

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ (ΗΜΜΥ)

Τεχνικές Βελτίωσης της Android Εφαρμογής Κοινωνικής Δικτύωσης e-Match

*Techniques for the Improvement of the e-Match Social Networking Android
Application*

Γεώργιος Τσακαλάκης

Επιτροπή

Κουτσάκης Πολυχρόνης (Αναπληρωτής Καθηγητής - Επιβλέπων)

Πατεράκης Μιχάλης (Καθηγητής)

Δελγιαννάκης Αντώνιος (Αναπληρωτής Καθηγητής)

ΠΕΡΙΛΗΨΗ

Η διπλωματική αυτή αφορά στο E-Match, μια νέα Android εφαρμογή για την εύρεση πιθανών φίλων (ανθρώπων με κοινά ενδιαφέροντα) στην περιοχή όπου βρίσκεται ο χρήστης. Αποτελεί συνέχεια της διπλωματικής της Γεωργίας Αθανασοπούλου, η οποία στην διπλωματική της είχε αναφερθεί εκτενώς στις λειτουργικές απαιτήσεις του συστήματος, στις περιπτώσεις χρήσης (use cases), στη βάση δεδομένων, στις αρχιτεκτονικές σχεδιασμού και στα εργαλεία, στη σχεδίαση διεπαφών χρήστη και στο matching (ταίριασμα των χρηστών της εφαρμογής). Σε αυτή τη διπλωματική γίνεται μια συνοπτική αναφορά στην ίδια την εφαρμογή και στο πώς σχεδιάστηκε και εστιάζουμε περισσότερο στα στοιχεία από τα νέα ερωτηματολόγια που συλλέχθηκαν και στο αλγοριθμικό κομμάτι του matching (του ταιριάσματος των χρηστών της εφαρμογής), στο οποίο έγιναν βελτιώσεις ούτως ώστε ο στόχος της εφαρμογής που είναι η εξεύρεση φίλων να γίνεται με μεγαλύτερη ακρίβεια και με ανάλυση στοιχείων σχετικών με την ηλικία και το φύλο των χρηστών. Κατόπιν, γίνεται λεπτομερής αναφορά στα αποτελέσματα που προέκυψαν από τα παραπάνω.

Στο πρώτο κομμάτι της διπλωματικής εργασίας, ο στόχος ήταν η επεξεργασία μεγάλου αριθμού νέων ερωτηματολογίων που συγκεντρώσαμε, και εξαγωγή αποτελεσμάτων για τους νέους χρήστες της εφαρμογής. Συγκεντρώσαμε επιπλέον 229 ερωτηματολόγια σε σχέση με την προηγούμενη διπλωματική, οπότε συνολικά έχουμε 286 συμπληρωμένα ερωτηματολόγια. Στη συνέχεια, ελέγξαμε την ισχύ των αποτελεσμάτων που είχαν εξαχθεί στην προηγούμενη διπλωματική για μικρό αριθμό χρηστών και εξερευνήσαμε νέες μετρικές και τεχνικές βελτίωσης του υπάρχοντος αλγορίθμου, με στόχο την βελτίωση του αλγορίθμου EgoSimilar του E-Match σε συγκεκριμένα σημεία. Τα αποτελέσματά μας επιβεβαιώνουν ότι ο EgoSimilar υπερτερεί σε σχέση με πλήθος άλλων μετρικών από την βιβλιογραφία.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή.....	4
2. Matching.....	5
2.1 EgoSimilar.....	5
2.2 Pearson Correlation.....	7
2.3 Cosine Similarity.....	7
2.4 Kappa Coefficient.....	8
2.5 Pi Coefficient.....	8
2.6 Jaccard Index.....	8
3. Αποτελέσματα.....	9
3.1 Σύγκριση Συντελεστών Ομοιότητας.....	9
3.2 Cluster Evaluation για $w_1 = 0.25$, $w_2 = 0.75$	12
3.3 Αλλαγή των βαρών w_1 και w_2	21
3.3.1 Evaluation για $w_1 = 0.5$, $w_2 = 0.5$	21
3.3.2 Evaluation για $w_1 = 0.75$, $w_2 = 0.25$	23
3.4 Matching βάσει των 2 πιο "αγαπημένων" κατηγοριών.....	25
3.5 Κατηγοριοποίηση χρηστών βάσει ηλικίας.....	26
3.6 Κατηγοριοποίηση χρηστών βάσει φύλου.....	30
4. Παρόμοιες Εφαρμογές και Συμπεράσματα.....	34
5. References.....	35

1. ΕΙΣΑΓΩΓΗ

Η εφαρμογή αυτή μπορεί να χρησιμοποιηθεί από χρήστες οι οποίοι έχουν στην κατοχή τους ένα Smartphone(Android). Για να εγκαταστήσει κάποιος χρήστης την εφαρμογή αυτή μπορεί να την βρει από το [1] και στην συνέχεια να ακολουθήσει την απαραίτητη διαδικασία εγκατάστασης εφαρμογών για Android.

Συνοπτικά, η δομή του συστήματος περιλαμβάνει τα εξής:

1. Τη βάση δεδομένων.
2. Την λογική στον server, δηλαδή ο κώδικας (Java) που τρέχει στον server.
3. Τη λογική της εφαρμογής, δηλαδή ο κώδικας (Java) που τρέχει στο κινητό.
4. Τις δυναμικές διεπαφές χρήστη, μέσω των οποίων όλες οι διαθέσιμες λειτουργίες του συστήματος θα παρέχονται προς τους τελικούς χρήστες.

Λεπτομέρειες για τις λειτουργικές απαιτήσεις του συστήματος, τις περιπτώσεις χρήσης, την βάση δεδομένων, τις αρχιτεκτονικές σχεδιασμού και τα εργαλεία της εφαρμογής και της σχεδίασης διεπαφών χρήστη υπάρχουν στην διπλωματική εργασία της Γεωργίας Αθανασοπούλου στο [2] και στο [3].

Τέλος, να σημειωθεί ότι η επικοινωνία με τον χρήστη μέσω των διεπαφών γίνεται στα Αγγλικά.

2. MATCHING

Ο αλγόριθμος που προτείνουμε είναι ο EgoSimilar, ο οποίος θα αναλυθεί παρακάτω και θα συγκριθεί με άλλες πέντε γνωστές μεθόδους ομοιότητας. Αυτές οι πέντε μέθοδοι είναι η μετρική Cosine (Cosine Measure), η συσχέτιση Pearson (Pearson Correlation) [4] και οι δείκτες Jaccard [5], Pi και Kappa [6]. Από τους τρεις τελευταίους δείκτες είχαμε καλύτερα και πιο ενδιαφέροντα αποτελέσματα από τον δείκτη Kappa, τον οποίο και εντέλει χρησιμοποιήσαμε για τις απαραίτητες συγκρίσεις και για τον οποίο θα κάνουμε εκτενέστερη αναφορά.

2.1. EgoSimilar

Ο αλγόριθμος EgoSimilar κάνει το ταίριασμα μεταξύ των χρηστών με "εγωκεντρικό τρόπο", επειδή κάθε χρήστης πρέπει να αναζητά φίλους με βάση τα δικά του κριτήρια και ενδιαφέροντα. Έτσι, το ποσοστό ταιριάσματος μεταξύ δύο χρηστών που θα εμφανιστεί στην οθόνη του κάθε χρήστη θα είναι πιθανότατα διαφορετικό. Αυτό μας φαίνεται λογικό, δεδομένου ότι στην πραγματική ζωή λειτουργούμε με τον ίδιο τρόπο, επιλέγοντας τους ανθρώπους γύρω μας βασισμένοι κυρίως στις ανάγκες και τα ενδιαφέροντα μας. Συνεπώς, αν για παράδειγμα ο χρήστης X έχει μία ενεργή κατηγορία ενδιαφέροντος ενώ ο χρήστης Y έχει πέντε, το ποσοστό αντιστοίχισης (X, Y) θα βασίζεται σε αυτή τη μία κατηγορία, ενώ το ποσοστό αντιστοίχισης (Y, X) και στις πέντε και έτσι τα αποτελέσματα που εμφανίζονται στην οθόνη του κάθε χρήστη είναι διαφορετικά. Κάθε χρήστης έχει τη δυνατότητα βαθμολόγησης μεταξύ 9 κατηγοριών: Movies, Music, Books, Games, Sports, Science, Shopping, Food, Travel. Έχει ακόμα τη δυνατότητα να βαθμολογήσει τα αντικείμενα που περιέχονται σε κάθε κατηγορία (και αν θέλει να προσθέσει και να βαθμολογήσει κι άλλα). Τα δημοφιλέστερα αντικείμενα (δημοφιλή με την έννοια ότι έχουν βαθμολογηθεί θετικά ή αρνητικά από πολλούς χρήστες) δεν θα πρέπει να επηρεάζουν τα αποτελέσματα ταιριάσματος όσο τα επηρεάζουν τα λιγότερο δημοφιλή αντικείμενα, εάν οι χρήστες "συμφωνούν" σε αυτά. Ο λόγος είναι ότι ακόμη και αν οι χρήστες μοιράζονται μια θετική γνώμη για μια πολύ γνωστή μπάντα, βιβλίο, ταινία κ.λπ., αυτό δεν δίνει μια ουσιαστική υπόδειξη ότι τα γούστα τους ταιριάζουν γενικά. Μια παρόμοια περίπτωση με μια σχετικά άγνωστη ταινία / βιβλίο δίνει μια πολύ ισχυρότερη ένδειξη κοινών ενδιαφερόντων.

Οι επιλογές αξιολόγησης των χρηστών κυμαίνονται στην κλίμακα από 1 έως 10. Συνεπώς η μέγιστη διαφορά βαθμολογίας θα είναι 9 και το βάρος μιας μονάδας στη διαφορά βαθμολογίας θα είναι $1/9 \approx 0.11$. Αυτό το βάρος πρέπει να περιλαμβάνεται, κατά την άποψή μας, στον υπολογισμό της ομοιότητας μεταξύ των χρηστών. Τα βήματα που ακολουθεί το eMatch για τον υπολογισμό της αντιστοίχισης μεταξύ χρηστών περιγράφονται παρακάτω. Τα πρώτα τρία βήματα ακολουθούνται ανεξάρτητα από την μέθοδο υπολογισμού ομοιότητας, η οποία εφαρμόζεται στο βήμα 4. Έστω X ο χρήστης που τρέχει την εφαρμογή, επομένως, η αντιστοίχιση (το ταίριασμα) γίνεται σύμφωνα με τα γούστα του X:

- 1) Έλεγχε αν η θέση του χρήστη είναι αποθηκευμένη. Αν όχι, ενημέρωσε το χρήστη, αλλιώς πήγαινε στο επόμενο βήμα.
- 2) Βρες χρήστες που βρίσκονται σε κοντινή γεωγραφική περιοχή με τον χρήστη X.
- 3) Βρες όλες τις ενεργές κατηγορίες ενδιαφέροντος του χρήστη X.
- 4) Το ταίριασμα στον EgoSimilar υπολογίζεται ως εξής: για κάθε χρήστη Y που βρίσκεται στο βήμα 2, υπολόγισε το Matching (X, Y) ως εξής:

$$\frac{1}{k_x} \sum_{c=1}^{k_x} \left[w_1 [1 - 0.11 d_1(X, Y, c)] + \frac{w_2}{N_x^c} \sum_{i=1}^{N_x^c} [1 - 0.11 d_2(X, Y, c, i)] \right] \quad (1)$$

Όπου:

k_x είναι ο αριθμός των ενεργών κατηγοριών του χρήστη X , $k_x \in [1, 9]$.

W_1 είναι το βάρος που αποδίδεται στη γενική βαθμολογία μιας κατηγορίας. Στην περίπτωση μας, $w_1 = 0.25$, καθώς θεωρούμε ότι το "γενικό" ταίριασμα των χρηστών (π.χ. και οι δύο αγαπούν τις ταινίες), είναι μικρότερης σημασίας, καθώς τα συγκεκριμένα γούστα τους σε αυτή την κατηγορία μπορεί να διαφέρουν σημαντικά ή τελείως.

W_2 είναι το βάρος των αξιολογήσεων όλων των επιμέρους αντικειμένων μιας κατηγορίας (στην περίπτωση μας, $w_2 = 0.75$). Τα w_1, w_2 πρέπει να αθροίζουν στον άσο. Στη συνέχεια, έγιναν πειράματα και για $W_1 = 0.5, W_2 = 0.5$ και για $W_1 = 0.75, W_2 = 0.25$.

n_x^c είναι ο αριθμός των στοιχείων που ο χρήστης X έχει εισαγάγει στην c -οστή κατηγορία.

Το $d1(X, Y, c)$ είναι μια συνάρτηση που υπολογίζει την απόλυτη διαφορά στις βαθμολογίες μεταξύ των χρηστών X και Y για την ενεργοποιημένη c -οστή κατηγορία του χρήστη X . Αν ο χρήστης Y έχει απενεργοποιήσει τη συγκεκριμένη κατηγορία, τότε το $(1 - 0.11 \cdot d1(X, Y, c))$ ισούται με το μηδέν.

Το $d2(X, Y, c, i)$ συνδέεται με το i -οστό στοιχείο που εισάγεται από το χρήστη X στην ενεργοποιημένη c -οστή κατηγορία και υποδηλώνει την απόλυτη διαφορά των αξιολογήσεων μεταξύ των χρηστών X και Y για το συγκεκριμένο στοιχείο. Ορίζουμε $(1 - 0.11 \cdot d2(X, Y, c, i))$ ίσο με το μηδέν αν ο χρήστης Y δεν έχει βαθμολογήσει αυτό το στοιχείο, διαφορετικά το $d2(X, Y, c, i)$ υπολογίζεται λαμβάνοντας υπόψη την δημοτικότητα του συγκεκριμένου αντικειμένου ως εξής:

α) Αρχικοποίησε το $d2(X, Y, c, i)$ ως την απόλυτη διαφορά στις βαθμολογίες μεταξύ χρηστών X και Y για αυτό το στοιχείο.

