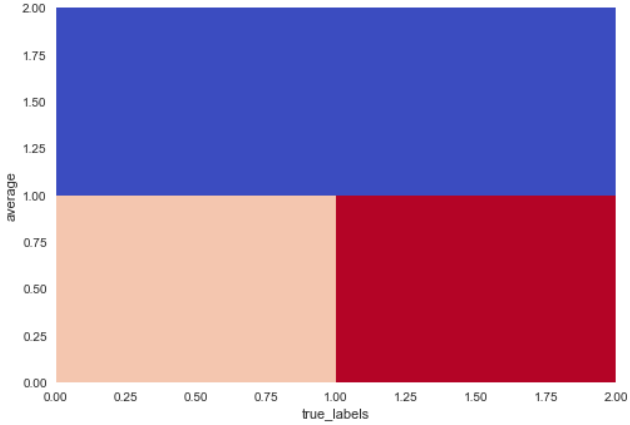
```
[[3089   19]
 [   0    0]]

PRECISION: [ 1.  0.]
RECALL: [ 0.99388674  0.        ]
F_MEASURE: [ 0.996934  0.      ]
```



Agglomerative → average

Σύγκριση με τα πραγματικά δεδομένα

```
[[1190 1918]
 [   0    0]]
```

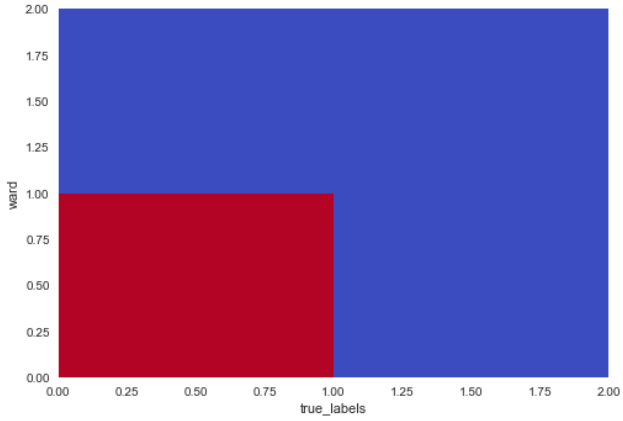


```
PRECISION: [ 1.  0.]
RECALL: [ 0.38288288  0.        ]
F_MEASURE: [ 0.55374593  0.        ]
```

Agglomerative → ward

Σύγκριση με τα πραγματικά δεδομένα

```
[[3106    2]
 [   0    0]]

PRECISION: [ 1.  0.]
RECALL: [ 0.9993565  0.       ]
F_MEASURE: [ 0.99967815  0.        ]
```



Agglomerative → single-link

Σύγκριση με τα πραγματικά δεδομένα

Η κλάση *sklearn.cluster* δεν περιέχει σαν *linkage = single* οπότε χρησιμοποίησα την βιβλιοθήκη *scipy.* Με την εντολή *sp.cluster.hierarchy.linkage(userVector, method='single', metric='cosine')* μου επιστρέφεται ένας πίνακας με την ιεραρχική συσσώρευση κωδικοποιημένη ως μήτρα σύνδεσης.

```
(3107, 4)
[[  3.96000000e+02   5.15000000e+02   3.35899173e-02   2.00000000e+00]
 [  5.30000000e+01   3.10800000e+03   4.95763033e-02   3.00000000e+00]
 [  3.20000000e+01   2.67000000e+02   9.39347561e-02   2.00000000e+00]
 ...,
 [  1.69600000e+03   6.21100000e+03   8.33500633e-01   3.10600000e+03]
 [  1.39400000e+03   6.21200000e+03   8.38277144e-01   3.10700000e+03]
 [  2.09900000e+03   6.21300000e+03   8.43741803e-01   3.10800000e+03]]
```

Δεν μπορώ να βρω πως παίρνω τις ετικέτες για κάθε έναν χρήστη για το σε ποια κλάση ανήκει οπότε δεν έχω κάποια αποτελέσματα να σας δείξω.

Τέλος, έχω 248 χρήστες που ανήκουν στην μία ομάδα(ομάδα Trump) και οι υπόλοιποι στην άλλοι, οι 248 χρήστες είναι του Trump, μερικοί από αυτούς τους χρήστες είναι οι ακόλουθοι,
['DonKeehotey', 'd_twit', 'AttyLeBlanc', 'a_degeatano', 'WholePlateWay', 'laura19191', 'scottyhasty', 'csilberman70', 'YellyYellerson', 'aliceinthewater']

Τα **30 hashtags/handles με τις μεγαλύτερες τιμές** για αυτήν την ομάδα είναι τα,
[('@realDonaldTrump', 3667), ('@HillaryClinton', 2865), ('@CNN', 1411), ('@FoxNews', 1219), ('@MSNBC', 1016), ('@YouTube', 1010), ('@JoyAnnReid', 928), ('@FBI', 839), ('@kurteichenwald', 809), ('@washingtonpost', 808),

('@jasoninthehouse', 798), ('@KellyannePolls', 743), ('@CNNPolitics', 705), ('@ABC', 699), ('@nytimes', 638), ('@wikileaks', 627), ('@megynkelly', 625), ('@thehill', 546), ('@GOP', 475), ('@POTUS', 475), ('@jaketapper', 462), ('#Trump', 452), ('@NBCNews', 425), ('@newtgingrich', 409), ('@CBSNews', 409), ('@politico', 406), ('@seanhannity', 391), ('@SpeakerRyan', 365), ('@SopanDeb', 355), ('#ImWithHer', 324)]

Για την άλλην ομάδα(ομάδα Clinton) κάποιοι από τους χρήστες της είναι οι,
['robert2266', 'CaptainNormal', 'UnBitterEnd2013', 'Jodyjtaylor', 'arjanomics', 'NHKeith', 'findit89', '2miche', 'jcjet5', 'brokenwing2005']

Και τα **30 καλύτερα hashtags/handles** είναι τα,
[('@realDonaldTrump', 14466), ('@HillaryClinton', 11083), ('@FoxNews', 6325), ('@YouTube', 4336), ('@megynkelly', 4222), ('#Trump', 4162), ('@CNN', 3720), ('#MAGA', 3607), ('#Hillary', 2869), ('@seanhannity', 2556), ('@newtgingrich', 2314), ('@wikileaks', 2168), ('@FBI', 2099), ('@KellyannePolls', 2058), ('#CrookedHillary', 1765), ('#TRUMP', 1641), ('@MSNBC', 1595), ('@POTUS', 1590), ('#DrainTheSwamp', 1548), ('#Clinton', 1539), ('@mitchellvii', 1460), ('@WDFx2EU7', 1437), ('@washingtonpost', 1386), ('@nytimes', 1360), ('#HillaryClinton', 1356), ('#ImWithHer', 1347), ('@ABC', 1294), ('@jasoninthehouse', 1274), ('@BreitbartNews', 1209), ('#TrumpPence16', 1167)]

Τώρα, ανάμεσα σε αυτά τα 2 clusters δεν φαίνεται ανάμεσα στα 30 hashtags/handles με τις μεγαλύτερες τιμές να υπάρχει κάποια διαφορά γιατί είναι σχεδόν όλα όμοια.
