



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Έξυπνο εξατομικευμένο λογισμικό ηλεκτρονικού βιβλιοπωλείου Intelligent personalized e-bookstore solution
Όνοματεπώνυμο Φοιτητή	Ιάκωβος Λαμπαδάκης
Πατρώνυμο	Σταμάτιος
Αριθμός Μητρώου	ΜΠΠΛ/17027
Επιβλέπων	Ευάγγελος Σακκόπουλος Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Νοέμβριος 2021**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευάγγελος Σακκόπουλος

Ευθύμιος Αλέπης

Διονύσιος Σωτηρόπουλος

Επίκουρος Καθηγητής

Αναπληρωτής Καθηγητής

Επίκουρος Καθηγητής

Περίληψη

Σκοπός της εργασίας είναι η κατασκευή ηλεκτρονικού βιβλιοπωλείου και η μελέτη αλγορίθμων για τη δημιουργία συστήματος που θα παρέχει στους χρήστες εξατομικευμένες προτάσεις για αγορά βιβλίων.

Abstract

The purpose of this paper is to build an online bookstore and study algorithms for creating a recommendation system that will provide to users personalized suggestions.

Πίνακας περιεχομένων

ΠΕΡΙΛΗΨΗ	3
ABSTRACT	3
1. ΕΙΣΑΓΩΓΗ	5
2. ΕΙΣΑΓΩΓΗ ΣΤΗ ΣΥΛΛΟΓΙΚΗ ΝΟΗΜΟΣΥΝΗ (COLLECTIVE INTELLIGENCE)	5
2.1 ΣΥΛΛΟΓΙΚΗ ΝΟΗΜΟΣΥΝΗ (COLLECTIVE INTELLIGENCE)	5
2.2 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ	7
2.2.1 ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	7
3. ΥΠΟΒΟΛΗ ΣΥΣΤΑΣΕΩΝ ΠΡΟΪΟΝΤΩΝ (MAKING RECOMMENDATIONS)	8
3.1 COLLABORATIVE FILTERING	8
3.2 ΕΥΡΕΣΗ ΠΑΡΟΜΟΙΩΝ ΧΡΗΣΤΩΝ (FINDING SIMILAR USERS)	10
3.3 ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ	10
3.4 ΣΥΣΧΕΤΙΣΗ PEARSON	14
3.5 ΚΑΤΑΤΑΞΗ ΤΩΝ ΚΡΙΤΙΚΩΝ	15
3.6 ΣΥΣΤΑΣΗ ΠΡΟΪΟΝΤΩΝ (RECOMMENDING ITEMS)	16
3.7 ΠΡΟΪΟΝΤΑ ΠΑΡΟΜΟΙΑ ΜΕΤΑΞΥ ΤΟΥΣ (MATCHING PRODUCTS)	19
4. ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΜΕ UML ΒΑΣΙΣΜΕΝΗ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ RUP	23
4.2 ΣΥΛΛΗΨΗ ΑΠΑΙΤΗΣΕΩΝ	23
4.3 ΑΝΑΛΥΣΗ-ΣΧΕΔΙΑΣΜΟΣ	24
ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ	24
ΔΙΑΓΡΑΜΜΑ ΤΑΞΕΩΝ	25
ΔΙΑΓΡΑΜΜΑ ΑΝΤΙΚΕΙΜΕΝΩΝ	25
ΔΙΑΓΡΑΜΜΑ ΣΥΝΕΡΓΑΣΙΑΣ	26
ΔΙΑΓΡΑΜΜΑ ΣΕΙΡΑΣ	27
ΔΙΑΓΡΑΜΜΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ	28
ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΣΤΑΣΕΩΝ	28
ΔΙΑΓΡΑΜΜΑ ΕΞΑΡΤΗΜΑΤΩΝ	30
ΔΙΑΓΡΑΜΜΑ ΔΙΑΝΟΜΗΣ	30
ΗΛΕΚΤΡΟΝΙΚΟ ΒΙΒΛΙΟΠΩΛΕΙΟ	31
4.4 ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ	34
4.5 ΣΥΜΠΕΡΑΣΜΑΤΑ	34
ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ	35

1.Εισαγωγή

Σκοπός της εργασίας είναι η κατασκευή ηλεκτρονικού βιβλιοπωλείου και η μελέτη αλγορίθμων για τη δημιουργία συστήματος που θα παρέχει στους χρήστες εξατομικευμένες προτάσεις για αγορά βιβλίων. Θα αναλυθούν έννοιες όπως το Collective Intelligence (Συλλογική Ευφυΐα), η Μηχανική Μάθηση (Machine learning) και το Collaborative Filtering.

2.Εισαγωγή στη Συλλογική Νοημοσύνη (Collective Intelligence)

Η συλλογική νοημοσύνη εφαρμόζεται σε διαφορετικά πεδία και μπορεί να χρησιμοποιηθεί για την εξαγωγή νέων συμπερασμάτων από δεδομένα (data) που συλλέγονται από πολλούς ανθρώπους. Το Netflix είναι μια πολυεθνική εταιρεία η οποία είναι ειδικευμένη στη διαδικτυακή ενοικίαση ταινιών, είτε DVD είτε Blu-Ray, καθώς και στην παροχή video streaming και επιτρέπει στους χρήστες να επιλέγουν ταινίες και κάνει προτάσεις βασισμένο στις ταινίες που οι πελάτες είχαν νοικιάσει στο παρελθόν. Στα τέλη του 2006 ανακοίνωσε ένα βραβείο στο πρώτο άτομο που θα βελτίωνε την ακρίβεια του συστήματός κατά 10%. Χιλιάδες ομάδες από όλο τον κόσμο συμμετείχαν στο διαγωνισμό. Τον Απρίλιο του 2007, η κορυφαία ομάδα κατάφερε να σημειώσει βελτίωση 7%. Χρησιμοποιώντας δεδομένα σχετικά με τις ταινίες που απολάμβανε κάθε πελάτης, το Netflix είναι σε θέση να προτείνει ταινίες σε άλλους πελάτες που μπορεί μέχρι τότε να μην τις γνώριζαν. Οποιοσδήποτε τρόπος για τη βελτίωση του συστήματος προτάσεων ταινιών αξίζει πολλά χρήματα. Η μηχανή αναζήτησης της Google ξεκίνησε το 1998, σε μια εποχή που υπήρχαν ήδη μεγάλες μηχανές αναζήτησης και πολλοί υπέθεσαν ότι μια ακόμη μηχανή αναζήτησης δε θα ήταν δυνατό να ανταγωνιστεί τις υπόλοιπες. Οι ιδρυτές της Google, ωστόσο, υιοθέτησαν μια εντελώς νέα προσέγγιση για την κατάταξη των αποτελεσμάτων αναζήτησης χρησιμοποιώντας τους συνδέσμους εκατομμυρίων ιστοτόπων να αποφασίσουν ποιες σελίδες ήταν πιο σχετικές. Τα αποτελέσματα αναζήτησης της Google ήταν πολύ καλύτερα από εκείνα των υπολοίπων που μέχρι το 2004 διέθεταν το 85% των αναζητήσεων στον παγκόσμιο Ιστό. Οι ιδρυτές της συγκαταλέγονται πλέον στους 10 πλουσιότερους ανθρώπους στον κόσμο. Τι κοινό έχουν αυτές οι δύο εταιρείες; Και οι δύο κατέληξαν σε νέα συμπεράσματα και δημιούργησαν νέες επιχειρηματικές ευκαιρίες χρησιμοποιώντας εξελιγμένους αλγόριθμους για το συνδυασμό δεδομένων που συλλέχθηκαν από πολλά διαφορετικά άτομα. Η ικανότητα συλλογής πληροφοριών και η υπολογιστική δύναμη για την ερμηνεία τους επέτρεψε μεγάλες ευκαιρίες συνεργασίας και καλύτερη κατανόηση των προτιμήσεων των χρηστών και των πελατών. Σχεδόν όλοι θέλουν να κατανοήσουν καλύτερα τους πελάτες τους και να δημιουργούν εξατομικευμένες διαφημίσεις. Αυτά είναι μερικά μόνο παραδείγματα στο συναρπαστικό πεδίο της συλλογικής νοημοσύνης. Η κατανόηση της μηχανικής μάθησης και των στατιστικών μεθόδων θα γίνει ολοένα και πιο σημαντική σε ένα ευρύ φάσμα τομέων, αλλά ιδιαίτερα στην ερμηνεία και την οργάνωση του τεράστιου όγκου πληροφοριών που δημιουργούνται από ανθρώπους σε όλο τον κόσμο [2].

