

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Στοχευμένη Πρόταση Προϊόντων σε Χρήστες
Κοινωνικών Δικτύων**

ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΩΝΣΤΑΝΤΟΠΟΥΛΟΣ Α.Μ.: 3960

Υπεύθυνος Καθηγητής:
κ. Αθανάσιος Τσακαλίδης

ΠΑΤΡΑ, ΙΟΥΝΙΟΣ 2013

Copyright © 2013 Κωνσταντίνος Κωνσταντόπουλος

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πατρών.

Ευχαριστίες

Ολοκληρώνοντας την παρούσα Διπλωματική Εργασία, θα ήθελα να ευχαριστήσω θερμά τον κ. Τσακαλίδη Αθανάσιο, Καθηγητή και Πρόεδρο του τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών, για την υποστήριξη και την καθοδήγησή του, κατά τη διάρκεια εκπόνησης της εργασίας. Ως επιβλέπων καθηγητής, συνέβαλλε σημαντικά στην ολοκλήρωσή της με συνέπεια και διάθεση.

Επιπλέον ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στους Δημήτρη Μητροβγένη, Πολυχρόνη Καρποδίνη και Αθανάσιο Νικολακόπουλο για την πολύτιμη βοήθειά τους και τις γνώσεις που απέκτησα κατά τη διάρκεια της συνεργασίας μας, οι οποίες με βοήθησαν σημαντικά στην ολοκλήρωση της εργασίας αυτής.

Τέλος, θα επιθυμούσα να ευχαριστήσω τους γονείς μου, οι οποίοι μου συμπαραστάθηκαν σε πολύ μεγάλο βαθμό, όχι μόνο κατά τη διάρκεια της Διπλωματικής μου Εργασίας, αλλά και καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Η συνεχώς αυξανόμενη τάση των Χρηστών του Διαδικτύου για συμμετοχή σε ένα ή περισσότερα Κοινωνικά Δίκτυα έχει οδηγήσει σε νέες μορφές δικτύωσης. Όλο και περισσότεροι Χρήστες περνούν, άλλοτε λιγότερο και άλλοτε περισσότερο, χρόνο συνδεδεμένοι σε αυτά ενώ αναπόφευκτα μοιράζονται δεδομένα. Τα Συστήματα Σύστασης προσπαθούν να εξομοιώσουν τις προτιμήσεις των Χρηστών έχοντας σαν στόχο την εκτίμηση του πόσο ενδιαφέροντα θα είναι κάποια αντικείμενα, πληροφορίες, ή ακόμα και υπηρεσίες για το Χρήστη και την υποβολή προτάσεων για στοιχεία που θα αγοράσει ένα άτομο ή θα εξετάσει. Αυτά τα συστήματα χρησιμοποιούνται συνεχώς στο διαδίκτυο και έχει γίνει βασικό συστατικό πλέον του ηλεκτρονικού εμπορίου και της ανάκτησης πληροφοριών, παρέχοντας προτάσεις που αποτελεσματικά φιλτράρουν μεγάλους χώρους πληροφοριών ώστε κάθε χρήστης να μπορεί να κατευθύνεται προς τα στοιχεία που ανταποκρίνονται με τον καλύτερο τρόπο στις ανάγκες και τις προτιμήσεις του. Η τρομακτική αύξηση των πληροφοριών που είναι διαθέσιμες στο διαδίκτυο, και κατ' επέκταση στα Κοινωνικά Δίκτυα, καθώς και ο συνεχώς αυξανόμενος αριθμός Χρηστών, θέτουν ενδιαφέροντες προκλήσεις. Για τις ανάγκες της παρούσας Διπλωματικής Εργασίας προτιμήθηκε η Πρόταση μιας Εκδήλωσης. Προκειμένου να γίνει δυνατή η υλοποίηση της εφαρμογής πρέπει να ληφθούν υπόψη τα δύο βασικά συστατικά της μέρη. Το πρώτο κομμάτι αφορά την υλοποίηση μιας εφαρμογής μέσω του Κοινωνικού Δικτύου Facebook ώστε να γίνει λήψη των δεδομένων. Το δεύτερο κομμάτι αφορά στην υλοποίηση του συστήματος προτάσεων που βασίζεται στην επεξεργασία των δεδομένων που ληφθήσαν από το Κοινωνικό Δίκτυο. Στόχος της παρούσας Διπλωματικής Εργασίας είναι η μελέτη και ανάπτυξη μιας εφαρμογής που θα επεξεργάζεται, τα διαθέσιμα από τα Κοινωνικά Δίκτυα, δεδομένα και τελικά θα προτείνει ένα ή περισσότερα προϊόντα στο Χρήστη του Κοινωνικού Δικτύου σύμφωνα με τα δεδομένα τόσο του ίδιου του Χρήστη όσο και από άλλους χρήστες.

Πίνακας Περιεχομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ	1
Ευχαριστίες.....	4
Περίληψη.....	5
Πίνακας Περιεχομένων	7
Πίνακας Εικόνων	9
1. Εισαγωγή	11
1.1 Συστήματα Συστάσεων.....	11
1.2 Κοινωνικά Δίκτυα	12
1.3 Η Επιλογή του Κοινωνικού Δικτύου Facebook.....	14
2. Τεχνολογίες που Χρησιμοποιήθηκαν.....	17
2.1 Εισαγωγή	17
2.2 Διεπαφή Χρήστη.....	17
2.2.1 PHP	17
2.2.2 HTML	19
2.2.3 CSS	21
2.2.4 MySQL.....	22
2.2.5 JavaScript.....	23
2.2.6 jQuery	24
2.3 Υλοποίηση του Αλγορίθμων της Συνεργατικής Διήθησης	25
2.3.1 MATLAB	25
3. Ανάλυση και Σχεδιασμός της Εφαρμογής.....	27
3.1 Εισαγωγή	27
3.2 Εφαρμογή στο Κοινωνικό Δίκτυο	27
3.2.1 Επιλογή Κοινωνικού Δικτύου και Δεδομένων.....	27
3.2.2 Ασφάλεια και Άδειες Δεδομένων	28
3.2.3 Σχεδιασμός Διεπαφής Χρήστη	29
3.2.4 Εργασίες στο Παρασκήνιο.....	30
3.2.5 Εξαγωγή Δεδομένων	32
3.2.6 Απαιτήσεις Συστήματος	32
3.3 Σύνολο Δεδομένων (Dataset).....	33
3.4 Αλγόριθμος Συνεργατικής Διήθησης	35

3.4.1	Προεπεξεργασία Δεδομένων	35
3.4.2	Περιγραφή Αλγορίθμου	37
3.4.3	Τελική Πρόταση και Μέτρηση Απόδοσης	43
4.	Υλοποίηση	47
4.1	Εισαγωγή	47
4.2	Δημιουργία Διεπαφής Χρήστη	47
4.2.1	Κατοχύρωση της Εφαρμογής στο Facebook	47
4.2.2	Ανάπτυξη Κώδικα	51
4.2.3	Τελικό Αποτέλεσμα	57
4.3	Υλοποίηση Αλγορίθμου Συνεργατικής Διήθησης	58
4.3.1	Προεπεξεργασία Δεδομένων	59
4.3.2	Αλγόριθμος Συνεργατικής Διήθησης	62
4.3.3	Τελική Πρόταση και Μέτρηση Απόδοσης	64
5.	Συμπεράσματα	69
6.	Επεκτάσεις και Μελλοντικές Εφαρμογές	71
6.1	Επεκτάσεις	71
6.2	Μελλοντικές Εφαρμογές	72
	Βιβλιογραφία	75

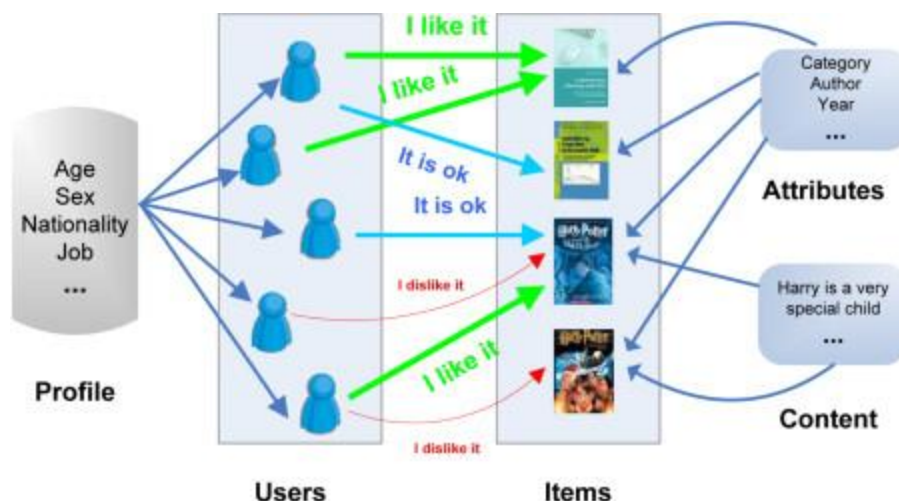
Πίνακας Εικόνων

Εικόνα 1.1: Απλό παράδειγμα Συστήματος Σύστασης.....	11
Εικόνα 1.2: Διάσημα Κοινωνικά Δίκτυα	13
Εικόνα 1.3: Λογότυπο Κοινωνικού Δικτύου Facebook.....	14
Εικόνα 2.1: Λογότυπο γλώσσας προγραμματισμού PHP.....	18
Εικόνα 2.2: Λογότυπο HTML5	20
Εικόνα 2.3: Λογότυπο CSS3	22
Εικόνα 2.4: Λογότυπο MySQL	22
Εικόνα 2.5: Λογότυπο MatLab	25
Εικόνα 3.1: Μορφή του Συνόλου Δεδομένων ως είσοδος για τον Αλγόριθμο Συνεργατικής Διήθησης	36
Εικόνα 3.2: Συσχέτιση Αντικειμένων στο Μητρώο Ομοιοτήτων	41
Εικόνα 4.1: Δημιουργία Facebook App - Βήμα 1	48
Εικόνα 4.2: Δημιουργία Facebook App - Βήμα 2	49
Εικόνα 4.3: Δημιουργία Facebook App - Βήμα 3	50
Εικόνα 4.4: ER διάγραμμα της Βάσης Δεδομένων	54
Εικόνα 4.5: Τελική εμφάνιση εφαρμογής - Προβολή Χρήστη	58
Εικόνα 4.6: Δισδιάστατος πίνακας δεδομένων Αλγορίθμου Συνεργατικής Διήθησης.....	59

1. Εισαγωγή

1.1 Συστήματα Συστάσεων

Το πρόβλημα της Πρότασης Προϊόντων σε Χρήστες (Κοινωνικών Δικτύων και μη) περιγράφεται πλήρως από τα *Συστήματα Συστάσεων*. Τα Συστήματα Συστάσεων (*Recommender Systems, RS*) είναι εργαλεία λογισμικού, τα οποία χρησιμοποιούνται από τα περισσότερα ηλεκτρονικά εμπορικά καταστήματα προκειμένου να κάνουν έξυπνες και γρήγορες εξατομικευμένες συστάσεις στους Χρήστες. Η ανάπτυξη τους ξεκίνησε από μια απλή παρατήρηση: τα άτομα συχνά βασίζονται στις συστάσεις που τους παρέχονται από άλλους, στη λήψη καθημερινών αποφάσεών τους. Για παράδειγμα, είναι σύνηθες άτομα τα οποία θέλουν να διαβάσουν ένα βιβλίο, να βασίζονται στις απόψεις ατόμων που το έχουν ήδη διαβάσει, οι εργοδότες κατά τη διαδικασία πρόσληψης να υπολογίζουν σε επιστολές για την απόφασή τους, και τέλος τα άτομα τα οποία επιθυμούν να δουν μια ταινία, να διαβάζουν και να βασίζονται στις κριτικές που έχει πάρει η ταινία αυτή. Τα συστήματα αυτά μπορούν να ιδωθούν σαν μέθοδοι φιλτραρίσματος πληροφορίας και απευθύνονται είτε σε Χρήστες που θα ήθελαν απλά να δοκιμάσουν ένα νέο «ξένο» σε αυτούς προϊόν είτε σε χρήστες που δε διαθέτουν επαρκή προσωπική εμπειρία ή την ικανότητα να αξιοποιήσουν τον συντριπτικό αριθμό των διαφορετικών αντικειμένων που υπάρχουν διαθέσιμα στο διαδίκτυο.



Εικόνα 1.1: Απλό παράδειγμα Συστήματος Σύστασης

Προκειμένου να παράξουν εξατομικευμένες συστάσεις, τα *Συστήματα Συστάσεων* συλλέγουν από τις προτιμήσεις των Χρηστών, οι οποίες είτε εκφράζονται ρητά, όπως είναι οι βαθμολογίες που δίνουν για τα αντικείμενα

(προϊόντα), ή συμπεραίνονται έμμεσα από την αλληλεπίδραση του χρήστη με το σύστημα. Για παράδειγμα, ένα Σύστημα Σύστασης μπορεί να μετρήσει την πλοήγηση ενός Χρήστη σε μια συγκεκριμένη σελίδα ως μια έμμεση ένδειξη της προτίμησής του προς τα αντικείμενα που εμφανίζονται σε αυτή τη σελίδα.

Για την παραγωγή *Προτάσεων* (ή *Συστάσεων*) έχουν αναπτυχθεί διάφορες τεχνικές. Η πλέον επιτυχημένη είναι η *Συνεργατική Διήθηση* (*Collaborative Filtering, CF*). Η μέθοδος της Συνεργατικής Διήθησης έχει μεγάλη απήχυσή και η ευρεία εφαρμογή της σε σημαντικά εμπορικά περιβάλλοντα έχει οδηγήσει στη σημαντική ανάπτυξη της θεωρίας, από την εμφάνισή της (την τελευταία δεκαετία), όπου μια ευρεία ποικιλία αλγορίθμων και μεθόδων έχουν προταθεί. Το σκεπτικό πίσω από τις μεθόδους της Συνεργατικής Διήθησης είναι πώς αν ο Χρήστης-1 συμφώνησε στο παρελθόν με κάποιους Χρήστες, τότε οι άλλες συστάσεις που προέρχονται από αυτούς θα είναι (μάλλον) σχετικές και κοντά στα ενδιαφέροντά του.

Άλλες σημαντικές τεχνολογίες Συστημάτων Συστάσεων, αποτελούν η *Διήθηση με Βάση το Περιεχόμενο* (*Content-Based Filtering*) και η *Υβριδική* (*Hybrid*) προσέγγιση. Στην περίπτωση της Διήθησης με Βάση το Περιεχόμενο, το σύστημα «μαθαίνει» να προτείνει αντικείμενα στο Χρήστη με βάση τα προϊόντα που έχει αγοράσει στο παρελθόν. Συγκεκριμένα, αναλύει το περιεχόμενο των ήδη βαθμολογημένων (από αυτόν) προϊόντων ή αντικειμένων και χτίζει το προφίλ του. Κατά τη διαδικασία παραγωγής συστάσεων, η Διήθηση με Βάση το Περιεχόμενο ταιριάζει τα χαρακτηριστικά (προτιμήσεις, ιστορικό κ.ά.) που υπάρχουν στο προφίλ του Χρήστη με τα χαρακτηριστικά του περιεχομένου των αντικειμένων που δεν έχει αλληλεπιδράσει ακόμα. Ένα Υβριδικό σύστημα συνδυάζει την Διήθηση με Βάση το Περιεχόμενο και την «κλασσική» Συνεργατική Διήθηση ώστε να χρησιμοποιήσει τα πλεονεκτήματα της μιας προκειμένου να διορθώσει τα μειονεκτήματα της άλλης. Για παράδειγμα, οι μέθοδοι Συνεργατικής Διήθησης υποφέρουν από το πρόβλημα των *Νέων Αντικειμένων* (*Cold-Start Problem*), δηλαδή οι μέθοδοι αυτοί δεν μπορούν να προτείνουν νέα αντικείμενα στους χρήστες εάν αυτά έχουν βαθμολογηθεί λίγο ή και καθόλου. Από την άλλη μεριά, οι τεχνικές Διήθησης με Βάση το Περιεχόμενο μπορούν να προτείνουν στους χρήστες αντικείμενα με πολύ διαφορετικό περιεχόμενο.

1.2 Κοινωνικά Δίκτυα

Ένα *Κοινωνικό Δίκτυο* είναι ένα θεωρητικό κατασκευάσμα χρήσιμο στις κοινωνικές επιστήμες για τη μελέτη των σχέσεων μεταξύ ατόμων, ομάδων, οργανισμών, ή ακόμη και ολόκληρες. Ο όρος χρησιμοποιείται για να περιγράψει μια κοινωνική δομή που καθορίζεται από τέτοιες αλληλεπιδράσεις. Οι δεσμοί μέσω του οποίου κάθε δεδομένη κοινωνική μονάδα συνδέεται,

αντιπροσωπεύουν τη σύγκλιση των διαφόρων κοινωνικών επαφών της εν λόγω μονάδας. Η θεωρητική αυτή προσέγγιση είναι, κατ' ανάγκην, σχεσιακή. Ένα αξίωμα της κοινωνικής προσέγγισης δικτύου για την κατανόηση της κοινωνικής αλληλεπίδρασης είναι ότι τα κοινωνικά φαινόμενα πρέπει πρωτίστως να συλληφθούν και να διερευνηθούν από τις ιδιότητες των σχέσεων μεταξύ και εντός των μονάδων, αντί για τις ιδιότητες αυτών των μονάδων. Έτσι, μια κοινή κριτική της θεωρίας των Κοινωνικών Δικτύων είναι ότι οι επιμέρους οργανισμοί αγνοούνται συχνά αν και αυτό δεν μπορεί να συμβαίνει στην πράξη. Ακριβώς επειδή πολλά διαφορετικά είδη σχέσεων, μεμονωμένες ή σε συνδυασμό, αποτελούν τις ρυθμίσεις του δικτύου, τα στοιχεία δικτύου είναι χρήσιμα σε ένα ευρύ φάσμα ερευνών. Στις κοινωνικές επιστήμες, οι τομείς της μελέτης περιλαμβάνουν (αλλά δεν περιορίζονται) την ανθρωπολογία, τη βιολογία, τις μελέτες της επικοινωνίας, την οικονομία, τη γεωγραφία, την επιστήμη των πληροφοριών, οργανωτικές μελέτες, την κοινωνική ψυχολογία, την κοινωνιολογία και την κοινωνιογλωσσολογία.

Το Κοινωνικό Δίκτυο, σε μια απλοϊκή μορφή, περιγράφεται ως μια κοινωνική δομή που αποτελείται από ένα σύνολο Κοινωνικών Φορέων (όπως τα Άτομα ή Οργανισμοί) και ένα περίπλοκο σύνολο των δυαδικών δεσμών μεταξύ αυτών των παραγόντων. Η προοπτική του Κοινωνικού Δικτύου παρέχει ένα σαφή τρόπο στην ανάλυση της δομής του συνόλου των Κοινωνικών Φορέων. Η μελέτη αυτών των δομών χρησιμοποιεί ανάλυση των κοινωνικών δικτύων για τον εντοπισμό των τοπικών και παγκόσμιων προτύπων, τον εντοπισμό Φορέων επιρροής και, τέλος, εξετάσει τη δυναμική του δικτύου.



Εικόνα 1.2: Διάσημα Κοινωνικά Δίκτυα

Τα Κοινωνικά δίκτυα και η ανάλυσή τους είναι ένας εγγενώς διεπιστημονικός και ακαδημαϊκός τομέας, που προέκυψε από την κοινωνική ψυχολογία, την κοινωνιολογία, τη στατιστική και θεωρία γράφων. Ο Georg Simmel συνέγραψε τις αρχικές διαρθρωτικές θεωρίες στην κοινωνιολογία τονίζοντας τον «ιστό των συνεργασιών της ομάδας» (*web of group affiliations*). Τη δεκαετία του 1930 ο Jacob Moreno πιστώνεται με την ανάπτυξη των πρώτων Γράφων Κοινωνικών Δικτύων (*Sociograms*) για να μελετήσει τις διαπροσωπικές σχέσεις. Αυτές οι προσεγγίσεις μαθηματικά επισημοποιήθηκαν τη δεκαετία του 1950 και διάφορες θεωρίες και μέθοδοι των κοινωνικών δικτύων γενικεύτηκαν στις κοινωνικές επιστήμες τη δεκαετία του 1980. Η ανάλυση των κοινωνικών δικτύων είναι πλέον ένα από τα σημαντικότερα παραδείγματα της σύγχρονης κοινωνιολογίας, και είναι επίσης εφαρμόζεται σε μια σειρά από άλλες κοινωνικές επιστήμες.

1.3 Η Επιλογή του Κοινωνικού Δικτύου Facebook

Την στιγμή που υλοποιήθηκε η παρούσα εργασία και συγγράφεται το κείμενο υπάρχουν διαθέσιμα δεκάδες Κοινωνικά Δίκτυα. Ωστόσο, για τις πραγματικές ανάγκες της υλοποίησης επιλέχθηκε το Κοινωνικό Δίκτυο Facebook.



Εικόνα 1.3: Λογότυπο Κοινωνικού Δικτύου Facebook

Αυτή η επιλογή έγινε για δύο (2) κυρίως λόγους:

1. *Ποιότητα και Ποσότητα Δεδομένων.* Τα Κοινωνικά Δίκτυα είναι μια πολύ μεγάλη δεξαμενή δεδομένων. Η απήχηση του Κοινωνικού Δικτύου Facebook είναι παγκόσμια. Είναι ένα από τα μεγαλύτερα και πολυπληθέστερα Κοινωνικά Δίκτυα και τη στιγμή που γράφεται η παρούσα Διπλωματική Εργασία αριθμεί περισσότερους από ένα δισεκατομμύριο Χρήστες και θεωρείται από τα πιο διαδεδομένα, ενώ καθημερινά οι Χρήστες αυξάνονται κατά χιλιάδες. Επίσης, τα δεδομένα παρέχονται είναι ποιοτικά αφού σε αυτό οι Χρήστες καταγράφουν δημογραφικά στοιχεία, ενδιαφέροντα, χόμπι, δραστηριότητες κ.ά.

Επίσης, μπορούν να προκύψουν χρήσιμα μετα-δεδομένα μέσω των συνδέσεων των Χρηστών με άλλους Χρήστες ή τα υπόλοιπα αντικείμενα του Κοινωνικού Δικτύου.

2. Η *Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface – API)* που παρέχει το συγκεκριμένο δίκτυο. Το Κοινωνικό Δίκτυο Facebook παρέχει το κατάλληλο προγραμματιστικό περιβάλλον (*Facebook API*) σε διάφορες γλώσσες προγραμματισμού. Έτσι, καλύπτεται ένα μεγάλο εύρος τεχνολογιών προσφέροντας ευκολία στους προγραμματιστές. Μέσω αυτής της διεπαφής, δίνεται η δυνατότητα για λήψη (όλων των διαθέσιμων) δεδομένων των Χρηστών και έτσι γίνεται δυνατή η υλοποίηση. Επίσης, παρέχονται η πλήρης έλεγχος και όλες οι δικλίδες ασφαλείας στο Χρήστη, καθώς βρίσκεται στην κρίση του εάν θα παραχωρήσει τα δεδομένα του και ποιά από αυτά.