β) Εστω m ο αριθμός των χρηστών που έχουν εισαγάγει αυτό το στοιχείο και n ο αριθμός των χρηστών που έχουν εισάγει στοιχεία στην ενεργοποιημένη c -οστή κατηγορία του χρήστη X . Το βάρος του πόσο δημοφιλές είναι το συγκεκριμένο αντικείμενο ορίζεται ως: $W = m / n$. Ένα στοιχείο θεωρείται ότι είναι δημοφιλές αν $W > 0.5$, πράγμα που σημαίνει ότι περισσότεροι από τους μισούς χρήστες που "ψήφισαν" για αυτήν την κατηγορία έχουν εισαγάγει το συγκεκριμένο στοιχείο (είτε ψήφισαν αρνητικά είτε θετικά).

γ) Το $d2(X, Y, c, i)$ προσαρμόζεται σε σχέση με τη δημοτικότητα του αντικειμένου και το σκεπτικό που εξηγείται παραπάνω ως εξής:

Αν $(W > 0.5$ και $d2(X, Y, c, i) < 5)$ τότε

$$d2(X, Y, c, i) = d2(X, Y, c, i) + W \cdot d2(X, Y, c, i)$$

Αυτό δηλώνει ότι δεδομένου ότι αυτό το στοιχείο είναι δημοφιλές και οι αξιολογήσεις των χρηστών είναι κοντά, τότε αυτό το στοιχείο δεν θα πρέπει να επηρεάσει τα αποτελέσματα αντιστοίχισης όσο τα επηρεάζουν τα λιγότερο δημοφιλέστερα στοιχεία. Επομένως, η απόσταση των βαθμολογήσεων μεταξύ των χρηστών X και Y πρέπει να αυξηθεί προκειμένου να μειωθεί το ταίριασμα. Η αύξηση αυτή πραγματοποιείται μέσω του βάρους W_{change} .

Αν $(W > 0.5$ και $d2(X, Y, c, i) \geq 5)$ τότε το $d2(X, Y, c, i)$ δεν αλλάζει.

Αν $(W < 0.5$ και $d2(X, Y, c, i) < 5)$ τότε

$$d2(X, Y, c, i) = d2(X, Y, c, i) - W \cdot d2(X, Y, c, i)$$

Αυτό δείχνει ότι, δεδομένου ότι αυτό το στοιχείο δεν είναι δημοφιλές και οι αξιολογήσεις των χρηστών είναι κοντά, τότε αυτό το στοιχείο θα πρέπει να επηρεάζει τα αποτελέσματα ταιριάσματος περισσότερο από ότι τα επηρεάζουν τα δημοφιλή αντικείμενα. Συνεπώς, η απόσταση των αξιολογήσεων μεταξύ των χρηστών X και Y πρέπει να μειωθεί προκειμένου να αυξηθεί το ταίριασμα. Αυτό γίνεται και πάλι μέσω του W_{change} .

Αν $(W \leq 0.5$ και $d2(X, Y, c, i) \geq 5)$ τότε

$$d2(X, Y, c, i) = d2(X, Y, c, i) + W \cdot d2(X, Y, c, i)$$

Ομοίως, στην περίπτωση που το στοιχείο δεν είναι δημοφιλές και οι αξιολογήσεις των χρηστών δεν είναι κοντά, συμπεραίνουμε ότι αυτό αποτελεί ένδειξη χρηστών που δεν έχουν κοινά ενδιαφέροντα. Έτσι, αυξάνοντας την απόσταση των αξιολογήσεών τους, το ταίριασμα τους μειώνεται.

Η πολυπλοκότητα του αλγορίθμου είναι: $O(pqr)$, όπου p είναι ο αριθμός των χρηστών, q ο αριθμός των κατηγοριών (στην περίπτωση μας εννέα) και r ο μέγιστος αριθμός των αντικειμένων που έχουν εισαχθεί σε μία από τις κατηγορίες.

2.2 Pearson Correlation

Η δεύτερη προσέγγιση για τον υπολογισμό της ομοιότητας των χρηστών, η οποία εφαρμόστηκε και δοκιμάστηκε στο eMatch, είναι η συσχέτιση Pearson. Η συσχέτιση Pearson δεν λαμβάνει υπόψη τη δημοτικότητα του αντικειμένου, όπως κάνει ο EgoSimilar. Μια θετική συσχέτιση παρουσιάζει ομοιότητες μεταξύ των χρηστών, ενώ μια αρνητική τιμή δείχνει ότι αυτοί οι χρήστες δεν ταιριάζουν και τα ενδιαφέροντά τους είναι διαφορετικά. Παρακάτω περιγράφεται πώς υπολογίζεται η συσχέτιση Pearson: Έστω X και Y δύο χρήστες του eMatch. Τα βήματα που ακολουθήθηκαν για να υπολογιστεί το ποσοστό ταιριάσματος είναι:

1) Βρες όλες τις ενεργές κατηγορίες ενδιαφερόντων του χρήστη X και όλα τα αντικείμενα που έχει εισάγει ο X σε κάθε κατηγορία και αποθήκευσε τις αξιολογήσεις κάθε στοιχείου σε ένα διάνυσμα x (ένα στοιχείο είναι είτε μια κατηγορία είτε ένα αντικείμενο, ένα διαφορετικό διάνυσμα κατασκευάζεται για τις κατηγορίες και ένα άλλο για τα αντικείμενα).

2) Δημιούργησε ένα διάνυσμα y που έχει ίσο μέγεθος με το διάνυσμα x . Για κάθε στοιχείο του προηγούμενου βήματος έλεγξε αν ο χρήστης Y το έχει βαθμολογήσει. Αν ναι, στην αντίστοιχη θέση του διανύσματος y εκχωρείται η αξιολόγηση του χρήστη Y , διαφορετικά εκχωρείται η τιμή μηδέν.

3) Έστω n το μήκος των διανυσμάτων x , y και ότι δηλώνουμε με \bar{x} τη μέση τιμή του διανύσματος x και με \bar{y} τη μέση τιμή του διανύσματος y . Τότε το Matching (X, Y) υπολογίζεται ως:

$$\text{Matching}(X, Y) = \frac{\sum_{i=1}^n (x[i] - \bar{x})(y[i] - \bar{y})}{\sqrt{\sum_{i=1}^n (x[i] - \bar{x})^2 \sum_{j=1}^n (y[j] - \bar{y})^2}} \quad (2)$$

2.3 Cosine Similarity

Μια άλλη πολύ γνωστή μετρική ομοιότητας μεταξύ δύο διανυσμάτων είναι η μετρική Cosine. Τα βήματα που ακολουθούνται είναι ακριβώς τα ίδια, όπως στην περίπτωση της συσχέτισης Pearson για την κατασκευή των διανυσμάτων x , y . Στη συνέχεια, το Matching (X, Y) υπολογίζεται ως εξής:

$$\text{Matching}(X, Y) = \frac{\sum_{i=1}^n x[i] y[i]}{\sqrt{\sum_{i=1}^n x[i]^2 \sum_{j=1}^n y[j]^2}} \quad (3)$$

Ένα μειονέκτημα της χρήσης της Cosine στο eMatch είναι ότι τα συγγραμικά διανύσματα οδηγούν σε 100% ομοιότητα. Επομένως, αν οι αξιολογήσεις του χρήστη X είναι 1 και του Y 10 τότε το αποτέλεσμα ταιριάσματος είναι 100% το οποίο είναι ένα σημαντικό σφάλμα, αν και στην πραγματικότητα αυτό το σενάριο σπανίως συναντάται.

2.4 Kappa Coefficient

Ο δείκτης Kappa μετρά τη συμφωνία μεταξύ δύο αξιολογητών που ταξινομούν ο καθένας N αντικείμενα σε C αμοιβαία αποκλειστικές κατηγορίες. Ο ορισμός του Kappa είναι:

$$\kappa = (A_o - A_e) / (1 - A_e) = 1 - (1 - A_o) / (1 - A_e), \quad (4)$$

όπου A_o είναι η σχετική παρατηρούμενη συμφωνία μεταξύ των κριτών (ταυτόσημο με την ακρίβεια), και A_e είναι η υποθετική πιθανότητα τυχαίας συμφωνίας, που χρησιμοποιεί τα παρατηρούμενα δεδομένα για τον υπολογισμό των πιθανοτήτων κάθε παρατηρητή να δει τυχαία κάθε κατηγορία. Αν οι κριτές συμφωνούν απόλυτα, τότε $\kappa = 1$. Αν δεν υπάρχει συμφωνία μεταξύ των κριτών, εκτός από αυτό που θα αναμενόταν τυχαία (όπως δίνεται από το A_e), $\kappa \leq 0$.

2.5 Pi Coefficient

Ο δείκτης Pi είναι παρόμοιος με τον δείκτη Kappa, καθώς και οι δύο βελτιώνουν την απλή παρατηρούμενη συμφωνία(A_o), εκμεταλλευόμενοι την συμφωνία η οποία αναμένεται τυχαία(A_e). Ωστόσο, σε κάθε στατιστική, η αναμενόμενη συμφωνία(A_e) υπολογίζεται ελαφρώς διαφορετικά. Ο δείκτης Pi κάνει την υπόθεση ότι οι αξιολογητές έχουν την ίδια κατανομή των απαντήσεων, γεγονός που καθιστά τον δείκτη Kappa ελαφρώς πιο πληροφοριακό. Ο ορισμός του Pi είναι ίδιος με αυτόν του Kappa(4). Ωστόσο, στο Pi το A_e υπολογίζεται χρησιμοποιώντας κοινές αναλογίες. Και οι δύο συντελεστές ορίζουν την αναμενόμενη συμφωνία(A_e) σαν την πιθανότητα ότι οι δύο αξιολογητές θα κατηγοριοποιήσουν ένα αυθαίρετο αντικείμενο στην ίδια κατηγορία. Αλλά ενώ το Pi υποθέτει ότι αυτή η πιθανότητα διέπεται από μια απλή κατανομή, το Kappa υποθέτει ότι ο κάθε αξιολογητής έχει μια διαφορετική κατανομή πιθανότητας. Έτσι, στον δείκτη Kappa το A_e έχει ελαφρώς μικρότερη τιμή από ότι στον δείκτη Pi και αυτό έχει ως συνέπεια ο δείκτης Kappa να έχει ελαφρώς μεγαλύτερη τιμή από τον δείκτη Pi. Κάτι το οποίο επιβεβαιώνεται και στα αποτελέσματα που παρουσιάζονται παρακάτω και το οποίο μας οδήγησε να επιλέξουμε τον δείκτη Kappa για την εξεύρεση τρόπων βελτίωσης του eMatch.

2.6 Jaccard Index

Ο δείκτης Jaccard είναι μια στατιστική η οποία χρησιμοποιείται για τη σύγκριση της ομοιότητας και της διαφορετικότητας συνόλων δειγμάτων. Ο συντελεστής Jaccard μετρά την ομοιότητα μεταξύ συνόλων πεπερασμένων δειγμάτων και ορίζεται ως το μέγεθος της τομής διαιρούμενο με το μέγεθος της ένωσης των συνόλων δειγμάτων(έστω A και B):

$$J(A, B) = |τομή A και B| / |ένωση A και B| = |τομή A και B| / (|A| + |B| - |τομή A και B|) \quad (5)$$

Αν τα A και B είναι και τα δύο κενά, ορίζουμε $J(A, B) = 1$, αλλιώς ισχύει $0 \leq J(A, B) \leq 1$.

3. ΑΠΟΤΕΛΕΣΜΑΤΑ

3.1 Σύγκριση Συντελεστών Ομοιότητας

Ο βασικότερός μας στόχος ήταν να βελτιώσουμε το average friends' placement του EgoSimilar. Το average friends' placement είναι μια μετρική που προσδιορίζει κατά μέσο όρο τη θέση εμφάνισης των φίλων των χρηστών (φίλοι πριν τη χρήση του eMatch) στη λίστα με τους υποψήφιους βάσει ταιριάσματος από το eMatch φίλων (σε φθίνουσα σειρά, τα υψηλά ποσοστά ταιριάσματος πρώτα και τα χαμηλά τελευταία, όπως είναι πιο φυσιολογικό να φαίνεται στην οθόνη του κινητού τηλεφώνου του χρήστη). Η συγκεκριμένη μετρική είναι πολύ σημαντική και έχει ουσιαστική επίδραση από την άποψη ότι θα επιλέξουμε τον αλγόριθμο ο οποίος στη λίστα με τους υποψήφιους φίλους θα εμφανίζει ψηλότερα τους πραγματικούς μας φίλους. Αν όντως συμβαίνει αυτό, τότε και οι υπόλοιποι χρήστες που είναι ψηλά στη λίστα και δεν είναι φίλοι μας, θα μπορούσαν πράγματι να αποτελέσουν δυνητικά φίλοι μας.

Παρακάτω παρουσιάζονται τα αποτελέσματα που προέκυψαν από την χρησιμοποίηση τριών καινούργιων δεικτών ομοιότητας (Jaccard, Pi, Kappa) με στόχο τη βελτίωση του matching (και κυρίως του average friends' placement) της εφαρμογής και συγκρίνονται με αυτά των μεθόδων EgoSimilar, Pearson και Cosine.

Ο Πίνακας 1 παρουσιάζει τα συνολικά μέσα αποτελέσματα ταιριάσματος των μεθόδων EgoSimilar (weight 0.1), Pearson, Cosine, Jaccard, Pi και Kappa για το συνολικό dataset των 286 χρηστών. Κάθε γραμμή του πίνακα αντιστοιχεί σε μια από τις προαναφερθείσες μεθόδους. Στον EgoSimilar χρησιμοποιούμε Wchange 0.1, γιατί για αυτό έχουμε τα καλύτερα ποσοστά ταιριάσματος.

ΠΙΝΑΚΑΣ 1. **OVERALL MATCHING RESULTS**
(Similarity Coefficients)

	<i>Matching Metrics</i>		
	<i>AM</i>	<i>AMC</i>	<i>AMnC</i>
EgoSimilar ($W_{\text{change}}=0.1$)	59.73%	64.74%	59.69%
Pearson	12.5(56.25%)	20.22(60.11%)	12.46(56.23%)
Cosine	81.0%	84.66%	80.97%
Jaccard_SingleRatings	6.69%	8.88%	6.68%
Jaccard_GroupedRatings	18.22%	22.23%	18.19%
Pi_SingleRatings	0.04(50.02%)	3.39(51.69%)	0.01(50.0%)
Pi_GroupedRatings	0.23(50.11%)	6.09(53.45%)	0.19(50.09%)
Kappa_SingleRatings	2.29(51.14%)	5.25(52.62%)	2.27(51.13%)
Kappa_GroupedRatings	5.24(52.62%)	9.88(54.94%)	5.2(52.6%)

Οι στήλες περιέχουν τις μετρικές που χρησιμοποιήθηκαν, οι οποίες είναι οι εξής:

1. AM (Average Matching): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των χρηστών της εκάστοτε μεθόδου
2. AMC (Average Matching of Connected users): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των συνδεδεμένων χρηστών της εκάστοτε μεθόδου
3. AMnC (Average Matching of not Connected users): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των χρηστών της εκάστοτε μεθόδου που δεν έχουν σύνδεση

Όσον αφορά στην Pearson, στο Pi και στο Kappa, τα μεγέθη AM, AMC, AMnC είναι υπολογισμένα στην κλίμακα [-100, 100] και [0, 100] μέσα στην παρένθεση.