2.1 Συλλογική Νοημοσύνη (Collective Intelligence)

Στο βιβλίο “Programming Collective Intelligence: Building Smart Web 2.0 Applications”, [2] ο Toby Segaran αναλύει την έννοια της συλλογικής νοημοσύνης (Collective Intelligence). Η φράση collective Intelligence χρησιμοποιείται για δεκαετίες και έχει γίνει όλο και πιο δημοφιλής και πιο σημαντική με την έλευση νέων τεχνολογιών. Οι τεχνολόγοι χρησιμοποιώντας αυτήν τη φράση συνήθως εννοούν το συνδυασμό συμπεριφοράς, προτιμήσεων ή ιδεών μιας ομάδας ανθρώπων με σκοπό να δημιουργήσουν νέες ιδέες.

Φυσικά, η συλλογική νοημοσύνη προϋπήρχε του Διαδικτύου. Πριν τον παγκόσμιο ιστό μπορούσαμε επίσης να συλλέξουμε δεδομένα από διαφορετικές ομάδες ατόμων, να τα συνδυάσουμε και να τα αναλύσουμε. Αυτό γινόταν με μια έρευνα ή απογραφή. Η συλλογή απαντήσεων από μια μεγάλη ομάδα ατόμων μάς επιτρέπει να εξαγάγουμε στατιστικά συμπεράσματα σχετικά με την ομάδα που κανένα μεμονωμένο μέλος δεν θα γνώριζε μόνο του. Η οικοδόμηση νέων συμπερασμάτων από διαφορετικές ομάδες ατόμων είναι στην πραγματικότητα η συλλογική νοημοσύνη (Collective Intelligence). Ένα πολύ γνωστό παράδειγμα που βρίσκει εφαρμογή είναι στο financial market, όπου η τιμή δεν καθορίζεται από ένα άτομο, αλλά από την εμπορική συμπεριφορά πολλών ανεξάρτητων ανθρώπων που όλοι ενεργούν σε αυτό που πιστεύουν ότι είναι για το δικό τους συμφέρον. Για παράδειγμα, με τον όρο future market, στο οποίο πολλοί ενδιαφερόμενοι επενδύουν με βάση τις πεποιθήσεις τους σχετικά με τις μελλοντικές τιμές των αγορών τους, θεωρούνται καλύτερες για την πρόβλεψη των τιμών, συγκριτικά με τους ειδικούς που κάνουν ανεξάρτητα προβλέψεις. Αυτό συμβαίνει επειδή οι αγορές συνδυάζουν τη γνώση, την εμπειρία και τη διορατικότητα χιλιάδων ανθρώπων για να δημιουργήσουν μια πρόβλεψη αντί να βασίζονται στην οπτική ενός ατόμου. Αν και υπήρχαν μέθοδοι συλλογικής νοημοσύνης πριν από το διαδίκτυο, η ικανότητα συλλογής πληροφοριών από χιλιάδες ή και εκατομμύρια ανθρώπους στον Ιστό έχει ανοίξει πολλές νέες δυνατότητες. Ανά πάσα στιγμή, οι άνθρωποι χρησιμοποιούν το διαδίκτυο για αγορές, έρευνα, αναζήτηση ψυχαγωγίας και δημιουργία δικών τους ιστοτόπων. Όλη αυτή η συμπεριφορά μπορεί να χρησιμοποιηθεί για την εξαγωγή πληροφοριών χωρίς να χρειάζεται να απασχολείται ο χρήστης, θέτοντας του ερωτήσεις. Υπάρχουν πολλοί τρόποι επεξεργασίας και ερμηνείας αυτών των πληροφοριών. Ακολουθούν μερικά βασικά παραδείγματα που δείχνουν αυτές τις προσεγγίσεις: Η Wikipedia είναι μια διαδικτυακή εγκυκλοπαίδεια που δημιουργήθηκε εξ ολοκλήρου από τις συνεισφορές των χρηστών. Οποιαδήποτε σελίδα, μπορεί να δημιουργηθεί ή να επεξεργαστεί από οποιονδήποτε καθώς υπάρχει ένας μικρός αριθμός διαχειριστών που παρακολουθούν τυχόν καταχρήσεις. Η Wikipedia έχει περισσότερες καταχωρήσεις από οποιαδήποτε άλλη εγκυκλοπαίδεια και παρά την κάποια παραβίαση από κακόβουλους χρήστες, γενικά θεωρείται ακριβής στα περισσότερα θέματα. Αυτό είναι ένα παράδειγμα συλλογικής νοημοσύνης, διότι κάθε άρθρο διατηρείται από μια μεγάλη ομάδα ανθρώπων και το αποτέλεσμα είναι μια εγκυκλοπαίδεια πολύ μεγαλύτερη από ό,τι μπόρεσε να δημιουργήσει κάθε συντονισμένη ομάδα. Το λογισμικό της Wikipedia δεν κάνει τίποτα ιδιαίτερα έξυπνο με τις συνεισφορές των χρηστών απλώς παρακολουθεί τις αλλαγές και εμφανίζει την τελευταία έκδοση. Το Google είναι η πιο δημοφιλής μηχανή αναζήτησης διαδικτύου στον κόσμο και ήταν η πρώτη μηχανή αναζήτησης που αξιολογεί ιστοσελίδες με βάση τον αριθμό άλλων σελίδων που συνδέονται με αυτές. Αυτή η μέθοδος αξιολόγησης λαμβάνει πληροφορίες σχετικά με όσα έχουν πει χιλιάδες άνθρωποι για μια συγκεκριμένη ιστοσελίδα και χρησιμοποιεί αυτές τις πληροφορίες για να ταξινομήσει τα αποτελέσματα σε μια αναζήτηση. Αυτό είναι ένα πολύ διαφορετικό παράδειγμα collective Intelligence. Όταν η Wikipedia προσκαλεί ρητά τους χρήστες του ιστότοπου να συνεισφέρουν, η Google εξάγει τις σημαντικές πληροφορίες από το τι κάνουν οι δημιουργοί περιεχομένου ιστού στους δικούς τους ιστότοπους και τις χρησιμοποιεί για να δημιουργήσει βαθμολογίες για τους χρήστες της. Ενώ η Wikipedia είναι ένας μεγάλος πόρος και ένα εντυπωσιακό παράδειγμα συλλογικής ευφυΐας, οφείλει την ύπαρξή της πολύ περισσότερο στη βάση χρηστών που συνεισφέρει πληροφορίες από ό,τι σε έξυπνους αλγόριθμους στο λογισμικό. Ορισμένα δεδομένα συλλέγονται για παράδειγμα ζητώντας από τους ανθρώπους να αξιολογήσουν, και μερικά συλλέγονται απλώς παρακολουθώντας τι αγοράζουν οι άνθρωποι. Και στις δύο περιπτώσεις το σημαντικό είναι όχι μόνο η συλλογή και προβολή των πληροφοριών αλλά η επεξεργασία τους με έξυπνο τρόπο και η δημιουργία νέων πληροφοριών.