2. Τεχνολογίες που Χρησιμοποιήθηκαν

2.1 Εισαγωγή

Η Πρόταση Προϊόντων είναι μια διαδικασία που αποτελείται από πολλά επιμέρους μέρη και βήματα. Στην παρούσα εργασία, μπορούμε να διακρίνουμε δύο (2) βασικά μέρη:

1. *Η Διεπαφή με το Χρήστη.* Αρχικά, θα χρησιμοποιήσουμε το Κοινωνικό Δίκτυο Facebook, και τα εργαλεία που αυτό μας παρέχει, για να δώσουμε στο Χρήστη ένα τρόπο να αλληλεπιδράσει με το σύστημα που θα δημιουργήσουμε και τελικά θα μπορέσουμε να δημιουργήσουμε ένα Σύνολο Δεδομένων (Dataset). Σε αυτό θα σημειώσουμε τους Χρήστες και τις προτιμήσεις τους, τα απαραίτητα στοιχεία για να γίνει το επόμενο βήμα.
2. *Υλοποίηση των αλγορίθμων της Συνεργατικής Διήθησης.* Οι αλγόριθμοι της Συνεργατικής Διήθησης στη συνέχεια θα επεξεργαστούν το Σύνολο Δεδομένων που δημιουργήσαμε και θα παράξουν την Πρόταση (ή Προτάσεις) Προϊόντων για το Χρήστη. Εξαιτίας του μεγέθους των δεδομένων και της επεξεργασίας απαιτούν, είναι επιτακτική η χρήση ισχυρών μαθηματικών μοντέλων και τεχνολογιών για να γίνει εφικτή η υλοποίηση.

2.2 Διεπαφή Χρήστη

Όπως αναφέραμε και παραπάνω η Διεπαφή με το Χρήστη είναι η πρώτη επαφή του Χρήστη με το σύστημα. Σε αυτή τη φάση χρησιμοποιήθηκαν οι εξής τεχνολογίες και γλώσσες προγραμματισμού.

2.2.1 PHP

Η *PHP* είναι μια *server-side scripting* γλώσσα προγραμματισμού ^[1] για την ανάπτυξη εφαρμογών Παγκόσμιου Ιστού, αλλά χρησιμοποιείται επίσης ως μια γενικής χρήσης γλώσσα προγραμματισμού. Η *PHP* έχει εγκατασταθεί σε περισσότερες από 244 εκατομμύρια ιστοσελίδες και 2,1 εκατομμύρια εξυπηρετητές διαδικτύου (web servers).

[1] Ως *server-side scripting* γλώσσες προγραμματισμού αναφέρονται οι γλώσσες που χρησιμοποιούνται ευρέως στον προγραμματισμό εφαρμογών διαδικτύου και διερμηνεύονται και υλοποιούνται σε εξυπηρετητές (servers).

Αρχικά δημιουργήθηκε από τον Rasmus Lerdorf το 1995, και πλέον η γλώσσα συντηρείται και διαχειρίζεται από τον Όμιλο PHP (The PHP Group). Ενώ η PHP ήταν αρχικά το ακρωνύμιο του Personal Home Page, τώρα στέκεται ως PHP: Hypertext Preprocessor.



Εικόνα 2.1: Λογότυπο γλώσσας προγραμματισμού PHP

Η ανάπτυξη της PHP ξεκίνησε το 1994, όταν ο προγραμματιστής Rasmus Lerdorf έγραψε μια σειρά από σενάρια κοινής διεπαφής Perl (Common Gateway Interface (CGI) Perl scripts), το οποίο χρησιμοποίησε για τη διατήρηση της προσωπικής ιστοσελίδας του. Αυτά τα πρώτα εργαλεία εκτελούσαν εργασίες, όπως η εμφάνιση του βιογραφικού και την καταγραφή της κίνησης της ιστοσελίδας του.

Έκτοτε, έχουν δημοσιευθεί αρκετές νεότερες εκδόσεις και συνεχώς δίνονται νέες εκδόσεις και επεκτάσεις (extensions) στη δημοσιότητα. Κύριο στοιχείο της είναι η αλληλεπίδραση με πολλές τεχνολογίες όπως είτε πρόκειται για Βάσεις Δεδομένων, όπως οι MySQL, PostgreSQL, είτε για επεκτάσεις που βοηθούν τους προγραμματιστές σε θέματα ασφαλείας (π.χ. OpenSSL και Curl) ή την επικοινωνία μεταξύ διαφορετικών πλατφόρμων (π.χ. php_soap).

Μια νέα σημαντική έκδοση βρίσκεται υπό ανάπτυξη για αρκετά χρόνια. Αυτή η έκδοση είχε αρχικά προγραμματιστεί να κυκλοφορήσει ως PHP 6, ως αποτέλεσμα των σημαντικών μεταβολών της.

Ο PHP κώδικας διερμηνεύεται από έναν εξυπηρετητή διαδικτύου (web server) με μία μονάδα επεξεργαστή PHP η οποία παράγει το αποτέλεσμα ιστοσελίδα: PHP εντολές μπορεί να ενσωματωθούν άμεσα σε ένα HTML έγγραφο και όχι καλώντας ένα εξωτερικό αρχείο για την επεξεργασία δεδομένων. Επίσης, έχει εξελιχθεί και περιλαμβάνει μια διεπαφή γραμμής εντολών και μπορεί να χρησιμοποιηθεί σε αυτόνομες γραφικές εφαρμογές.

Η PHP είναι ελεύθερο λογισμικό που διατίθεται βάσει της Άδειας PHP (PHP License), η οποία είναι ασυμβίβαστη με την GNU General Public License (GPL)

λόγω των περιορισμών σχετικά με τη χρήση του όρου PHP. Η PHP μπορεί να αναπτυχθεί στους περισσότερους εξυπηρετητές διαδικτύου (web servers) και επίσης ως αυτόνομη τεχνολογία σχεδόν σε κάθε λειτουργικό σύστημα και την πλατφόρμα, χωρίς χρέωση. Επίσης, οι διερμηνείς PHP είναι διαθέσιμοι και στις δύο εκδόσεις των λειτουργικών συστημάτων (32-bit και 64-bit).

2.2.1.1 Facebook PHP SDK

Η Facebook προσφέρει εργαλεία, υπηρεσίες και τα *Software Development Kits* (SDK) για προγραμματιστές που θέλουν να ενσωματώσουν κοινωνικές εμπειρίες σε εφαρμογές τους. Το Facebook PHP SDK είναι μια βιβλιοθήκη της PHP που προσφέρει εργαλεία για να καταστήσει την ολοκλήρωση εύκολη την επικοινωνία με το Κοινωνικό Δίκτυο. Επιτρέπει τη δημιουργία εφαρμογών στο κοινωνικό δίκτυο (apps) και εξυπηρετεί τον απώτερο στόχο της πρότασης προϊόντων. Έτσι, χρησιμοποιούμε την ισχυρή γλώσσα προγραμματισμού PHP και τη συνδυάζουμε με το Facebook PHP SDK για να πετύχουμε το βέλτιστο αποτέλεσμα.

2.2.2 HTML

Η HTML (ακρωνύμιο για το *HyperText Markup Language*) είναι η κύρια γλώσσα σήμανσης για τη δημιουργία ιστοσελίδων και άλλες πληροφορίες που μπορεί να εμφανίζεται σε ένας Φυλλομετρητής (πρόγραμμα περιήγησης στο διαδίκτυο).

Ένα έγγραφο HTML είναι γραμμένο σε μορφή HTML και τα στοιχεία του αποτελούνται από ετικέτες περικλείονται σε αγκύλες (π.χ. <html>), στο περιεχόμενο της ιστοσελίδας. Οι HTML ετικέτες (*tags*) πιο συχνά έρχονται σε ζεύγη, όπως h1 και </ h1>, αν και ορισμένες ετικέτες, που είναι γνωστά ως κενά στοιχεία, είναι αταίριαστα, για παράδειγμα το . Η πρώτη ετικέτα σε ένα ζευγάρι είναι η ετικέτα έναρξης, και η δεύτερη ετικέτα είναι η ετικέτα τέλους (είναι επίσης ονομάζεται το άνοιγμα και το κλείσιμο της ετικέτας). Ανάμεσα σε αυτές τις ετικέτες οι σχεδιαστές ιστοσελίδων μπορούν να προσθέσουν κείμενο, άλλες ετικέτες, σχόλια και άλλα είδη κειμένου με βάση το περιεχόμενο.

Ο σκοπός ενός φυλλομετρητή (web browser) είναι να διαβάσει HTML έγγραφα και να συνθέτει σε οπτική ή ηχητική μορφή την ιστοσελίδα ώστε να είναι εύκολα αντιληπτή από το χρήστη. Το πρόγραμμα περιήγησης δεν εμφανίζει τις ετικέτες HTML, αλλά χρησιμοποιεί τις ετικέτες για να ερμηνεύσει το περιεχόμενο της σελίδας. Τα HTML στοιχεία αποτελούν τα δομικά στοιχεία του συνόλου των ιστοχώρων. Η χρήση της HTML επιτρέπει εικόνες και αντικείμενα να είναι ενσωματωμένα και μπορούν να χρησιμοποιηθούν για τη δημιουργία διαδραστικών μορφών. Η HTML παρέχει ένα μέσο για τη δημιουργία δομημένων εγγραφών, υποδηλώνει τη διαρθρωτική σημασιολογία για το κείμενο, όπως

επικεφαλίδες, παραγράφους, λίστες, συνδέσεις, αποσπάσματα και άλλα αντικείμενα. Επίσης, μπορεί να ενσωματώσει scripts γραμμένα σε γλώσσες όπως η JavaScript που επηρεάζουν τη συμπεριφορά των HTML ιστοσελίδων ή να ορίσει την εμφάνιση και τη διάταξη του κειμένου με χρήση των πρότυπων CSS.

Η HTML εμφανίστηκε πρώτη φορά το 1980, όταν ο φυσικός Tim Berners-Lee, ο οποίος ήταν εργολάβος στο CERN, προτείνει το ENQUIRE, ένα σύστημα για το CERN όπου οι ερευνητές να χρησιμοποιούν και να μοιράζονται έγγραφα. Το 1989, ο Berners-Lee έγραψε ένα σημείωμα προτείνοντας ένα Internet-based σύστημα *υπερκειμένου*. Έτσι, καθορίζεται η HTML και έγραψε το πρώτο πρόγραμμα περιήγησης και το λογισμικό διακομιστή στα τέλη του 1990.

Η περαιτέρω ανάπτυξη της HTML, βρίσκεται υπό την αιγίδα της IETF, ενώ έχει καθυστέρησε λόγω των ανταγωνιστικών συμφερόντων. Από το 1996, οι προδιαγραφές της HTML έχουν διατηρηθεί, με σχόλια και προτάσεις από την εμπορικούς προμηθευτές λογισμικού και το World Wide Web Consortium (W3C). Ωστόσο, το 2000, η HTML, επίσης, έγινε ένα διεθνές πρότυπο (ISO/IEC 15445:2000). Η HTML 4.01 δημοσιεύθηκε στα τέλη του 1999, με περαιτέρω διορθώσεις που δημοσιεύθηκε το 2001. Το 2004 η ανάπτυξη ξεκίνησε η ανάπτυξη της HTML5 από το Web Hypertext Application Technology Working Group (WHATWG). Η HTML5 αναμένεται να κυριαρχήσει στα επόμενα χρόνια στο διαδίκτυο.



Εικόνα 2.2: Λογότυπο HTML5

2.2.3 CSS

Ο όρος *CSS* είναι ακρωνύμιο για το *Cascading Style Sheets*. Πρόκειται για μια γλώσσα που δίνει μορφή και χρησιμοποιείται για την περιγραφή (εμφάνιση και μορφοποίηση) ενός εγγράφου γραμμένο σε μια γλώσσα σήμανσης, όπως η HTML. Πιο κοινή εφαρμογή του είναι να μορφοποιεί ιστοσελίδες γραμμένες σε HTML και XHTML, αλλά η γλώσσα μπορεί επίσης να εφαρμοστεί σε οποιοδήποτε είδος του εγγράφου XML, συμπεριλαμβανομένου του απλού XML, SVG και XUL.

Η CSS έχει σχεδιαστεί για να επιτρέπει κυρίως το διαχωρισμό του περιεχομένου εγγράφου (γραμμένο σε HTML ή παρόμοια γλώσσα σήμανσης) από την παρουσίαση του εγγράφου, συμπεριλαμβανομένων στοιχείων, όπως η διάταξη, τα χρώματα και τις γραμματοσειρές. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, παρέχει μεγαλύτερη ευελιξία και έλεγχο των προδιαγραφών των χαρακτηριστικών παρουσίασης, ώστε πολλές σελίδες για να μοιραστούν τη μορφοποίηση, και να μειώσουν την πολυπλοκότητα και την επανάληψη της διάρθρωσης του περιεχόμενου. Η CSS μπορεί επίσης να επιτρέψει την ίδια (ιστο)σελίδα να παρουσιάζεται σε διαφορετικά στυλ για τις διάφορες μεθόδους επεξεργασίας, όπως επί της οθόνης, σε έντυπη μορφή, με τη φωνή (όταν διαβάζεται από μια ομιλία που βασίζεται πρόγραμμα περιήγησης ή λογισμικό ανάγνωσης οθόνης) και σε Braille (συσκευές αφής). Μπορεί επίσης να χρησιμοποιηθεί για να επιτρέψει στην ιστοσελίδα να εμφανιστεί με διαφορετικό τρόπο ανάλογα με το μέγεθος της οθόνης ή συσκευής, στην οποία προβάλλεται το αντικείμενο αυτό. Ενώ ο συντάκτης ενός εγγράφου συνδέει τυπικά το έγγραφο σε ένα φύλλο στυλ CSS, οι αναγνώστες μπορούν να χρησιμοποιούν διαφορετικό φύλλο στυλ, ίσως ένα για το δικό τους υπολογιστή, για να παρακάμψετε αυτό ο συγγραφέας έχει καθορίσει. Η CSS καθορίζει ένα σύστημα προτεραιότητας για να καθοριστεί ποιοι κανόνες στυλ ισχύουν, αν περισσότερες από μία εγγραφές αφορούν ένα συγκεκριμένο στοιχείο. Σε αυτό το λεγόμενο καταρράκτη, οι προτεραιότητες και τα βάρη υπολογίζεται και αποδίδεται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμο.

Οι προδιαγραφές CSS που τηρείται από το World Wide Web Consortium (W3C). Ο Internet Media τύπος (MIME type) είναι text /css και έχει καταχωρηθεί για χρήση με CSS από το RFC 2318 (Μάρτιος 1998), ενώ λειτουργούν και δωρεάν υπηρεσίες επικύρωσης CSS.

Επόμενη έκδοση της CSS, είναι η CSS3 που βρίσκεται ακόμα σε δοκιμαστικό στάδιο.



Εικόνα 2.3: Λογότυπο CSS3

2.2.4 MySQL

Η *MySQL* είναι (από το 2008) το πιο ευρέως χρησιμοποιούμενο ανοιχτού κώδικα (open source) σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) του κόσμου, που λειτουργεί ως ένας διακομιστής που παρέχει πρόσβαση πολλών χρηστών σε μια σειρά από βάσεις δεδομένων. Πήρε το όνομά της από το όνομα της κόρης του συνιδρυτή Michael Widenius (μεταφράζεται ως «SQL μου», όπου η φράση *SQL* σημαίνει *Structured Query Language*).



Εικόνα 2.4: Λογότυπο MySQL

Το σχέδιο ανάπτυξης της MySQL έχει να κάνει με τον πηγαίο κώδικα ο οποίος είναι διαθέσιμος υπό τους όρους της General Public License (GNU), καθώς και κάτω από μια ποικιλία των ιδιοκτητών συμφωνιών. MySQL ανήκει και χρηματοδοτείται από μία και μόνη επιχείρηση, τη σουηδική εταιρεία MySQL AB, που τώρα ανήκει στην Oracle Corporation.

Η MySQL είναι μια δημοφιλής επιλογή της βάσης δεδομένων για χρήση σε εφαρμογές διαδικτύου και αποτελεί βασικό συστατικό στοιχείο της (χρησιμοποιείται ευρέως) LAMP (ανοικτού κώδικα στοίβα λογισμικού εφαρμογών). Η LAMP είναι ένα αρκτικόλεξο για το Linux, Apache, MySQL, Perl/PHP/Python. Πρόκειται για ένα ελεύθερου και ανοικτού κώδικα σύστημα που απαιτεί ένα πλήρως εξοπλισμένο σύστημα διαχείρισης βάσεων δεδομένων που είναι η MySQL.

Για εμπορική χρήση, είναι διαθέσιμες πολλές εκδόσεις και προσφέρουν επιπλέον λειτουργικότητα. Οι εφαρμογές που χρησιμοποιούν MySQL βάσεις δεδομένων περιλαμβάνουν: TYPO3, Joomla, WordPress, phpBB, Drupal και άλλα λογισμικά. Επίσης, η MySQL χρησιμοποιείται σε πολλές υψηλού προφίλ και μεγάλης κλίμακας εφαρμογές παγκόσμιου Ιστού, συμπεριλαμβανομένης της Wikipedia, Google (αν και όχι για τις αναζητήσεις), Facebook, Twitter, Flickr, Nokia.com και το YouTube.

2.2.5 JavaScript

Η *JavaScript* (JS) είναι μια ερμηνευμένη γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών. Αρχικά είχε εφαρμοστεί ως μέρος των φυλλομετρητών (web browsers), έτσι ώστε να είναι δυνατή η εκτέλεση client-side scripts για να μπορούν να αλληλεπιδράσουν με το χρήστη, τον έλεγχο του προγράμματος περιήγησης, να επικοινωνούν ασύγχρονα, και να αλλάξει το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι ένα πρωτότυπο που βασίζεται σε scripting γλώσσα που είναι δυναμική, ασθενώς δακτυλογραφημένη και έχει πρώτης τάξεως λειτουργίες. Η σύνταξή της έχει επηρεαστεί από τη γλώσσα C. Η JavaScript αντιγράφει πολλά ονόματα και τις συμβάσεις ονομασίας από την Java, αλλά οι δύο γλώσσες είναι διαφορετικές, άσχετες και έχουν πολύ διαφορετική σημασία. Πρόκειται για μια πολυ-πρότυπη γλώσσα προγραμματισμού, η οποία είναι αντικειμενοστραφής (Object-Oriented).

Επιπλέον, κοινή είναι η χρήση της JavaScript σε εφαρμογές εκτός από ιστοσελίδες, για παράδειγμα, σε έγγραφα PDF, site-specific προγράμματα περιήγησης, καθώς και desktop widgets. Τα νεότερα και ταχύτερα JavaScript Virtual-Machines (VMs) και τα πλαίσια ανάπτυξης εφαρμογών (frameworks) που έχουν αναπτυχθεί (κυρίως το Node.js) αυξήσει τη δημοτικότητα της Javascript ακόμη και για server-side εφαρμογές διαδικτύου.

Η JavaScript επισημοποιήθηκε στην τυπική γλώσσα ECMAScript και κατά κύριο λόγο αυτή χρησιμοποιείται ως μέρος ενός προγράμματος περιήγησης στο web (client-side JavaScript). Αυτό επιτρέπει την πρόσβαση μέσω προγραμματισμού στα υπολογιστικά αντικείμενα μέσα σε ένα περιβάλλον υποδοχής.

Πλέον, η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού για το διαδίκτυο. Αρχικά, αν και πολλοί επαγγελματίες προγραμματιστές δυσφήμησαν τη γλώσσα, διότι το κοινό που στόχευε η γλώσσα οι ερασιτέχνες προγραμματιστές διαδικτύου. Η έλευση της τεχνικής Ajax επέστρεψε τη JavaScript στο προσκήνιο και τράβηξε την προσοχή επαγγελματιών προγραμματισμού. Το αποτέλεσμα ήταν ο πολλαπλασιασμός διαθέσιμων πλαισίων και των βιβλιοθηκών, με βελτιωμένες πρακτικές προγραμματισμού JavaScript, και αυξημένη χρήση αυτής έξω από τα προγράμματα περιήγησης.

Τον Ιανουάριο του 2009, το project CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης, κυρίως για την ανάπτυξη JavaScript έξω από το πρόγραμμα περιήγησης.

2.2.6 jQuery

Η *jQuery* είναι μια βιβλιοθήκη *JavaScript* που έχει σχεδιαστεί για να απλοποιήσει το client-side scripting της HTML και έχει τη δυνατότητα να εκτελεστεί από όλους τους φυλλομετρητές (web browsers). Κυκλοφόρησε τον Ιανουάριο του 2006 από τη BarCamp NYC με κύριο προγραμματιστή τον John Resig. Σήμερα αναπτύσσεται από μια ομάδα προγραμματιστών με επικεφαλής τον Dave Methvin. Χρησιμοποιείται από πάνω από το 55% των 10.000 πιο δημοφιλών ιστοσελίδων, ενώ η *jQuery* είναι η πιο δημοφιλής βιβλιοθήκη *JavaScript* που χρησιμοποιούνται σήμερα.

Η *jQuery* είναι δωρεάν, ανοιχτού κώδικα λογισμικό, κάτω από την άδεια MIT License. Η σύνταξη της *jQuery* έχει σχεδιαστεί για να καταστήσει ευκολότερη την πλοήγηση ένα έγγραφο, επιλέγει τα DOM στοιχεία, δημιουργεί κινούμενες εικόνες, χειρίζεται τα γεγονότα, και την ανάπτυξη εφαρμογών Ajax. Η *jQuery* παρέχει επίσης δυνατότητες για τους προγραμματιστές να δημιουργήσουν πρόσθετα (plug-ins) στην βασισμένα σε αυτή τη *JavaScript* βιβλιοθήκη. Αυτό επιτρέπει στους προγραμματιστές να δημιουργήσουν αφαιρέσεις για χαμηλού επιπέδου αλληλεπιδράσεις και κινήσεις (animation) και προηγμένα εφέ υψηλού επιπέδου. Η προσέγγιση της *jQuery* βιβλιοθήκης, επιτρέπει τη δημιουργία ισχυρών δυναμικών ιστοσελίδων και διαδικτυακών εφαρμογών.

Η σημασία της *jQuery* φαίνεται από την απόφαση των Microsoft και Nokia να ανακοινώσουν τα σχέδιά τους για να συνδυάσουν *jQuery* στις πλατφόρμες τους. Η Microsoft αρχικά το υιοθέτησε στο Visual Studio, για χρήση εντός της των ASP.NET AJAX και ASP.NET MVC frameworks ενώ η Nokia την έχει ενσωματώσει στην Run-Time πλατφόρμα ανάπτυξης widget.