Εκεί που αναγράφεται Jaccard_SingleRatings, Pi_SingleRatings και Kappa_SingleRatings, έχουμε υπολογίσει τα αποτελέσματα θεωρώντας ως ξεχωριστή κατηγορία κάθε rating (από 1-10), ενώ εκεί που αναγράφεται Jaccard_GroupedRatings, Pi_GroupedRatings και Kappa_GroupedRatings έχουμε υπολογίσει τα αποτελέσματα ομαδοποιώντας τις κατηγορίες(τα ratings) ως εξής: 1-2, 3-5, 6-8, 9-10. Τα ίδια ισχύουν και όσον αφορά στον υπολογισμό του average friends' placement παρακάτω.

Για την υλοποίηση όλων των προαναφερθέντων μεθόδων Jaccard, Pi και Kappa θεωρήσαμε ως ενιαίο σύνολο τις κατηγορίες και τα αντικείμενα, χωρίς να τους δώσουμε ξεχωριστά βάρη, όπως στις μεθόδους EgoSimilar, Pearson και Cosine(0.25 στις κατηγορίες και 0.75 στα αντικείμενα).

Όσον αφορά στα αποτελέσματα για τις τρεις πρώτες μεθόδους, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις τρεις υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Η Pearson και η Cosine παρουσιάζουν τα χειρότερα αποτελέσματα όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Χρησιμοποιώντας τον EgoSimilar, η διαφορά μεταξύ AMC και AMnC είναι περίπου 5% ανεξάρτητα από την τιμή του Wchange, ενώ με την χρήση της μεθόδου Pearson η διαφορά είναι 3.9% και με τη χρήση της μεθόδου Cosine 3.7%. Συνεπώς, η Pearson και η Cosine(περισσότερο) υστερούν στο να ανακαλύψουν ποιοι χρήστες είναι ήδη συνδεδεμένοι.
3. Παρατηρούμε ότι (όπως συμβαίνει και με την περίπτωση που έχουμε λίγες χρήστες στην εφαρμογή, [3]) η Cosine παρουσιάζει καλύτερα αποτελέσματα από την Pearson και από τον EgoSimilar και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων. Ακόμη, ο EgoSimilar έχει καλύτερα αποτελέσματα από την Pearson αναφορικά με το AM, το AMC και το AMnC.

Όσον αφορά στα αποτελέσματα για τις επόμενες μεθόδους, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις έξι υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Ο δείκτης Kappa_SingleRatings παρουσιάζει τα χειρότερα αποτελέσματα όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών και ο δείκτης Jaccard_SingleRatings τα καλύτερα. Χρησιμοποιώντας τον Jaccard_SingleRatings, η διαφορά μεταξύ AMC και AMnC είναι περίπου 2.2%, ενώ με την χρήση του δείκτη Kappa_SingleRatings η διαφορά είναι 1.49% και με τη χρήση του δείκτη Pi_SingleRatings 1.69%. Αντίστοιχα, χρησιμοποιώντας τον Jaccard_GroupedRatings, η διαφορά μεταξύ AMC και AMnC είναι 4.4%, ενώ με την χρήση του δείκτη Kappa_GroupedRatings η διαφορά είναι 2.34% και με τη χρήση του δείκτη Pi_GroupedRatings 3.36%.
3. Παρατηρούμε ότι ο δείκτης Kappa_SingleRatings παρουσιάζει καλύτερα αποτελέσματα από τους άλλους δύο δείκτες και για το μέσο ποσοστό ταιριάσματος

όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων. Ακόμη, ο Pi_SingleRatings έχει καλύτερα αποτελέσματα από τον Jaccard_SingleRatings αναφορικά με το AM, το AMC και το AMnC. Η διαφορά στα αποτελέσματα μεταξύ των δεικτών Pi_SingleRatings και Kappa_SingleRatings είναι μικρή, αλλά έχουμε αναμενόμενα καλύτερα αποτελέσματα για τον Kappa_SingleRatings (λόγω της διαφοράς στον υπολογισμό του A_e), ενώ η διαφορά στα αποτελέσματα των δύο παραπάνω δεικτών είναι εμφανώς μεγαλύτερη συγκρίνοντας με τα αποτελέσματα που παρουσιάζει ο δείκτης Jaccard_SingleRatings. Τα ίδια ισχύουν και για τα αποτελέσματα που παρουσιάζουν οι Jaccard_GroupedRatings, Pi_GroupedRatings και Kappa_GroupedRatings.

Ο Πίνακας 2 παρουσιάζει το average friends' placement των μεθόδων EgoSimilar(weight 0.3), Pearson, Cosine, Jaccard, Pi και Kappa. Στον EgoSimilar χρησιμοποιούμε Wchange 0.3, γιατί για το average friends' placement αυτό έχει το καλύτερο αποτέλεσμα. Θέλουμε να είναι όσο το δυνατόν μικρότερο και το καλύτερο αποτέλεσμα πετυχαίνει ο δείκτης Jaccard_SingleRatings (χωρίς την ομαδοποίηση των ratings). Παρατηρούμε ότι όλες οι υλοποιήσεις των νέων δεικτών παρουσιάζουν καλύτερο average friends' placement από τις μεθόδους EgoSimilar, Pearson και Cosine. Επίσης, όσον αφορά στους δείκτες Pi και Kappa παρατηρούμε ότι η ομαδοποίηση των ratings οδηγεί σε μείωση του average friends' placement, ενώ το αντίθετο συμβαίνει με τον δείκτη Jaccard.

ΠΙΝΑΚΑΣ 2. AVERAGE FRIENDS' PLACEMENT
(Similarity Coefficients)

	<i>Placement</i>
EgoSimilar ($W_{\text{change}}=0.3$)	109.92
Pearson	101.79
Cosine	119.23
Jaccard_SingleRatings	92.44
Jaccard_GroupedRatings	97.41
Pi_SingleRatings	100.39
Pi_GroupedRatings	93.46
Kappa_SingleRatings	96.39
Kappa_GroupedRatings	93.06

Με βάση τα αποτελέσματα των Πινάκων 1,2 προχωρήσαμε στην μελέτη μας χρησιμοποιώντας τον δείκτη Kappa, ο οποίος έχει συνολικά καλύτερα αποτελέσματα από τους δείκτες Pi, Jaccard στο average friends placement και στα matching results. Συγκεκριμένα, επιλέξαμε τον δείκτη Kappa_GroupedRatings (με την ομαδοποίηση των κατηγοριών(των ratings) ως εξής: 1-2, 3-5, 6-8, 9-10), γιατί παρουσιάζει καλύτερα αποτελέσματα matching και καλύτερο average friends' placement από τον δείκτη Kappa_SingleRatings (όπου θεωρούμε καθένα από τα 10 πιθανά ratings ως ξεχωριστή κατηγορία).

3.2 Cluster Evaluation για $w_1 = 0.25$, $w_2 = 0.75$

Οι παρακάτω πίνακες(3 - 18) παρουσιάζουν τα αποτελέσματα των μεθόδων Cosine, Pearson και EgoSimilar όταν χρησιμοποιούμε K-means clustering για $K = 5, 7$ και 10 αντίστοιχα. Κάθε γραμμή αντιστοιχεί σε ένα cluster/group χρηστών(συνολικά 286 χρήστες, εκ των οποίων οι 272 έχουν δίκτυο). Ο μέσος όρος των βαθμολογήσεων των 286 αυτών χρηστών είναι 6.6 και η τυπική απόκλιση 2.5.

Οι στήλες περιέχουν τις μετρικές που χρησιμοποιήθηκαν, οι οποίες είναι οι εξής:

1. id: cluster id, όπου $id \in [1, K]$
2. N1: ο αριθμός των χρηστών στο cluster
3. N2: ο αριθμός των χρηστών στο cluster που έχουν δίκτυο(οι οποίοι έχουν σύνδεση με τουλάχιστον έναν άλλο χρήστη που μπορεί να βρίσκεται σε αυτό το cluster ή σε άλλο)
4. N3: ο αριθμός των χρηστών στο cluster που έχουν σύνδεση στην πραγματικότητα(σαν φίλοι ή ζευγάρια)
5. CP(Cluster Precision): Αυτό είναι η ακρίβεια που παρουσιάζει το cluster(πόσους χρήστες έχουμε στο N3 σε σχέση με το πόσους έχουμε στο N1 και το N2)
6. AVC (Average Valid Connections): για κάθε χρήστη στο cluster υπολογίσαμε το ποσοστό των συνδέσεων που συμπεριλαμβάνονται στο συγκεκριμένο cluster, και εξήχθη το μέσο ποσοστό
7. AVC*N2: Πολλαπλασιάζουμε το άνωθι ποσοστό με τον αριθμό των χρηστών στο cluster που έχουν δίκτυο(N2), ώστε να έχουμε μια επιπλέον συγκρίσιμη ποσότητα(όχι ποσοστό)
8. AM (Average Matching): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των χρηστών στο συγκεκριμένο cluster
9. AMC (Average Matching of Connected users): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των συνδεδεμένων χρηστών στο συγκεκριμένο cluster
10. AMnC (Average Matching of not Connected users): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των χρηστών στο συγκεκριμένο cluster που δεν έχουν σύνδεση

ΠΙΝΑΚΑΣ 3. **COSINE** (cid = 5)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	75	74	19	0.7	35.2	2608.3	85.42	91.28	85.38
2	48	42	10	1.1	27.4	1150.0	77.3	82.55	77.25
3	106	103	57	1.0	39.6	4081.7	86.67	87.54	86.66
4	52	48	16	1.4	23.6	1134.8	75.14	62.52	75.29
5	5	5	0	0.0	0.0	0.0	49.8	-	49.8
Σ	286	272	92						

ΠΙΝΑΚΑΣ 4. **COSINE** (cid = 7)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	69	64	18	0.8	30.2	1933.3	81.99	88.29	81.94
2	94	89	22	0.5	33.5	2983.3	85.04	90.8	85.01
3	23	20	2	1.0	8.8	175.0	81.3	93.03	81.21
4	18	17	3	2.2	10.0	170.0	71.73	87.53	71.42
5	36	36	4	0.6	9.2	331.0	78.67	76.13	78.69
6	11	9	0	0.0	0.0	0.0	71.14	-	71.14
7	23	23	5	1.9	25.7	591.7	87.45	91.36	87.37
Σ	286	272	54						

ΠΙΝΑΚΑΣ 5. **COSINE** (cid = 10)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	22	20	4	2.1	12.5	250.0	82.06	87.15	81.97
2	51	49	10	0.8	26.4	1291.7	85.61	89.07	85.58
3	51	47	12	1.1	34.6	1625.0	85.05	88.69	85.01
4	6	6	0	0.0	0.0	0.0	83.38	-	83.38
5	5	5	0	0.0	0.0	0.0	70.73	-	70.73
6	25	24	3	1.0	8.8	211.9	83.5	81.21	83.52
7	34	33	7	1.3	26.3	866.7	78.8	86.72	78.7
8	6	5	0	0.0	0.0	0.0	83.23	-	83.23
9	85	82	45	1.3	39.3	3221.0	82.19	87.42	82.12
10	1	1	0	-	0.0	0.0	-	-	-
Σ	286	272	81						

ΠΙΝΑΚΑΣ 6. PEARSON (cid = 5)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	63	62	24	1.2	35.6	2208.3	62.36	66.23	62.31
2	34	30	6	1.3	21.7	650.0	57.38	59.1	57.36
3	51	50	10	0.8	16.2	810.2	56.93	61.09	56.9
4	79	74	25	0.9	25.6	1892.1	61.32	64.05	61.3
5	59	56	17	1.1	36.3	2033.3	65.03	71.46	64.96
Σ	286	272	82						

ΠΙΝΑΚΑΣ 7. PEARSON (cid = 7)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	19	16	1	0.8	4.2	66.7	56.27	61.61	56.24
2	75	71	25	1.0	27.9	1980.0	61.26	65.71	61.22
3	31	29	4	0.9	13.2	383.3	61.48	67.53	61.43
4	30	28	12	3.1	29.2	816.7	61.38	62.32	61.35
5	13	13	1	1.2	7.7	100.0	58.45	72.88	58.27
6	20	20	1	0.5	2.5	50.0	53.31	63.44	53.25
7	98	95	32	0.7	38.1	3618.6	62.73	68.7	62.69
Σ	286	272	76						

ΠΙΝΑΚΑΣ 8. PEARSON (cid = 10)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	42	39	7	0.9	14.9	581.7	61.02	66.52	60.98
2	35	35	8	1.3	26.9	940.0	62.6	71.02	62.49
3	21	19	3	1.7	12.7	241.7	62.07	67.98	61.99
4	22	20	3	1.5	15.8	316.7	58.27	58.95	58.26
5	18	17	3	2.2	16.7	283.3	62.06	68.8	61.93
6	20	20	3	1.5	15.8	316.7	63.14	66.58	63.08
7	35	33	7	1.3	24.7	814.3	61.82	66.39	61.77
8	25	24	6	2.1	19.6	470.0	60.53	68.58	60.37
9	21	20	2	1.0	3.7	73.3	61.34	65.43	61.3
10	47	45	9	0.9	32.2	1450.0	64.17	73.32	64.1
Σ	286	272	51						

ΠΙΝΑΚΑΣ 9. EGOSIMILAR ($W_{\text{change}} = 0.1$, cid = 5)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	89	86	29	0.7	36.1	3101.2	63.62	66.62	63.6
2	92	86	41	1.1	37.8	3253.3	59.94	64.14	59.9
3	17	17	4	2.9	32.8	558.3	68.76	68.78	68.76
4	41	37	8	1.2	24.1	890.0	59.54	66.98	59.47
5	47	46	20	1.9	34.4	1583.3	63.33	73.15	63.14
Σ	286	272	102						