2.2 Μηχανική Μάθηση

Μηχανική μάθηση είναι πεδίο της επιστήμης των υπολογιστών [4] που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη [4]. Το 1959, ο Άρθουρ Σάμουελ ορίζει τη μηχανική μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί". Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να «μαθαίνουν» από τα δεδομένα [6] και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις ως αποτέλεσμα. Η μηχανική μάθηση είναι στενά συνδεδεμένη και συχνά συγχέεται με υπολογιστική στατιστική, ένας κλάδος που επίσης επικεντρώνεται στην πρόβλεψη μέσω της χρήσης των υπολογιστών. Έχει ισχυρούς δεσμούς με το μαθηματικό προγραμματισμό, ο οποίος παρέχει μεθόδους, τη θεωρία και τομείς εφαρμογής. Η Μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός, όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ανέφικτος. Παραδείγματα εφαρμογών αποτελούν, τα φίλτρα spam (spam filtering), η οπτική αναγνώριση χαρακτήρων (OCR),[5] οι μηχανές αναζήτησης και η υπολογιστική όραση. Η Μηχανική μάθηση μερικές φορές συγχέεται με την εξόρυξη δεδομένων, όπου η τελευταία επικεντρώνεται περισσότερο στην εξερευνητική ανάλυση των δεδομένων, γνωστή και ως μη επιτηρούμενη μάθηση. Στο πεδίο της ανάλυσης δεδομένων η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόηση πολύπλοκων μοντέλων και αλγορίθμων που οδηγούν στην πρόβλεψη. Τα αναλυτικά μοντέλα επιτρέπουν στους ερευνητές, στους επιστήμονες δεδομένων, στους μηχανικούς και στους αναλυτές να παράγουν αξιόπιστες αποφάσεις και αποτελέσματα.

2.2.1 Παραδείγματα Μηχανικής Μάθησης

Όπως αναφέρει στο βιβλίο του "Programming Collective Intelligence: Building Smart Web 2.0 Applications", [2] ο Toby Segaran υπάρχουν πολλοί ιστότοποι στο διαδίκτυο που συλλέγουν δεδομένα από πολλά διαφορετικά άτομα και χρησιμοποιούν μηχανική μάθηση και στατιστικές μεθόδους για να επωφεληθούν από αυτά. Η Google όχι μόνο χρησιμοποιεί συνδέσμους ιστού για την κατάταξη των σελίδων αλλά συλλέγει συνεχώς πληροφορίες σχετικά με τον αριθμό των επισκέψεων μοναδικών χρηστών στις διαφημίσεις, γεγονός που της επιτρέπει να στοχεύει αποτελεσματικότερα στη διαφήμιση. Άλλα παραδείγματα περιλαμβάνουν ιστότοπους με συστήματα προτάσεων. Ιστότοποι όπως η Amazon και το Netflix χρησιμοποιούν πληροφορίες σχετικά με πράγματα που αγοράζουν ή ενοικιάζουν οι άνθρωποι για να προσδιορίσουν ποια άτομα ή αντικείμενα είναι παρόμοια μεταξύ τους και στη συνέχεια, κάνουν προτάσεις με βάση το ιστορικό των αγορών. Άλλοι ιστότοποι, όπως το Pandora και το Last.fm χρησιμοποιούν τις αξιολογήσεις των χρηστών για διαφορετικές μπάντες και τραγούδια για να δημιουργήσουν προσαρμοσμένους στις μουσικές προτιμήσεις των χρηστών ραδιοφωνικούς σταθμούς. Οι προβλέψεις αγορών (Prediction Markets) είναι επίσης μια μορφή συλλογικής νοημοσύνης.

Product marketing

Το product marketing είναι μια στρατηγική λειτουργία μάρκετινγκ που γεφυρώνει το χάσμα μεταξύ διαχείρισης προϊόντων και επικοινωνιών μάρκετινγκ. Ο πρωταρχικός ρόλος ενός διαχειριστή μάρκετινγκ προϊόντων είναι ο καθορισμός και το μέγεθος των αγορών στόχων και των προτάσεων αξίας.

Financial fraud detection

Οι εταιρείες πιστωτικών καρτών αναζητούν συνεχώς νέους τρόπους για να εντοπίσουν, εάν οι συναλλαγές είναι δόλιες. Για το σκοπό αυτό, έχουν χρησιμοποιήσει τεχνικές όπως, νευρωνικά δίκτυα και επαγωγική λογική, για να επαληθεύσουν τις συναλλαγές και να αποφύγουν οικονομικές απάτες. Οι εφαρμογές της Τεχνητής Νοημοσύνης βελτιώνουν την εμπειρία του χρήστη για το γεγονός, ότι οι διαφημίσεις τους είναι εξατομικευμένες.

Εξατομίκευση

Η εξατομίκευση λογισμικού είναι η ικανότητα να χρησιμοποιείς τα δεδομένα των πελατών (customer data) για να προσαρμόζεις το περιεχόμενο ενός ηλεκτρονικού καταστήματος στις προτιμήσεις του καταναλωτή, διευκολύνοντάς τον να βρει ευκολότερα το προϊόν που επιθυμεί αλλά και να του προσφέρει ολοκληρωμένες προτάσεις που άπτονται των προτιμήσεων και των συνήθειών του, οδηγώντας τον σε επαναλαμβανόμενες αγορές.