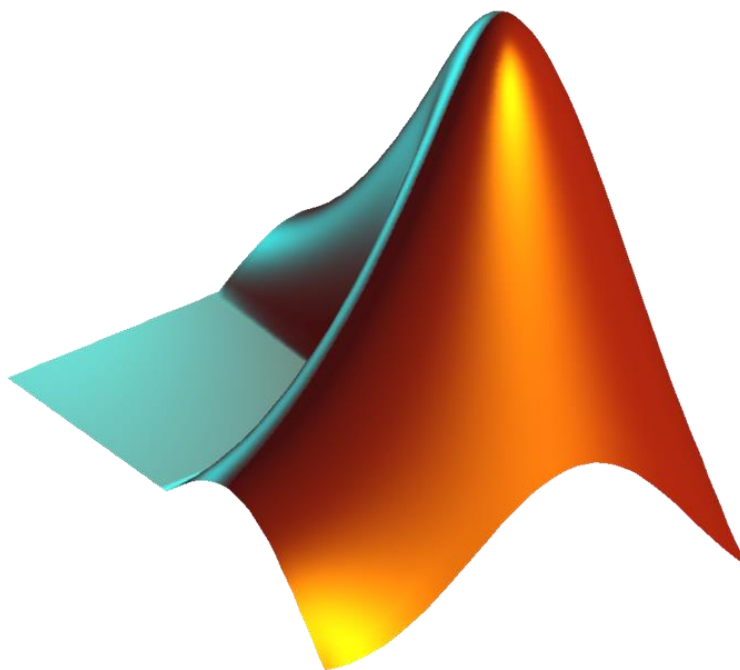
jQuery έχει επίσης χρησιμοποιηθεί σε αλλαγές από την έκδοση 1.16 και ύστερα.

2.3 Υλοποίηση του Αλγορίθμων της Συνεργατικής Διήθησης

Το δεύτερο μέρος της εφαρμογής που αναπτύχθηκε αφορά την υλοποίηση των αλγορίθμων Συνεργατικής Διήθησης. Η επίλυση του συγκεκριμένου προβλήματος ανάγεται σε πρόβλημα μητρώων καθώς χρειάζεται να γίνουν πολλαπλασιασμοί μητρώων και διανυσμάτων και κάποιες αναδρομικές διαδικασίες. Για αυτό το λόγο επιλέχθηκε ειδικό λογισμικό που ειδικεύεται σε αριθμητικές πράξεις και υπολογισμούς γραμμικής άλγεβρας. Συγκεκριμένα:

2.3.1 MATLAB

Η *MATLAB* (*MA*Trix *LAB*oratory) είναι ένα αριθμητικό υπολογιστικό περιβάλλον και τέταρτης γενιάς γλώσσα προγραμματισμού. Αναπτύχθηκε από τη MathWorks και το MATLAB επιτρέπει χειρισμούς μητρώων, αποτύπωση των λειτουργιών και των δεδομένων, την εφαρμογή των αλγορίθμων, δημιουργία διεπαφών χρήστη, και διασυνδέεται με προγράμματα που είναι γραμμένα σε άλλες γλώσσες, όπως C, C++, Java, και Fortran.



Εικόνα 2.5: Λογότυπο MatLab

Ο Cleve Moler, ο πρόεδρος του Τμήματος Επιστήμης των Υπολογιστών στο Πανεπιστήμιο του Νέου Μεξικού, ξεκίνησε την ανάπτυξη της MATLAB στα τέλη της δεκαετίας του 1970. Αρχικά, την είχε σχεδιάσει για να δώσουν στους

μαθητές την πρόσβαση στο LINPACK και Eispack χωρίς να χρειάζεται να μάθουν Fortran. Γρήγορα εξαπλώθηκε σε άλλα πανεπιστήμια και βρήκε ένα ισχυρό κοινό μέσα στην κοινότητα των εφαρμοσμένων μαθηματικών. Ο Jack Little, ένας μηχανικός, είχε μια πρώτη επαφή με τη MATLAB κατά τη διάρκεια μιας επίσκεψης που πραγματοποίησε ο Moler στο Πανεπιστήμιο του Stanford το 1983. Αναγνωρίζοντας εμπορικό δυναμικό της, ένωσε τις δυνάμεις του με τους Moler και Steve Bangert. Ξαναέγραψαν τη MATLAB σε C και ίδρυσαν την εταιρία MathWorks το 1984 για να συνεχίσει την ανάπτυξή της. Οι βιβλιοθήκες που ξαναέγραψαν έγιναν γνωστές ως JACKPAC. Το 2000, το MATLAB ξαναγράφηκε για να χρησιμοποιήσει μια νεότερη σειρά των βιβλιοθηκών για τη, εύκολο χειρισμό μητρώων (LAPACK).

Η MATLAB υιοθετήθηκε για πρώτη φορά από τους ερευνητές και τους επαγγελματίες στο χώρο της μηχανικής αλλά γρήγορα εξαπλώθηκε σε πολλούς άλλους τομείς. Τώρα επίσης χρησιμοποιείται στην εκπαίδευση, ιδίως τη διδασκαλία της γραμμικής άλγεβρας και αριθμητικής ανάλυσης, και είναι δημοφιλής μεταξύ των επιστημόνων που συμμετέχουν στην επεξεργασία εικόνας.

Αν και η MATLAB προορίζεται κυρίως για την αριθμητική υπολογιστών, μια προαιρετική εργαλειοθήκη χρησιμοποιεί την MuPAD συμβολική μηχανή, επιτρέποντας την πρόσβαση σε συμβολικές υπολογιστικές δυνατότητες. Ένα πρόσθετο πακέτο, το Simulink, προσθέτει γραφικές προσομοιώσεις και σχεδιασμό Model-Based για δυναμικά και ενσωματωμένα συστήματα.

Το 2004, η MATLAB είχε περίπου ένα εκατομμύριο χρήστες σε όλη τη βιομηχανία και τον ακαδημαϊκό κόσμο. Οι χρήστες του MATLAB προέρχονται από διάφορους χώρους της μηχανικής, των θετικών επιστημών και την οικονομία. Χρησιμοποιείται ευρέως τόσο σε ακαδημαϊκά όσο και σε ερευνητικά ιδρύματα, καθώς και βιομηχανικές επιχειρήσεις.

3. Ανάλυση και Σχεδιασμός της Εφαρμογής

3.1 Εισαγωγή

Οι *Αλγόριθμοι Συνεργατικής Διήθησης* έχουν ευρεία εφαρμογή στα *Συστήματα Συστάσεων*. Η τάση για εκτεταμένη χρήση των συγκεκριμένων αλγορίθμων στα Συστήματα Συστάσεων ξεκίνησε μετά τη δημοσίευση του Γεώργιου Καρύπη και των συνεργατών του (*Item-Based Collaborative Filtering Recommendation Algorithms*), οι οποίοι πρότειναν ένα απλό, αλλά ταυτόχρονα πολύ ισχυρό μαθηματικό μοντέλο για την απόφαση μιας σύστασης σε ένα Χρήστη. Βασισμένοι σε προηγούμενες δημοσιεύσεις και αποτελέσματα, επιλέξαμε να χρησιμοποιήσουμε αυτό το μοντέλο καθώς κρίθηκε ότι καλύπτει τις ανάγκες της Διπλωματικής Εργασίας.

Τα δεδομένα που παρέχονται από τα Κοινωνικά Δίκτυα σε συνδυασμό με το παραπάνω μοντέλο μπορούν να οδηγήσουν σε μια υλοποιήσιμη εφαρμογή με πολύ καλά αποτελέσματα και πολλές δυνατότητες διαφοροποίησης και επέκτασης.

Στο κεφάλαιο που ακολουθεί θα περιγράψουμε τον σχεδιασμό της εφαρμογής και τα βήματα που ακολουθήθηκαν.

3.2 Εφαρμογή στο Κοινωνικό Δίκτυο

Όπως έχουμε αναφέρει και στις παραπάνω παραγράφους η υλοποίηση μπορεί να χωριστεί σε δύο (2) βασικά συστατικά μέρη: τη Διεπαφή του Χρήστη με το Κοινωνικό Δίκτυο και την υλοποίηση του Συστήματος Σύστασης. Σε αυτή την παράγραφο θα περιγράψουμε το πρώτο μέρος, της Διεπαφής του Χρήστη με το Κοινωνικό Δίκτυο.

3.2.1 Επιλογή Κοινωνικού Δικτύου και Δεδομένων

Τα Κοινωνικά Δίκτυα που είναι διαθέσιμα είναι δεκάδες. Επιλέξαμε το Κοινωνικό Δίκτυο Facebook επειδή:

- *Προτιμάται από πολλούς Χρήστες.* Καθημερινά εκατομμύρια Χρήστες επισκέπτονται το Κοινωνικό Δίκτυο και εγγράφονται νέοι. Αναπόφευκτα μοιράζονται δεδομένα και δημιουργούν συνδέσεις με άλλους Χρήστες ή Αντικείμενα του Κοινωνικού Δικτύου, που οδηγεί σε μεγάλο πλήθος διαθέσιμων δεδομένων.
- *Πλήθος Δεδομένων.* Λόγω των διαθέσιμων επιλογών που παρέχει το Κοινωνικό Δίκτυο οι Χρήστες μπορούν να προσθέσουν σε αυτό διάφορα δεδομένα, όπως τις *Τοποθεσίες* που έχουν επισκεφθεί (*Checkins*),

Φωτογραφίες τους (Photos), Εκδηλώσεις που έχουν παρευρεθεί (Events), Συνδέσμους (Links) κ.ά. Το πλήθος των διαθέσιμων δεδομένων σε σχέση το εξαιρετικά μεγάλο πλήθος των Χρηστών οδηγούν σε μια τεράστια πηγή δεδομένων.

- *Ποιότητα Δεδομένων.* Τα δεδομένα που μοιράζεται ο Χρήστης είναι προσωπικά και τον χαρακτηρίζουν, αφού τα Κοινωνικά Δίκτυα είναι μια ηλεκτρονική μικρογραφία του πραγματικού κόσμου. Επομένως, το Κοινωνικό Δίκτυο μπορεί να θεωρηθεί μια «αξιόπιστη» πηγή δεδομένων.

Οι παραπάνω παρατηρήσεις οδήγησαν στην επιλογή του Κοινωνικού Δικτύου Facebook για την Διπλωματική Εργασία.

Αναφέραμε και παραπάνω ότι στο Κοινωνικό Δίκτυο περιέχονται πολλά δεδομένα του Χρήστη. Για τις ανάγκες της Εργασίας επιλέξαμε το δεδομένο *Εκδήλωση (Event)*, καθώς η *Πρόταση μιας Εκδήλωσης* ενδιαφέρει το μεγαλύτερο πλήθος των Χρηστών, καθώς μπορεί να μεταφραστεί σε μια ευχάριστη έξοδο. Ο σύγχρονος τρόπος ζωής δεν αφήνει αρκετό ελεύθερο χρόνο στους ανθρώπους (άρα και Χρήστες της εφαρμογής μας) και οι όποιες ευκαιρίες για κάποια έξοδο είναι οάσεις στην μονοτονία της καθημερινότητας. Έτσι, προκύπτει η ανάγκη για επιτυχημένη πρόταση μιας Εκδήλωσης (ενός προϊόντος ή μιας υπηρεσίας σε τελική ανάλυση) που αποσκοπεί στην ευχαρίστηση του Χρήστη, προσωποποιημένη για τον καθένα από αυτούς για να τους ικανοποιεί.

Επίσης, με βάση τα διαθέσιμα από το Κοινωνικό Δίκτυο δεδομένα η Εκδήλωση μπορεί να καλύψει τις διάφορες προτιμήσεις του Χρήστη καθώς μπορεί να την χαρακτηρίσει ως *Ενδιαφέρουσα, Λιγότερο ενδιαφέρουσα ή Αδιάφορη (Attending, Maybe, Declined)*. Αυτό το γεγονός θα μας βοηθήσει να «γνωρίσουμε» καλύτερα το Χρήστη αφού γνωρίζουμε τι προτιμά και τι όχι.

3.2.2 Ασφάλεια και Άδειες Δεδομένων

Το Κοινωνικό Δίκτυο Facebook παρέχει πολύ υψηλό επίπεδο ασφαλείας επειδή σε αυτό καθημερινά εκατομμύρια Χρηστών μοιράζονται προσωπικά δεδομένα. Αυτά τα δεδομένα καλύπτονται από Διεθνείς Κανονισμούς Ασφαλείας και δεν είναι διαθέσιμα στον καθένα από εμάς. Συγκεκριμένα ο Χρήστης πρέπει να δώσει την *Άδειά του (Permission)* ^[2] ώστε να μπορέσουμε να έχουμε πρόσβαση, να προβάλλουμε ή να λάβουμε τα δεδομένα του. Σύμφωνα με το Κοινωνικό

[2] Το Κοινωνικό Δίκτυο Facebook έχει ορίσει διαφορετικά επίπεδα *Αδειών (Permissions)* ανάλογα με το δεδομένο που επιθυμούμε να μας παρέχει ο Χρήστης. Οι Άδειες χωρίζονται σε *Άδειες Χρηστών (User Permissions)*, *Άδειες Φύλων (Friend Permissions)* και *Εκτεταμένες Άδειες (Extended Permissions)*.

Δίκτυο πρέπει κατά τον σχεδιασμό να επιλέξουμε τα δεδομένα που επιθυμούμε να έχουμε πρόσβαση και στη συνέχεια να ζητήσουμε και να λάβουμε την κατάλληλη Άδεια από το Χρήστη.

Για τη λήψη των δεδομένων Εκδήλωση του Χρήστη χρησιμοποιούμε την Άδεια Χρήστη (*User Permission*) και συγκεκριμένα την Άδειας Εκδηλώσεων (*user_events*).

Σεβόμενοι πάντα τα προσωπικά στοιχεία και δεδομένα του Χρήστη ζητούμε από το Χρήστη μόνο τα δεδομένα που χρειαζόμαστε για την εφαρμογή μας και τίποτε περισσότερο. Η «απληστία» ως προς το πλήθος των δεδομένων που απαιτούμε από το Χρήστη έχει υπέρ και κατά. Ο πιο συνετός λόγος για να ζητήσουμε «πολλά» δεδομένα από το Χρήστη είναι η καλύτερη ανάλυση και επεξεργασία αυτών με σκοπό να οδηγήσει σε πιο επιτυχημένη τελική απόφαση/πρόταση. Αντίθετα, η απαίτηση περισσότερων προσωπικών δεδομένων από το Χρήστη μπορεί να οδηγήσει σε δυσφορία και καχυποψία στο Χρήστη (πιθανή είσοδος στην προσωπική του ζωή) και τελικά ο Χρήστης να μη χρησιμοποιήσει ποτέ την εφαρμογή μας και έτσι να «χάσουμε» πολύτιμα δεδομένα. Για τις ανάγκες της Διπλωματικής Εργασίας αρκούν μόνο τα δεδομένα Εκδηλώσεων του Χρήστη καθώς η επεξεργασία των δεδομένων που θα ακολουθήσει είναι δυνατό να παράγει τα επιθυμητά αποτελέσματα.

Συγκεκριμένα, να ξεκαθαρίσουμε ότι ο μόνος τρόπος για αν λάβουμε δεδομένα του Χρήστη, όποια και αν είναι αυτά, είναι μέσω μιας Εφαρμογής Facebook (Facebook App) η οποία ζητά πρόσβαση στα δεδομένα του Χρήστη. Αυτόν ακριβώς το στόχο εξυπηρετεί ο σχεδιασμός και ανάπτυξη της Διεπαφής του Χρήστη με το Κοινωνικό Δίκτυο.

3.2.3 Σχεδιασμός Διεπαφής Χρήστη

Η Διεπαφή της εφαρμογής μας με το Χρήστη δεν επηρεάζει τη λειτουργία του συστήματος. Εξυπακούεται ότι, μια σύγχρονη σχεδίαση θα κεντρίσει το ενδιαφέρον του Χρήστη και θα τον «τραβήξει» να χρησιμοποιήσει την εφαρμογή μας με ότι αυτό συνεπάγεται (άδεια για χρήση και λήψη των δεδομένων του, επεξεργασία και τελική πρόταση).

Τελικά, προτιμήθηκε η σχεδίαση μιας απλής και λιτής διεπαφής χρήστη καθώς κρίθηκε ότι είναι αρκετή για τις ανάγκες της παρούσας Διπλωματικής Εργασίας.

Μπορείτε να δείτε το τελικό αποτέλεσμα σε επόμενη παράγραφο όπου αναλύεται η διαδικασία της υλοποίησης.

3.2.4 Εργασίες στο Παρασκήνιο

Από τη στιγμή που ο Χρήστης επιλέξει να προβάλει την εφαρμογή μας και παράσχει σε αυτή την απαραίτητη Άδεια για να έχουμε πρόσβαση στα δεδομένα του ξεκινά η πρώτη φάση της σύστασης. Αυτή είναι η *Συλλογή των Δεδομένων* του Χρήστη. Συνοπτικά η συλλογή δεδομένων αναφέρεται στις υπο-διεργασίες:

- Αίτημα προς το Κοινωνικό Δίκτυο
- Λήψη δεδομένων
- Αποθήκευση των δεδομένων

Οι παραπάνω διεργασίες εκτελούνται στο παρασκήνιο αυξάνοντας έτσι την εμπειρία του Χρήστη ως προς τη χρήση της εφαρμογής.

Στη συνέχεια, θα περιγράψουμε αναλυτικά η κάθε μια από αυτές τις διεργασίες.

3.2.4.1 Αίτημα προς το Κοινωνικό Δίκτυο

Τα δεδομένα που μοιράζονται οι Χρήστες στο Κοινωνικό Δίκτυο βρίσκονται αποθηκευμένα σε εξυπηρετητές διαδικτύου (web servers). Από την στιγμή που ο Χρήστης δώσει την Άδειά του να τα χρησιμοποιήσουμε μπορούμε να τα λάβουμε. Για να γίνει αυτό πρέπει να μεταφέρουμε τα δεδομένα από τους εξυπηρετητές του Κοινωνικού Δικτύου στους δικούς μας. Αυτό απαιτεί από μεριά μας ένα *Αίτημα (Request)* προς το Κοινωνικό Δίκτυο. Το Κοινωνικό Δίκτυο θα λάβει το Αίτημά μας, θα το επεξεργαστεί, θα εξετάσει εάν η Άδεια που μας έχει δώσει ο Χρήστης μπορεί να δώσει τα δεδομένα που αιτούμαστε και τελικά θα μας επιστρέψει τα δεδομένα που χρειαζόμαστε.

Προτιμάται να γίνεται πάντα ορθή και συνετή χρήση των διαθέσιμων συστημάτων και εργαλείων ώστε να αποφεύγονται πιθανά σφάλματα και λάθη.

3.2.4.2 Λήψη Δεδομένων

Η απάντηση του Κοινωνικού Δικτύου στο Αίτημα μας, εφόσον γίνει σωστά και ικανοποιεί τους περιορισμούς που περιγράφουμε παραπάνω, είναι ένα σύνολο από δεδομένα. Τα ακατέργαστα αυτά δεδομένα περιέχουν, μεταξύ άλλων, τα δεδομένα που μας ενδιαφέρουν. Δεν είναι αναγκαίο να περιγράψουμε όλα τα δεδομένα που επιστρέφονται, αλλά θα περιγράψουμε αυτά που θα χρησιμοποιήσουμε και για ποιο σκοπό. Συγκεκριμένα, τα δεδομένα που τελικά αποθηκεύουμε είναι:

- Ο Χρήστης. Για να μπορέσουμε να εξάγουμε αποτελέσματα για τον Χρήστη πρέπει να γνωρίζουμε ποιος είναι αυτός. Το Κοινωνικό Δίκτυο του αναθέτει ένα μοναδικό αναγνωριστικό (*User ID*) και εμείς με τη σειρά μας το αποθηκεύουμε για να προχωρήσουμε τη διαδικασία. Να

θυμίσουμε ότι όλοι οι Χρήστες και τα δεδομένα τους συμμετέχουν στο Σύστημα Σύστασης.

- Η *Εκδήλωση*. Στο Κοινωνικό Δίκτυο κάθε Εκδήλωση αποθηκεύεται με ένα μοναδικό αριθμό αναγνώρισης (*Event ID*). Αυτός ο αριθμός είναι ξεχωριστός για κάθε Εκδήλωση (ουσιαστικά για κάθε οντότητα του Κοινωνικού Δικτύου) και χρησιμοποιείται για να γνωρίσουμε την Εκδήλωση που παρευρεθεί ο Χρήστης.
- Η *Απάντηση του Χρήστη για την Εκδήλωση*. Όπως έχουμε αναφέρει και παραπάνω ο Χρήστης χαρακτηρίζει την κατάστασή του ως προς την Εκδήλωση ως *Ενδιαφέρουσα*, *Λιγότερο ενδιαφέρουσα* ή *Αδιάφορη*. Εμείς αποθηκεύουμε αυτή την επιλογή επειδή θα μας βοηθήσει να «γνωρίσουμε» το Χρήστη. Οι απαντήσεις αυτές θα μας παρουσιάσουν τα ενδιαφέροντά του Χρήστη και τις συνήθειές του, αποκαλύπτοντας τη συμπεριφορά του ως προς τις Εκδηλώσεις που τον ενδιαφέρουν, οδηγώντας έτσι σε επιτυχημένη πρόταση ή προτάσεις.

Η εφαρμογή μας σέβεται απόλυτα τα προσωπικά δεδομένα του Χρήστη. Τα δεδομένα που αποθηκεύουμε, αν και αντιστοιχούν σε συγκεκριμένο Χρήστη, είναι ανώνυμα καθώς στη συνέχεια με τυχαία διαδικασία αντιστοιχίζονται σε άλλες τιμές, π.χ. το Αναγνωριστικό κάθε Χρήστη αντιστοιχίζεται τυχαία στο σύνολο [1, #πλήθος_χρηστών]. Τελικά, η απόφαση του Συστήματος μας θα αποφασίσει για τον ψευδοτυχαίο Αναγνωριστικό Χρήστη.

3.2.4.3 Αποθήκευση των Δεδομένων

Τελικά, τα δεδομένα που αναφέραμε στην παραπάνω παράγραφο (μόνο τα δεδομένα που μας ενδιαφέρουν και μόνον αυτά) αποθηκεύονται στη Βάση Δεδομένων μας, για να είναι άμεσα διαθέσιμα όποτε τα χρειαζόμαστε. Αυτό είναι ένα πολύ σημαντικό βήμα, επειδή θέλουμε να κρατήσουμε τα στοιχεία του Χρήστη για να τα χρησιμοποιήσουμε τόσο για τις δικές του προτάσεις όσο και για προτάσεις που θα γίνουν σε μελλοντικούς Χρήστες.