ΠΙΝΑΚΑΣ 10. EGOSIMILAR ($W_{\text{change}} = 0.2$, cid = 5)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	38	33	8	1.5	18.7	618.6	57.65	63.27	57.59
2	29	28	5	1.3	22.3	625.0	57.79	68.51	57.66
3	58	57	9	0.5	27.0	1540.0	64.3	69.56	64.28
4	116	111	55	0.9	50.4	5598.3	61.51	66.29	61.47
5	45	43	22	2.4	45.0	1933.3	61.97	70.26	61.78
Σ	286	272	99						

ΠΙΝΑΚΑΣ 11. EGOSIMILAR ($W_{\text{change}} = 0.3$, cid = 5)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	71	70	36	1.4	54.2	3791.7	66.89	71.46	66.82
2	32	30	4	0.9	12.9	386.7	57.93	67.2	57.85
3	114	108	66	1.1	55.5	5997.1	61.19	64.33	61.16
4	36	32	4	0.8	15.8	506.0	53.26	58.57	53.22
5	33	32	10	2.0	32.9	1053.3	54.07	62.0	53.92
Σ	286	272	120						

ΠΙΝΑΚΑΣ 12. EGOSIMILAR ($W_{\text{change}} = 0.5$, cid = 5)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	27	27	12	3.4	35.5	958.3	62.6	72.1	62.3
2	33	29	5	1.2	14.3	415.2	52.3	52.4	52.3
3	56	54	14	0.9	26.9	1453.3	57.4	61.6	57.4
4	21	18	2	1.3	14.4	258.3	59.9	74.0	59.8
5	149	144	98	0.9	62.5	9001.9	58.7	63.0	58.7
Σ	286	272	131						

ΠΙΝΑΚΑΣ 13. EGOSIMILAR ($W_{\text{change}} = 0.1$, $\text{cid} = 7$)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	92	89	35	0.8	44.9	3991.7	62.76	69.99	62.7
2	18	16	2	1.6	14.6	233.3	51.37	62.03	51.23
3	58	56	15	0.9	33.1	1854.3	65.73	66.97	65.72
4	11	11	2	3.6	36.4	400.0	65.33	69.52	65.17
5	50	45	19	1.9	34.5	1554.3	57.92	61.99	57.86
6	28	26	5	1.5	29.5	766.7	61.44	64.34	61.41
7	29	29	15	3.6	39.0	1131.7	71.28	75.75	71.1
Σ	286	272	93						

ΠΙΝΑΚΑΣ 14. EGOSIMILAR ($W_{\text{change}} = 0.2$, $\text{cid} = 7$)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	33	28	6	1.5	22.8	639.3	55.98	62.13	55.91
2	22	21	5	2.3	22.9	481.9	61.41	58.24	61.48
3	28	28	5	1.3	33.9	950.0	68.95	75.53	68.86
4	139	132	83	0.9	60.1	7931.7	59.66	65.33	59.62
5	5	5	1	10.0	40.0	200.0	67.95	70.47	67.67
6	32	31	11	2.3	29.0	900.0	68.01	71.53	67.93
7	27	27	7	1.9	16.5	445.0	59.97	65.14	59.86
Σ	286	272	118						

ΠΙΝΑΚΑΣ 15. EGOSIMILAR ($W_{\text{change}} = 0.3$, $\text{cid} = 7$)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	3	3	0	0.0	0.0	0.0	60.37	-	60.37
2	74	68	12	0.5	26.7	1814.3	58.66	61.1	58.65
3	81	79	31	1.0	43.3	3423.3	61.19	64.26	61.16
4	15	14	1	1.0	3.8	53.3	58.39	57.83	58.4
5	27	27	6	1.7	24.6	665.0	62.97	67.23	62.89
6	47	43	19	2.1	34.9	1501.2	56.67	61.48	56.58
7	39	38	16	2.2	32.2	1225.0	66.84	70.49	66.76
Σ	286	272	85						

ΠΙΝΑΚΑΣ 16. EGOSIMILAR ($W_{\text{change}} = 0.1$, $\text{cid} = 10$)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	3	3	0	0.0	0.0	0.0	26.14	-	26.14
2	3	2	0	0.0	0.0	0.0	23.27	-	23.27
3	64	62	29	1.5	41.4	2565.0	64.64	66.34	64.62
4	24	24	5	1.8	17.5	420.0	63.19	66.13	63.14
5	16	15	3	2.8	18.9	283.3	65.24	69.87	65.13
6	20	19	4	2.3	14.4	273.3	60.0	61.36	59.97
7	43	41	18	2.1	37.4	1533.3	68.1	72.52	68.01
8	27	25	5	1.6	21.7	541.7	54.54	69.05	54.33
9	54	49	11	0.9	28.0	1372.6	62.3	72.81	62.21
10	32	32	6	1.2	28.9	925.0	68.77	71.91	68.73
Σ	286	272	81						

ΠΙΝΑΚΑΣ 17. EGOSIMILAR ($W_{\text{change}} = 0.2$, $\text{cid} = 10$)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	17	17	3	2.2	35.3	600.0	63.33	70.66	63.17
2	4	4	0	0.0	0.0	0.0	40.29	-	40.29
3	37	36	13	2.0	25.5	916.7	67.39	75.22	67.23
4	9	8	0	0.0	0.0	0.0	57.96	-	57.96
5	83	79	35	1.1	42.3	3340.0	62.28	65.43	62.25
6	21	18	2	1.3	13.8	247.6	54.08	49.07	54.13
7	35	33	6	1.1	18.1	598.3	61.58	65.12	61.54
8	8	8	1	3.5	9.4	75.0	64.96	81.79	64.34
9	56	55	9	0.6	23.8	1308.3	62.65	71.29	62.6
10	16	14	1	1.0	14.3	200.0	55.63	80.65	55.42
Σ	286	272	70						

ΠΙΝΑΚΑΣ 18. EGOSIMILAR ($W_{\text{change}} = 0.3$, $\text{cid} = 10$)

id	Matching Metrics per cluster								
	<i>NI</i>	<i>N2</i>	<i>N3</i>	<i>CP</i>	<i>AVC</i> (%)	<i>AVC*N2</i>	<i>AM</i> (%)	<i>AMC</i> (%)	<i>AMnC</i> (%)
1	13	10	0	0.0	0.0	0.0	48.64	-	48.64
2	5	5	0	0.0	0.0	0.0	43.27	-	43.27
3	58	52	22	1.6	42.3	2200.0	58.69	58.77	58.68
4	10	8	2	7.1	20.8	166.7	54.94	61.28	54.64
5	32	32	13	2.6	30.9	990.0	66.78	70.71	66.68
6	2	2	0	0.0	0.0	0.0	41.15	-	41.15
7	70	69	24	1.0	35.6	2453.6	60.87	66.94	60.81
8	34	33	4	0.7	24.2	800.0	68.97	74.59	68.93
9	38	37	6	0.9	21.4	792.6	59.37	65.82	59.31
10	24	24	3	1.0	10.0	240.0	64.24	68.85	64.19
Σ	286	272	74						

Αυτό που παρατηρούμε είναι ότι η μέθοδος Cosine παρουσιάζει υψηλότερο ποσοστό ταιριάσματος μεταξύ των cluster της συγκρινόμενη με τη μέθοδο Pearson, και για τους συνδεδεμένους και για τους μη συνδεδεμένους χρήστες, για όλες τις τιμές του K . Επίσης, η μέθοδος Cosine υπερσχύει της μεθόδου Pearson στα πλαίσια της μετρικής N3 για $K = 5$ και $K = 10$, με τη μεγαλύτερη διαφορά να παρατηρείται για $K = 10$ (81 έναντι 51). Τα αποτελέσματα αυτά είναι αντίστοιχα με την περίπτωση των 57 χρηστών [3].

Όσον αφορά στον EgoSimilar, παρατηρούμε ότι υπερσχύει των μεθόδων Cosine και Pearson στα πλαίσια της μετρικής N3 για όλες τις τιμές του K και του W_{change} , εκτός της Cosine για $K = 10$. Για $K = 10$ υπερσχύει της Pearson, αλλά παρουσιάζει ίδιο N3(81) με την Cosine για $W_{\text{change}} = 0.1$ και μικρότερο για $W_{\text{change}} = 0.2$ και 0.3 . Το ότι υπερσχύει των παραπάνω μεθόδων στα πλαίσια της μετρικής N3 σημαίνει ότι καταφέρνει να συμπεριλάβει στο ίδιο cluster περισσότερους ανθρώπους που είναι ήδη συνδεδεμένοι σαν φίλοι ή ζευγάρια πριν τη χρήση του eMatch. Αποτελέσματα τα οποία συμφωνούν με τα αντίστοιχα για την περίπτωση που είχαμε μόλις 57 χρήστες στην εφαρμογή, με την εξαίρεση ότι για τους λιγότερους χρήστες ο EgoSimilar υπερσχύει της Cosine μόνο για μια τιμή του W_{change} και όχι όπως περιγράφηκε παραπάνω. Συγκρίνοντας τον EgoSimilar με τις άλλες δύο μεθόδους στα πλαίσια των μετρικών AVC, AM, AMC και AMnC, παρατηρούμε ότι ο EgoSimilar υπερσχύει της μεθόδου Pearson για όλες τις τιμές του W_{change} και του K (ενώ για τους λίγους χρήστες τα αποτελέσματα που παρουσιάζουν οι δύο μέθοδοι είναι περίπου ίδια) και παρουσιάζει μικρότερα στατιστικά ταιριάσματος για όλες τις τιμές του W_{change} και του K από την μέθοδο Cosine (όπως συμβαίνει και με τους λιγότερους χρήστες). Σε κάποια cluster παρατηρούμε λιγότερους συνδεδεμένους από ότι μη συνδεδεμένους χρήστες (AMC <= AMnC), φαινόμενο το οποίο παρατηρείται κάποιες λίγες φορές στα αποτελέσματα της Cosine και του EgoSimilar.

Αναφορικά με τον αριθμό των cluster που χρησιμοποιούμε, η διαφορά που παρατηρείται είναι ότι η αύξηση του αριθμού των cluster καταλήγει στο να έχουμε κάποια clusters με λίγους συνολικούς και συνδεδεμένους χρήστες (και κάποιες φορές κανένα χρήστη που να έχει σύνδεση στην πραγματικότητα, N3), αφού υπάρχουν περισσότερα cluster για να γίνει η κατανομή των χρηστών, και επομένως το ποσοστό ταιριάσματος να μειώνεται τις περισσότερες φορές.

Σε όλους τους πίνακες και στις αναλύσεις των αποτελεσμάτων παρακάτω, για να μπορέσουμε να κάνουμε μια ουσιαστική σύγκριση μεταξύ του EgoSimilar και των μεθόδων Pearson, Cosine και Kappa (με την ομαδοποίηση των κατηγοριών (των ratings) ως εξής: 1-2, 3-5, 6-8, 9-10), υπολογίσαμε το ταίριασμα μεταξύ των κατηγοριών και το ταίριασμα μεταξύ των αντικειμένων ξεχωριστά με τη χρήση των παραπάνω μεθόδων χρησιμοποιώντας τα ίδια βάρη $w1$ και $w2$ (0,25, 0,75) και στη συνέχεια (0,5, 0,5) και (0,75, 0,25) όπως στον EgoSimilar.

Ακόμα, σε όλους τους πίνακες και στις αναλύσεις των αποτελεσμάτων παρακάτω, εκεί που αναγράφεται Kappa, έχουμε υπολογίσει το $\text{matching}(X,Y)$ σύμφωνα με τα στοιχεία (κατηγορίες και αντικείμενα) του χρήστη X και το $\text{matching}(Y,X)$ σύμφωνα με τα στοιχεία του χρήστη Y , ενώ εκεί που αναγράφεται Kappa2 έχουμε υπολογίσει ένα matching μεταξύ των δύο χρηστών σύμφωνα με τα κοινά στοιχεία τους. Εξ' ορισμού, ο δείκτης Kappa υπολογίζεται όπως έχουμε υπολογίσει το Kappa2 (για τα κοινά στοιχεία των δύο χρηστών) και εξάγεται ένα ποσοστό ταιριάσματος κι όχι δύο, αλλά για να έχουμε μια απευθείας σύγκριση με τις άλλες μεθόδους που υπολογίζουν δύο ποσοστά ταιριάσματος, εφαρμόσαμε και μια παραλλαγή του δείκτη Kappa, όπου υπολογίζονται δύο ποσοστά ταιριάσματος μεταξύ των δύο χρηστών.

Ο Πίνακας 19 παρουσιάζει τα συνολικά μέσα αποτελέσματα ταιριάσματος των μεθόδων EgoSimilar (weight 0.1, 0.2, 0.3, 0.5), Pearson, Cosine, Kappa και Kappa2 για το συνολικό dataset των 286 χρηστών για $w1 = 0.25$ και $w2 = 0.75$. Κάθε γραμμή του πίνακα αντιστοιχεί σε μια από τις προαναφερθείσες μεθόδους. Υπολογίσαμε αποτελέσματα και για $W_{\text{change}} = 0.5$ με σκοπό την καλυτέρευση του average friends' placement, αλλά εντέλει χρησιμοποιήθηκε

$W_{\text{change}} = 0.3$, το οποίο έχει λίγο χειρότερο average friends' placement από ότι για 0.5, αλλά καλύτερα ποσοστά ταιριάσματος(όσο μειώνεται το συγκεκριμένο βάρος έχουμε καλύτερα ποσοστά).