3. Υποβολή Συστάσεων Προϊόντων (Making Recommendations)

Ξεκινώντας την αναφορά σχετικά με τη «συλλογική νοημοσύνη», [2] θα μελετήσουμε τρόπους, όπου χρησιμοποιώντας δεδομένα με προτιμήσεις μιας ομάδας ανθρώπων, σε διάφορες κατηγορίες, όπως βιβλία, ταινίες, μουσική, μπορούμε να κάνουμε συστάσεις σε άλλα άτομα. Υπάρχουν πολλές εφαρμογές που χρησιμοποιούν αντίστοιχες μεθόδους, όπως η υποβολή προτάσεων προϊόντων για διαδικτυακές αγορές με βάση τα ενδιαφέροντα των χρηστών ή η παροχή βοήθειας στους χρήστες να βρουν μουσική και ταινίες. Σε αυτό το κεφάλαιο γίνεται αναφορά σε συστήματα προτάσεων, για την εύρεση ατόμων, που μοιράζονται τις ίδιες προτιμήσεις με τις δικές μας και η δημιουργία αυτόματων προτάσεων με βάση προϊόντα που αρέσουν στους άλλους. Πλέον συναντάμε όλο και πιο συχνά ιστότοπους για διαδικτυακές αγορές που διαθέτουν μηχανές προτάσεων, όπως η Amazon. Η Amazon παρακολουθεί τις αγοραστικές συνήθειες όλων των αγοραστών της και όταν ο χρήστης συνδέεται στον ιστότοπο, χρησιμοποιεί αυτές τις πληροφορίες για να προτείνει προϊόντα που μπορεί να τον ενδιαφέρουν. Η Amazon μπορεί ακόμη να προτείνει και ταινίες στο χρήστη που μπορεί να τον ενδιαφέρουν, ακόμα κι αν έχει αγοράσει μόνο βιβλία στο παρελθόν. Ορισμένα διαδικτυακά πρακτορεία εισιτηρίων συναυλιών θα δουν το ιστορικό των εκπομπών που έχει πραγματοποιήσει ο χρήστης πριν και θα τον ειδοποιήσουν για επερχόμενες εκπομπές που μπορεί να τον ενδιαφέρουν. Από αυτά τα παραδείγματα, μπορούμε να συμπεράνουμε ότι οι προτιμήσεις των χρηστών μπορούν να συλλεχθούν με πολλούς διαφορετικούς τρόπους.

3.1 Collaborative Filtering

Το Collaborative Filtering [2] είναι μια τεχνική που χρησιμοποιείται από συστήματα προτάσεων. Είναι μια μέθοδος αυτόματης πρόβλεψης (φιλτράρισμα) για τα ενδιαφέροντα ενός χρήστη, συλλέγοντας προτιμήσεις ή πληροφορίες από πολλούς χρήστες (συνεργαζόμενοι). Ο όρος Collaborative Filtering χρησιμοποιήθηκε για πρώτη φορά από τον David Goldberg στο Xerox PARC το 1992 σε μια εφημερίδα με τίτλο «Using collaborative filtering to weave an information tapestry» Σχεδίασε ένα σύστημα που ονομάζεται Tapestry που επιτρέπει στους χρήστες να σχολιάζουν έγγραφα, είτε ενδιαφέροντα είτε μη, και χρησιμοποίησε αυτές τις πληροφορίες για να φιλτράρει έγγραφα για άλλα άτομα. Υπάρχουν τώρα εκατοντάδες ιστότοποι που χρησιμοποιούν κάποιο είδος Collaborative Filtering για ταινίες, μουσική, βιβλία, αγορές, podcast, άρθρα κλπ. Το Collaborative Filtering χρησιμοποιεί

ομοιότητες μεταξύ χρηστών και στοιχείων ταυτόχρονα για την παροχή συστάσεων, όπως μοντέλα συνεργατικού φιλτραρίσματος μπορούν να προτείνουν ένα στοιχείο στο χρήστη A με βάση τα ενδιαφέροντα ενός παρόμοιου χρήστη B. Ένας τρόπος, να αναπαραστήσουμε ένα σύνολο προτιμήσεων διαφορετικών χρηστών, στην Python, γίνεται με τη χρήση των λεξικών. Τα λεξικά είναι ένας τύπος δεδομένων της Python, ο οποίος μπορεί να θεωρηθεί κατά ένα τρόπο ως γενίκευση του τύπου δεδομένων λίστας. Τα λεξικά, όπως και οι λίστες, είναι δομές δεδομένων που μπορούν να αποθηκεύσουν αντικείμενα κάθε τύπου. Στο συγκεκριμένο παράδειγμα το λεξικό χρησιμοποιεί ένα εύρος βαθμολογίας από 1 έως 5, έτσι ώστε να εκφράσουν οι χρήστες τις προτιμήσεις τους σε διαφορετικούς τύπους βιβλίων. Μπορούμε να δημιουργήσουμε ένα αρχείο (file), να το ονομάσουμε recommendations.py και να προσθέσουμε ενδεικτικά τη βαθμολογία που έχουν δώσει σε τίτλους βιβλίων διάφοροι χρήστες. Στην παρακάτω εικόνα παρατηρούμε ενδεικτικά τη βαθμολογία ορισμένων χρηστών.

Εικόνα 1. Βαθμολογία χρηστών σε τίτλους βιβλίων.

```

1  # A dictionary of book critics and their ratings of a small
2  # set of books
3  critics = {
4      'Ferdinand Lewsey': {
5          'Foundation Mathematics for Computer Science': 2.5,
6          'Fundamentals of Discrete Math for Computer Science': 3.5,
7          'Database Systems': 3.0,
8          'Java-Object Oriented Programming': 3.5,
9          'Computer Networks': 2.5,
10         'Domain-Driven DESIGN': 3.0,
11     },
12     'Griselda Regi': {
13         'Foundation Mathematics for Computer Science': 3.0,
14         'Fundamentals of Discrete Math for Computer Science': 3.5,
15         'Database Systems': 1.5,
16         'Java-Object Oriented Programming': 5.0,
17         'Domain-Driven DESIGN': 3.0,
18         'Computer Networks': 3.5,
19     },
20     'Caterina Farr': {
21         'Foundation Mathematics for Computer Science': 2.5,
22         'Fundamentals of Discrete Math for Computer Science': 3.0,
23         'Java-Object Oriented Programming': 3.5,
24         'Domain-Driven DESIGN': 4.0,
25     },
26     'Kelly Navarro': {
27         'Fundamentals of Discrete Math for Computer Science': 3.5,
28         'Database Systems': 3.0,
29         'Domain-Driven DESIGN': 4.5,
30         'Java-Object Oriented Programming': 4.0,
31         'Computer Networks': 2.5,
32     },
33     'Keven Bearn': {
34         'Foundation Mathematics for Computer Science': 3.0,
35         'Fundamentals of Descrete Math for Computer Science': 4.0,
36         'Database Systems': 2.0,
37         'Java-Object Oriented Programming': 3.0,
38         'Domain-Driven DESIGN': 3.0,
39         'Computer Networks': 2.0,
40     },
41     'Kelvin Challenger': {
42         'Foundation Mathematics for Computer Science': 3.0,
43         'Fundamentals of Discrete Math for Computer Science': 4.0,
44         'Domain-Driven DESIGN': 3.0,
45         'Java-Object Oriented Programming': 5.0,
46         'Computer Networks': 3.5,
47     },
48     'Olympia Farr': {
49         'Fundamentals of Discrete Math for Computer Science': 4.5,
50         'Computer Networks': 1.0,
51         'Java-Object Oriented Programming': 4.0
52     }
53 }
```

Παρατηρούμε ενδεικτικά τη βαθμολογία ορισμένων χρηστών, μέσω του τερματικού κόνοντας χρήση του λεξικού της Python.