Μείζονος σημασίας είναι σε αυτή φάση η σχεδίαση της Βάσης Δεδομένων. Η Βάση Δεδομένων έχει σχεδιαστεί με κύριο στοιχείο της το Χρήστη. Για αυτόν θα εξαχθεί το όποιο αποτέλεσμα και με βάση τα δικά του στοιχεία. Έτσι, έχουμε σχεδιάσει ένα (1) πίνακα που αποθηκεύονται οι Χρήστες της εφαρμογής (ποίοι είναι αυτοί) και ένα (1) ακόμη πίνακα όπου αποθηκεύονται οι Εκδηλώσεις για τις οποίες έχει απαντήσει ο Χρήστης και ποία είναι η απάντηση που έχει δώσει. Με άλλα λόγια, αποθηκεύουμε τους Χρήστες και σε ποιες Εκδηλώσεις έχει μεταβεί ο καθένας. Επίσης, έχουμε ένα ακόμα βοηθητικό πίνακα όπου αποθηκεύονται οι Φίλοι των Χρηστών.

Σημαντικό για την αποφυγή λαθών είναι να μην υπάρχουν διπλότυπες ή τριπλότυπες κ.λπ. εγγραφές στη Βάση Δεδομένων καθώς θα οδηγήσει σε

λανθασμένες αποφάσεις. Ο Χρήστης μπορεί να επισκεφτεί την εφαρμογή μας περισσότερες από μία (1) φορές και δεν θέλουμε να ξανα-αποθηκεύσουμε τα δεδομένα του (τουλάχιστον όχι αυτά που ήδη γνωρίζουμε). Αυτή η ενέργεια θα οδηγήσει στην άσκοπη χρήση πόρων του συστήματος και δεν συνιστάται η χρήση κάποιας τέτοιας μεθόδου όσο και αν αυτή φαίνεται να εξυπηρετεί την υλοποίηση. Άρα, αρκεί να αποθηκεύσουμε τα «νέα» (άγνωστα για εμάς), κάθε φορά, δεδομένα ώστε να γνωρίσουμε το Χρήστη καλύτερα. Έτσι, το παραπάνω πρόβλημα προβλέφθηκε, αναλύθηκε διεξοδικά και τελικά επιλύθηκε κατά τη διαδικασία του Σχεδιασμού της Βάσης Δεδομένων.

3.2.5 Εξαγωγή Δεδομένων

Από τη στιγμή που αποθηκεύτηκαν τα δεδομένα στη αποθηκευμένα στη Βάση Δεδομένων μας, έχει ολοκληρωθεί η επαφή του Χρήστη με το Κοινωνικό Δίκτυο, και πλέον η διαδικασία θα συνεχιστεί με τον αλγόριθμο Συνεργατικής Διήθησης για να καταλήξει στη απόφαση για Πρόταση.

Ο αλγόριθμος απαιτεί τα δεδομένα να βρίσκονται σε συγκεκριμένη μορφή για να είναι εύκολη για αυτόν η διαδικασία επίλυσης. Συγκεκριμένα, ο αλγόριθμος απαιτεί τα δεδομένα να βρίσκονται σε τριάδες της μορφής:

Χρήστης, Δεδομένο, Τιμή

Επομένως, στην περίπτωση μας αυτό μεταφράζεται σε:

Χρήστης, Εκδήλωση, Απάντηση_Χρήστη

Δηλαδή, από τον πίνακα της Βάσης Δεδομένων εξάγουμε τα στοιχεία εκείνα που χρειαζόμαστε για να οδηγηθούμε σε απόφαση.

Προκειμένου να καλυφτεί αυτή η ανάγκη σχεδιάστηκε υποσύστημα που θα εξάγει σε αυτή τη μορφή τα απαιτούμενα δεδομένα. Το αποτέλεσμα που παράγεται είναι ένα σύνολο από «τριάδες» (όπως περιγράφονται παραπάνω και συγκεκριμένα το πλήθος τους είναι όσες και οι εγγραφές στον πίνακα), το οποίο δίνεται ως είσοδος στον αλγόριθμο της Συνεργατικής Διήθησης.

3.2.6 Απαιτήσεις Συστήματος

Όπως έχουμε αναφέρει στο Κεφάλαιο 2 χρησιμοποιήθηκαν διάφορες τεχνολογίες. Όλες οι επιλογές έγιναν με βάση την ικανοποίηση των αναγκών του Χρήστη ώστε να έχει την καλύτερη δυνατή εμπειρία από τη χρήση της εφαρμογής. Έτσι, επιλέχθηκαν τεχνολογίες που χρησιμοποιούνται από τις μεγαλύτερες εταιρίες που ασχολούνται στον ευρύτερο κλάδο της Πληροφορικής και είναι γνωστές για την ευελιξία και τις δυνατότητες τους.

Οι πιο σημαντικές από αυτές είναι η χαμηλή κατανάλωση πόρων και γρήγορη εκτέλεση των εντολών. Αυτές οι δύο (2) «αρετές» είναι πάρα πολύ σημαντικές για την υλοποίηση και ομαλή λειτουργία του συνολικού συστήματος. Το σύστημα εξυπηρετεί δεκάδες Χρήστες στους οποίους δύναται να προταθούν εκατοντάδες Εκδηλώσεις. Επομένως, είναι μέγιστης σημασίας η συνετή χρήση των πόρων του εξυπηρετητή (server) που φιλοξενεί το σύστημα για να αποφευχθεί πιθανή πτώση του συστήματος (system failure). Τον ίδιο σκοπό έχει η συσταδοποίηση και ομαδοποίηση των δεδομένων ώστε να γίνεται επεξεργασία κατά ομάδες δεδομένων, και όχι όλων των δεδομένων μαζί, ώστε να ελαχιστοποιείται η χρήση *προσωρινής μνήμης (RAM memory)*.

Όλες οι προαναφερθείσες λειτουργίες αναπτύχθηκαν και περιγράφεται αναλυτικά η διαδικασία που ακολουθήθηκε στο Κεφάλαιο 4 (βλ. παρακάτω).

Σε αυτή τη φάση έχει ολοκληρωθεί η πρώτη φάση της Διπλωματική Εργασίας που περιλαμβάνει την πρώτη επαφή του Χρήστη με το σύστημά μας μέσω του Κοινωνικού Δικτύου με τελικό στόχο τη συλλογή δεδομένων από το Κοινωνικό Δίκτυο.

3.3 Σύνολο Δεδομένων (Dataset)

Προκειμένου να λειτουργήσει το σύστημα και να μετρήσουμε την απόδοσή του χρειάστηκε να δημιουργήσουμε ένα *Σύνολο Δεδομένων (Dataset)*. Αυτό το Σύνολο Δεδομένων δημιουργήθηκε από τους πρώτους Χρήστες της εφαρμογής και χρησιμοποιήθηκε για τον αρχικό καθορισμό των παραμέτρων και της μέτρησης της απόδοσης του συνολικού συστήματος. Ο αλγόριθμος που υλοποιούμε επεξεργάζεται, με κάποιες επαναληπτικές διαδικασίες, το συγκεκριμένο Σύνολο Δεδομένων και τελικά μέσα από αυτό προκύπτει η τελική πρόταση για το Χρήστη.

Αυτή τη στιγμή υπάρχουν διαθέσιμα δεκάδες Σύνολα Δεδομένων. Είναι συνήθης και αποδεκτή τακτική η χρήση ενός εξ' αυτών για τον έλεγχο των Συστημάτων Σύστασης, καθώς είναι εύκολη η σύγκριση αποτελεσμάτων διαφορετικών αλγορίθμων και τεχνικών από την στιγμή που αναφερόμαστε στο ίδιο Σύνολο Δεδομένων. Χαρακτηριστικό παράδειγμα είναι το Σύνολο Δεδομένων MovieLens (αποτελείται από 1000 Χρήστες, 1700 Αντικείμενα και 100000 Απαντήσεις Χρηστών, συχνά αναφέρονται και ως Βαθμολογίες) που χρησιμοποιείται από τις περισσότερες ερευνητικές εργασίες. Ωστόσο, στην περίπτωσή μας επιλέξαμε να δημιουργήσουμε ένα νέο Σύνολο Δεδομένων καθώς η παρούσα Διπλωματική Εργασία είναι πρωτότυπη και δεν υπάρχει κάποιο γενικά αποδεκτό Σύνολο Δεδομένων που να αναφέρεται σε Χρήστες Κοινωνικών Δικτύων ώστε να μπορούμε να χρησιμοποιήσουμε. Επίσης, το Σύνολο Δεδομένων που

δημιουργήσαμε δεν μπορεί να αμφισβητηθεί καθώς προέρχεται από πραγματικά και ανώνυμα δεδομένα Χρηστών και ως εκ τούτου εμφανίζει τα χαρακτηριστικά και δραστηριότητες που περιγράφουμε παρακάτω.

Με τη διαδικασία που περιγράψαμε παραπάνω συγκεντρώθηκαν τα δεδομένα και προέκυψαν τα εξής:

Σύνολο Χρηστών	Σύνολο Εκδηλώσεων	Απάντηση Χρήστη στην Εκδήλωση
943	6178	«1» ή «5»

Συνοπτικός Πίνακας Συνόλου Δεδομένων

Σε αυτό το σημείο θα πρέπει να σας τονίσουμε και ενημερώσουμε για τις συμβάσεις που έγιναν κατά τη δημιουργία του Συνόλου Δεδομένων και τις όποιες λεπτομέρειες:

- Η Απάντηση του Χρήστη στην Εκδήλωση αντιστοιχίζεται στις τιμές «1» ή «5». Η τιμή «1» σημειώνεται όταν ο Χρήστης δεν ενδιαφέρεται για την Εκδήλωση, ενώ η τιμή «5» όταν αυτός ενδιαφέρεται (βλ. παραπάνω σχετικά με την απάντηση του Χρήστη). Οι τιμές αυτές αποφασίστηκαν ως οι ιδανικές ύστερα από τους κύκλους εκπαίδευσης που έγιναν. Επιπλέον, το σύστημα υποστηρίζει διαφορετική έξοδο των παραπάνω τιμών κατ' επιλογή του διαχειριστή του συστήματος και ανάλογα με τις ανάγκες του προβλήματος.
- Κάθε Χρήστης έχει απαντήσει σε τουλάχιστον μία (1) εκδήλωση (όποια και αν είναι αυτή η απάντηση). Αυτή είναι μια σύμβαση που κάναμε κατά τη δημιουργία του Συνόλου Δεδομένων που εξυπηρετεί στην καλύτερη γνώση του Χρήστη, ώστε να οδηγήσει σε επιτυχημένη (κατάλληλη) πρόταση για αυτόν.

Το Σύνολο Δεδομένων είναι σημαντικό στοιχείο του Συστήματος Προτάσεων που περιγράφουμε. Η ορθή και αμερόληπτη δημιουργία του, σε συνδυασμό με επιλογή των δεδομένων οδηγεί σε ένα αποτέλεσμα που θα είναι αξιόπιστο και δεν θα υπάρχουν περιθώρια αμφισβήτησης της διαδικασίας Πρότασης, και του Συστήματος Σύστασης γενικότερα, που υλοποιούμε.

Επίσης, ανάλογα με το μέγεθός του μπορούμε να εξετάσουμε την απόδοσή του ως προς την κατανάλωση πόρων και τον συνολικό χρόνο που απαιτείται για να εξαχθεί το αποτέλεσμα.

Το παραπάνω Σύνολο δεδομένων βρίσκεται αποθηκευμένο στη Βάση Δεδομένων και εισάγεται (με τη μορφή που περιγράψαμε) στο σύστημα της δεύτερης φάσης (Αλγόριθμος Συνεργατικής Διήθησης) του Συστήματος Σύστασης.

Τέλος το Σύνολο Δεδομένων θα είναι διαθέσιμο σε όποιον το χρειαστεί και έτσι και εμείς με τη σειρά μας συνεισφέρουμε στη διάδοση πληροφοριών βοηθώντας ερευνητές να συνεχίσουν και να επεκτείνουν το έργο τους .

3.4 Αλγόριθμος Συνεργατικής Διήθησης

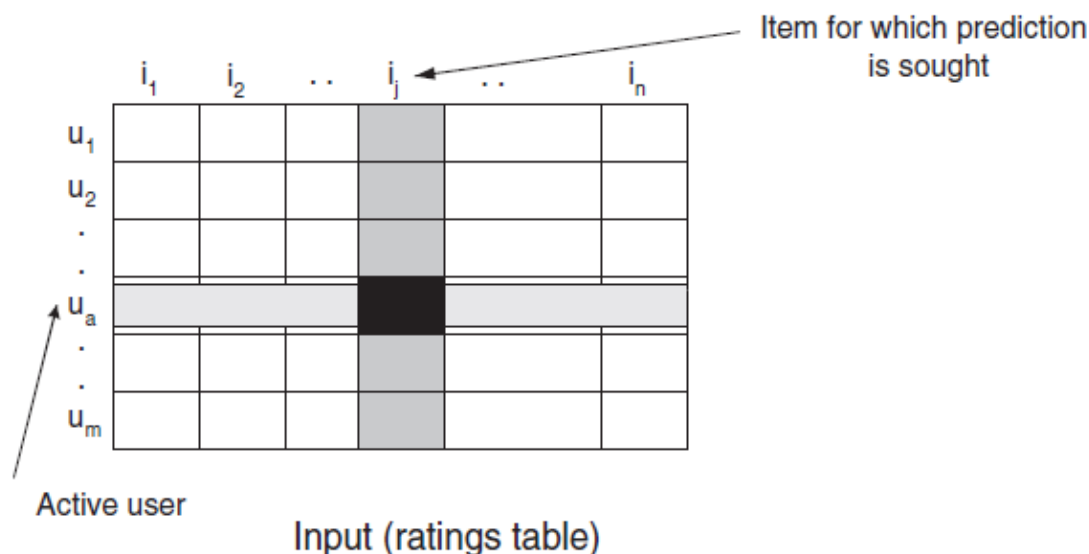
Το Σύστημα Σύστασης που περιγράφουμε θα χρησιμοποιήσει έναν *Αλγόριθμο Συνεργατικής Διήθησης (Collaborative Filtering Algorithm)*. Οι συστάσεις, ή προτάσεις, βασίζονται σε αξιολογήσεις των Χρηστών, οι οποίοι έχουν κοινά ενδιαφέροντα και προτιμήσεις. Το όλο εγχείρημα είναι θεμελιωμένο πάνω στην ιδέα ότι το σύνολο των Χρηστών που έκαναν τις ίδιες επιλογές, πιθανότατα να έχουν και τις ίδιες γενικότερες προτιμήσεις. Έτσι, θα μπορούσαμε σε κάποιο Χρηστή, μέσα από το σύνολο αυτό, να του προτείνουμε Αντικείμενα που υπόλοιποι τα έχουν αξιολογήσει ως ικανοποιητικά στο παρελθόν. Οι γνώμες μπορούν να εκφραστούν σε βαθμολογίες από τους Χρήστες με κλίμακες κακό έως καλό, ή επίσης και με καταγραφή από το σύστημα διαφόρων ενεργειών ή αντιδράσεων του Χρήστη. Για παράδειγμα, αν ο Χρήστης θα διάβαζε κάποιο βιβλίο ή θα έβλεπε κάποια κινηματογραφική ταινία θα είχε θετική αξιολόγηση, αλλιώς αν την αγνοούσε αρνητική. Όπως επίσης, αν πέρναγε αρκετή ώρα διαβάζοντας κάποιο άρθρο ή απλά του «έριχνε μια ματιά» και προσχωρούσε στο επόμενο.

Αναλύοντας τις προτιμήσεις ή τη συμπεριφορά των Χρηστών, μπορούμε να εξάγουμε κάποιο συμπέρασμα για το βαθμό ομοιότητας των προτιμήσεών τους, χωρίς να δίνεται βαρύτητα στο είδος της πληροφορίας. Σαν αποτέλεσμα, το μεγαλύτερο πλεονέκτημα της μεθόδου είναι η δυνατότητα να παράγει συστάσεις για πολλές κατηγορίες αντικειμένων όπως μουσική, ταινίες, βιβλία καθώς και πολλά άλλα παραδείγματα.

3.4.1 Προεπεξεργασία Δεδομένων

Το Σύνολο Δεδομένων που δημιουργήσαμε ουσιαστικά περιέχει όλη την πληροφορία που χρειαζόμαστε για να ολοκληρώσουμε τη διαδικασία της σύστασης. Ωστόσο, η πληροφορίες που περιέχονται σε αυτό δεν βρίσκονται στην κατάλληλη μορφή ώστε να τις επεξεργαστεί ο αλγόριθμος. Άρα, θα πρέπει να μετατρέψουμε τα δεδομένα σε «ευανάγνωστη» μορφή για τον αλγόριθμο. Μέσω μιας απλής διαδικασίας προεπεξεργασίας των δεδομένων που έχουμε εξάγει στο προηγούμενο βήμα (βλ. παραπάνω, σύνολο από τριάδες).

Ο αλγόριθμος απαιτεί τα δεδομένα να βρίσκονται σε ένα *Δισδιάστατο Πίνακα* όπου στις *γραμμές* του πίνακα θα αντιστοιχίζονται οι Χρήστες (Χρήστες της Εφαρμογής - *Users*) και οι *στήλες* αντιστοιχούν στα προς πρόταση Προϊόντα (Εκδηλώσεις της Εφαρμογής - *Items*). Έτσι, οι τριάδες που παίρνουμε από τη βάση μετατρέπονται σε ένα *Δισδιάστατο Πίνακα (2D Array)*, με τη σχεδίαση και τα χαρακτηριστικά που περιγράψαμε. Συνοπτικά, θα προκύψει ένας πίνακας που θα μοιάζει κάπως έτσι:



Εικόνα 3.1: Μορφή του Συνόλου Δεδομένων ως είσοδος για τον Αλγόριθμο Συνεργατικής Διήθησης

Στην παραπάνω εικόνα παρατηρούμε τον Δισδιάστατο Πίνακα που δημιουργούμε καθώς τον απαιτεί ο Αλγόριθμος Συνεργατικής Διήθησης που χρησιμοποιούμε. Παρατηρούμε, ότι κάθε *γραμμή* του πίνακα σε ένα (1) Χρήστη και κάθε *στήλη* σε μία (1) *Εκδήλωση*. Το στοιχείο του πίνακα:

$$(Χρήστης_i, Εκδήλωση_j)$$

αντιστοιχεί στην απάντηση του Χρήστη («1» ή «5») για τη συγκεκριμένη Εκδήλωση. Εξυπακούεται και είναι λογικό να μην έχει απαντήσει ο Χρήστης σε όλες τις Εκδηλώσεις αλλά σε ένα υποσύνολο αυτών, δηλαδή να υπάρχουν «άγνωστες» στο Χρήστη Εκδηλώσεις. Αυτό σε βάθος, είναι και το ζητούμενο του Συστήματος Σύστασης, να προτείνουμε στο Χρήστη κάποιο Προϊόν (Εκδήλωση στην περίπτωση μας) που δεν γνωρίζει αλλά θα τον ενδιέφερε.

Τέλος, για τον καθορισμό των παραμέτρων το Σύνολο Δεδομένων χωρίστηκε σε δύο (2) υποσύνολα, τα:

- *Υποσύνολο Εκπαίδευσης (Train Set)*. Στο Σύνολο Εκπαίδευσης περιλαμβάνεται το μεγαλύτερο ποσοστό (80%) των δεδομένων και μέσω επαναληπτικών διαδικασιών που θα εφαρμοστούν σε αυτό θα καθοριστεί η τελική συμπεριφορά του Συστήματος Σύστασης.
- *Υποσύνολο Ελέγχου (Test Set)*. Στο Σύνολο Ελέγχου περιλαμβάνεται το υπόλοιπο μέρος των δεδομένων (20%), που δεν περιέχεται στο παραπάνω σύνολο. Με αυτό θα γίνουν οι απαραίτητοι έλεγχοι για να διαπιστωθεί η απόδοση του συστήματος και κατά πόσο η απόφαση-πρόταση συμφωνεί με τα στοιχεία του Χρήστη.

Αυτός ο διαχωρισμός έγινε για τον Καθορισμό των Παραμέτρων του Συστήματος, οποίος θα γίνει μέσα από τις επαναληπτικές διαδικασίες του αλγορίθμου, και τον Έλεγχο των Εξαγόμενων Αποτελεσμάτων. Το *Υποσύνολο Ελέγχου* ουσιαστικά θα χρησιμοποιηθεί για να ελεγχθούν τα αποτελέσματα του αλγορίθμου (μέσω μιας απλής σύγκρισης αφού γνωρίσουμε ήδη την απάντηση του Χρήστη, θα συγκρίνουμε την απάντηση του με το αποτέλεσμα του Συστήματος).

3.4.2 Περιγραφή Αλγορίθμου

Ουσιαστικά, ο αλγόριθμος της Συνεργατικής Διήθησης είναι μια ακολουθία από βήματα και στοιχειώδεις υπό εργασίες. Από τη στιγμή που έχουμε ολοκληρώσει τη διαδικασία Προεπεξεργασίας των Δεδομένων, μπορούμε να εφαρμόσουμε τα βήματα για να καταλήξουμε στο αποτέλεσμα (πρόταση ενός ή περισσότερων προϊόντων).

Για την περίπτωση μας, επιλέξαμε μια λύση που βασίζεται στον *Αλγόριθμο Βαθμολόγησης Ιστοσελίδων (PageRank Algorithm)*^[3]. Ο αλγόριθμος αυτός, αναπαριστά τις ιστοσελίδες ως ένα διασυνδεδεμένο γράφο, όπου οι συνδέσεις του γράφου αναπαριστούν τους υπερσυνδέσμους ανάμεσα στις ιστοσελίδες. Έτσι, προσπαθώντας να προβλέψουμε το επόμενο «βήμα» του *Περιηγητή* στο γράφο, βασισμένοι στις προηγούμενες επιλογές του Περιηγητή και άλλων περιηγητών που είχαν κοινές επιλογές, ακολουθούμε τα βήματα του αλγορίθμου. Στην περίπτωση μας, μπορεί να γίνει αναγωγή του προβλήματος

[3] Ο PageRank είναι ένας αλγόριθμος ανάλυσης συνδέσμων, που πήρε το όνομά του από τον Larry Page, και χρησιμοποιείται από τη μηχανή αναζήτησης Google στον παγκόσμιο ιστό. Αυτός, αποδίδει μια αριθμητική στάθμιση σε κάθε στοιχείο ενός συνδεδεμένου γράφου, όπως το Διαδίκτυο, με σκοπό μέτρηση της σχετικής σημασίας του στοιχείου μέσα στο σύνολο. Ο αλγόριθμος μπορεί να εφαρμοστεί σε οποιαδήποτε συλλογή των οντοτήτων με αμοιβαία αποσπάσματα και αναφορές.

μας σε PageRank πρόβλημα χωρίς να επηρεάζεται το αποτέλεσμα και να γίνεται άρση της γενικότητας.