ΠΙΝΑΚΑΣ 19. **OVERALL MATCHING RESULTS**

(w1, w2) = (0.25, 0.75)

	<i>Matching Metrics</i>		
	<i>AM</i>	<i>AMC</i>	<i>AMnC</i>
EgoSimilar ($W_{\text{change}} = 0.1$)	59.73%	64.74%	59.69%
EgoSimilar ($W_{\text{change}} = 0.2$)	58.79%	63.8%	58.75%
EgoSimilar ($W_{\text{change}} = 0.3$)	57.85%	62.85%	57.82%
EgoSimilar ($W_{\text{change}} = 0.5$)	56.0%	61.0%	55.9%
Pearson	12.5(56.25%)	20.22(60.11%)	12.46(56.23%)
Cosine	81.0%	84.66%	80.97%
Kappa	3.56(51.78%)	7.6(53.8%)	3.52(51.76%)
Kappa2	4.36(52.18%)	8.7(54.35%)	4.34(52.17%)

Οι στήλες περιέχουν τις μετρικές που χρησιμοποιήθηκαν, οι οποίες είναι οι εξής:

1. AM (Average Matching): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των χρηστών της εκάστοτε μεθόδου
2. AMC (Average Matching of Connected users): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των συνδεδεμένων χρηστών της εκάστοτε μεθόδου
3. AMnC (Average Matching of not Connected users): Αυτό είναι το μέσο ποσοστό ταιριάσματος όλων των χρηστών της εκάστοτε μεθόδου που δεν έχουν σύνδεση

Όσον αφορά στην Pearson και στο Kappa, τα μεγέθη AM, AMC, AMnC είναι υπολογισμένα στην κλίμακα [-100, 100] και [0, 100] μέσα στην παρένθεση.

Όσον αφορά στα αποτελέσματα για τις πέντε πρώτες υλοποιήσεις, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις πέντε υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ότι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Η Pearson και η Cosine παρουσιάζουν τα χειρότερα αποτελέσματα όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Χρησιμοποιώντας τον EgoSimilar, η διαφορά μεταξύ AMC και AMnC είναι περίπου 5% ανεξάρτητα από την τιμή του W_{change} , ενώ με την χρήση της μεθόδου Pearson η διαφορά είναι 3.9% και με τη χρήση της μεθόδου Cosine 3.7%. Συνεπώς, η Pearson και η Cosine(περισσότερο) υστερούν στο να ανακαλύψουν ποιοι χρήστες είναι ήδη συνδεδεμένοι.
3. Όσο αυξάνεται το W_{change} , τόσο μειώνονται τα συνολικά στατιστικά. Η αύξηση του W_{change} προκαλεί μείωση των ποσοστών και αυτό οφείλεται στο ότι οι περισσότεροι χρήστες έχουν ψηφίσει τα default items και δεν έχουν εισάγει δικά τους. Αυτό κάνει τα default items δημοφιλή και τα δημοφιλή αντικείμενα όταν χρησιμοποιούμε τον EgoSimilar οδηγούν σε μείωση των ποσοστών(αν οι χρήστες έχουν σημαντικά διαφορετικές απόψεις πάνω σε αυτά) ή σε καμία αλλαγή στα

ποσοστά. Παρατηρούμε επίσης ότι η Cosine παρουσιάζει καλύτερα αποτελέσματα από την Pearson και από τον EgoSimilar και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων. Ακόμη, ο EgoSimilar έχει καλύτερα αποτελέσματα από την Pearson αναφορικά με το AM, το AMC και το AMnC για $W_{change} = 0.1, 0.2$ και 0.3 (αποτελέσματα καλύτερα από όταν είχαμε λιγότερους χρήστες), ενώ για $W_{change} = 0.5$ παρουσιάζει λίγο χειρότερα αποτελέσματα αναφορικά με το AM και το AMnC, αλλά καλύτερα όσον αφορά στο AMC.

Όσον αφορά στα αποτελέσματα για τις μεθόδους Kappa και Kappa2, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις δύο υλοποιήσεις το ταιρίασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Ο δείκτης Kappa παρουσιάζει χειρότερα αποτελέσματα από τον Kappa2 όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Χρησιμοποιώντας τον Kappa, η διαφορά μεταξύ AMC και AMnC είναι 2.04%, ενώ με την χρήση του δείκτη Kappa2 η διαφορά είναι 2.18%.
3. Παρατηρούμε ότι ο δείκτης Kappa2 παρουσιάζει καλύτερα αποτελέσματα από τον Kappa και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων.

Παρατηρούμε ότι οι δείκτες Kappa και Kappa2 παρουσιάζουν χειρότερα αποτελέσματα από τις μεθόδους EgoSimilar, Pearson και Cosine αναφορικά με τον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Επίσης, παρουσιάζουν χειρότερα αποτελέσματα από τις παραπάνω μεθόδους και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων.

Ο Πίνακας 20 παρουσιάζει το average friends' placement των μεθόδων EgoSimilar (weight 0.1, 0.2, 0.3, 0.5), Pearson, Cosine, Kappa και Kappa2. Θέλουμε να είναι όσο το δυνατόν μικρότερο και το καλύτερο αποτέλεσμα πετυχαίνει το Kappa2. Ο EgoSimilar παρουσιάζει καλύτερα αποτελέσματα (και για τα τέσσερα weights) από την μέθοδο Cosine, αλλά όχι και από την Pearson. Επίσης, όσο αυξάνουμε το W_{change} , τόσο καλύτερα (μικρότερα) αποτελέσματα έχουμε.

ΠΙΝΑΚΑΣ 20. AVERAGE FRIENDS' PLACEMENT

(w1, w2) = (0.25, 0.75)

	<i>Placement</i>
EgoSimilar ($W_{change}=0.1$)	110.94
EgoSimilar ($W_{change}=0.2$)	110.44
EgoSimilar ($W_{change}=0.3$)	109.92
EgoSimilar ($W_{change}=0.5$)	109.05
Pearson	101.79
Cosine	119.23
Kappa	101.11
Kappa2	99.65

Παρατηρούμε ότι οι δείκτες Kappa και Kappa2 παρουσιάζουν καλύτερο average friends' placement από τις μεθόδους EgoSimilar, Pearson και Cosine.

3.3 Αλλαγή των βαρών w_1 και w_2

3.3.1 Evaluation για $w_1 = 0.5$, $w_2 = 0.5$

Ο Πίνακας 21 παρουσιάζει τα συνολικά μέσα αποτελέσματα ταιριάσματος των μεθόδων EgoSimilar(weight 0.3), Pearson, Cosine, Kappa και Kappa2 για το συνολικό dataset των 286 χρηστών για $w_1 = 0.5$ και $w_2 = 0.5$. Κάθε γραμμή του πίνακα αντιστοιχεί σε μια από τις προαναφερθείσες μεθόδους. Από τη στιγμή που ο βασικότερος στόχος μας είναι η βελτίωση του average friends' placement, στα αποτελέσματα από εδώ και κάτω(όπου αυξάνουμε το βάρος των κατηγοριών και αντιστοίχως μειώνεται το βάρος των αντικειμένων) έχουμε εξάγει αποτελέσματα για τον EgoSimilar για το W_{change} που πετυχαίνει το καλύτερο average friends' placement, συνεπώς για $W_{\text{change}} = 0.3$.

ΠΙΝΑΚΑΣ 21. **OVERALL MATCHING RESULTS**
(w_1, w_2) = (0.5, 0.5)

	<i>Matching Metrics</i>		
	<i>AM</i>	<i>AMC</i>	<i>AMnC</i>
EgoSimilar ($W_{\text{change}} = 0.3$)	63.93%	68.87%	63.89%
Pearson	12.26(56.13%)	20.1(60.05%)	12.2(56.1%)
Cosine	85.3%	88.2%	85.28%
Kappa	3.02(51.51%)	7.46(53.73%)	2.98(51.49%)
Kappa2	3.56(51.78%)	8.1(54.05%)	3.54(51.77%)

Όσον αφορά στα αποτελέσματα για τις μεθόδους EgoSimilar, Pearson και Cosine, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις τρεις υλοποιήσεις το ταιρίασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Η Pearson και η Cosine παρουσιάζουν τα χειρότερα αποτελέσματα όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Χρησιμοποιώντας τον EgoSimilar, η διαφορά μεταξύ AMC και AMnC είναι περίπου 5% ανεξάρτητα από την τιμή του W_{change} , ενώ με την χρήση της μεθόδου Pearson η διαφορά είναι 3.9% και με τη χρήση της μεθόδου Cosine 3%. Συνεπώς, η Pearson και η Cosine(περισσότερο) υστερούν στο να ανακαλύψουν ποιοι χρήστες είναι ήδη συνδεδεμένοι.
3. Παρατηρούμε ότι η Cosine παρουσιάζει καλύτερα αποτελέσματα από την Pearson και από τον EgoSimilar και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων. Ακόμη, ο EgoSimilar έχει καλύτερα αποτελέσματα από την Pearson αναφορικά με το AM, το AMC και το AMnC.

Όσον αφορά στα αποτελέσματα για τις μεθόδους Kappa και Kappa2, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις δύο υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Ο δείκτης Kappa παρουσιάζει χειρότερα αποτελέσματα από τον Kappa2 όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Χρησιμοποιώντας τον Kappa, η διαφορά μεταξύ AMC και AMnC είναι 2.24%, ενώ με την χρήση του δείκτη Kappa2 η διαφορά είναι 2.28%.
3. Παρατηρούμε ότι ο δείκτης Kappa2 παρουσιάζει καλύτερα αποτελέσματα από τον Kappa και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων.

Παρατηρούμε ότι οι δείκτες Kappa και Kappa2 παρουσιάζουν χειρότερα αποτελέσματα από τις μεθόδους EgoSimilar, Pearson και Cosine αναφορικά με τον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Επίσης, παρουσιάζουν χειρότερα αποτελέσματα από τις παραπάνω μεθόδους και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων.

Ο Πίνακας 22 παρουσιάζει το average friends' placement των μεθόδων EgoSimilar(weight 0.3), Pearson, Cosine, Kappa και Kappa2. Θέλουμε να είναι όσο το δυνατόν μικρότερο και το καλύτερο αποτέλεσμα πετυχαίνει ο EgoSimilar. Ο EgoSimilar παρουσιάζει καλύτερα αποτελέσματα και από την μέθοδο Cosine και από την Pearson, το οποίο σημαίνει ότι όταν δίνουμε βάρος στη βαθμολόγηση των κατηγοριών του χρήστη ο EgoSimilar ξεπερνά την Pearson.

ΠΙΝΑΚΑΣ 22. AVERAGE FRIENDS' PLACEMENT
(w1, w2) = (0.5, 0.5)

	<i>Placement</i>
EgoSimilar ($W_{change}=0.3$)	105.74
Pearson	111.86
Cosine	119.35
Kappa	109.73
Kappa2	108.28

Παρατηρούμε ότι οι δείκτες Kappa και Kappa2 παρουσιάζουν καλύτερο average friends' placement από τις μεθόδους Pearson και Cosine και χειρότερο από τον EgoSimilar. Συνεπώς, όταν αυξάνουμε το βάρος των κατηγοριών ο δείκτης Kappa είναι καλύτερος από τη μέθοδο Pearson.

Συγκρίνοντας με τα αποτελέσματα για $w1=0.25$ και $w2=0.75$, ο EgoSimilar και η Cosine έχουν βελτιωμένα ποσοστά ταιριάσματος και οι δύο μέθοδοι Kappa και η Pearson χειρότερα. Επίσης, το average friends' placement του EgoSimilar είναι βελτιωμένο, της Cosine περίπου ίδιο και των δύο μεθόδων Kappa και της Pearson χειρότερο. Παρατηρούμε λοιπόν ότι η αύξηση του βάρους στις κατηγορίες οδηγεί σε βελτίωση του EgoSimilar, ο οποίος υπερσχύει των υπόλοιπων μεθόδων.

3.3.2 Evaluation για $w_1 = 0.75$, $w_2 = 0.25$

Ο Πίνακας 23 παρουσιάζει τα συνολικά μέσα αποτελέσματα ταιριάσματος των μεθόδων EgoSimilar(weight 0.3), Pearson, Cosine, Kappa και Kappa2 για το συνολικό dataset των 286 χρηστών. Κάθε γραμμή του πίνακα αντιστοιχεί σε μια από τις προαναφερθείσες μεθόδους.

Σε όλες τις μεθόδους του πίνακα το βάρος για τις κατηγορίες είναι 0.75 και για τα αντικείμενα 0.25. Τα ίδια ισχύουν και όσον αφορά στον υπολογισμό του average friends' placement παρακάτω.

ΠΙΝΑΚΑΣ 23. **OVERALL MATCHING RESULTS**

(w_1, w_2) = (0.75, 0.25)

	<i>Matching Metrics</i>		
	<i>AM</i>	<i>AMC</i>	<i>AMnC</i>
EgoSimilar ($W_{change} = 0.3$)	70.0%	74.89%	69.97%
Pearson	12.0(56.0%)	20.0(60.0%)	11.94(55.97%)
Cosine	89.6%	91.74%	89.58%
Kappa	2.48(51.24%)	7.3(53.65%)	2.44(51.22%)
Kappa2	2.76(51.38%)	7.5(53.75%)	3.46(51.37%)

Όσον αφορά στα αποτελέσματα για τις μεθόδους EgoSimilar, Pearson και Cosine, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Και στις τρεις υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Η Pearson και η Cosine παρουσιάζουν τα χειρότερα αποτελέσματα όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών. Χρησιμοποιώντας τον EgoSimilar, η διαφορά μεταξύ AMC και AMnC είναι περίπου 5% ανεξάρτητα από την τιμή του Wchange, ενώ με την χρήση της μεθόδου Pearson η διαφορά είναι 4% και με τη χρήση της μεθόδου Cosine 2.16%. Συνεπώς, η Pearson και η Cosine(περισσότερο) υστερούν στο να ανακαλύψουν ποιοι χρήστες είναι ήδη συνδεδεμένοι.
3. Παρατηρούμε ότι(όπως συμβαίνει και με την περίπτωση που έχουμε λίγες χρήστες στην εφαρμογή, [3]) η Cosine παρουσιάζει καλύτερα αποτελέσματα από την Pearson και από τον EgoSimilar και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων. Ακόμη, ο EgoSimilar έχει καλύτερα αποτελέσματα από την Pearson αναφορικά με το AM, το AMC και το AMnC.

Όσον αφορά στα αποτελέσματα για τις μεθόδους Kappa και Kappa2, έχουμε τις εξής παρατηρήσεις:

1. Και στις δύο υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Ο δείκτης Kappa παρουσιάζει καλύτερα αποτελέσματα(περίπου το ίδιο αποτέλεσμα) από τον Kappa2 όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη

συνδεδεμένων χρηστών. Χρησιμοποιώντας τον Kappa, η διαφορά μεταξύ AMC και AMnC είναι 2.43%, ενώ με την χρήση του δείκτη Kappa2 η διαφορά είναι 2.38%.