Εικόνα 2. Ενδεικτική Βαθμολογία χρηστών με χρήση του τερματικού (terminal).

```
Python 2.7.15 (default, Aug 22 2018, 16:36:18)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> from recommendations import critics ]
[>>> critics['Caterina Farr']['Foundation Mathematics for Computer Science'] ]
2.5
[>>> critics['Olympia Farr'] ]
{'Computer Networks': 1.0, 'Fundamentals of Discrete Math for Computer Science':
 4.5, 'Java-Object Oriented Programming': 4.0}
>>>
```

3.2 Εύρεση Παρόμοιων Χρηστών (Finding Similar Users)

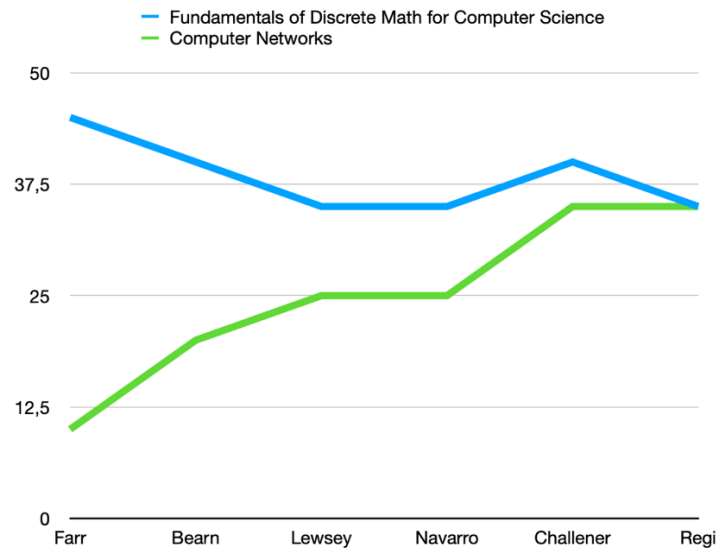
Αφού γίνει συλλογή με δεδομένα σχετικά με τους τίτλους των βιβλίων που βαθμολόγησαν οι χρήστες, χρειαζόμαστε έναν τρόπο για να προσδιορίσουμε πόσα άτομα έχουν παρόμοιες προτιμήσεις. Αυτό επιτυγχάνεται συγκρίνοντας κάθε άτομο με κάθε άλλο άτομο και υπολογίζοντας έναν αριθμό ομοιότητας (similarity cost). Υπάρχουν αρκετοί τρόποι για να επιτευχθεί αυτό και σε αυτήν την ενότητα θα γίνει αναφορά σε δύο συστήματα για τον υπολογισμό των αποτελεσμάτων ομοιότητας, στην Ευκλείδεια απόσταση (Euclidean distance) και στη συσχέτιση του Pearson (Pearson correlation) [2].

3.3 Ευκλείδεια Απόσταση

Ένας πολύ απλός τρόπος να υπολογίσουμε το βαθμό ομοιότητας (similarity cost) είναι να χρησιμοποιήσουμε την Ευκλείδεια απόσταση η οποία παίρνει τους κοινούς τίτλους βιβλίων που οι χρήστες έχουν βαθμολογήσει και τους χρησιμοποιεί ως άξονες στο γράφημα. Στη συνέχεια, απεικονίζουμε τους χρήστες αυτούς στο γράφημα και παρατηρούμε πόσο κοντά είναι.

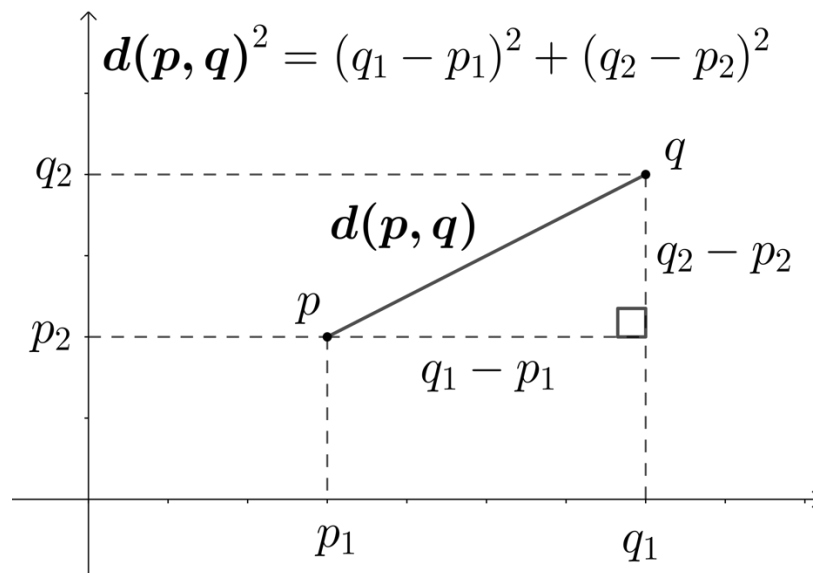
Πίνακας 1

	Fundamentals of Discrete Math for Computer Science	Computer Networks
Farr	45	10
Bearn	40	20
Lewsey	35	25
Navarro	35	25
Challener	40	35
Regi	35	35



Η ευκλείδεια μετρική είναι μία συνάρτηση: $d: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ που αντιστοιχεί σε δύο διανύσματα x, y του n -διάστατου διανυσματικού χώρου \mathbb{R}^n , $x = \{x_1, \dots, x_n\}$, $y = \{y_1, \dots, y_n\}$ τον αριθμό $d(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \dots + (y_n - x_n)^2} = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$. [7] Η συνάρτηση μετράει την Ευκλείδεια απόσταση μεταξύ δύο σημείων στο επίπεδο, n -διάστατο χώρο κάνοντας επανειλημμένη χρήση του Πυθαγορείου θεωρήματος.

Εικόνα 3. Ευκλείδεια Απόσταση [9].



Όσο πιο κοντά βρίσκονται δύο άτομα στο χώρο προτιμήσεων, τόσο πιο παρόμοιες είναι οι προτιμήσεις τους. Για να υπολογίσουμε την απόσταση μεταξύ Farr και Beam στο γράφημα, εφαρμόζουμε τον τύπο της Ευκλείδειας απόστασης, όπως φαίνεται παρακάτω.

Εικόνα 4. Εφαρμογή Ευκλείδειας Απόστασης.

```
Python 2.7.15 (default, Aug 22 2018, 16:36:18)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> from recommendations import critics
[>>> critics['Caterina Farr']['Foundation Mathematics for Computer Science']
2.5
[>>> critics['Olympia Farr']
{'Computer Networks': 1.0, 'Fundamentals of Discrete Math for Computer Science':
 4.5, 'Java-Object Oriented Programming': 4.0}
[>>> from math import sqrt
[>>> sqrt(pow(1-2,2)+pow(4.5-4.0,2))
1.118033988749895
[>>> 1/(1+sqrt(pow(1-2,2)+pow(4.5-4.0,2)))
0.4721359549995794
>>>
```

Προσθέτουμε την παρακάτω συνάρτηση στο αρχείο recommendations.py που έχουμε δημιουργήσει. Η συγκεκριμένη συνάρτηση η οποία αναφέρεται στο βιβλίο “Programming Collective Intelligence: Building Smart Web 2.0 Applications”, [2] υπολογίζει το βαθμό ομοιότητας δυο χρηστών και επιστρέφει πάντα μια τιμή μεταξύ 0 και 1, όπου η τιμή 1 σημαίνει ότι δύο άτομα έχουν πανομοιότυπες προτιμήσεις.