Τα βήματα που ακολουθεί ο Αλγόριθμος είναι:

1. Υπολόγισε το *Μητρώο Ομοιοτήτων (Similarities Matrix)*. Με βάση το μητρώο που έχουμε δημιουργήσει κατά την προεπεξεργασία δεδομένων (βλ. παραπάνω), υπολογίζουμε το Μητρώο Ομοιοτήτων, το οποίο καθορίζει τις συσχετίσεις και τις ομοιότητες μεταξύ των Αντικειμένων/Προϊόντων που έχουμε στη διάθεσή μας (στην περίπτωση μας είναι οι Εκδηλώσεις). Το μητρώο αυτό τελικά, θα περιέχει αριθμητικές τιμές που ορίζουν τη συσχέτιση μεταξύ των αντικειμένων. Οι τιμές που προέκυψαν, βασίζονται στις προτιμήσεις των Χρηστών, που έχουμε εξαγει από το Κοινωνικό Δίκτυο. Ο υπολογισμός των τιμών που περιέχονται στο Μητρώο Ομοιοτήτων υπολογίζονται με διαφορετικό τρόπο, το οποίο μας βοηθά να συγκρίνουμε τα αποτελέσματά μας και τελικά να επιλέξουμε την πιο αποδοτική μέθοδο για το πρόβλημα που επιλύουμε. Ο υπολογισμός του μητρώου δεν επηρεάζει τα βήματα του αλγορίθμου, παρά μόνο τα τελικά αποτελέσματα, και για αυτό θα περιγράψουμε τις διαδικασίες σε επόμενη παράγραφο (βλ. παρακάτω).
2. Υπολόγισε το μητρώο:

$$P = a S + (1 - a) (e w^T)$$

Όπου, S είναι το Μητρώο Ομοιοτήτων που υπολογίσαμε, w^T είναι το (κανονικοποιημένο) ανάστροφο διάνυσμα (διάνυσμα «γραμμή» πλέον) με τις επιλογές του Χρήστη, a είναι η σταθερά του Αλγορίθμου PageRank και e είναι το διάνυσμα «στήλη» που εξυπηρετεί την πραγματοποίηση των πράξεων μεταξύ μητρώων. Εξαιτίας της αναγωγής του προβλήματος μας σε PageRank, υιοθετούμε όλες τις συμβάσεις του αλγορίθμου για να καλύψουμε τις ανάγκες που προκύπτουν.

3. Εφάρμοσε την *Επαναληπτική Μέθοδο Power (Power Iteration)* στο μητρώο P (που υπολόγισες στο προηγούμενο βήμα) σε συνδυασμό με το κανονικοποιημένο διάνυσμα w , που περιέχει τις επιλογές του Χρήστη. Η *Επαναληπτική Μέθοδος Power* είναι ένας αλγόριθμος ιδιοτιμών. Συγκεκριμένα, δοθέντος μητρώου A , ο αλγόριθμος θα παράγει ένα αριθμό λ (ιδιοτιμή), και ένα μη μηδενικό διάνυσμα v (ιδιοδιάνυσμα), τέτοιο ώστε:

$$A v = \lambda v$$

Η Επαναληπτική Μέθοδος Power είναι ένας πολύ απλός αλγόριθμος. Δεν υπολογίζει διάσπαση μητρώου, και ως εκ τούτου μπορεί να χρησιμοποιηθεί όταν το A είναι ένα πολύ μεγάλο και αραιό μητρώο.

Ωστόσο, θα καταλήξει τελικά σε μόνο μία ιδιοτιμή (αυτή με τη μεγαλύτερη απόλυτη τιμή) και μπορεί να συγκλίνει μόνο αργά.

Η έξοδος της παραπάνω επαναληπτικής διαδικασίας, είναι ένα διάνυσμα που περιέχει τιμές για κάθε Προϊόν (Εκδήλωση), οι οποίες αντικατοπτρίζουν την απόφαση του Αλγορίθμου για το κάθε Προϊόν με βάση τις επιλογές του Χρήστη και τις συσχετίσεις των Προϊόντων μεταξύ τους.

4. Τέλος, ταξινόμησε σε φθίνουσα σειρά το διάνυσμα – έξοδο του προηγούμενου βήματος. Το ταξινομημένο διάνυσμα, περιέχει την απόφαση του αλγορίθμου για τον συγκεκριμένο Χρήστη, όπου στην πρώτη θέση βρίσκεται το Προϊόν που, σύμφωνα με τον αλγόριθμο, είναι πιο πιθανό να ενδιαφέρει το Χρήστη και στην τελευταία αυτό που θα είναι αδιάφορο (ή θα ενδιαφέρει ελάχιστα) το Χρήστη. Η τελική έξοδος της εφαρμογής θα είναι ένα υποσύνολο των *Πρώτων-N* θέσεων του διανύσματος, δηλαδή η εφαρμογή θα προτείνει τις Πρώτες-N Εκδηλώσεις με τη μεγαλύτερη βαθμολογία.

Αναφέραμε παραπάνω ότι σε ορισμένα βήματα χρησιμοποιούνται κανονικοποιημένα διανύσματα. Οι κανονικοποιήσεις των διανυσμάτων είναι ζωτικής σημασίας, καθώς εξυπηρετούν λογικές αλλά και υπολογιστικές ανάγκες. Λόγω της αναπαράστασης των αριθμών κινητής υποδιαστολής στις υπολογιστικές μονάδες, μεταφέρονται μικρές τιμές (κοντά στο μηδέν (0)) οι οποίες τελικά, αν δεν κανονικοποιηθούν, καθιστούν την υλοποίηση αδύνατη.

Όπως φαίνεται από τον αλγόριθμο που μόλις περιγράψαμε, διαδικασία εξαγωγής αποτελέσματος αποτελείται από μια επαναληπτική διαδικασία και πράξεις γραμμικής άλγεβρας (πολλαπλασιασμοί μητρώων/διανυσμάτων). Προηγούμενες δημοσιεύσεις, έρευνες και μελέτες έχουν καταλήξει στο συμπέρασμα ότι ο αλγόριθμος είναι εύκολο να υλοποιηθούν σε «πραγματικό χρόνο», δηλαδή η εξαγωγή του αποτελέσματος - Πρόταση μπορεί να υπολογιστεί την ώρα που επισκέπτεται ο Χρήστης την εφαρμογή καθώς η υπολογιστή ισχύ που απαιτείται, τόσο σε επίπεδο πόρων όσο και σε επίπεδο χρόνου, το καθιστούν εύκολο να υλοποιηθεί. Από την άλλη μεριά, ο υπολογισμός του Μητρώου Ομοιοτήτων είναι μια επίπονη διαδικασία (βλ. παρακάτω) καθώς εκτελεί υπολογισμούς και πράξεις σε υπερβολικά μεγάλο πλήθος (ανάλογα με την περίπτωση ακόμη και $O(NM)$, όπου N το πλήθος των αντικειμένων (Εκδηλώσεις) και M το πλήθος των Χρηστών της εφαρμογής).

Κύριο χαρακτηριστικό του Αλγορίθμου είναι η χρήση της Επαναληπτικής Μεθόδου Power. Η χρήση της μεθόδου καθιστά δυνατή πραγματοποίηση και υλοποίηση του συστήματος και τη μεταφορά του από θεωρητικό επίπεδο σε

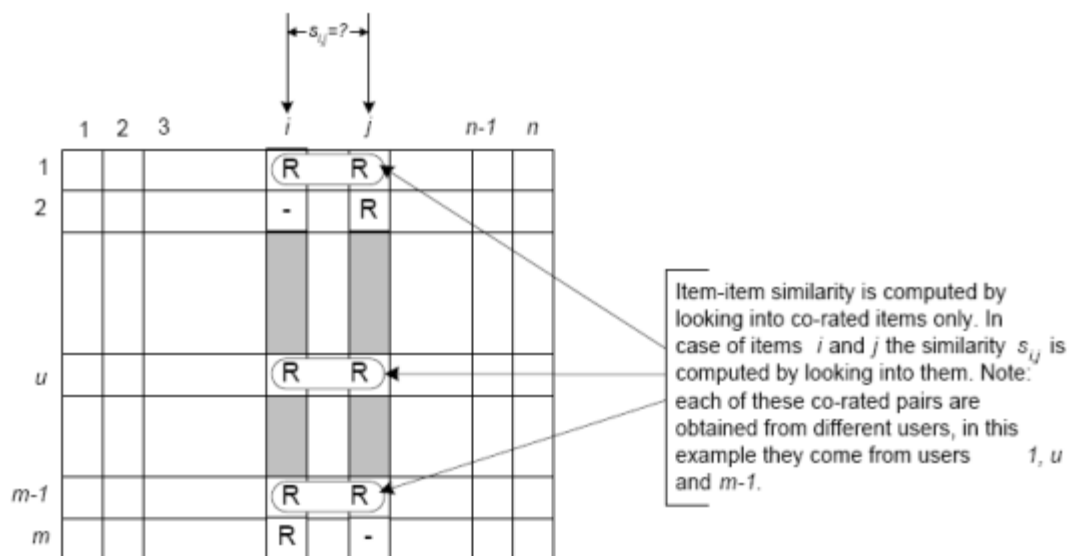
πραγματική εφαρμογή που μπορεί να υλοποιηθεί από υπολογιστικά συστήματα που συναντούμε παγκοσμίως. Αν και με βάση το θεωρητικό επίπεδο της μεθόδου, μπορούμε να κάνουμε την εκτίμηση ότι η μέθοδος μπορεί να συγκλίνει αργά, ωστόσο η κατάλληλη επιλογή των σταθερών οδηγεί σε γρήγορη σύγκλιση της μεθόδου. Συγκεκριμένα, σύμφωνα με τον αλγόριθμο PageRank, στον οποίο έχουμε βασιστεί, ορίζουμε τη μεταβλητή α στη τιμή 0,85, όπως προτείνεται, και επιτυγχάνουμε γρήγορη σύγκλιση (ύστερα από 2-5 επαναλήψεις με βάση τα πειραματικά αποτελέσματα) της Power Μεθόδου. Έτσι, η υλοποίηση ενός ιδιαίτερα καθοριστικού βήματος του Αλγορίθμου γίνεται δυνατή και μπορούμε να προχωρήσουμε τη διαδικασία υλοποίησης.

Οι παραπάνω παρατηρήσεις, επιβεβαιώνονται και από τα δικά μας πειράματα, μετρήσεις και αποτελέσματα. Επομένως, καταλήγουμε στο συμπέρασμα ότι μπορεί να γίνει ο υπολογισμός του Μητρώου Ομοιοτήτων ασύγχρονα, με την έννοια ότι μπορεί να υπολογιστεί σε αδιάφορη (για την εφαρμογή) χρονική στιγμή, και το κομμάτι του Αλγορίθμου Συνεργατικής Διήθησης μπορεί να εκτελείται την στιγμή που αλληλεπιδρά ο Χρήστης με την εφαρμογή μας. Επίσης, καθώς η συνεχής ενημέρωση των δεδομένων της εφαρμογής οδηγεί σε καλύτερα και πιο αξιόπιστα αποτελέσματα, προτείνουμε να υπολογίζεται το Μητρώο Ομοιοτήτων ασύγχρονα, σε στιγμή που το υπολογιστικό σύστημα, που υλοποιεί τον υπολογισμό, έχει αρκετούς ελεύθερους πόρους και γενικά μπορεί να διαθέσει τους απαιτούμενους πόρους και υπολογιστική ισχύ για τον υπολογισμό του μητρώου. Έτσι, εκμεταλλευόμαστε στο έπακρο ακόμη και το υλικό (*hardware*) που έχουμε στη διάθεσή μας.

3.4.2.1 Υπολογισμός Μητρώου Ομοιοτήτων

Όπως έχουμε αναφέρει και στην προηγούμενη παράγραφο, σημαντικό στοιχείο κατά την εξαγωγή του αποτελέσματος και εξέχουσας σημασίας αποτελεί το *Μητρώο Ομοιοτήτων*. Το μητρώο αυτό περιέχει πληροφορία που προκύπτει από τις επιλογές των Χρηστών, που γνωρίζει το σύστημα, δηλαδή τα δεδομένα των Χρηστών που συλλέγουμε από το Κοινωνικό Δίκτυο.

Αναλυτικά, το *Μητρώο Ομοιοτήτων* περιέχει την *ομοιότητα* που έχει κάθε αντικείμενο με οποιοδήποτε άλλο, που περιέχεται στο σύνολο δεδομένων που έχουμε στη διάθεσή μας. Η ομοιότητα (*τιμή* της ομοιότητας) μεταξύ των αντικειμένων εξάγεται παρατηρώντας τη συμπεριφορά όλων των Χρηστών που έχουν αξιολογήσει τα αντικείμενα (Εκδηλώσεις). Όπως φαίνεται στο παρακάτω διάγραμμα, η ομοιότητα μεταξύ των δύο στοιχείων εξαρτάται από τις αξιολογήσεις που έχουν δοθεί στα αντικείμενα από τους Χρήστες οι οποίοι έχουν αξιολογήσει και τα δύο (ή περισσότερα από αυτά).



Εικόνα 3.2: Συσχέτιση Αντικειμένων στο Μητρώο Ομοιοτήτων

Υπάρχει διαθέσιμο ένα μεγάλο πλήθος διαφορετικών μαθηματικών τυποποιήσεων που μπορούν να χρησιμοποιηθούν για τον υπολογισμό της Ομοιότητας μεταξύ δύο αντικειμένων. Για τις ανάγκες της Διπλωματικής Εργασίας επιλέξαμε τις τρεις (3) πιο δημοφιλείς από αυτές τις μεθόδους. Αυτές είναι η Ομοιότητα Συνημίτονου (*Cosine-Based Similarity*), η Προσαρμοσμένη Ομοιότητα Συνημίτονου (*Adjusted Cosine Similarity*) και η Συσχέτιση κατά Pearson (*Pearson Correlation*). Οι παραπάνω μέθοδοι, έχουν προταθεί και χρησιμοποιούνται σε όλες τις δημοσιεύσεις στα διασημότερα συνέδρια, σχετικά με τη Συνεργατική Διήθηση, π.χ. το Συνέδριο *RecSys* του Οργανισμού ACM. Προφανώς, υπάρχουν και άλλοι τρόποι υπολογισμού του Μητρώου Ομοιοτήτων, αλλά οι ανάγκες της παρούσας Διπλωματικής Εργασίας οι παραπάνω μέθοδοι καλύπτονται πλήρως από τις παραπάνω μεθόδους χωρίς όμως να επηρεάζεται η συνολική απόδοση του συστήματος που προτείνουμε. Στη συνέχεια, περιγράφουμε αναλυτικά τον τρόπο υπολογισμού καθεμιάς από τις παραπάνω μεθόδους.

Όπως μπορεί να φανεί στον παρακάτω τύπος, κάθε τύπος περιλαμβάνει τους όρους οι οποίοι αθροίζονται πάνω στο σύνολο των κοινών Χρηστών.

- *Ομοιότητα Συνημίτονου (Cosine-Based Similarity)*

Η πρώτη μορφή που χρησιμοποιούμε είναι η Ομοιότητα Συνημίτονου. Πρόκειται για την πρώτη μορφή ομοιότητας που προτάθηκε ποτέ για χρήση σε Συστήματα Συστάσεων. Αν και είναι απλοϊκή η ιδέα και η υλοποίηση της, πρόκειται για ένα ισχυρό μοντέλο που, σύμφωνα με προηγούμενες δημοσιεύσεις, έχει πολύ καλά αποτελέσματα. Η υλοποίησή της, ουσιαστικά είναι ο υπολογισμός συνημίτονου δύο (2) διανυσμάτων,

τα διανύσματα 2 αντικειμένων που έχουν βαθμολογηθεί τον ίδιο προϊόν. Η μέθοδος είναι επίσης γνωστή ως ομοιότητα διανυσμάτων, η διατύπωση αυτή θεωρεί τα δύο αντικείμενα (τις βαθμολογίες τους) ως διανύσματα, και ορίζει την ομοιότητα μεταξύ τους, ως τη «γωνία» μεταξύ αυτών. Έτσι η Ομοιότητα Συνημίτονου ορίζεται ως:

$$\text{sim}(i, j) = \cos(i, j) = \frac{i \cdot j}{\|i\|_2 * \|j\|_2}$$

- *Προσαρμοσμένη Ομοιότητα Συνημίτονου (Adjusted Cosine Similarity)*
Η Προσαρμοσμένη Ομοιότητα Συνημίτονου, αποτελεί παραλλαγή της προηγούμενης μεθόδου, με σκοπό να δώσει καλύτερα αποτελέσματα. Αυτό δεν αναιρεί τις δυνατότητες της Ομοιότητας Συνημίτονου, απλά αποτελεί μια εναλλακτική πρόταση, που προσαρμόζεται στις γενικότερες αποφάσεις των Χρηστών. Ουσιαστικά, αυτή η μέτρηση ομοιότητας είναι μια τροποποιημένη μορφή του διανύσματος ομοιότητας όταν λάβουμε υπόψη το γεγονός ότι οι διάφοροι χρήστες έχουν διαφορετικά συστήματα αξιολόγησης. Με άλλα λόγια, ορισμένοι χρήστες ενδέχεται να βαθμολογούν τα αντικείμενα με υψηλές βαθμολογίες, σε γενικές γραμμές, και άλλοι μπορεί να επιλέγουν να δώσουν χαμηλότερες βαθμολογίες στα αντικείμενα. Για να καταργήσουμε αυτό το μειονέκτημα από το διάνυσμα ομοιότητας, αφαιρούμε τη μέση βαθμολογία για κάθε χρήστη από τη βαθμολογία κάθε χρήστη για το ζεύγος των εν λόγω αντικειμένων. Έτσι, η Προσαρμοσμένη Ομοιότητα Συνημίτονου υπολογίζεται από τον τύπο:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

- *Συσχέτιση κατά Pearson (Pearson Correlation)*
Αρκετά διαδεδομένη και με πολύ συχνή χρήση είναι η Συσχέτιση κατά Pearson. Η Συσχέτιση κατά Pearson, μοιάζει αρκετά με την Προσαρμοσμένη Ομοιότητα Συνημίτονου που μόλις περιγράψαμε, αλλά υπάρχει μια σημαντική διαφορά ανάμεσα σε αυτές. Σε αυτή, βασική ιδέα του υπολογισμού της ομοιότητας, είναι ο υπολογισμός του μέτρου με βάση το πόσο οι αξιολογήσεις από τους κοινούς χρήστες, για ένα ζευγάρι αντικειμένων (δεδομένου ότι και οι δύο (2) Χρήστες έχουν βαθμολογήσει και το ίδιο ζεύγος προϊόντων), αποκλίνουν από τη μέση βαθμολογία για τα στοιχεία. Έτσι, ο υπολογισμός του παραπάνω Μητρώου Ομοιότητας γίνεται ως:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}$$

Σε αυτό το σημείο, να τονίσουμε ότι καθεμιά από τις παραπάνω μεθόδους επιλύει το πρόβλημα μας, αλλά έχει διαφορετική απόδοση. Για τις ανάγκες της Διπλωματικής Εργασίας, υλοποιήσαμε και τις τρεις (3) μεθόδους, που μόλις περιγράψαμε, και ανάλογα με τα αποτελέσματα που θα προκύψουν, θα επιλέξουμε την *καταλληλότερη* τελικά μέθοδο για το πρόβλημά μας, την Στοχευμένη Πρόταση Προϊόντων σε Χρήστες Κοινωνικών Δικτύων. Αξίζει να αναφέρουμε ότι υπάρχουν και άλλες μέθοδοι υπολογισμού του Μητρώου Ομοιοτήτων, αλλά η μελέτη τους ξεφεύγει από τα όρια της παρούσας Διπλωματικής Εργασίας και οι μέθοδοι που υλοποιήθηκαν καλύπτουν πλήρως τις προθέσεις μας.

3.4.3 Τελική Πρόταση και Μέτρηση Απόδοσης

Πια, το τελικό βήμα που απομένει είναι η εύρεση των Εκδηλώσεων που θα προτείνουμε στο Χρήστη. Η πλέον σύγχρονη τάση στα Συστήματα Συστάσεων, τη στιγμή που γράφεται η παρούσα Διπλωματική Εργασία, είναι η πρόταση των *N-επικρατέστερων Προτάσεων (Top-N Recommendation)*.

Η σύσταση των N-επικρατέστερων Προτάσεων είναι μια παραλλαγή ή αλλιώς λειτουργία αξιολόγησης, της Σύστασης βασισμένη στα Αντικείμενα. Σκοπός της, είναι να προβλέψει τις καλύτερες N επιλογές. Μόνο τα N στοιχεία θα παρουσιάζονται στο χρήστη ή θα ληφθούν υπόψη για τον υπολογισμό του αποτελέσματος της αξιολόγησης. Για παράδειγμα, στο Χρήστη δεν θα προτείνουμε μόνο μία (1) αλλά ένα σύνολο από περισσότερες (N) Εκδηλώσεις στις οποίες θα βρίσκονται οι Εκδηλώσεις που υπολογίσαμε ότι έχουν το «καλύτερο αποτέλεσμα».

Με την παραπάνω μέθοδο επιλύονται προβλήματα όπως η Πρόταση ενός (1) και μόνο Αντικειμένου που μπορεί λόγω των υπολογισμών, ή μη επαρκούς γνώσης για το Χρήστη, να είναι λανθασμένη. Έτσι, με τη χρήση της μεθόδου των N-επικρατέστερων Προτάσεων ευελπιστούμε πως στο σύνολο των N Εκδηλώσεων που θα προτείνουμε να βρίσκονται μία (1) ή περισσότερες Εκδηλώσεις οι οποίες θα ενδιαφέρουν το Χρήστη.

Η επιλογή αυτών γίνεται εύκολα, απλά ταξινομώντας σε φθίνουσα σειρά τις Εκδηλώσεις με βάση το αποτέλεσμα της Μεθόδου Power, για κάθε Χρήστη.

Σε αυτό το σημείο να τονίσουμε ότι στη τελική Πρόταση δεν περιέχονται οι Εκδηλώσεις για τις οποίες έχει ήδη απαντήσει (είτε θετικά είτε αρνητικά) ο Χρήστης. Η παραπάνω παραδοχή είναι απόλυτα συμβατή με την πραγματικότητα καθώς για να έχουμε μια αποδοτική Πρόταση, η οποία ανταποκρίνεται στις πραγματικές ανάγκες του Χρήστη, δεν θα πρέπει να προτείνουμε μια Εκδήλωση ο Χρήστης που έχει ήδη επισκεφτεί ή έχει αποφασίσει εάν θα την επισκεφτεί (ή όχι).