3. Παρατηρούμε ότι ο δείκτης Kappa2 παρουσιάζει καλύτερα αποτελέσματα από τον Kappa και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων.

Αναφορικά με τον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών, παρατηρούμε ότι οι δείκτες Kappa και Kappa2 παρουσιάζουν χειρότερα αποτελέσματα από τις μεθόδους EgoSimilar και Pearson και καλύτερα από την Cosine. Επίσης, παρουσιάζουν χειρότερα αποτελέσματα από όλες τις παραπάνω μεθόδους και για το μέσο ποσοστό ταιριάσματος όλων των χρηστών και για αυτό των συνδεδεμένων και για αυτό των μη συνδεδεμένων.

Ο Πίνακας 24 παρουσιάζει το average friends' placement των μεθόδων EgoSimilar(weight 0.3), Pearson, Cosine, Kappa και Kappa2. Θέλουμε να είναι όσο το δυνατόν μικρότερο και το καλύτερο αποτέλεσμα πετυχαίνει ο EgoSimilar. Ο EgoSimilar(όπως και στην περίπτωση $(w1, w2) = (0.5, 0.5)$) παρουσιάζει καλύτερα αποτελέσματα και από την μέθοδο Cosine και από την Pearson, το οποίο σημαίνει ότι όταν δίνουμε βάρος στη βαθμολόγηση των κατηγοριών του χρήστη ο EgoSimilar ξεπερνά την Pearson. Παρατηρούμε ότι οι δείκτες Kappa και Kappa2 παρουσιάζουν καλύτερο average friends' placement από τις μεθόδους Pearson και Cosine και χειρότερο από τον EgoSimilar. Συνεπώς, όπως και στην προηγούμενη περίπτωση, όταν αυξάνουμε το βάρος των κατηγοριών ο δείκτης Kappa είναι καλύτερος από τη μέθοδο Pearson.

ΠΙΝΑΚΑΣ 24. AVERAGE FRIENDS' PLACEMENT
(w1, w2) = (0.75, 0.25)

	<i>Placement</i>
EgoSimilar ($W_{change}=0.3$)	102.44
Pearson	117.98
Cosine	119.66
Kappa	115.57
Kappa2	116.06

Συγκρίνοντας με τα αποτελέσματα για $w1 = 0.25$ και $w2 = 0.75$ (και για $w1 = 0.5$ και $w2 = 0.5$), ο EgoSimilar και η Cosine έχουν βελτιωμένα ποσοστά ταιριάσματος και οι δύο μέθοδοι Kappa και η Pearson χειρότερα. Επίσης, το average friends' placement του EgoSimilar είναι βελτιωμένο, της Cosine περίπου ίδιο και των δύο μεθόδων Kappa και της Pearson χειρότερο. Επομένως, παρατηρούμε και σε αυτή την περίπτωση ότι η αύξηση του βάρους στις κατηγορίες οδηγεί σε βελτίωση του EgoSimilar, ο οποίος υπερσχύει των υπόλοιπων μεθόδων.

3.4 Matching βάσει των 2 πιο "αγαπημένων" κατηγοριών

Όσον αφορά στον βασικότερο στόχο μας που είναι η βελτίωση του EgoSimilar, του αλγορίθμου που τρέχει η εφαρμογή eMatch, παρατηρούμε ότι όσο αυξάνουμε το βάρος στην αξιολόγηση των κατηγοριών των χρηστών της εφαρμογής(και αντιστοίχως μειώνουμε αυτό των αντικειμένων) τόσο καλύτερα αποτελέσματα έχουμε και στα πλαίσια των μετρικών AM, AMC και AMnC και στα πλαίσια του average friends' placement. Συνεπώς, η αρχική μας θεώρηση να δώσουμε βάρος στα αντικείμενα έχει από τη μία χειρότερα αποτελέσματα, αλλά το σκεπτικό ότι το "γενικό" ταίριασμα(ταίριασμα στις κατηγορίες) των χρηστών είναι μικρότερης σημασίας, καθώς τα συγκεκριμένα γούστα τους σε κάποια κατηγορία μπορεί να διαφέρουν σημαντικά ή τελείως, ενώ στα αντικείμενα να διαφέρουν λιγότερο είναι σωστό από τη στιγμή που έχουμε μικρό αριθμό κατηγοριών και πολλά αντικείμενα σε αυτές(και συνολικά).

Για να υπάρχει λοιπόν μια καλύτερη εικόνα όσον αφορά στο τι συμβαίνει με τις κατηγορίες που ψηφίζουν οι χρήστες της εφαρμογής, εξάγαμε αποτελέσματα για το matching(X, Y) των χρηστών X και Y μόνο με βάση τις 2 πιο "αγαπημένες" κατηγορίες(κατηγορίες με την υψηλότερη βαθμολογία) του χρήστη X.

Ο Πίνακας 25 παρουσιάζει τα συνολικά μέσα αποτελέσματα ταιριάσματος της μεθόδου EgoSimilar($W_{\text{change}} = 0.3$) για το συνολικό dataset των 286 χρηστών.

Εκεί που αναγράφεται EgoSimilar2, έχουμε υπολογίσει το matching(X, Y) μόνο με βάση τις 2 πιο "αγαπημένες" κατηγορίες(κατηγορίες με την υψηλότερη βαθμολογία) του χρήστη X. Το W_c που αναγράφεται στην παρένθεση είναι το βάρος των κατηγοριών. Τα ίδια ισχύουν και όσον αφορά στον υπολογισμό του average friends' placement παρακάτω.

ΠΙΝΑΚΑΣ 25. **OVERALL MATCHING RESULTS**

(2 favorite categories)

	Matching Metrics		
	AM	AMC	AMnC
EgoSimilar ($W_c = 0.25$)	57.85%	62.85%	57.82%
EgoSimilar ($W_c = 0.5$)	63.93%	68.87%	63.89%
EgoSimilar ($W_c = 0.75$)	70.0%	74.89%	69.97%
EgoSimilar2 ($W_c = 0.25$)	62.09%	68.75%	62.04%
EgoSimilar2 ($W_c = 0.5$)	66.62%	73.2%	66.57%
EgoSimilar2 ($W_c = 0.75$)	71.15%	77.66%	71.1%

Όσον αφορά στα αποτελέσματα, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Σε όλες τις υλοποιήσεις το ταίριασμα είναι υψηλότερο μεταξύ των συνδεδεμένων χρηστών απ' ό,τι μεταξύ των μη συνδεδεμένων χρηστών, κάτι το οποίο είναι αναμενόμενο, αφού οι συνδεδεμένοι χρήστες έχουν ήδη συνάψει σχέσεις μεταξύ τους.
2. Χρησιμοποιώντας τον EgoSimilar για να προτείνει φίλους με βάση όλες τις ενεργές κατηγορίες του χρήστη X, η διαφορά μεταξύ AMC και AMnC είναι περίπου 5% ανεξάρτητα από την τιμή του W_{change} , ενώ όταν τον χρησιμοποιούμε για να προτείνει φίλους με βάση μόνο τις 2 πιο "αγαπημένες" κατηγορίες του χρήστη X η διαφορά είναι περίπου 6.7%. Συνεπώς, στη δεύτερη περίπτωση έχουμε καλύτερα

αποτελέσματα όσον αφορά στον διαχωρισμό των συνδεδεμένων από των μη συνδεδεμένων χρηστών.

3. Παρατηρούμε ότι όταν χρησιμοποιούμε τον EgoSimilar για να προτείνει φίλους με βάση μόνο τις 2 πιο "αγαπημένες" κατηγορίες του χρήστη X, τα αποτελέσματα αναφορικά με το AM, το AMC και το AMnC είναι καλύτερα από όταν τον χρησιμοποιούμε για να προτείνει φίλους με βάση όλες τις ενεργές κατηγορίες του χρήστη X.

Ο Πίνακας 26 παρουσιάζει το average friends' placement των μεθόδων EgoSimilar($W_c = 0.3$). Θέλουμε να είναι όσο το δυνατόν μικρότερο και πετυχαίνουμε το καλύτερο αποτέλεσμα όταν χρησιμοποιούμε τον EgoSimilar για να προτείνει φίλους με βάση μόνο τις 2 πιο "αγαπημένες" κατηγορίες του χρήστη X και το βάρος των κατηγοριών είναι 0.75. Παρατηρούμε ότι όταν χρησιμοποιούμε τον EgoSimilar για να προτείνει φίλους με βάση μόνο τις 2 πιο "αγαπημένες" κατηγορίες του χρήστη X, τα αποτελέσματα για το average friends' placement είναι καλύτερα(μικρότερα) από όταν τον χρησιμοποιούμε για να προτείνει φίλους με βάση όλες τις ενεργές κατηγορίες του χρήστη X.

ΠΙΝΑΚΑΣ 26. **AVERAGE FRIENDS' PLACEMENT**
(2 favorite categories)

	<i>Placement</i>
EgoSimilar ($W_c = 0.25$)	109.92
EgoSimilar ($W_c = 0.5$)	105.74
EgoSimilar ($W_c = 0.75$)	102.44
EgoSimilar2 ($W_c = 0.25$)	108.13
EgoSimilar2 ($W_c = 0.5$)	101.89
EgoSimilar2 ($W_c = 0.75$)	98.34

Συνεπώς, όταν δίνουμε βάρος στις κατηγορίες(είτε αυξάνοντας το w_1 είτε προτείνοντας φίλους χρησιμοποιώντας μόνο τις δύο κατηγορίες των χρηστών με την υψηλότερη βαθμολογία) ο EgoSimilar παρουσιάζει καλύτερα αποτελέσματα.

3.5 Κατηγοριοποίηση χρηστών βάσει ηλικίας

Μια άλλη ενδιαφέρουσα εκδοχή του EgoSimilar είναι να χρησιμοποιηθεί για να προτείνει φίλους βάσει της ηλικίας τους.

Οι παρακάτω πίνακες παρουσιάζουν τα συνολικά αποτελέσματα για το συνολικό dataset των 286 χρηστών αναφορικά με το πόσους χρήστες από κάθε κατηγορία ηλικίας(τέσσερις κατηγορίες) έχουν οι χρήστες στο top 10% των προτεινόμενων φίλων(29 από τους 285 προτεινόμενους για κάθε χρήστη) από τις εξής μεθόδους: EgoSimilar(weight 0.3), Pearson, Cosine και Kappa. Χωρίσαμε τις ηλικίες των χρηστών στις εξής τέσσερις κατηγορίες: 17 - 20, 21 - 22, 23 - 26, > 26.

Σε όλες τις παραπάνω μεθόδους οι κατηγορίες έχουν βάρος 0.25 και τα αντικείμενα βάρος 0.75.

Στις ηλικίες 17 - 20 έχουμε 84 χρήστες, στις ηλικίες 21 - 22 102 χρήστες, στις ηλικίες 23 - 26 66 χρήστες και στις ηλικίες > 26 33 χρήστες.

ΠΙΝΑΚΑΣ 27. EgoSimilar (Overall Age Statistics)

	<i>17 - 20</i>	<i>21 - 22</i>	<i>23 - 26</i>	<i>> 26</i>
<i>17 - 20</i>	900	858	562	116
<i>21 - 22</i>	1046	1059	693	160
<i>23 - 26</i>	633	681	470	130
<i>> 26</i>	299	322	246	85

Στον EgoSimilar παρατηρούμε ότι οι χρήστες ηλικίας 17 - 20 και 21 - 22 έχουν περισσότερους προτεινόμενους φίλους στο top 10% από την ίδια κατηγορία ηλικίας και λιγότερους από τους χρήστες ηλικίας > 26. Οι χρήστες ηλικίας 23 - 26 και > 26 έχουν περισσότερους προτεινόμενους φίλους από την κατηγορία ηλικίας 21 - 22(στην οποία ανήκουν οι περισσότεροι χρήστες) και στις υπόλοιπες κατηγορίες ηλικίας τα αποτελέσματα είναι ανάλογα με το πόσους χρήστες έχουμε σε αυτές(όπως αναγράφεται παραπάνω).

ΠΙΝΑΚΑΣ 28. Pearson (Overall Age Statistics)

	<i>17 - 20</i>	<i>21 - 22</i>	<i>23 - 26</i>	<i>> 26</i>
<i>17 - 20</i>	799	851	527	257
<i>21 - 22</i>	919	1091	682	260
<i>23 - 26</i>	559	700	419	232
<i>> 26</i>	256	287	259	153

Στην Pearson παρατηρούμε ότι οι χρήστες ηλικίας 17 - 20 και 21 - 22 και 23 - 26 έχουν περισσότερους προτεινόμενους φίλους στο top 10% σε όλες τις κατηγορίες ηλικίας ανάλογα με το πόσους χρήστες έχουμε σε αυτές. Το ίδιο συμβαίνει και με τους χρήστες ηλικίας > 26, με τη διαφορά ότι έχουν περισσότερους προτεινόμενους φίλους στην κατηγορία ηλικίας 23 - 26 από ότι στην κατηγορία 17 - 20.

ΠΙΝΑΚΑΣ 29. Cosine (Overall Age Statistics)

	<i>17 - 20</i>	<i>21 - 22</i>	<i>23 - 26</i>	<i>> 26</i>
<i>17 - 20</i>	760	812	626	238
<i>21 - 22</i>	864	1023	783	288
<i>23 - 26</i>	563	646	510	195
<i>> 26</i>	276	310	242	123

Στην Cosine παρατηρούμε ότι οι χρήστες ηλικίας 17 - 20 και 21 - 22 και 23 - 26 έχουν περισσότερους προτεινόμενους φίλους στο top 10% σε όλες τις κατηγορίες ηλικίας ανάλογα με το πόσους χρήστες έχουμε σε αυτές.

ΠΙΝΑΚΑΣ 30. Kappa (Overall Age Statistics)

	<i>17 - 20</i>	<i>21 - 22</i>	<i>23 - 26</i>	<i>> 26</i>
<i>17 - 20</i>	767	911	536	217
<i>21 - 22</i>	955	1040	710	248
<i>23 - 26</i>	547	699	450	216
<i>> 26</i>	252	296	257	149

Στην Kappa παρατηρούμε ότι οι χρήστες ηλικίας 17 - 20 και 21 - 22 και 23 - 26 έχουν περισσότερους προτεινόμενους φίλους στο top 10% σε όλες τις κατηγορίες ηλικίας ανάλογα με το πόσους χρήστες έχουμε σε αυτές. Το ίδιο συμβαίνει και με τους χρήστες ηλικίας > 26, με τη διαφορά ότι έχουν περισσότερους προτεινόμενους φίλους στην κατηγορία ηλικίας 23 - 26 από ότι στην κατηγορία 17 - 20.