Εικόνα 5. Συνάρτηση υπολογισμού βαθμού ομοιότητας με Python.

```
55 from math import sqrt
56
57 # Returns a distance-based similarity score for person1 and person2
58 def sim_distance(prefs, person1, person2):
59     # Get the list of shared_items
60     si={}
61     for item in prefs[person1]:
62         if item in prefs[person2]: si[item]=1
63
64     # if they have no ratings in common, return 0
65     if len(si)==0: return 0
66
67     # Add up the squares of all the differences
68     sum_of_squares=sum([pow(prefs[person1][item]-prefs[person2][item],2)
69                       for item in prefs[person1] if item in prefs[person2]])
70
71     return 1/(1+sum_of_squares)
```

Παρακάτω γίνεται και η χρήση αυτής της συνάρτησης με το τερματικό υπολογίζοντας το βαθμό ομοιότητας (similarity score) δύο χρηστών, του Ferdinand Lewsey και της Griselda Regi.

Εικόνα 6. Εύρεση βαθμού ομοιότητας δυο χρηστών με χρήση του τερματικού.

```
Python 2.7.15 (default, Aug 22 2018, 16:36:18)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> from recommendations import critics ]
[>>> from math import sqrt ]
[>>> import recommendations ]
[>>> reload(recommendations) ]
<module 'recommendations' from 'recommendations.pyc'>
[>>> recommendations.sim_distance(recommendations.critics,'Ferdinand Lewsey','Griselda Regi') ]
0.14814814814814814
[>>> ]
```

Παρακάτω βλέπουμε την αντίστοιχη υλοποίηση υπολογισμού του βαθμού ομοιότητας (similarity distance) δυο χρηστών με χρήση της PHP. Η συγκεκριμένη συνάρτηση υπολογίζει το βαθμό ομοιότητας δυο χρηστών και επιστρέφει πάντα μια τιμή μεταξύ 0 και 1, όπου η τιμή 1 σημαίνει ότι δύο άτομα έχουν πανομοιότυπες προτιμήσεις [8].

Εικόνα 7. Συνάρτηση για υπολογισμό βαθμού ομοιότητας (similarity distance) με PHP.

```
1  <?php
2  function similarity_distance($matrix,$person1,$person2)
3  {
4      //var_dump($matrix);
5      $similar=array();
6      $sum=0;
7      if( isset($matrix[$person1]) && is_array($matrix[$person1])){
8          foreach($matrix[$person1] as $key=> $value)
9          {
10             //var_dump($value);
11             if(array_key_exists($key,$matrix[$person2]))
12             {
13                 $similar[$key]=1;
14             }
15         }
16     }
17     if($similar==0)
18     {
19         return 0;
20     }
21     if( isset($matrix[$person1]) && is_array($matrix[$person1])){
22         foreach($matrix[$person1] as $key=> $value)
23         {
24             if(array_key_exists($key,$matrix[$person2]))
25             {
26                 $sum=$sum+pow($value-$matrix[$person2][$key],2);
27             }
28         }
29     }
30     return 1/(1+sqrt($sum));
31 }
32 ?>
```

3.4 Συσχέτιση Pearson

Ένας διαφορετικός τρόπος για τον προσδιορισμό της ομοιότητας μεταξύ των ενδιαφερόντων των χρηστών είναι η χρήση του συντελεστή συσχέτισης Pearson. Ο συντελεστής συσχέτισης είναι ένα μέτρο για το πόσο καλά δύο σύνολα δεδομένων ταιριάζουν σε μια ευθεία γραμμή. Ο συντελεστής συσχέτισης Pearson είναι πιο περίπλοκη μέθοδος από την Ευκλείδεια απόσταση αλλά τείνει να δίνει καλύτερα αποτελέσματα σε καταστάσεις όπου τα δεδομένα δεν είναι καλά ομαλοποιημένα. Για να απεικονίσουμε αυτήν τη μέθοδο, μπορούμε να σχεδιάσουμε τις βαθμολογίες από την κριτική δύο βιβλίων σε ένα γράφημα. Το βιβλίο Database Systems βαθμολογήθηκε με 3 από τον Kelly Navarro και 5 από το χρήστη Griselda Regi, οπότε τοποθετείται στο (3,5) στο γράφημα. Σχετικά με το ποια μέθοδος είναι προτιμότερο να εφαρμόζεται, μέχρι τώρα έγιναν αναφορές σε δύο συναρτήσεις για δύο διαφορετικές μετρήσεις, αλλά στην πραγματικότητα υπάρχουν πολλοί περισσότεροι τρόποι για τη μέτρηση της ομοιότητας μεταξύ δύο συνόλων δεδομένων. Σύμφωνα με το βιβλίο “Programming Collective Intelligence: Building Smart Web 2.0 Applications”, [2] το τι είναι προτιμότερο να χρησιμοποιήσουμε εξαρτάται από την εφαρμογή μας. Αξίζει να εφαρμοστούν και η απόσταση Pearson και η Ευκλείδεια απόσταση ή κάποια διαφορετική για να δούμε ποια δίνει καλύτερα αποτελέσματα. Προσθέτουμε την παρακάτω συνάρτηση [2] στο αρχείο recommendations.py όπου εφαρμόζεται η απόσταση Pearson στο αρχείο που έχουμε δημιουργήσει.

Εικόνα 8. Συνάρτηση Συσχέτισης Pearson με Python.

```

73 def sim_pearson(prefs, p1, p2):
74
75     # Returns the Pearson correlation coefficient for p1 and p2.
76
77     # Get the list of mutually rated items
78     si = {}
79     for item in prefs[p1]:
80         if item in prefs[p2]:
81             si[item] = 1
82     # If they are no ratings in common, return 0
83     if len(si) == 0:
84         return 0
85     # Sum calculations
86     n = len(si)
87     # Sums of all the preferences
88     sum1 = sum([prefs[p1][it] for it in si])
89     sum2 = sum([prefs[p2][it] for it in si])
90     # Sums of the squares
91     sum1Sq = sum([pow(prefs[p1][it], 2) for it in si])
92     sum2Sq = sum([pow(prefs[p2][it], 2) for it in si])
93     # Sum of the products
94     pSum = sum([prefs[p1][it] * prefs[p2][it] for it in si])
95     # Calculate r (Pearson score)
96     num = pSum - sum1 * sum2 / n
97     den = sqrt((sum1Sq - pow(sum1, 2) / n) * (sum2Sq - pow(sum2, 2) / n))
98     if den == 0:
99         return 0
100    r = num / den
101    return r
102

```

Παρακάτω βλέπουμε στην πράξη τον υπολογισμό του αριθμού ομοιότητας (similarity score) δύο χρηστών κάνοντας χρήση της απόστασης Pearson μέσω του τερματικού. Αυτή η συνάρτηση θα επιστρέφει μια τιμή μεταξύ -1 και 1 . Μια τιμή 1 σημαίνει ότι τα δύο άτομα έχουν ακριβώς τις ίδιες βαθμολογίες για κάθε στοιχείο.