Για τη μέτρηση της Απόδοσης του Συστήματος, θα χρησιμοποιήσουμε τις Μετρικές *Ανάκληση (Recall)* και *Ακρίβεια (Precision)*. Οι μετρικές αυτές, θα μας δείξουν, ποιό από τα Μητρώα Ομοιότητας που χρησιμοποιούμε παράγει τα καλύτερα αποτελέσματα («ταιριάζει» καλύτερα) στο Πρόβλημά μας, δίνοντας καλύτερα τελικά αποτελέσματα. Έτσι, στο εξής θα χρησιμοποιούμε το συγκεκριμένο μητρώο για την Πρόταση Εκδηλώσεων στους Χρήστες Κοινωνικών Δικτύων. Συγκεκριμένα, για να ελέγξουμε την απόδοση, χωρίσαμε τα δεδομένα από το Κοινωνικό Δίκτυο σε δύο (2) μεγάλα υποσύνολα, το Σύνολο Εκπαίδευσης (Train Set) και το Σύνολο Ελέγχου (Test Set), όπως έχουμε ήδη αναφέρει και παραπάνω. Οι αποφάσεις-προτάσεις που εξάγουμε, για κάθε Χρήστη του Συνόλου Δεδομένων που χρησιμοποιούμε, αντιπαραβάλλονται με τις πραγματικές αποφάσεις του ίδιου Χρήστη που περιέχονται στο Σύνολο Ελέγχου.

Οι μετρικές Ακρίβεια και Ανάκληση υπολογίζονται ως προς το σύνολο των σωστών Προτάσεων που επεστράφησαν σε κάθε περίπτωση. Συγκεκριμένα, η Ακρίβεια ορίζεται ως το πλήθος των αποφάσεων (Εκδηλώσεων στην περίπτωση μας) διά το πλήθος των συνολικών αποφάσεων που έπρεπε να επιστρέψει η Εφαρμογή ανάλογα με το N που έχουμε ορίσει. Βασισμένη στη ίδια λογική, η Ανάκληση ορίζεται ως το πλήθος των αποφάσεων του Χρήστη διά το πλήθος των αποφάσεων που έπρεπε να επιστρέψει η Εφαρμογή (ανάλογα με τις επιλογές του χρήστη που γνωρίζουμε. Συνοπτικά, οι μαθηματικοί τύποι είναι:

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

Όπου, και οι δύο (2) τιμές βρίσκονται στο εύρος $[0, 1]$.

Για τον καλύτερο δυνατό έλεγχο των αποτελεσμάτων και προς απόδειξη στους δύσπιστους για τα αποτελέσματα της Εφαρμογής, οι τελικές τιμές της Ακρίβειας και της Ανάκλησης υπολογίζονται με τη μέθοδο της 5-πλης Διασταυρωμένης Επικύρωσης (5-Fold Cross Validation). Η Μέθοδος της Διασταυρωμένης Επικύρωσης είναι μια τεχνική επικύρωσης για την εκτίμηση του πώς τα αποτελέσματα μιας στατιστικής ανάλυσης γενικεύονται σε ανεξάρτητο σύνολο

δεδομένων. Χρησιμοποιείται κυρίως σε χώρους όπου στόχος είναι η πρόβλεψη, και κάποιος επιθυμεί να εκτιμήσει την ακρίβεια ενός προγνωστικού μοντέλου στην πράξη. Στην περίπτωση μας, θα χρησιμοποιήσουμε πέντε (5) διαφορετικά σύνολα για να καταλήξουμε στην τελική τιμή. Η χρήση πέντε (5) τιμών είναι μια τιμή που χρησιμοποιείται ευρέως και είναι αποδεκτή για την επιστημονική και ερευνητική κοινότητα.

Τέλος, σε μια πραγματική εφαρμογή θα υπάρχουν πάντα οι περιπτώσεις των Χρηστών οι οποίοι είναι νέοι (ή «ξένοι») προς το Σύστημα. Αναφερόμαστε στους Χρήστες, για τους οποίους το σύστημα δεν έχει (ακόμα) γνώση των προηγούμενων προτιμήσεών τους. Έτσι, δεν μπορεί να γίνει προσωποποιημένη πρόταση. Σε αυτή την περίπτωση, θα προτείνουμε στο Νέο Χρήστη Εκδηλώσεις με βάση τα δεδομένα των υπολοίπων Χρηστών. Συγκεκριμένα, για κάθε νέο Χρήστη, προτείνουμε τις Εκδηλώσεις που προτίμησαν περισσότερο οι υπόλοιποι Χρήστες, δηλαδή τις Εκδηλώσεις με τις περισσότερες «θετικές» αποφάσεις των Χρηστών. Έτσι, για τον υπολογισμό αυτών των Εκδηλώσεων, απλά προσθέτουμε τις θετικές αποφάσεις (βαθμολογία «5») ανά Εκδήλωση και, τελικά, η Εκδήλωση με το υψηλότερο άθροισμα θα είναι η επικρατέστερη.

Εννοείται ότι για κάθε νέο Χρήστη οι Προτάσεις θα είναι ίδιες. Αυτή η εκτίμηση δεν αναμένουμε να έχει την απόδοση των προαναφερθέντων προσποιημένων μεθόδων και χρησιμοποιείται για να καλύψουμε όλους τους Χρήστες και κάθε δυνατή περίπτωση που μπορεί να προκύψει.

4. Υλοποίηση

4.1 Εισαγωγή

Στην εργασία προτιμήθηκε η πρόταση του προϊόντος *Εκδήλωση (Event)* για τους Χρήστες της εφαρμογής και κατ' επέκταση του Κοινωνικού Δικτύου. Η επιλογή αυτή έγινε καθώς οι Εκδηλώσεις έχουν απήχηση στο ευρύ κοινό καθώς η τελική πρόταση μπορεί να είναι μια συναυλία ή μια βραδινή έξοδος ανάλογα με τις προτιμήσεις του Χρήστη. Οι Χρήστες της εφαρμογής παρέχουν πρόσβαση στα δεδομένα αυτών και των φίλων τους και τελικά πληθυσμώνεται η Βάση Δεδομένων με τις προτιμήσεις τους (Χρηστών και Φίλων), είτε θετικές είτε αρνητικές, ώστε τελικά αυτά τα δεδομένα να εξεργαστούν και να δώσουν την τελική πρόταση στο Χρήστη μας.

Όπως έχουμε ήδη αναφέρει αρκετές φορές η εργασία μπορεί να χωριστεί σε δύο (2) κύρια συστατικά μέρη: την Διεπαφή του Χρήστη και την υλοποίηση του αλγορίθμου Συνεργατικής Διήθησης.

Η ακριβής διαδικασία που ακολουθήθηκε περιγράφεται στις παρακάτω παραγράφους, ενώ στο Παράρτημα παρατίθεται ο κώδικας που αναπτύχθηκε επαρκώς σχολιασμένος.

4.2 Δημιουργία Διεπαφής Χρήστη

Το πρώτο μέρος απαιτεί την δημιουργία μιας Εφαρμογής στο Κοινωνικό Δίκτυο Facebook (Facebook app), ώστε να γίνει λήψη των δεδομένων του Χρήστη. Με αυτό τον τρόπο μπορούμε να πάρουμε τα δεδομένα του Χρήστη και έτσι να μπορέσουμε να «γνωρίσουμε» καλύτερα το Χρήστη για να μπορέσουμε να κάνουμε μια «επιτυχή» πρόταση. Όσα περισσότερα δεδομένα έχουμε για το Χρήστη ή όσους περισσότερους Χρήστες έχουμε τόσο «καλύτερη» (πιο κοντά στις προτιμήσεις του Χρήστη) θα είναι η πρόταση που θα παράγει το σύστημα.

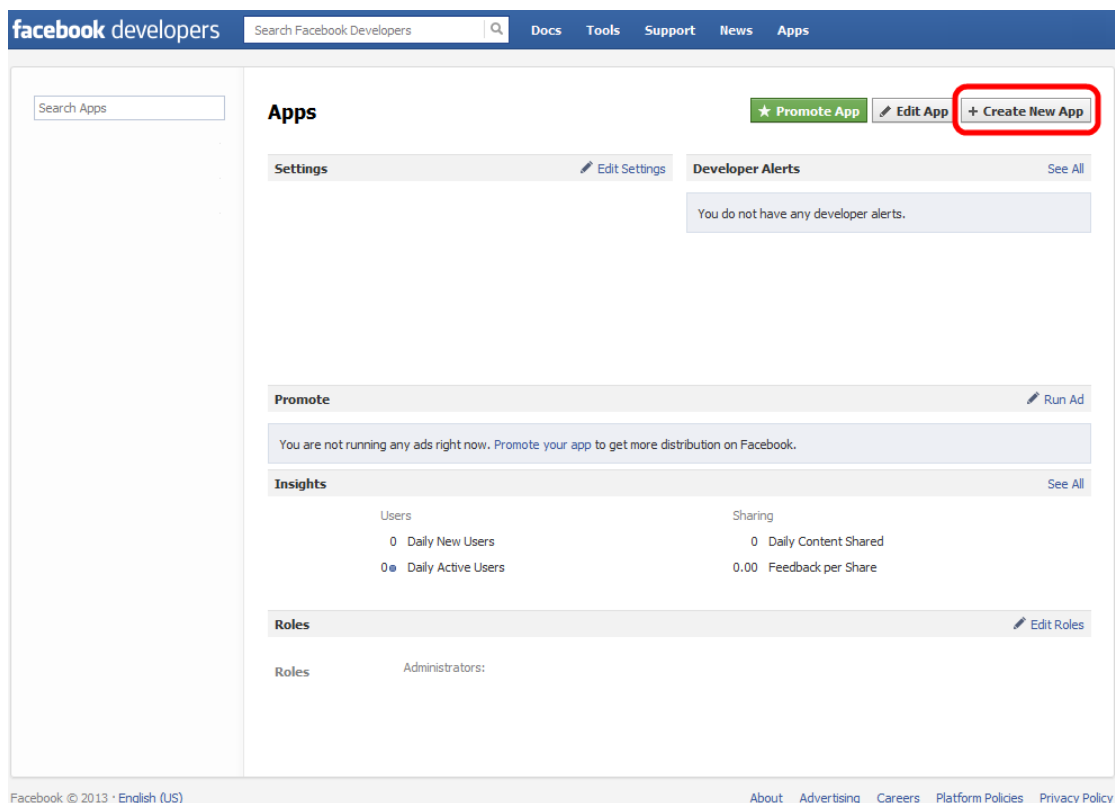
Η διαδικασία αυτή μπορεί και αυτή με τη σειρά της να χωριστεί σε 2 επιμέρους διαδικασίες.

4.2.1 Κατοχύρωση της Εφαρμογής στο Facebook

Η εφαρμογή που αναπτύξαμε είναι διαθέσιμη μέσω του Κοινωνικού Δικτύου Facebook. Έτσι, θα πρέπει να την κατοχυρώσουμε σε αυτό και να συμπληρώσουμε κατάλληλα τα πεδία ώστε να γίνει δυνατή η εκτέλεση της μέσα από το Κοινωνικό Δίκτυο.

Τα βήματα που ακολουθήσαμε είναι:

1. Μεταβαίνουμε στην ηλεκτρονική διεύθυνση:
<https://developers.facebook.com/apps>.
και ύστερα επιλέγουμε την **Create New App**.



Εικόνα 4.1: Δημιουργία Facebook App - Βήμα 1

Σε αυτή τη φάση να υπενθυμίσουμε ότι τόσο ο συγγραφέας της παρούσας εργασίας όσο και ο Χρήστης θα πρέπει να είναι και χρήστες του Κοινωνικού Δικτύου Facebook, ώστε να χρησιμοποιήσουν την εφαρμογή μας και να μπορέσουν να προχωρήσουν στα υπόλοιπα βήματα της διαδικασίας.

2. Στο νέο παράθυρο που εμφανίζεται, συμπληρώνουμε το πεδίο **App Name** με το όνομα που επιθυμούμε να έχει η εφαρμογή μας (επιλέχθηκε το *Evnt Fndr*) και επιλέγουμε **Continue**. Έπειτα, συνεχίζουμε συμπληρώνοντας τον έλεγχο ασφαλείας και προχωρούμε στο επόμενο βήμα.

Εικόνα 4.2: Δημιουργία Facebook App - Βήμα 2

3. Στο επόμενο βήμα συμπληρώνουμε κατάλληλα τα πεδία που αφορούν την αλληλεπίδραση της εφαρμογής μας με το Κοινωνικό Δίκτυο. Θα πρέπει να συμπληρώσουμε τα πεδία με βάση τα στοιχεία που μας έχει δώσει ο πάροχος του εξυπηρετητή διαδικτύου (web server). Το Κοινωνικό Δίκτυο Facebook έχει επιλέξει να μην συμμετέχει στη διαδικασία φιλοξενίας των εφαρμογών, που τρέχουν σε αυτό, αλλά συμπεριλαμβάνει στις σελίδες του το περιεχόμενο από τον εξυπηρετητή που φιλοξενεί την εφαρμογή. Τέλος, επιλέγουμε **Save Changes** για να ολοκληρωθεί η διαδικασία.

The screenshot shows the Facebook Developers interface for the 'Evtnt Fndr' app. The left sidebar contains navigation links for Settings (Basic, Developer Roles, Permissions, Payments, Realtime Updates, Advanced), App Details (App Details, Review Status, Open Graph, Localize, Alerts, Insights, Promote), and Related links (Use Debug Tool, Use Graph API Explorer, Use Object Browser, See App Timeline View, Delete App). The main content area is titled 'Apps > Evtnt Fndr > Basic'. It displays the app's logo, ID, and secret, along with a status indicator 'This app is live'. The 'Basic Info' section includes fields for Display Name, Namespace, Contact Email, App Domains, Hosting URL, and Sandbox Mode. The 'Select how your app integrates with Facebook' section contains three expandable panels: 'Website with Facebook Login' (with Site URL), 'App on Facebook' (with Canvas Page, Canvas URL, Secure Canvas URL, Canvas Width, and Canvas Height), and 'Page Tab' (with Page Tab Name, Page Tab URL, Secure Page Tab URL, Page Tab Edit URL, Page Tab Image, and Page Tab Width). At the bottom, there are checkboxes for 'Mobile Web', 'Native iOS App', and 'Native Android App', and a 'Save Changes' button.

Εικόνα 4.3: Δημιουργία Facebook App - Βήμα 3

Πλέον, το Κοινωνικό Δίκτυο Facebook κάθε φορά που θα ζητείται πρόσβαση στην εφαρμογή μας θα εμφανίζει το περιεχόμενο από τον εξυπηρετητή (web server) που έχουμε ορίσει.

4.2.2 Ανάπτυξη Κώδικα

Μόλις ολοκληρώσουμε τα παραπάνω βήματα μπορούμε να αναπτύξουμε την Διεπαφή του Χρήστη και τα «λειτουργικά» μέρη της εφαρμογής που είναι υπεύθυνα κυρίως για τη λήψη των δεδομένων του χρήστη και την αποθήκευση αυτών.

4.2.2.1 Διεπαφή

Η Διεπαφή που εμφανίζεται στο Χρήστη γίνεται με χρήση HTML και JavaScript, όπως και κάθε άλλη σελίδα του παγκόσμιου ιστού. Ένα PHP αρχείο παράγει τον HTML κώδικα. Για παράδειγμα:

Κώδικας

```
...
<header class="clearfix">
<?php if (isset($basic)) { ?>
    <div id="picture" style="background-image:
url(https://graph.facebook.com/<?php echo he($user_id);
?>/picture?type=normal)"></div>

    <div>
        <h1>Welcome <?php echo he(idx($basic,
'first_name')); ?></h1>

        <div id="share-app">
            <p>Spread the word:</p>
            <ul>
                <li>
                    <a href="#"
class="facebook-button" id="postToWall" data-url="<?php
echo AppInfo::getUrl(); ?>">
                <span class="plus">Post to Wall</span>
            </li>
        </ul>
    </div>
</header>
...
```

Με χρήση της JavaScript γίνονται οι ασύγχρονες κλήσεις από και προς το Κοινωνικό Δίκτυο ώστε να λάβουμε την απαραίτητη άδεια (με βάση τα στοιχεία της εφαρμογής μας) ώστε να συνεχιστεί η διαδικασία και να μπορέσουμε να έχουμε πρόσβαση στα δεδομένα του χρήστη. Για παράδειγμα:

Κώδικας

```

...
<script type="text/javascript">
    window.fbAsyncInit = function() {
        FB.init({
            appId      : '<?php echo AppInfo::appID();
?>', // App ID
            channelUrl : '//'<?php echo
$_SERVER["HTTP_HOST"]; ?>/channel.html', // Channel File
            status     : true, // check login status
            cookie     : true, // enable cookies to
allow the server to access the session
            xfbml      : true // parse XFBML
        });

        // Listen to the auth.login which will be
called when the user logs in
        // using the Login button
        FB.Event.subscribe('auth.login',
function(response) {
            // We want to reload the page now so PHP
can read the cookie that the
            // Javascript SDK sat. But we don't want
to use
            // window.location.reload() because if
this is in a canvas there was a
            // post made to this page and a reload
will trigger a message to the
            // user asking if they want to send data
again.
            window.location = window.location;
        });

        FB.Canvas.setAutoGrow();
    };

    // Load the SDK Asynchronously
    (function(d, s, id) {
        var js, fjs = d.getElementsByTagName(s)[0];
        if (d.getElementById(id)) return;
        js = d.createElement(s); js.id = id;
        js.src = "//connect.facebook.net/en_US/all.js";
        fjs.parentNode.insertBefore(js, fjs);
    }(document, 'script', 'facebook-jssdk'));
</script>
...

```

Επίσης, στην αρχή του αρχείου απαιτούμε να συμπεριληφθούν τα βοηθητικά αρχεία που είναι απαραίτητα καθώς περιέχουν βοηθητικές συναρτήσεις.

Κώδικας

```
...
//Require files

    // This provides access to helper functions defined
in 'utils.php'
    require_once('utils.php');

    //Database funtions
    require_once('model.php');

    // Provides access to app specific values such as
your app id and app secret defined in 'AppInfo.php'
    require_once('AppInfo.php');

    //Connect to database
    $con = connect2db();
...
```

Τέλος, με χρήση της jQuery βιβλιοθήκης γίνεται ασύγχρονη κλίση των PHP αρχείων που υλοποιούν τις κλήσεις προς το Κοινωνικό Δίκτυο, για τη λήψη των δεδομένων, και αποθηκεύουν τα ληφθέντα δεδομένα.

Κώδικας

```
...
    <script>

        $.get("get_events.php", {appuser: "<?php echo
$user_id; ?>"}, function() {
            //alert("success");
        });

        $.get("get_checkins.php", {appuser: "<?php echo
$user_id; ?>"}, function() {
            //alert("success");
        });

    </script>
...
```

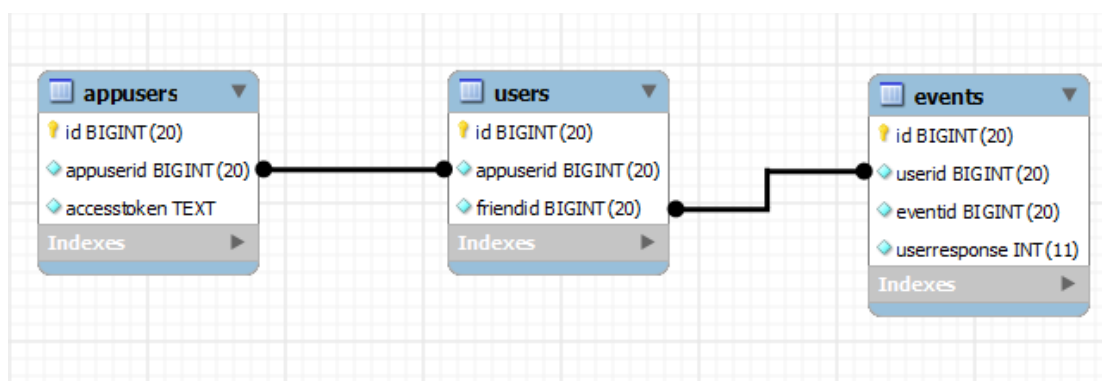
4.2.2.2 Σχήμα Βάσης Δεδομένων

Όπως έχουμε αναφέρει και πιο πάνω στην εργασία επιλέχτηκε χρήση βάσης δεδομένων MySQL.

Συνοπτικά, η Βάση Δεδομένων αποτελείται από τρεις (3) πίνακες που περιέχουν τους:

- Χρήστες της εφαρμογής (*appusers*),
- τους Φίλους των Χρηστών (*users*) και
- τα κυρίως δεδομένα από τα οποία πρόκειται να εξάγουμε τις προτάσεις (*events*)

Το *Διάγραμμα Συσχετίσεων Οντοτήτων (ER Διάγραμμα)* της Βάσης Δεδομένων είναι:



Εικόνα 4.4: ER διάγραμμα της Βάσης Δεδομένων

4.2.2.3 Λήψη, Αποθήκευση και Εξαγωγή Αρχικών Δεδομένων

Η λήψη, αποθήκευση και εξαγωγή των δεδομένων γίνεται με χρήση της γλώσσας προγραμματισμού PHP. Η επιλογή αυτή δεν είναι τυχαία καθώς η γλώσσα παρέχει πολλές δυνατότητες και μεγάλη ευελιξία. Η διαδικασία που περιγράφεται παρακάτω έχει υλοποιηθεί με την αναφερθείσα γλώσσα.

Αρχικά, στέλνουμε τα *Αιτήματα (Requests)* για να μας «απαντήσει» το Κοινωνικό Δίκτυο με τις προηγούμενες προτιμήσεις του Χρήστη.

Κώδικας

```

...
$events_raw = $facebook->api('/', $user .
'?fields=events.type(' . $event_resp . ').since(' .
strtotime('-2 month') . ').limit(100)');
...
  
```

Τα δεδομένα επιστρέφονται σε μορφή PHP-Πίνακα (PHP array) και συμπεριλαμβάνουν επιπλέον πληροφορία που για το πρόβλημά μας δεν

χρειάζονται. Έτσι, κρατάμε μόνο τα δεδομένα που μας ενδιαφέρουν. Η επεξεργασία αυτή γίνεται εξ' ολοκλήρου με PHP και εκμεταλλευόμενοι τα χαρακτηριστικά της γλώσσας μπορούμε να πετύχουμε ένα πολύ καλό αποτέλεσμα με πολύ καλή επίδοση τόσο από άποψη πόρων αλλά και ταχύτητας. Σε αυτό το σημείο να υπενθυμίσουμε ότι είναι σημαντικό σημείο η χρησιμοποίηση όσο το δυνατόν λιγότερων πόρων καθώς και η γρήγορη ολοκλήρωση όλων των επιμέρους λειτουργιών και διαδικασιών καθώς το πρόβλημα της σύστασης εξ' ορισμού απαιτεί μεγάλο όγκο δεδομένων που πρέπει να επεξεργαστεί για να έχουμε μια πρόταση με τα καλύτερα δυνατά αποτελέσματα.

Κώδικας

```
...
if ( array_key_exists('events', $events_raw) ){

    $events = array();
    $events = $events_raw['events']['data'];

    foreach ($events as $event){

        $event_name = $event['name'];
        $event_id = $event['id'];
        $event_start_time = $event['start_time'];

        //echo 'Event Name: ' . $event_name;
        //echo ' (ID: ' . $event_id . ' ) ';
        //echo ' Start time: ' . $event_start_time . '
( or ' . strtotime($event_start_time) . ' ) <br />' ;

        insert_event( $con, $user, $event_id, $resp_id
    );

        //echo '<br />';echo '<br />';

    }

}
...
```

Τελικά, αποθηκεύουμε τα δεδομένα στη Βάση Δεδομένων.