ΠΙΝΑΚΑΣ 31. Kappa2 (Overall Age Statistics)

	<i>17 - 20</i>	<i>21 - 22</i>	<i>23 - 26</i>	<i>> 26</i>
<i>17 - 20</i>	725	836	595	274
<i>21 - 22</i>	893	985	780	291
<i>23 - 26</i>	509	659	499	243
<i>> 26</i>	237	279	264	172

Στην Kappa2 παρατηρούμε ότι οι χρήστες ηλικίας 17 - 20 και 21 - 22 και 23 - 26 έχουν περισσότερους προτεινόμενους φίλους στο top 10% σε όλες τις κατηγορίες ηλικίας ανάλογα με το πόσους χρήστες έχουμε σε αυτές. Το ίδιο συμβαίνει και με τους χρήστες ηλικίας > 26, με τη διαφορά ότι έχουν περισσότερους προτεινόμενους φίλους στην κατηγορία ηλικίας 23 - 26 από ότι στην κατηγορία 17 - 20.

Οι παρακάτω πίνακες παρουσιάζουν τα συνολικά μέσα αποτελέσματα για το συνολικό dataset των 286 χρηστών αναφορικά με το πόσους χρήστες από κάθε κατηγορία ηλικίας έχουν οι χρήστες στο top 10% των προτεινόμενων φίλων (29 από τους 285 προτεινόμενους για κάθε χρήστη) από τις εξής μεθόδους: EgoSimilar(weight 0.3), Pearson, Cosine και Kappa. Παρουσιάζουν για κάθε ηλικία το ποσοστό των προτεινόμενων φίλων ανά χρήστη.

Με βάση τον αριθμό των χρηστών σε κάθε ηλικιακή κατηγορία, τα ποσοστά επί του συνόλου των χρηστών είναι: 29.47% (17-20), 35.78% (21-22), 23.15% (23-26), 11.57% (>26).

ΠΙΝΑΚΑΣ 32. EgoSimilar (Age Statistics / user)

	<i>17 - 20</i>	<i>21 - 22</i>	<i>23 - 26</i>	<i>> 26</i>
<i>17 - 20</i>	36.93%	35.2%	23.06%	4.75%
<i>21 - 22</i>	35.34%	35.79%	23.41%	5.37%
<i>23 - 26</i>	33.06%	35.55%	24.55%	6.75%
<i>> 26</i>	31.24%	33.62%	25.68%	8.86%

Παρατηρούμε και στις τέσσερις κατηγορίες ότι το ποσοστό που αντιστοιχεί στην κατηγορία 17 - 20 είναι υψηλότερο από το 29.47% που είναι η αναλογία των χρηστών 17 - 20 επί του συνόλου. Επίσης, παρατηρούμε και στις τέσσερις κατηγορίες ότι το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι αρκετά χαμηλότερο από το 11.57% που είναι η αναλογία των χρηστών > 26 επί του συνόλου.

ΠΙΝΑΚΑΣ 33. Pearson (Age Statistics / user)

	17 - 20	21 - 22	23 - 26	> 26
17 - 20	32.79%	34.93%	21.62%	10.51%
21 - 22	31.03%	36.86%	23.03%	8.75%
23 - 26	29.17%	36.55%	21.86%	12.1%
> 26	26.72%	29.96%	27.03%	15.96%

Παρατηρούμε στις δύο πρώτες κατηγορίες ότι το ποσοστό που αντιστοιχεί στην κατηγορία 17 - 20 είναι υψηλότερο από το 29.47% που είναι η αναλογία των χρηστών 17 - 20 επί του συνόλου και χαμηλότερο στις επόμενες δύο. Επίσης, παρατηρούμε στην κατηγορία > 26 ότι το ποσοστό που αντιστοιχεί στην κατηγορία 21 - 22 είναι μικρότερο από το 35.78% που είναι η αναλογία των χρηστών 21 - 22 επί του συνόλου και ότι το ποσοστό που αντιστοιχεί στην κατηγορία 23 - 26 είναι μεγαλύτερο από το 23.15% που είναι η αναλογία των χρηστών 23 - 26 επί του συνόλου. Ακόμα, παρατηρούμε στην κατηγορία > 26 ότι το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι αρκετά μεγαλύτερο από το 11.57% που είναι η αναλογία των χρηστών > 26 επί του συνόλου. Στην κατηγορία 21 - 22 το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι μικρότερο από το 11.57% που είναι η αναλογία των χρηστών > 26 επί του συνόλου.

ΠΙΝΑΚΑΣ 34. Cosine (Age Statistics / user)

	17 - 20	21 - 22	23 - 26	> 26
17 - 20	31.17%	33.31%	25.68%	9.75%
21 - 22	29.2%	35.17%	26.44%	9.72%
23 - 26	29.41%	33.72%	26.62%	10.17%
> 26	28.82%	32.37%	25.27%	12.82%

Συγκρίνοντας με τις υπόλοιπες μεθόδους, δεν παρατηρούμε μεγάλες διαφορές των ποσοστών που αντιστοιχούν στις κατηγορίες αναφορικά με την αναλογία των χρηστών κάθε κατηγορίας επί του συνόλου. Παρατηρούμε στην κατηγορία 17 - 20 ότι το ποσοστό που αντιστοιχεί στην κατηγορία 17 - 20 είναι λίγο υψηλότερο από το 29.47% που είναι η αναλογία των χρηστών 17 - 20 επί του συνόλου. Επίσης, και στις τέσσερις κατηγορίες το ποσοστό που αντιστοιχεί στην κατηγορία 23 - 26 είναι μεγαλύτερο από το 23.15% που είναι η αναλογία των χρηστών 23 - 26 επί του συνόλου.

ΠΙΝΑΚΑΣ 35. Kappa (Age Statistics / user)

	17 - 20	21 - 22	23 - 26	> 26
17 - 20	31.48%	37.37%	22.0%	8.89%
21 - 22	32.27%	35.13%	24.0%	8.37%
23 - 26	28.55%	36.51%	23.48%	11.27%
> 26	26.31%	30.89%	26.82%	15.55%

Παρατηρούμε στις δύο πρώτες κατηγορίες ότι το ποσοστό που αντιστοιχεί στην κατηγορία 17 - 20 είναι λίγο υψηλότερο από το 29.47% που είναι η αναλογία των χρηστών 17 - 20 επί του συνόλου και στην κατηγορία > 26 ότι είναι λίγο χαμηλότερο. Επίσης, στην κατηγορία > 26 το ποσοστό που αντιστοιχεί στην κατηγορία 21 - 22 είναι αρκετά μικρότερο από το 35.78% που είναι η αναλογία των χρηστών 21 - 22 επί του συνόλου και το ποσοστό που αντιστοιχεί στην κατηγορία 23 - 26 είναι μεγαλύτερο από το 23.15% που είναι η αναλογία των χρηστών 23 - 26 επί του συνόλου. Ακόμη, στις δύο πρώτες κατηγορίες το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι μικρότερο από το 11.57% που είναι η αναλογία των χρηστών > 26 επί

του συνόλου και στην κατηγορία > 26 το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι μεγαλύτερο από το 11.57.

ΠΙΝΑΚΑΣ 36. Kappa2 (Age Statistics / user)

	17 - 20	21 - 22	23 - 26	> 26
17 - 20	29.75%	34.31%	24.41%	11.24%
21 - 22	30.17%	33.27%	26.34%	9.82%
23 - 26	26.58%	34.41%	26.06%	12.68%
> 26	24.75%	29.13%	27.58%	17.96%

Παρατηρούμε στις δύο τελευταίες κατηγορίες ότι το ποσοστό που αντιστοιχεί στην κατηγορία 17 - 20 είναι χαμηλότερο από το 29.47% που είναι η αναλογία των χρηστών 17 - 20 επί του συνόλου. Επίσης, στην κατηγορία > 26 το ποσοστό που αντιστοιχεί στην κατηγορία 21 - 22 είναι αρκετά μικρότερο από το 35.78% που είναι η αναλογία των χρηστών 21 - 22 επί του συνόλου. Και στις τέσσερις κατηγορίες τα ποσοστά που αντιστοιχούν στην κατηγορία 23 - 26 είναι μεγαλύτερα από το 23.15% που είναι η αναλογία των χρηστών 23 - 26 επί του συνόλου, με τη μεγαλύτερη διαφορά να παρατηρείται στην κατηγορία > 26. Ακόμη, στην κατηγορία 21 - 22 το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι μικρότερο από το 11.57% που είναι η αναλογία των χρηστών > 26 επί του συνόλου και στην κατηγορία > 26 το ποσοστό που αντιστοιχεί στην κατηγορία > 26 είναι αρκετά μεγαλύτερο από το 11.57%.

Αν διατρέξουμε τα αποτελέσματα στους παραπάνω πίνακες «διαγώνια» εξετάζοντας τα ποσοστά των χρηστών ίδιας ηλικίας που προτείνονται στο top 10% των πιθανών φίλων, βλέπουμε ότι ο EgoSimilar βρίσκεται στη μέση (3^{ος}, πίσω από την Pearson και την Kappa2) στο άθροισμα των αντίστοιχων ποσοστών, με τη διαφορά πάντως να είναι μικρή. Αυτό δείχνει ότι η κοινή ηλικία των χρηστών παίζει ρόλο στην πρόταση φίλων από τον EgoSimilar, αλλά ο ρόλος αυτός δεν είναι κυρίαρχος.

3.6 Κατηγοριοποίηση χρηστών βάσει φύλου

Αντίστοιχα με την προηγούμενη εκδοχή του EgoSimilar, μπορούμε να τον χρησιμοποιήσουμε για να προτείνει φίλους βάσει του φύλου τους.

Οι παρακάτω πίνακες παρουσιάζουν τα συνολικά αποτελέσματα για το συνολικό dataset των 286 χρηστών αναφορικά με το πόσοι χρήστες από το top 10% των προτεινόμενων φίλων (29 από τους 285 προτεινόμενους για κάθε χρήστη) είναι από το ίδιο ή το αντίθετο φύλο από τις εξής μεθόδους: EgoSimilar(weight 0.3), Pearson, Cosine και Kappa. Οι άντρες είναι 183 και οι γυναίκες 103.

Σε όλες τις παραπάνω μεθόδους οι κατηγορίες έχουν βάρος 0.25 και τα αντικείμενα βάρος 0.75.

ΠΙΝΑΚΑΣ 37. EgoSimilar (Overall Gender Statistics)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	3926	1381
<i>Women</i>	1360	1627

Στον EgoSimilar παρατηρούμε ότι οι άντρες έχουν περισσότερους προτεινόμενους φίλους στο top 10% οι οποίοι προέρχονται από το ίδιο φύλο με αυτούς. Το ίδιο συμβαίνει και με τις γυναίκες, όπου παρατηρείται μικρότερη διαφορά ανάμεσα στο αν οι φίλοι που προτείνονται είναι από το ίδιο ή το αντίθετο φύλο.

ΠΙΝΑΚΑΣ 38. Pearson (Overall Gender Statistics)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	4437	870
<i>Women</i>	951	2036

Στην Pearson παρατηρούμε ότι οι άντρες έχουν περισσότερους προτεινόμενους φίλους στο top 10% οι οποίοι προέρχονται από το ίδιο φύλο με αυτούς. Το ίδιο συμβαίνει και με τις γυναίκες, όπου παρατηρείται μικρότερη διαφορά ανάμεσα στο αν οι φίλοι που προτείνονται είναι από το ίδιο ή το αντίθετο φύλο. Συγκρίνοντας με τον EgoSimilar, παρατηρούμε και στους άντρες και στις γυναίκες περισσότερους προτεινόμενους φίλους από το ίδιο φύλο.

ΠΙΝΑΚΑΣ 39. Cosine (Overall Gender Statistics)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	3784	1523
<i>Women</i>	1035	1952

Στην Cosine παρατηρούμε ότι οι άντρες έχουν περισσότερους προτεινόμενους φίλους στο top 10% οι οποίοι προέρχονται από το ίδιο φύλο με αυτούς. Το ίδιο συμβαίνει και με τις γυναίκες, όπου παρατηρείται μικρότερη διαφορά ανάμεσα στο αν οι φίλοι που προτείνονται είναι από το ίδιο ή το αντίθετο φύλο. Συγκρίνοντας με την Pearson, παρατηρούμε και στους άντρες και στις γυναίκες μικρότερη διαφορά στο αν οι προτεινόμενοι φίλοι είναι από το ίδιο φύλο ή το αντίθετο.

ΠΙΝΑΚΑΣ 40. Kappa (Overall Gender Statistics)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	4056	1251
<i>Women</i>	1354	1633

Στην Kappa παρατηρούμε ότι οι άντρες έχουν περισσότερους προτεινόμενους φίλους στο top 10% οι οποίοι προέρχονται από το ίδιο φύλο με αυτούς. Το ίδιο συμβαίνει και με τις γυναίκες, όπου παρατηρείται μικρότερη διαφορά ανάμεσα στο αν οι φίλοι που προτείνονται είναι από το ίδιο ή το αντίθετο φύλο. Συγκρίνοντας με τον EgoSimilar, παρατηρούμε στους άντρες μεγαλύτερη διαφορά στο αν οι προτεινόμενοι φίλοι είναι από το ίδιο φύλο ή το αντίθετο. Συγκρίνοντας με την Cosine, παρατηρούμε στις γυναίκες μικρότερη διαφορά στο αν οι προτεινόμενοι φίλοι είναι από το ίδιο φύλο ή το αντίθετο και στους άντρες μεγαλύτερη διαφορά.