Εικόνα 9. Εφαρμογή Συσχέτισης Pearson στο τερματικό.

```
Python 2.7.15 (default, Aug 22 2018, 16:36:18)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> from recommendations import critics
[>>> critics['Caterina Farr']['Foundation Mathematics for Computer Science']
2.5
[>>> critics['Olympia Farr']
{'Computer Networks': 1.0, 'Fundamentals of Discrete Math for Computer Science':
 4.5, 'Java-Object Oriented Programming': 4.0}
[>>> from math import sqrt
[>>> sqrt(pow(1-2,2)+pow(4.5-4.0,2))
1.118033988749895
[>>> 1/(1+sqrt(pow(1-2,2)+pow(4.5-4.0,2)))
0.4721359549995794
[>>> import recommendations
[>>> reload(recommendations)
<module 'recommendations' from 'recommendations.pyc'>
[>>> print recommendations.sim_pearson(recommendations.critics, 'Ferdinand Lewsey'
, 'Griselda Regi')
0.396059017191
>>>
```

3.5 Κατάταξη των Κριτικών

Εκτός από τις συναρτήσεις που υπολογίζουν και συγκρίνουν τις προτιμήσεις δυο χρηστών μπορούμε να υπολογίσουμε μέσω συνάρτησης τις προτιμήσεις των υπολοίπων χρηστών και να τις συγκρίνουμε με τις προτιμήσεις του συνδεδεμένου χρήστη. Αυτό γίνεται όταν μας ενδιαφέρει να μάθουμε ποιοι κριτικοί βιβλίων έχουν παρόμοιες απόψεις με εμάς ώστε να γνωρίζουμε τις κριτικές των χρηστών που θα συμβουλευόμαστε, όταν αποφασίσουμε να αγοράσουμε για παράδειγμα ένα βιβλίο. Η συνάρτηση υλοποίησης [2] φαίνεται παρακάτω:

Εικόνα 10. Συνάρτηση που επιστρέφει μια ταξινομημένη λίστα ατόμων με παρόμοιες προτιμήσεις με αυτές του χρήστη.

```
104 def topMatches(prefs, person, n=5, similarity=sim_pearson):
105
106     # Returns the best matches for person from the prefs dictionary.
107     # Number of results and similarity function are optional params.
108
109
110     scores = [(similarity(prefs, person, other), other) for other in prefs
111               if other != person]
112     scores.sort()
113     scores.reverse()
114     return scores[0:n]
115
```

Η μέθοδος `sort()` είναι μια (built in) μέθοδος της Python που ταξινομεί (by default) τη λίστα κατά αύξουσα σειρά. Η μέθοδος `reverse()` αντιστρέφει τα στοιχεία μια λίστας στην Python. Γίνεται εφαρμογή της συνάρτησης στο τερματικό.

Εικόνα 11. Εφαρμογή της συνάρτησης topMatches στο τερματικό.

```
[>>> recommendations.topMatches(recommendations.critics,'Olympia Farr',n=3)
[(0.9912407071619299, 'Ferdinand Lewsey'), (0.9244734516419049, 'Keven Bearn'),
(0.8934051474415647, 'Kelly Navarro')]
>>> ]
```

Η κλήση αυτής της συνάρτησης δίνει μια λίστα με κριτικούς βιβλίων και βαθμολογίες ομοιότητας (similarity scores) με το χρήστη που είναι συνδεδεμένος στην εφαρμογή. Από αυτά βγαίνει το συμπέρασμα ότι ο συνδεδεμένος χρήστης θα πρέπει να διαβάσει κριτικές από τον Ferdinand Lewsey, τον Keven Bearn και την Kelly Navarro καθώς οι προτιμήσεις τους τείνουν να μοιάζουν με του συνδεδεμένου χρήστη.

3.6 Σύσταση Προϊόντων (Recommending Items)

Σχετικά με τη σύσταση προϊόντων [2] θα μπορούσαμε απλώς να κοιτάξουμε το άτομο που έχει τις ίδιες προτιμήσεις με τη δικές μας και να αναζητήσουμε ένα βιβλίο που του αρέσει και το οποίο δεν έχουμε διαβάσει ακόμα αλλά μια τέτοια προσέγγιση δεν παρέχει την ποιότητα των αποτελεσμάτων που θα θέλαμε. Μια τέτοια προσέγγιση θα μπορούσε κατά λάθος να εμφανίσει κριτικούς που δεν έχουν βαθμολογήσει κάποια από τα βιβλία που θα μπορούσαν να μας ενδιαφέρουν. Θα μπορούσε επίσης να εμφανίσει έναν χρήστη που παραδόξως του άρεσε ένα βιβλίο που έλαβε κακές κριτικές από όλους τους άλλους κριτικούς που εμφανίζονται στον πίνακα των topMatches. Για να λυθούν αυτά τα ζητήματα, πρέπει να βαθμολογήσουμε τα στοιχεία δημιουργώντας μια σταθμισμένη βαθμολογία που κατατάσσει τους κριτικούς. Παίρνουμε τις ψήφους όλων των άλλων κριτικών και πολλαπλασιάζουμε τον αριθμό ομοιότητας με τη βαθμολογία που έδωσαν σε κάθε βιβλίο. Ο Πίνακας 2 δείχνει πώς λειτουργεί αυτή η διαδικασία.

Πίνακας 2

Critic	Similarity	Domain-Driven Design	S.xDomain-Driven Design	Maths for Computer Science	S.xMaths for computer Science	Database Systems	S.xDatabase Systems
Lewsey	0.99	3.0	2.97	2.5	2.48	3.0	2.97
Regi	0.38	3.0	1.14	3.0	1.14	1.5	0.57
Navarro	0.89	4.5	4.02			3.0	2.68
Bearn	0.92	3.0	2.77	3.0	2.77	2.0	1.85
Challener	0.66	3.0	1.99	3.0	1.99		
Total			12.89		8.38		8.07
Sim.Sum			3.84		2.95		3.18
Total/Sim.Sum			3.35		2.83		2.53

Αυτός ο πίνακας δείχνει βαθμολογίες συσχέτισης για κάθε κριτικό και τις βαθμολογίες που έδωσαν στα τρία βιβλία (Domain-Driven Design, Maths for Computer Science και Database Systems) που ο συνδεδεμένος χρήστης δεν έχει βαθμολογήσει. Οι στήλες που ξεκινούν με S.x δίνουν την ομοιότητα πολλαπλασιασμένη με τη βαθμολογία, οπότε ένα άτομο που είναι παρόμοιο με το χρήστη θα συμβάλει περισσότερο στη συνολική βαθμολογία από ένα άτομο που είναι διαφορετικό. Η σειρά Total δείχνει το άθροισμα όλων αυτών των αριθμών. Θα μπορούσαμε απλά να χρησιμοποιήσουμε τα τελικά αποτελέσματα για να υπολογιστούν οι βαθμολογίες αλλά στη συνέχεια ένα βιβλίο με κριτική από περισσότερους ανθρώπους θα είχε ένα μεγαλύτερο πλεονέκτημα. Για να διορθωθεί αυτό, πρέπει να διαιρέσουμε το τελικό αποτέλεσμα με το άθροισμα όλων των τιμών ομοιότητας για τα άτομα που έκαναν κριτική σε αυτήν την ταινία (η σειρά Sim.Sum στον πίνακα). Επειδή το Domain-Driven Design υποβλήθηκε σε κριτική από όλους, το σύνολο του διαιρείται με το άθροισμα όλων των ομοιοτήτων.