Κώδικας

```
...
//insert new event
function insert_event_query( $con, $user_id, $event_id,
$user_response ){

    $q = mysqli_query( $con, 'INSERT INTO `events`
VALUES ( NULL , '$user_id.', ' . $event_id . ', ' .
$user_response . ' ) ON DUPLICATE KEY UPDATE userresponse
= ' . $user_response );
    return $q;
}
...
```

Στη Βάση Δεδομένων πλέον έχουμε τα ακατέργαστα δεδομένα όπως αυτά επιστρέφονται από το Κοινωνικό Δίκτυο. Αυτά τα δεδομένα πρόκειται να υποστούν επεξεργασία και για να είναι πιο εύκολη η μετέπειτα διαχείριση τους αναπτύχθηκε κώδικας που τα εξάγει σε διαχωριζόμενη μορφή (comma separated values - csv) τα δεδομένα.

4.2.2.4 Βοηθητικά Αρχεία και Λεπτομέρειες

Παράλληλα με τα παραπάνω, αναπτύχθηκε επιπλέον κώδικας που εξυπηρετεί σε ορισμένες ενέργειες και βοηθά στην ορθή δομή και διάρθρωση του συνολικού πηγαίου κώδικα που αναπτύχθηκε. Συγκεκριμένα, ακολουθήθηκε μορφή Αρχιτεκτονικής 3 Επιπέδων (3 Layer Architecture). Κάθε αρχείο του ενός επιπέδου ανταλλάσσει δεδομένα και «επικοινωνεί» μόνο με αρχεία του ακριβώς επόμενου ή προηγούμενου επιπέδου. Αυτό δίνει τη ευελιξία στην εφαρμογή μας και βοηθά στην εύκολη επίλυση προβλημάτων που μπορεί να προκύψουν.

Κώδικας

```
...

function select_friend( $con, $user_id, $limit, $offset
){

    $res = select_friend_query( $con, $user_id, $limit,
$offset );
    return $res;
    /*
    $result = $res;
    free_result($res);
    return $result;
    */
}
```



```
        */
    }
    // end ---

    // Get Long Lived Access Token for the appuser
    function get_long_lived_access_token( $accesstoken ) {

        $extend_url =
        'https://graph.facebook.com/oauth/access_token?grant_type
        =fb_exchange_token&client_id=' . AppInfo::appID() .
        '&client_secret=' . AppInfo::appSecret() .
        '&fb_exchange_token=' . $accesstoken;

        $response = file_get_contents( $extend_url );
        parse_str( $response, $parsed_resp );

        $new_access_token = $parsed_resp['access_token'];

        return $new_access_token;
    }
    // end ---

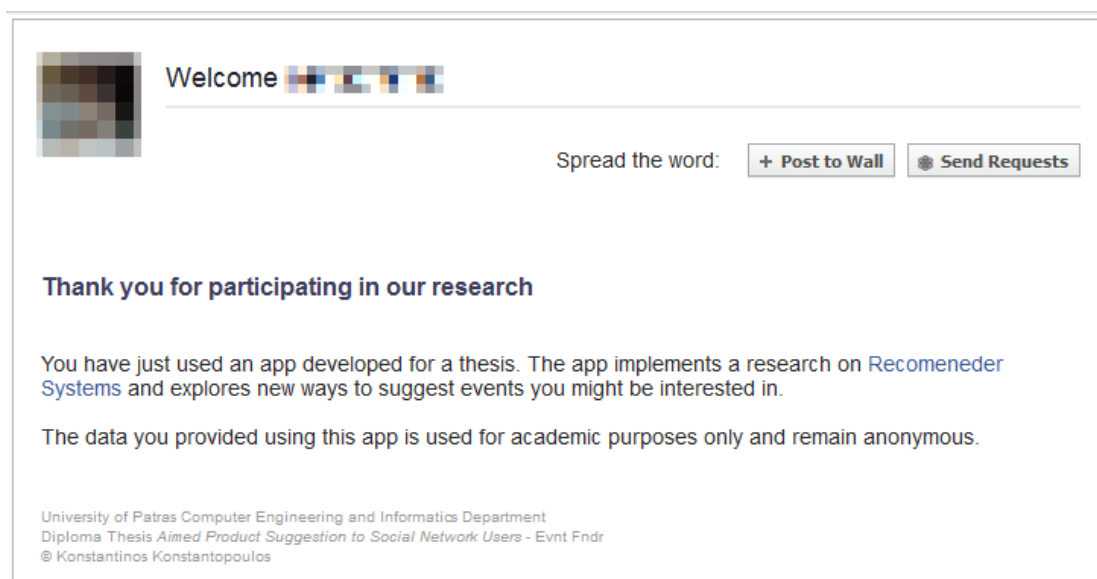
    // Preparing output files
    function get_all_data_output( $con, $table, $offset ){

        if ( $table == 'events' )
            $res = select_all_events_output( $con, 0,
            $offset );
        else if ( $table == 'checkins' )
            $res = select_all_checkins_output( $con, 0,
            $offset );

        return $res;
        /*
        $result = $res;
        free_result($res);
        return $result;
        */
    }
    ...
}
```

4.2.3 Τελικό Αποτέλεσμα

Το τελικό αποτέλεσμα που εμφανίζεται στο Χρήστη έχει την παρακάτω μορφή:



Εικόνα 4.5: Τελική εμφάνιση εφαρμογής - Προβολή Χρήστη

Ο Χρήστης μόλις επιλέξει την εφαρμογή εμφανίζεται η παραπάνω προβολή και δεν επηρεάζεται καθόλου η συνολική εμπειρία, αφού για παράδειγμα δεν περιμένει να γίνουν τα αιτήματα στο Κοινωνικό Δίκτυο. Όλες οι υπόλοιπες διαδικασίες γίνονται στο παρασκήνιο και δεν απαιτείται καμιά άλλη ενέργεια από το Χρήστη.

4.3 Υλοποίηση Αλγορίθμου Συνεργατικής Διήθησης

Στο δεύτερο μέρος της Διπλωματικής Εργασίας υλοποιήσαμε τον Αλγόριθμο Συνεργατικής Διήθησης. Ο αλγόριθμος αυτός είναι υπεύθυνος να εξάγει το τελικό αποτέλεσμα, την Πρόταση του Προϊόντος στο Χρήστη. Λαμβάνει ως είσοδο τα δεδομένα από το Κοινωνικό Δίκτυο και με κατάλληλη επεξεργασία αυτών καταλήγει στην καλύτερη επιλογή για το Χρήστη, βασισμένος στα δεδομένα του. Η επεξεργασία είναι μια πολύπλοκη και επίπονη διαδικασία και για αυτό το λόγο κρίθηκε κατάλληλη η επιλογή της Matlab. Αυτή είναι μια επιλογή κοινά αποδεκτή παγκοσμίως, καθώς όλα τα μεγάλα εργαστήρια και, σχεδόν, οι δημοσιεύσεις που έχουν γίνει, και σχετίζονται με αλγορίθμους Συνεργατικής Διήθησης, χρησιμοποιούν την παραπάνω τεχνολογία.

Αρχικά, γίνεται μια προεπεξεργασία των δεδομένων, που προήλθαν από το Κοινωνικό Δίκτυο, ώστε να βρίσκονται σε κατάλληλη μορφή για να τα επεξεργαστεί ο αλγόριθμος. Τελικά μέσω μιας επαναληπτικής διαδικασίας εξάγεται η τελική απόφαση, Πρόταση Προϊόντος στο Χρήστη.

4.3.1 Προεπεξεργασία Δεδομένων

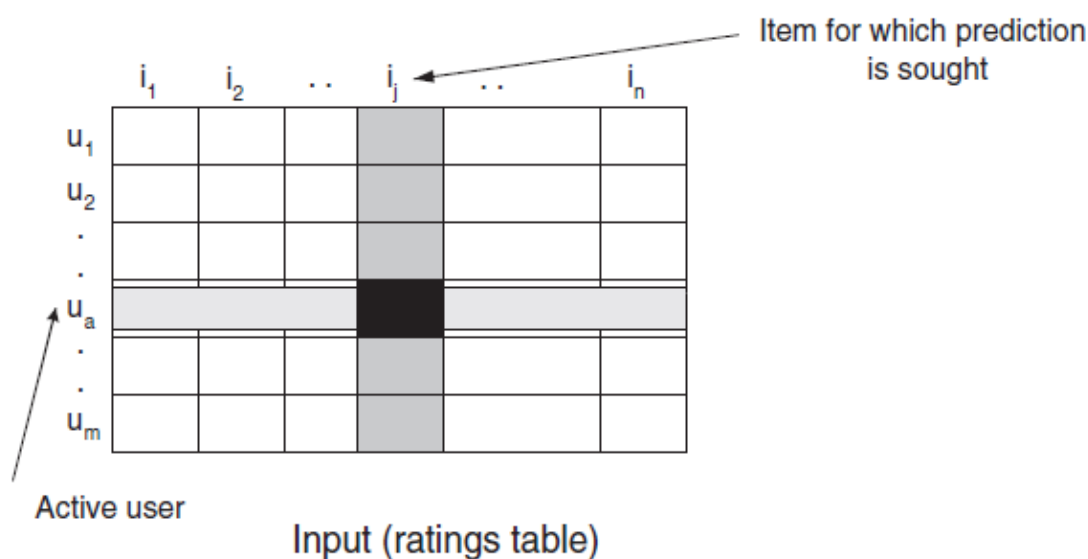
Η Προεπεξεργασία των Δεδομένων έχει στόχο την μετατροπή των δεδομένων, των Χρηστών του Κοινωνικού Δικτύου, στη μορφή που απαιτεί ο Αλγόριθμος Συνεργατικής Διήθησης για να λειτουργήσει.

Η διαδικασία είναι σχετικά απλή καθώς μετατρέπει τα δεδομένα, που έχουν εξαχθεί από το Κοινωνικό Δίκτυο (οι Τριάδες που έχουμε αναφέρει), σε ένα Δισδιάστατο Πίνακα (2D Array).

Συγκεκριμένα, από το Κοινωνικό Δίκτυο εξάγουμε τα δεδομένα σε τριάδες της μορφής:

Χρήστης, Εκδήλωση, Απάντηση_Χρήστη

Μετατρέπουμε αυτά τα δεδομένα σε ένα Δισδιάστατο Πίνακα οι Γραμμές του οποίου αντιστοιχίζονται στους Χρήστες της εφαρμογής και οι Στήλες στα Αντικείμενα προς Πρόταση, δηλαδή στις Εκδηλώσεις. Η τελική μορφή του πίνακα είναι:



Εικόνα 4.6: Δισδιάστατος πίνακας δεδομένων Αλγορίθμου Συνεργατικής Διήθησης

Κάθε στοιχείο του πίνακα αντιστοιχεί στην απάντηση του Χρήστη προς την Εκδήλωση, όπως έχουμε αναφέρει και παραπάνω, δηλαδή έχει την τιμή «1» ή «5». Εάν ο Χρήστης δεν έχει απαντήσει για κάποια Εκδήλωση τότε το στοιχείο έχει την τιμή «0».

Αυτή η διαδικασία υλοποιείται από τον κώδικα:

Κώδικας

```

% Import raw data from input file
R = dlmread('events.csv', ',');

% Create Item-User matrix
temp = R(1,1);
count = 1;
R(1,1) = count;
for i=2:length(R(:,1))
    if R(i,1)~= temp
        temp = R(i,1);
        count = count+1;
    end
    R(i,1)=count;
end

% R = [R(:,2) R(:,1) R(:,3)];
[~, I] = sort(R(:,2));
R = R(I,:);

temp = R(1,2);
count = 1;
R(1,2) = count;
for i=2:length(R(:,1))
    if R(i,2) ~= temp
        temp = R(i,2);
        count = count+1;
    end
    R(i,2)=count;
end

 [~, I] = sort(R(:,1));
R = R(I,:);

% Final matrix
X = sparse(R(:,2), R(:,1),R(:,3));

```

Εδώ να τονίσουμε ότι ο συγκεκριμένος πίνακας που πρικήπτει, είναι εξαιρετικά *Αραιός* (*πυκνότητα* δεδομένων περίπου 1%) και για λόγους που καθιστούν εφικτή την υλοποίηση ο πίνακας που δημιουργήσαμε αντιμετωπίζεται ως τέτοιος. Είναι λογικό να είναι αραιός και αυτή η αντιμετώπιση βοηθά στη γρήγορη εξαγωγή αποτελεσμάτων με χαμηλή κατανάλωση πόρων.

Αφού ολοκληρωθεί αυτή η διαδικασία πρέπει να δημιουργήσουμε τα *Υποσύνολα Εκπαίδευσης και Έλεγχου*. Αυτά τα σύνολα, όπως έχουμε ήδη αναφέρει, χρησιμοποιούνται για τον αρχικό καθορισμό των παραμέτρων (Υποσύνολο Εκπαίδευσης) και τον έλεγχο της απόδοσης (Υποσύνολο Ελέγχου) του συστήματος.

Ένα ποσοστό της τάξης του 80% χρησιμοποιείται στο Υποσύνολο Εκπαίδευσης και το υπόλοιπο 20% στο Υποσύνολο Ελέγχου.

Έτσι, προκειμένου να γίνει ο παραπάνω διαχωρισμός, αναπτύχθηκε ο κώδικας:

Κώδικας

```
% Find Train - Test Sets dimensions
M = max(R(:,1));
N = max(R(:,2));

% Create Train - Test Sets
% Create Test Sets
TestSet1 = sparse(N,M);

...

[rows, cols, vals] = find(X);
for i=1:size(rows)
    chooseTestSet = rand;

    if chooseTestSet < 0.2
        TestSet1(rows(i), cols(i)) = vals(i);
        continue
    end

...
end

% Create Train Sets
TrainSet1 = X - TestSet1;
...

TrainSet1 = sparse(TrainSet1);
...

% Verification
% nnz(X)
% nnz(TestSet1) + nnz(TrainSet1)

...
```

Από την στιγμή που τα δεδομένα μας είναι έτοιμα, προχωρούμε με την υλοποίηση του Αλγορίθμου Συνεργατικής Διήθησης.

4.3.2 Αλγόριθμος Συνεργατικής Διήθησης

Η υλοποίηση του αλγορίθμου συντονίζεται από ένα κύριο αρχείο, το οποίο καλεί τις συναρτήσεις που δημιουργήσαμε και υπολογίζουν τις επιμέρους διεργασίες. Ύστερα από την προετοιμασία των δεδομένων, που περιγράψαμε στην προηγούμενη παράγραφο, προχωρούμε με τον υπολογισμό των Μητρώων Ομοιότητας (ένα για κάθε μέθοδο που προτείνουμε).

Στη συνέχεια, προχωρούμε με τα βήματα του αλγορίθμου υπολογίζοντας το μητρώο που θα χρησιμοποιήσουμε (δώσουμε ως είσοδο) στην Επαναληπτική Διαδικασία Power, και καλούμε τη συνάρτηση που υλοποιεί αναδρομικά την επαναληπτική διαδικασία.

Κώδικας

```
...  
P = a .* SimilaritesArray + (1-a) .* ( ones(length(w), 1)  
* w' );  
  
% Power Iteration  
P_new = doPowerMethod(P, w, 0, maxIts, errorTol);  
...
```

Σημαντικός παράγοντας, που επηρεάζει σε πολύ μεγάλο βαθμό την υλοποίηση ή μη του συστήματος, είναι οι ενδιάμεσες κανονικοποιήσεις των διανυσμάτων που χρησιμοποιούνται στους υπολογισμούς. Η κανονικοποίηση που επιλέγουμε και υλοποιούμε σε κάθε περίπτωση βασίζεται, στο γεγονός ότι επιλέγουμε το αποτέλεσμα να είναι στοχαστικό κατά στήλες. Για να επιτύχουμε το παραπάνω, απλά διαιρούμε κάθε στοιχείο του διανύσματος με το άθροισμα των στοιχείων του διανύσματος, και υλοποιείται με το εξής τμήμα κώδικα:

Κώδικας

```
...  
Pref = sparse(m,n);  
  
for i=1:n  
    outdeg = sum(TrainSet(:,i));  
    if outdeg  
        Pref(:,i) = TrainSet(:,i)/outdeg;  
    end  
end
```

```
end  
end  
...
```

Επόμενο βήμα, είναι ο υπολογισμός της Διαδικασίας Power.

Σε αυτό το σημείο να αναφέρουμε, ότι αυστηρά η Διαδικασία Power, ορίζεται ως επαναληπτική μέθοδος. Ωστόσο, υλοποιήσαμε αυτή την επαναληπτική διαδικασία με μια αναδρομική σχέση που έχει παρόμοια συμπεριφορά και τα ίδια ακριβώς αποτελέσματα. Η επιλογή αυτή έγινε καθώς, στα υπολογιστικά συστήματα, η χρήση αναδρομικών σχέσεων έχει καλύτερη απόδοση, από άποψη χρήσης πόρων και ταχύτητας εκτέλεσης, από τη συμβατική χρήση ενός βρόγχου επανάληψης. Έτσι, υλοποιήσαμε τη συνάρτηση:

Κώδικας

```
function Result = doPowerMethod(P, P_old, iteration,  
maxIteration, errorTolerance)  
  
    % Show iteration - Debug only  
    iteration;  
  
    % Check if reached Max Iterations number  
    reachedMax = 0;  
    if iteration == maxIteration  
        reachedMax = 1;  
    end  
  
    % Calculate Power Method  
    P = sparse(P);  
    P_new = P*P_old;  
  
    % Normalize Vector (if at least 1 value grater than  
0)  
    if max(P_new) ~= 0  
        P_new = ( 1/sum(P_new) ) .* P_new;  
    end  
  
    % Check Error Tolerance  
    % Norm  
    errorOK = 0;  
    error = abs ( norm(P_old) - norm(P_new) );  
    if error <= errorTolerance  
        errorOK = 1;  
    end
```

```
        if ~reachedMax & ~errorOK
            Result = doPowerMethod(P, P_new, iteration+1,
maxIteration, errorTolerance);
        else
            Result = P_new;
        end
    end
end
```

4.3.3 Τελική Πρόταση και Μέτρηση Απόδοσης

Στη συνέχεια, και τελικό βήμα της μεθόδου, είναι η τελική επιλογή των Εκδηλώσεων που θα προτείνει στον εκάστοτε Χρήστη η Εφαρμογή.

Όπως αναφέραμε και στο Κεφάλαιο 3 (Παράγραφος 3.4.3) οι Προτάσεις προς το Χρήστη δεν θα περιλαμβάνουν τις Εκδηλώσεις τις οποίες ήδη γνωρίζει (έχει αποφασίσει για αυτές είτε θετικά είτε αρνητικά) ο Χρήστης. Τα παραπάνω, υλοποιούνται στον κώδικα:

Κώδικας

```
...
eventid_test_set = find(Test_Set(:,i) > 4);

...

eventid_train_set = find(Train_Set(:,i)>0);
...
```

Ουσιαστικά με το παραπάνω τμήμα κώδικα, εξετάζουμε και βρίσκουμε τις Εκδηλώσεις για τις οποίες έχει αποφασίσει θετικά ο Χρήστης για τα δύο (2) Υποσύνολα Δεδομένων Εκπαίδευσης και Ελέγχου, και εξετάζουμε ποιες από αυτές τις Εκδηλώσεις περιέχονται και στα δύο (2) Υποσύνολα.

Έπειτα, όπως έχουμε αναφέρει, ταξινομούμε σε φθίνουσα σειρά τις Εκδηλώσεις (φθίνουσα με βάση το αποτέλεσμα της Power Μεθόδου), και η τελική πρόταση της Εφαρμογής μας θα περιλαμβάνει τις πρώτες N Εκδηλώσεις, όπως αυτές προέκυψαν μετά την ταξινόμηση. Σε αυτό το σημείο, να υπενθυμίσουμε στον αναγνώστη ότι, επιλέγει η σύσταση των N-επικρατέστερων Προτάσεων (Top-N Recommendation), η οποία ορίζει ότι στο Χρήστη θα προταθεί ένα σύνολο από N Εκδηλώσεις, τις οποίες θεωρεί ότι είναι ιδανικές για αυτόν. Αυτή η επιλογή, δηλαδή η εν λόγω ταξινόμηση, χάρη στις δυνατότητες της MATLAB, επιτυγχάνεται απλά από το παρακάτω τμήμα κώδικα:

Κώδικας

```
...  
[Sorted_PI, Sorted_Codes]=sort(PI(:,1),'descend');  
  
topN_events= Sorted_Codes(1:N);  
...
```

Τέλος, απομένει η μέτρηση της Απόδοσης της Εφαρμογής. Η μέτρηση της Απόδοσης γίνεται με τη χρήση των μετρικών Ακρίβεια και Ανάκληση. Ο τρόπος υλοποίησης αυτών είναι απλός καθώς απλά υπολογίζουμε το πλήθος των επιλογών που αντιστοιχούν σε κάθε μετρική (οι μετρικές έχουν περιγραφεί πλήρως παραπάνω στο κείμενο και για αυτό δεν περιγράφεται ξανά η διαδικασία υπολογισμού τους καθώς η υλοποίηση είναι εξαιρετικά απλή). Έτσι, για να υπολογίσουμε τις μετρικές, αρκεί να προσδιορίσουμε ότι μια επιλογή οποία περιέχεται και στο Σύνολο Εκπαίδευσης και στο Σύνολο Ελέγχου. Τότε, είναι μια «σωστή» και επιτυχημένη Πρόταση για το Χρήστη.

Κώδικας

```
...  
for j=1:R  
    for k=1:N  
        if eventid_test_set(j) == topN_tainies(k)  
            hit=hit+1;  
            break;  
        end  
    end  
end  
  
% Metrics for Each User  
recall = hit/R;  
precision = hit/N ;  
recall_overall(i) = recall;  
precision_overall(i)= precision;  
...
```

Για τον ακριβή προσδιορισμό των μεθόδων, χρησιμοποιήσαμε την τεχνική της 5-πλης Διασταυρωμένης Επικύρωσης (5-Fold Cross Validation). Η υλοποίηση της μεθόδου γίνεται απλά υπολογίζοντας τον μέσο όρο των 5 τιμών που υπολογίστηκαν.