ΠΙΝΑΚΑΣ 41. Kappa2 (Overall Gender Statistics)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	3009	2298
<i>Women</i>	2553	434

Στην Kappa2 παρατηρούμε ότι οι άντρες έχουν περισσότερους προτεινόμενους φίλους στο top 10% οι οποίοι προέρχονται από το ίδιο φύλο με αυτούς. Δε συμβαίνει το ίδιο με τις γυναίκες, οι οποίες έχουν με μεγάλη διαφορά περισσότερους προτεινόμενους φίλους από το αντίθετο φύλο(το οποίο δε συμβαίνει στις άλλες μεθόδους). Συγκρίνοντας με τις άλλες μεθόδους, παρατηρούμε στους άντρες μικρότερη διαφορά στο αν οι προτεινόμενοι φίλοι είναι από το ίδιο φύλο ή το αντίθετο.

Οι παρακάτω πίνακες παρουσιάζουν τα συνολικά μέσα αποτελέσματα για το συνολικό dataset των 286 χρηστών αναφορικά με το πόσοι χρήστες από το top 10% των προτεινόμενων φίλων(29 από τους 285 προτεινόμενους για κάθε χρήστη) είναι από το ίδιο ή το αντίθετο φύλο από τις εξής μεθόδους: EgoSimilar(weight 0.3), Pearson, Cosine και Kappa. Παρουσιάζουν το ποσοστό των προτεινόμενων φίλων ανά χρήστη. Με βάση τον αριθμό των χρηστών ανά φύλο, τα ποσοστά επί του συνόλου των χρηστών είναι: 63.98% (άντρες) και 36.01% (γυναίκες).

ΠΙΝΑΚΑΣ 42. EgoSimilar (Gender Statistics / user)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	73.96%	26.0%
<i>Women</i>	45.51%	54.44%

Παρατηρούμε στους προτεινόμενους φίλους των αντρών ότι η διαφορά στο ποσοστό των αντρών είναι περίπου 10% συγκρίνοντας με το 63.98% που είναι η αναλογία των αντρών επί του συνόλου. Ίδια είναι περίπου και η διαφορά στο ποσοστό των γυναικών συγκρίνοντας με το 36.01% που είναι η αναλογία των γυναικών επί του συνόλου. Όσον αφορά στους προτεινόμενους φίλους των γυναικών, έχουμε να κάνουμε αντίστοιχες παρατηρήσεις. Η διαφορά στα ποσοστά και των δύο συγκρίνοντας με την αναλογία επί του συνόλου είναι περίπου 18%.

ΠΙΝΑΚΑΣ 43. Pearson (Gender Statistics / user)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	83.58%	16.37%
<i>Women</i>	31.82%	68.13%

Παρατηρούμε στους προτεινόμενους φίλους των αντρών ότι η διαφορά στο ποσοστό των αντρών είναι περίπου 20% συγκρίνοντας με το 63.98% που είναι η αναλογία των αντρών επί του συνόλου. Ίδια είναι περίπου και η διαφορά στο ποσοστό των γυναικών συγκρίνοντας με το 36.01% που είναι η αναλογία των γυναικών επί του συνόλου. Όσον αφορά στους προτεινόμενους φίλους των γυναικών, έχουμε να κάνουμε αντίστοιχες παρατηρήσεις. Η διαφορά στα ποσοστά και των δύο συγκρίνοντας με την αναλογία επί του συνόλου είναι περίπου 32%.

ΠΙΝΑΚΑΣ 44. Cosine (Gender Statistics / user)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	71.27%	28.68%
<i>Women</i>	34.62%	65.34%

Παρατηρούμε στους προτεινόμενους φίλους των αντρών ότι η διαφορά στο ποσοστό των αντρών είναι περίπου 8% συγκρίνοντας με το 63.98% που είναι η αναλογία των αντρών επί του συνόλου. Ίδια είναι περίπου και η διαφορά στο ποσοστό των γυναικών συγκρίνοντας με το 36.01% που είναι η αναλογία των γυναικών επί του συνόλου. Όσον αφορά στους προτεινόμενους φίλους των γυναικών, έχουμε να κάνουμε αντίστοιχες παρατηρήσεις. Η διαφορά στα ποσοστά και των δύο συγκρίνοντας με την αναλογία επί του συνόλου είναι περίπου 29%.

ΠΙΝΑΚΑΣ 45. Kappa (Gender Statistics / user)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	76.41%	23.55%
<i>Women</i>	45.31%	54.65%

Παρατηρούμε στους προτεινόμενους φίλους των αντρών ότι η διαφορά στο ποσοστό των αντρών είναι περίπου 13% συγκρίνοντας με το 63.98% που είναι η αναλογία των αντρών επί του συνόλου. Ίδια είναι περίπου και η διαφορά στο ποσοστό των γυναικών συγκρίνοντας με το 36.01% που είναι η αναλογία των γυναικών επί του συνόλου. Όσον αφορά στους προτεινόμενους φίλους των γυναικών, έχουμε να κάνουμε αντίστοιχες παρατηρήσεις. Η διαφορά στα ποσοστά και των δύο συγκρίνοντας με την αναλογία επί του συνόλου είναι περίπου 18%.

ΠΙΝΑΚΑΣ 46. Kappa2 (Gender Statistics / user)

	<i>Men</i>	<i>Women</i>
<i>Men</i>	56.68%	43.27%
<i>Women</i>	85.44%	14.51%

Παρατηρούμε στους προτεινόμενους φίλους των αντρών ότι η διαφορά στο ποσοστό των αντρών είναι περίπου 7% συγκρίνοντας με το 63.98% που είναι η αναλογία των αντρών επί του συνόλου. Ίδια είναι περίπου και η διαφορά στο ποσοστό των γυναικών συγκρίνοντας με το 36.01% που είναι η αναλογία των γυναικών επί του συνόλου. Όσον αφορά στους προτεινόμενους φίλους των γυναικών, έχουμε να κάνουμε αντίστοιχες παρατηρήσεις. Η διαφορά στα ποσοστά και των δύο συγκρίνοντας με την αναλογία επί του συνόλου είναι περίπου 22%. Η διαφορά με τους προτεινόμενους φίλους των αντρών είναι ότι οι γυναίκες έχουν περισσότερους προτεινόμενους φίλους οι οποίοι προέρχονται από το αντίθετο φύλο.

Τα αποτελέσματα στους παραπάνω πίνακες δείχνουν ότι σε σχέση με όλες τις υπόλοιπες μετρικές, ο αλγόριθμος του EgoSimilar δίνει την πιο «ισορροπημένη» πρόταση φίλων σε σχέση με το φύλο τους, στους χρήστες του e-Match.

4. ΠΑΡΟΜΟΙΕΣ ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

Υπάρχουν παρόμοιες εφαρμογές με το e-Match, των οποίων η κατεύθυνση στρέφεται περισσότερο προς τον εντοπισμό των φίλων επάνω στον χάρτη και πολύ λιγότερο προς το ταίριασμα των φίλων. Μια τέτοια εφαρμογή είναι η [7]. Ακόμη και το facebook πρόσθεσε για ένα χρονικό διάστημα την επιλογή του χάρτη όμως την απενεργοποίησε για λόγους ασφαλείας, όπως αναφέρεται στο [8] και στο [9].

Όπως εξηγήθηκε και στο [2], οι διαφορές του eMatch με αυτές τις εφαρμογές είναι ότι το eMatch επικεντρώνεται στο matching των ενδιαφερόντων και ότι την τοποθεσία την αποθηκεύει για καθαρά πρακτικούς λόγους. Δηλαδή ο λόγος δεν είναι να μαρκάρει πάνω στον χάρτη την θέση κάθε χρήστη και έπειτα να την εμφανίζει στους υπόλοιπους χρήστες, αλλά να χρησιμοποιεί αυτή την θέση ώστε να βρίσκει πιθανούς φίλους σε ίδιες περιοχές. Αυτό χρησιμεύει στην διευκόλυνση αυτών των πιθανών φίλων να συναντηθούν και να γνωριστούν. Πέραν αυτού όμως, η καθαρή θέση που έχει ένας χρήστης δεν αποκαλύπτεται πουθενά μέσα στην εφαρμογή. Δηλαδή η μόνη πληροφορία που παρέχεται είναι ότι ο άλλος χρήστης βρίσκεται κάπου κοντά (σε ακτίνα 16 χιλιομέτρων).

Βέβαια έχουν παρθεί κάποια μέτρα για να μην παρουσιαστούν προβλήματα ασφαλείας. Το πρώτο είναι ότι δίνεται η επιλογή στον χρήστη να μπει σε κατάσταση «Non Visible». Όταν ενεργοποιηθεί αυτή η επιλογή, ακόμη και αν ο χρήστης έχει ανοιχτή την εφαρμογή δεν θα εμφανίζεται σε κανένα αποτέλεσμα matching. Το δεύτερο είναι ότι στα άτομα τα οποία δεν είναι συνδεδεμένα με σχέση φιλίας αποκρύπτεται η πληροφορία των ενδιαφερόντων από τα προφίλ, οπότε και να θελήσει κάποιος χρήστης κακοπροαίρετα να προσεγγίσει έναν άλλο δεν θα μπορεί να έχει καμία πρόσβαση στα ενδιαφέροντα του και να εκμεταλλευτεί αυτή τη πληροφορία και να εισάγει παρόμοια ενδιαφέροντα ώστε να πετύχει πλαστό υψηλό matching. Αυτό αποτελεί στοιχειώδη προστασία, καθώς την φιλία πρέπει να την αποδεχτούν και οι δύο χρήστες.

Στην παρούσα διπλωματική εστίασαμε στην επεξεργασία των στοιχείων από αντλήθηκαν από ένα σημαντικό αριθμό (229) νέων ερωτηματολογίων που συλλέχθηκαν από χρήστες διαφορετικών ηλικιών, με στόχο την εύρεση ανθρώπων με κοινά ενδιαφέροντα μέσω του e-Match. Ο στόχος της διπλωματικής ήταν να ελέγξει την ισχύ των αποτελεσμάτων που είχαν εξαχθεί στα [2, 3] για περιορισμένο αριθμό χρηστών και να εξερευνήσει νέες μετρικές και τεχνικές που θα μπορούσαν να βελτιώσουν τον αλγόριθμο EgoSimilar της εφαρμογής e-Match.

Τα αποτελέσματά μας επιβεβαιώνουν ότι ο EgoSimilar υπερτερεί σε σχέση με πλήθος άλλων μετρικών από την βιβλιογραφία, όπως είναι η συσχέτιση Pearson, η μετρική Cosine, οι συντελεστές Kappa και Pi και ο δείκτης Jaccard. Μετρικές οι οποίες χρησιμοποιούνται ευρέως για τη μέτρηση ομοιότητας.

Σε κάποια σημεία παρατηρήσαμε να υπερέχουν άλλες μετρικές και όχι ο EgoSimilar, αλλά αν εξετάσουμε συνολικά τα αποτελέσματα και τις αναλύσεις τους προτιμούμε τον EgoSimilar για να πετύχουμε το σκοπό μας. Για παράδειγμα, όσον αφορά στο cluster evaluation για βάρος κατηγοριών $w1 = 0.25$ και βάρος αντικειμένων $w2 = 0.75$ και όσον αφορά στα συνολικά στατιστικά ταιριάσματος για $(w1, w2) = (0.25, 0.75), (0.5, 0.5), (0.75, 0.25)$ παρατηρούμε ότι ο EgoSimilar είναι δεύτερος πίσω από την Cosine (αλλά καλύτερος από τις υπόλοιπες μεθόδους). Όσον αφορά στο average friends' placement όμως, το οποίο μας ενδιαφέρει περισσότερο από όλα, για όλα τα προαναφερθέντα $(w1, w2)$ η μέθοδος Cosine έχει τα χειρότερα αποτελέσματα μεταξύ των υπολοίπων μεθόδων.

Για $(w1, w2) = (0.25, 0.75)$ οι δύο μέθοδοι Kappa και η Pearson παρουσιάζουν καλύτερο average friends' placement από τον EgoSimilar. Αν αυξήσουμε το βάρος στις κατηγορίες όμως $(w1 = 0.5, 0.75)$, ο EgoSimilar πετυχαίνει το καλύτερο average friends' placement από όλες τις μετρικές. Όταν χρησιμοποιούμε τον EgoSimilar επίσης για να υπολογίσει το matching βάσει μόνο των δύο πιο «αγαπημένων» κατηγοριών των χρηστών, τα

αποτελέσματά του είναι βελτιωμένα και το average friends' placement ακόμα μικρότερο. Ακόμα, είδαμε όταν κάνουμε κατηγοριοποίηση των χρηστών βάσει της ηλικίας τους ότι η κοινή ηλικία των χρηστών παίζει ρόλο στην πρόταση φίλων από τον EgoSimilar, ο οποίος με μικρή διαφορά δεν είναι πρώτος στην πρόταση αυτή, ενώ στην κατηγοριοποίηση βάσει φύλου δίνει όσον αφορά στα ποσοστά των προτεινόμενων φίλων ανά χρήστη την πιο «ισορροπημένη» πρόταση φίλων στους χρήστες του e-Match.

REFERENCES

- [1] [Online]: <https://play.google.com/store/apps/details?id=com.tuc.eMatch>
- [2] Γεωργία Αθανασοπούλου, “E-match: A new Android Application for finding friends in your location”, Διπλωματική Εργασία, Τμήμα ΗΜΜΥ Πολυτεχνείου Κρήτης, 2012.
- [3] G. Athanasopoulou and P. Koutsakis, “eMatch: An Android Application for Finding Friends in your Location”, Mobile Information Systems Journal (Hindawi Publ.), Vol. 2015, Article ID 463791.
- [4] L. Mekouar, Y. Iraqi, and R. Boutaba, “An analysis of peer similarity for recommendations in P2P systems,” Multimedia Tools and Applications, vol. 60, no. 2, pp. 277–303, 2012.
- [5] P. Jaccard, "Nouvelles Recherches Sur la Distribution Florale," Bulletin de la Societe Vaudoise des Sciences Naturelles, Vol. 44, 1908, pp. 223-270.
- [6] R. Artstein and M. Poesio, “Bias decreases in proportion to the number of annotators”, [Online]: <http://www.artstein.org/publications/bias-a5.pdf>
- [7][Online]: http://www.androidzoom.com/android_applications/travel_and_local/nearby-friend-finder_bhhtq.html
- [8] [Online]: <http://www.wired.com/gadgetlab/2012/06/facebook-quietly-releases-find-friends-nearby-then-quietly-pulls-it/>
- [9] [Online]: <http://www.dailymail.co.uk/sciencetech/article-2164366/Facebook-kills-Find-Friends-Nearby-feature-stalking-fears-GPS-app.html>