Κώδικας

```
...  
  
cosine_recall = mean(cos_recall)  
cosine_precision = mean(cos_precision)  
  
...
```

Συνοπτικά, ο υπολογισμός π.χ. για το Μητρώο Ομοιότητας Συνημίτονου, γίνεται ως:

Κώδικας

```
...  
% Define Constants  
a = 0.85;  
errorTolerance = 10^-6;  
maxIterations = 200;  
  
COS1_RES = getResult(Pref1, COS1, a, errorTolerance,  
maxIterations);  
COS2_RES = getResult(Pref2, COS2, a, errorTolerance,  
maxIterations);  
COS3_RES = getResult(Pref3, COS3, a, errorTolerance,  
maxIterations);  
COS4_RES = getResult(Pref4, COS4, a, errorTolerance,  
maxIterations);  
COS5_RES = getResult(Pref5, COS5, a, errorTolerance,  
maxIterations);  
  
% Evaluating Method  
cos_recall = zeros(1, 5);  
cos_precision = zeros(1, 5);  
[cos_recall(1), cos_precision(1)] = calculateMetrics(  
TestSet1, TrainSet1, COS1_RES, 943, 4);  
[cos_recall(2), cos_precision(2)] = calculateMetrics(  
TestSet2, TrainSet2, COS2_RES, 943, 4);  
[cos_recall(3), cos_precision(3)] = calculateMetrics(  
TestSet3, TrainSet3, COS3_RES, 943, 4);  
[cos_recall(4), cos_precision(4)] = calculateMetrics(  
TestSet4, TrainSet4, COS4_RES, 943, 4);  
[cos_recall(5), cos_precision(5)] = calculateMetrics(  
TestSet5, TrainSet5, COS5_RES, 943, 4);  
  
cosine_recall = mean(cos_recall)  
cosine_precision = mean(cos_precision)  
  
...
```

Όπως παρατηρήσατε, στο παραπάνω τμήμα κώδικα, για κάθε ένα από τα πέντε (5) ζεύγη υποσυνόλων Εκπαίδευσης και Ελέγχου του Μητρώου Ομοιοτήτων Συνημίτονου, επαναλαμβάνεται η ίδια διαδικασία, όπως αυτή περιγράφεται παραπάνω. Τελικά, υπολογίζονται οι μετρικές απόδοσης για τα ζεύγη μητρώων και τελικά ο μέσος όρος αυτών δίνει την τελική απόδοση του συστήματος.

Όμοια, υλοποιείται η διαδικασία και για τα εναπομείναντα Μητρώα Ομοιοτήτων Προσαρμοσμένης Ομοιότητας Συνημίτονου και Συσχέτισης κατά Pearson.

Να σημειωθεί, ότι τα παραπάνω υλοποιούνται με χρήση συναρτήσεων και η διαδικασία γίνεται για κάθε Χρήστη που συμπεριλαμβάνεται στο Σύνολο Δεδομένων μας.

Όλα τα αρχεία που περιέχουν τον πηγαίο κώδικα υλοποιήθηκε συμπεριλαμβάνονται σε ηλεκτρονική μορφή σε δίσκο CD που δόθηκε μαζί με τη Διπλωματική Εργασία.

5. Συμπεράσματα

Στα πλαίσια της παρούσας Διπλωματικής Εργασίας σχεδιάστηκε και αναπτύχθηκε σύστημα που έχει σκοπό τη Στοχευμένη Πρόταση Προϊόντων σε Χρήστες Κοινωνικών Δικτύων. Τα Κοινωνικά Δίκτυα είναι διαδικτυακοί χώροι όπου οι Χρήστες ανταλλάσσουν προσωπικά δεδομένα και πληροφορίες, και σε συνδυασμό με την απήχηση που έχουν αποτελούν την καταλληλότερη πηγή για λήψη δεδομένων.

Έτσι, δημιουργήθηκε ένα Σύστημα Συστάσεων, το οποίο συλλέγει τις πληροφορίες των Χρηστών, ανώνυμα και σεβόμενο πάντα τα προσωπικά τους δεδομένα, και τελικά προτείνει σε αυτούς προϊόντα βασισμένο στα δεδομένα και τη γνώση που έχει για τον εκάστοτε Χρήστη και το σύνολο των Χρηστών γενικότερα.

Η Εφαρμογή που αναπτύξαμε είναι βασισμένη σε μοντέλα και δημοσιεύσεις για τα οποία έχει αποδειχθεί η αποδοτικότητά τους και οι δυνατότητες επίλυσης παρόμοιων προβλημάτων. Οπότε, η λύση που προτείνουμε επεκτείνει αυτές τις μεθόδους σε μια νέα διάσταση, αυτή των Κοινωνικών Δικτύων. Για τις ανάγκες της Διπλωματικής Εργασίας επιλέγει το Κοινωνικό Δίκτυο Facebook και η Πρόταση που εξάγει η Εφαρμογή μας, αφορά Εκδηλώσεις (π.χ. συναυλίες, ειδικές επιδείξεις σε μουσεία, βραδινές εξόδους κ.λπ.) οι οποίες ενδιαφέρουν το Χρήστη.

Η εργασία μπορεί να χωριστεί σε δύο (2) βασικά μέρη: τη Διεπαφή του Χρήστη με το Κοινωνικό Δίκτυο και την Υλοποίηση των Αλγορίθμων Συνεργατικής Διήθησης. Το πρώτο μέρος, αφορά τη γνωριμία και αλληλεπίδραση του Χρήστη με την Εφαρμογή και τη συλλογή πληροφοριών με την άδεια του. Το δεύτερο μέρος αφορά την υλοποίηση του Αλγορίθμου Συνεργατικής Διήθησης, οποίος εξάγει το τελικό αποτέλεσμα, δηλαδή τις προσωποποιημένες και εξατομικευμένες Προτάσεις - Εκδηλώσεις που θα προτείνουμε στο Χρήστη. Για να βρούμε την καλύτερη δυνατή λύση στο πρόβλημά μας, υλοποιήθηκαν τρεις (3) διαφορετικές υλοποιήσεις (Μητρώα Ομοιοτήτων) και συγκρίθηκαν οι αποδόσεις σε κάθε περίπτωση.

Συνοπτικά η απόδοση κάθε Μητρώου Ομοιοτήτων περιέχεται στον πίνακα:

	Ομοιότητα Συνημίτονου	Προσαρμοσμένη Ομοιότητα Συνημίτονου	Συσχέτιση κατά Pearson
Ακρίβεια	$5,5591 \cdot 10^{-4}$	$4,4006 \cdot 10^{-4}$	$6,3312 \cdot 10^{-4}$
Ανάκληση	0,2108	0,1676	0,2561

Συγκεντρωτικός Πίνακας Αποδόσεων

Από τον παραπάνω πίνακα συμπεραίνουμε ότι προτιμότερο για το πρόβλημα μας είναι να χρησιμοποιήσουμε Μητρώο Ομοιοτήτων υπολογισμένο σύμφωνα με τη Συσχέτιση κατά Pearson. Σε αυτό το σημείο να τονίσουμε ότι, η μέτρηση της απόδοσης της κάθε μεθόδου θέλουμε να έχει τη μεγαλύτερη δυνατή τιμή. Αν και σε απόλυτο μέγεθος οι τιμές εμφανίζονται «μικρές» αυτό δεν επηρεάζει την τελική μας απόφαση. Επίσης, να θυμίσουμε ότι, και οι τρεις (3) μέθοδοι θα επιλύσουν το πρόβλημά μας, με τη Συσχέτιση κατά Pearson όμως, να εμφανίζει τα καλύτερα αποτελέσματα.

Με την ολοκλήρωση της παρούσας Διπλωματικής Εργασίας ανοίγονται νέοι δρόμοι, τόσο σε ερευνητικό όσο και σε επιστημονικό επίπεδο, για το συνδυασμό των Κοινωνικών Δικτύων με τα Συστήματα Συστάσεων.

6. Επεκτάσεις και Μελλοντικές Εφαρμογές

6.1 Επεκτάσεις

Η εφαρμογή που έχουμε υλοποιήσει μπορεί να εξάγει Προτάσεις, εξατομικευμένες για κάθε Χρήστη, για Εκδηλώσεις τις οποίες θα ενδιαφέρεται (σύμφωνα με τα αποτελέσματα που προκύπτουν) να παρακολουθήσει ο Χρήστης.

Η εφαρμογή στη μορφή που είναι τώρα εξυπηρετεί καθαρά ακαδημαϊκούς σκοπούς, αλλά πολύ εύκολα μπορεί να επεκταθεί ώστε να λειτουργεί ως ένα αυτόνομο Σύστημα Σύστασης ή να ενσωματωθεί σε ένα άλλο εξωτερικό σύστημα που έχει στόχο την πρόταση προϊόντων και να παρέχει πρόσθετα δεδομένα και γνώση στο εξωτερικό σύστημα. Γενικότερα η Πρόταση Προϊόντων είναι μια σύνθετη διαδικασία και τα περισσότερα δεδομένα οδηγούν σε «καλύτερα» αποτελέσματα, εφόσον γίνει σωστή επιλογή και επεξεργασία αυτών. Να θυμίσουμε ότι «καλύτερα» θεωρούμε τα αποτελέσματα όπου η πρόταση (ή προτάσεις) που εξάγεται (εξάγονται) όντως συμφωνούν με τις προτιμήσεις του Χρήστη.

Η εφαρμογή μας μπορεί να επεκταθεί σε ένα αυτόνομο σύστημα, με την έννοια ότι θα μπορεί να:

- Ενημερώνεται. Μπορεί να σχεδιαστεί κατάλληλα βοηθητικό σύστημα που θα αναλάβει την περιοδική ανανέωση-ενημέρωση των στοιχείων των Χρηστών στην Βάση Δεδομένων, εξασφαλίζοντας ότι το σύστημα θα είναι ενημερωμένο με τις τελευταίες προτιμήσεις του Χρήστη και τελικά εξάγοντας τα καλύτερα δυνατά αποτελέσματα. Αυτό το βοήθημα θα μπορεί να λειτουργεί ασύγχρονα, π.χ. μία (1) φορά ανά εβδομάδα.
- Υπολογίζει και εξάγει αποτελέσματα. Όπως έχουμε αναφέρει και παραπάνω στο σύγγραμμα η τελική πρόταση είναι μια διαδικασία η οποία είναι χρονοβόρα και απαιτεί μεγάλη υπολογιστική ισχύ. Ιδανικά το σύστημα θα μπορεί να προχωράει στο επόμενο (κάθε φορά) βήμα ενώ θα χρησιμοποιεί τους ελάχιστους πόρους.

Εδώ να σημειώσουμε ότι οι προαναφερθείσες διαδικασίες θα γίνονται αυτόματα, όπως επίσης θα γίνονται αυτόματα και οι επιμέρους δημιουργίες αρχείων και μεταπτώσεις δεδομένων που είναι απαραίτητες για την επικοινωνία των επιμέρους υπο-συστημάτων.

Το τελικό αποτέλεσμα θα είναι ένα πλήρες Σύστημα Σύστασης που θα λαμβάνει δεδομένα από Χρήστες Κοινωνικών Δικτύων, θα ενημερώνεται αυτόματα και θα εξάγει τις βέλτιστες Προτάσεις για το Χρήστη χωρίς την παραμικρή ενέργεια ή επιτήρηση από τον Διαχειριστή της εφαρμογής.

6.2 Μελλοντικές Εφαρμογές

Το Πρόβλημα της Σύστασης αναφέρεται στην καλύτερη δυνατή πρόταση οποιουδήποτε ενός προϊόντος (υλικού και μη) ή υπηρεσίας στο Χρήστη με κύριο στόχο ο Χρήστης να ενδιαφέρεται για το προϊόν που προτάθηκε.

Παρατηρούμε και καταλαβαίνουμε πολύ εύκολα τη χρησιμότητα των συγκεκριμένων συστημάτων σε υπηρεσίες όπως το *Ηλεκτρονικό Εμπόριο*, τη *Διαφήμιση* και το *Marketing*. Με τους συνεχώς αυξανόμενους ρυθμούς των Κοινωνικών Δικτύων οι παραπάνω τομείς μπορούν να βοηθηθούν πολύ από τη χρήση ενός συστήματος, όπως αυτό που αναπτύχτηκε.

Επομένως, μπορούμε να θεωρήσουμε ότι το σύστημά μας μπορεί πολύ εύκολα να χρησιμοποιηθεί ως ένα *Πρόσθετο (Plugin)* σε ένα Ιστότοπο που προτείνει *Εξόδους*, όπως π.χ. ένας *ηλεκτρονικός οδηγός πόλης*. Με βάση τα στοιχεία από το Κοινωνικό Δίκτυο, ο οδηγός θα μπορεί να προτείνει την κατάλληλη, για παράδειγμα, *βραδινή έξοδο* για το Χρήστη με βάση τα στοιχεία του και τις προηγούμενες προτιμήσεις του. Η εφαρμογή που μόλις αναφέραμε μπορεί να υλοποιηθεί πολύ εύκολα ήδη με το υπάρχον σύστημα που αναπτύξαμε. Πόσο μάλλον, αν ολοκληρωθεί η επέκταση της που περιγράψαμε στην παραπάνω παράγραφο (*Παράγραφος 6.1 Επεκτάσεις*) μπορούμε να μιλάμε για μια ολοκληρωμένη πρόταση η οποία θα μπορεί να αξιοποιηθεί άμεσα από επιχειρήσεις που ασχολούνται στον τομέα των προτάσεων και διαφήμισης μέσω διαδικτύου.

Όταν αναφερόμαστε στα Συστήματα Συστάσεων αυτά μπορούν να εξάγουν αποτέλεσμα για κάθε είδος προϊόντος ή υπηρεσίας. Για τις ανάγκες υλοποίησης της παρούσας Διπλωματικής Εργασίας, και μόνο, προτιμήθηκε η πρόταση μιας *Εκδήλωσης*, καθώς έχει ιδιαίτερη απήχηση στο ευρύ κοινό. Εύκολα και απλά, μπορεί να αλλάξει το αντικείμενο του αποτελέσματος σε ένα *εισιτήριο για τον κινηματογράφο*, ένα *προϊόν ενός ηλεκτρονικού καταστήματος* ή το επόμενο *βιβλίο που θα δανειστούμε από τη Δημοτική Βιβλιοθήκη*. Οι αλλαγές που πρέπει να γίνουν για να καλυφθούν αυτά τα προβλήματα είναι ελάχιστες και αφορούν κυρίως τους «σταθερούς» όρους. Εδώ να επισημάνουμε ότι είναι αναγκαίες αυτές οι αλλαγές που μόλις αναφέραμε καθώς κάθε πρόβλημα, άρα και προϊόν, αντιμετωπίζεται διαφορετικά από τους Χρήστες και έτσι πρέπει να γίνουν όλα τα βήματα της διαδικασίας της ανάλυσης για να πετύχουμε τα βέλτιστα αποτελέσματα.

Τέλος, η εφαρμογή μας, ως υποσύστημα ενός μεγαλύτερου συστήματος, θα μπορεί να υποστηρίξει την έκδοση αποτελεσμάτων δίνοντας πιο σύνθετες και επιτυχείς προτάσεις. Για παράδειγμα, μπορεί να χρησιμοποιηθεί παράλληλα με ένα υπάρχον *ηλεκτρονικό κατάστημα (e-shop)* για να οδηγήσει σε πιο εύστοχες προτάσεις για το *κινητό τηλέφωνο* που θα αγοράσετε μέχρι και τις *καλοκαιρινές σας διακοπές*.

Γενικότερα, οι εφαρμογές είναι κυριολεκτικά χιλιάδες, ανάλογα κάθε φορά με το προϊόν ή τον τομέα που απευθυνόμαστε. Τα Κοινωνικά Δίκτυα είναι μια εξαιρετική πηγή πληροφοριών καθώς αντικατοπτρίζουν πλήρως τους Χρήστες και έτσι μπορούμε να εξάγουμε ασφαλή συμπεράσματα που θα οδηγήσουν σε επιτυχημένες προβλέψεις.

Βιβλιογραφία

1. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, 'Item-Based Collaborative Filtering Recommendation Algorithms', WWW10, May 1-5, 2001
2. Jianming He and Wesley W. Chu, 'A Social Network-Based Recommender System (SNRS)'
3. Ido Guy and David Carmel, 'Social Recommender Systems', WWW 2011, March 28–April 1, 2011
4. Satnam Alag 2009, 'Collective Intelligence in Action', MANNING
5. Matthew A. Russell 2011, 'Mining the Social Web', O'Reilly Media
6. Charu C. Aggarwal 2011, 'Social Network Data Analytics', Springer
7. Toby Segaran 2007, 'Programming Collective Intelligence', O'Reilly Media
8. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, 'Evaluating Collaborative Filtering Recommender Systems', ACM
9. Bonhard P. and Sasse M.A., "Knowing me, knowing you" – using profiles and social networking to improve recommender systems.', BT Technology Journal, July 2006
10. He J., Chu W.W. and Liu Z, 'Inferring privacy information from social networks.',
11. IEEE Intelligence and Security Informatics Conference (ISI 2006), May 2006
12. Cong Li, Li Ma and Ke Dong, 'Collaborative Filtering Cold-Start Recommendation Based on Dynamic Browsing Tree Model in E-commerce', International Conference on Web Information Systems and Mining, 2009
13. Ilham Esslimani, Armelle Brun and Anne Boyer, 'From Social Networks to Behavioral Networks in Recommender Systems', Advances in Social Network Analysis and Mining, 2009
14. Robin Burke, 'Integrating Knowledge-based and Collaborative-filtering Recommender Systems', Workshop on AI and Electronic Commerce, AAAI 1999
15. J. Ben Schafer, Dan Frankowski, Jon Herlocker and Shilad Sen, 'Collaborative Filtering Recommender Systems', The Adaptive Web, Springer 2007
16. Georg Groh and Christian Ehmig, 'Recommendations in Taste Related Domains: Collaborative Filtering vs. Social Filtering', ACM
17. Prem Melville, Raymond J. Mooney and Ramadass Nagarajan, 'Content-Boosted Collaborative Filtering for Improved Recommendations', American Association for Artificial Intelligence, 2002

18. Greg Linden, Brent Smith, and Jeremy York, 'Amazon.com Recommendations: Item-to-Item Collaborative Filtering', IEEE Computer Society, JANUARY • FEBRUARY 2003
19. Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, 'Collaborative Filtering Recommender Systems', Now the essence of knowledge, 2011
20. Yehuda Koren, 'The BellKor Solution to the Netflix Grand Prize', August 2009
21. Andreas Tröbscher and Michael Jahrer, 'The BigChaos Solution to the Netflix Grand Prize', September 5, 2009
22. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, 'Evaluating Collaborative Filtering Recommender Systems', ACM Transactions on Information Systems, January 2004
23. Henry Kautz, Bart Selman and Mehul Shah, 'Combining Social Networks and Collaborative Filtering', COMMUNICATIONS OF THE ACM, March 1997
24. Wei Chen and Simon Fong, 'Social Network Collaborative Filtering Framework and Online Trust Factors: a Case Study on Facebook'
25. Georgios Lappas, 'From Web Mining to Social Multimedia Mining', International Conference on Advances in Social Networks Analysis and Mining, 2011
26. Ziqi Wang, Ming Zhang, Yuwei Tan, Wenqing Wang, Yuexiang Zhang and Ling Chen, 'Recommendation Algorithm Based on Graph-Model Considering User Background Information', Ninth International Conference on Creating, Connecting and Collaborating through Computing, 2011
27. Claudia Canali, Sara Casolari and Riccardo Lancellotti, 'A quantitative methodology to identify relevant users in social networks', IEEE, 2010
28. Ming Chen, 'Research on Recommender Technology in E-commerce Recommendation System', 2nd International Conference on Education Technology and Computer (ICETC), 2010
29. Χ.Μακρής, 'Ανάκτηση Πληροφορίας Πανεπιστημιακές Σημειώσεις'
30. Κουνέλη Μαριάννα, 'Ανάπτυξη συστήματος συστάσεων συνεργατικής διήθησης με χρήση ιεραρχικών αλγορίθμων κατάταξης' 2012
31. Michael J. Pazzani, 'A Framework for Collaborative, Content-Based and Demographic Filtering', Artificial Intelligence Review, 1999
32. Πληροφορίες για τη γλώσσα προγραμματισμού PHP, <https://en.wikipedia.org/wiki/PHP>
33. Πληροφορίες για τα Κοινωνικά Δίκτυα, http://en.wikipedia.org/wiki/Social_network
34. Πληροφορίες για τη γλώσσα προγραμματισμού HTML, <http://en.wikipedia.org/wiki/HTML>
35. Πληροφορίες για τη γλώσσα μορφοποίησης CSS, https://en.wikipedia.org/wiki/Cascading_Style_Sheets

36. Πληροφορίες για τις σχετικά με τις βάσεις δεδομένων MySQL, <http://en.wikipedia.org/wiki/MySQL>
37. Πληροφορίες για τη γλώσσα προγραμματισμού JavaScript, <http://en.wikipedia.org/wiki/JavaScript>
38. Πληροφορίες για τη βιβλιοθήκη jQuery, <http://en.wikipedia.org/wiki/JQuery>
39. Πληροφορίες για το περιβάλλον MATLAB, <https://en.wikipedia.org/wiki/MATLAB>
40. Πληροφορίες για τη Συνεργατική Διήθηση, http://en.wikipedia.org/wiki/Collaborative_filtering
41. Πληροφορίες και περιγραφή της επαναληπτικής μεθόδου Power, http://en.wikipedia.org/wiki/Power_iteration
42. Αρχική σελίδα για τους προγραμματιστές από το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/>
43. Περιγραφή δεδομένου Εκδήλωση στο Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/docs/reference/api/event/>
44. Ηλεκτρονικός βοηθός περιγραφής και επίλυσης σφαλμάτων που σχετίζονται με το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/tools/debug/>
45. Ηλεκτρονικός προσομοιωτής ερωτημάτων που σχετίζονται με το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/tools/explorer/>
46. Ιστότοπος συνεδρίων σχετικά με τα Συστήματα Σύστησης από το ν οργανισμό ACM, <http://recsys.acm.org/>
47. Πληροφορίες για τη χρήση και υλοποίηση της λήψης δεδομένων από το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/docs/reference/php/facebook-setAccessToken/>
48. Πληροφορίες για τη χρήση και υλοποίηση της λήψης δεδομένων από το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/docs/reference/php/facebook-getAccessToken/>
49. Πληροφορίες για τη χρήση και υλοποίηση της λήψης δεδομένων από το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/docs/reference/login/>
50. Ηλεκτρονικός οδηγός για τη χρήση και ορισμό μιας νέας εφαρμογής στο Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/docs/guides/appcenter/>
51. Πληροφορίες για τη χρήση και υλοποίηση ερωτημάτων με σκοπό τη λήψη δεδομένων από το Κοινωνικό Δίκτυο Facebook, <https://developers.facebook.com/docs/reference/api/batch/>

52. Η κοινότητα «Stack Overflow» παρέχει ένα μέρος για ερωτήσεις και απαντήσεις σε πολλά προβλήματα που μπορεί να συναντήσει ένας προγραμματιστής, <http://stackoverflow.com/>
53. Παραδείγματα για την μέθοδο Power, http://people.sc.fsu.edu/~jburkardt/m_src/power_method/power_method.html
54. Παράδειγμα εφαρμογής Κοινωνικού Δικτύου Facebook, <https://developers.facebook.com/docs/samples/meals-with-friends/>