Το Maths for Computer Science, ωστόσο, δεν έλαβε κριτική από τον Navarro, οπότε η βαθμολογία της ταινίας διαιρείται με το άθροισμα όλων των άλλων ομοιοτήτων. Η τελευταία σειρά δείχνει τα αποτελέσματα αυτής της διαίρεσης. Παρακάτω βλέπουμε και ποιοι τίτλοι βιβλίων θα ενδιέφεραν το χρήστη κάνοντας χρήση της παρακάτω συνάρτησης [2].

Εικόνα 12. Συνάρτηση σύστασης βιβλίων στο χρήστη με Python.

```

117 #Gets recommendations for a person by using a weighted average
118 #of every other user's rankings
119 def getRecommendations(prefs, person, similarity=sim_pearson):
120
121     totals = {}
122     simSums = {}
123     for other in prefs:
124         # Don't compare me to myself
125         if other == person:
126             continue
127         sim = similarity(prefs, person, other)
128         # Ignore scores of zero or lower
129         if sim <= 0:
130             continue
131         for item in prefs[other]:
132             # Only score books I haven't read yet
133             if item not in prefs[person] or prefs[person][item] == 0:
134                 # Similarity * Score
135                 totals.setdefault(item, 0)
136                 # The final score is calculated by multiplying each item by the
137                 # similarity and adding these products together
138                 totals[item] += prefs[other][item] * sim
139                 # Sum of similarities
140                 simSums.setdefault(item, 0)
141                 simSums[item] += sim
142     # Create the normalized list
143     rankings = [(total / simSums[item], item) for (item, total) in
144                 totals.items()]
145     # Return the sorted list
146     rankings.sort()
147     rankings.reverse()
148     return rankings

```

Η εκτέλεση της παραπάνω συνάρτησης στο τερματικό φαίνεται στην παρακάτω εικόνα.

Εικόνα 13. Εφαρμογή συνάρτησης σύστασης βιβλίων στο τερματικό.

```

[>>> recommendations.getRecommendations(recommendations.critics,'Olympia Farr')
[(3.3477895267131013, 'Domain-Driven DESIGN'), (2.832549918264162, 'Foundation Mathematics for
Computer Science'), (2.5309807037655645, 'Database Systems')]
>>> recommendations.getRecommendations(recommendations.critics,'Olympia Farr',similarity=recomm
endations.sim_distance)
[(3.5002478401415877, 'Domain-Driven DESIGN'), (2.7561242939959363, 'Foundation Mathematics for
Computer Science'), (2.461988486074373, 'Database Systems')]
>>>

```

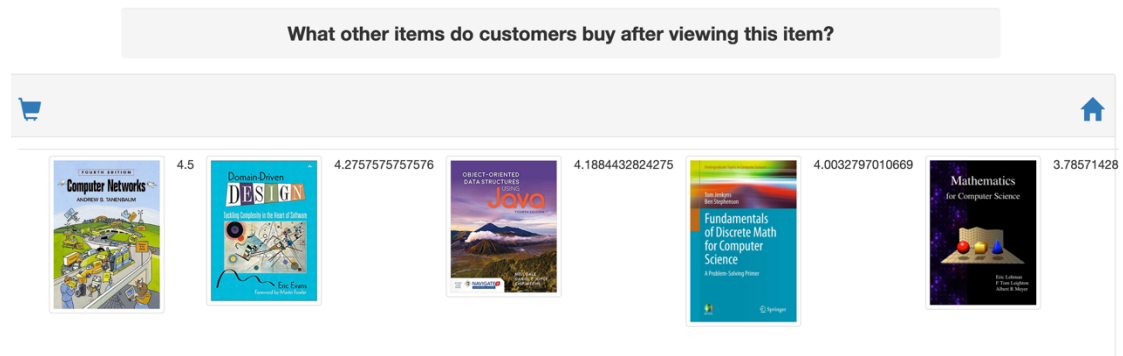
Παρακάτω βλέπουμε τους τίτλους βιβλίων που θα ενδιέφεραν το χρήστη κάνοντας χρήση της παρακάτω συνάρτησης με PHP. [8]

Εικόνα 14. Συνάρτηση σύστασης βιβλίων στο χρήστη με PHP.

```
68 function getRecommendation($matrix,$person)
69 {
70     $total=array();
71     $simsum=array();
72     $ranks=array();
73
74     foreach($matrix as $otherperson=>$value)
75     {
76         //Don't compare me to myself
77         if($otherperson!=$person)
78         {
79             $sim=similarity_distance($matrix,$person,$otherperson);
80             //var_dump($sim);
81
82             foreach($matrix[$otherperson] as $key=>$value)
83             {
84                 if(!array_key_exists($key,$matrix[$person]))
85                 {
86                     if(!array_key_exists($key,$total))
87                     {
88                         $total[$key]=0;
89                     }
90                     //The final score is calculated by multiplying each item by the
91                     //similarity and adding these products together
92                     $total[$key]+=$matrix[$otherperson][$key]*$sim;
93
94                     if(!array_key_exists($key,$simsum))
95                     {
96                         $simsum[$key]=0;
97                     }
98                     //Sum of similarities
99                     $simsum[$key]+=$sim;
100                 }
101             }
102         }
103     }
104
105     foreach($total as $key=>$value)
106     {
107         $ranks[$key]=$value/$simsum[$key];
108     }
109
110     //Return the sorted list
111     array_multisort($ranks, SORT_DESC);
112     return $ranks;
113 }
114
115 ?>
```

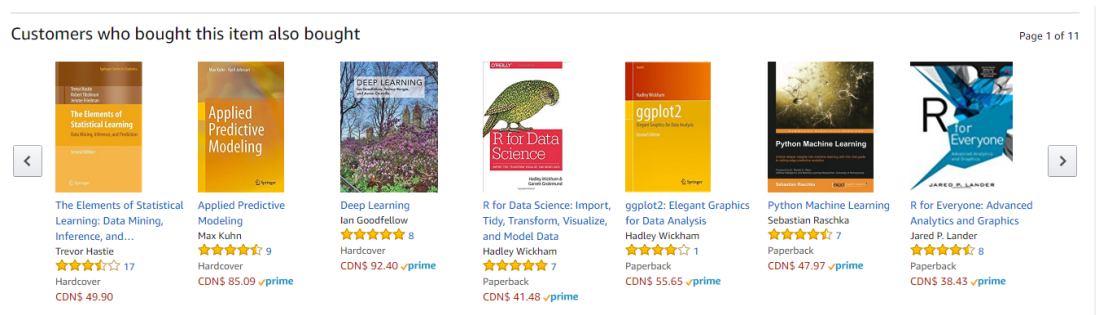
Βλέπουμε ενδεικτικά ένα στιγμιότυπο σύστασης βιβλίων στο χρήστη του ηλεκτρονικού βιβλιοπωλείου κάνοντας χρήση της παραπάνω συνάρτησης.

Εικόνα 15. Στιγμιότυπο σύστασης βιβλίων από την εφαρμογή.



3.7 Προϊόντα Παρόμοια Μεταξύ τους (Matching Products)

Παραπάνω εξετάσαμε τρόπους να βρίσκουμε άτομα που έχουν παρόμοιες προτιμήσεις με εμάς και η εφαρμογή να μας προτείνει προϊόντα που μας ενδιαφέρουν. Μπορούμε να εξετάσουμε και ποια προϊόντα είναι παρόμοια μεταξύ τους. Αυτό το συναντάμε όταν ο ιστότοπος δεν έχει συλλέξει πολλές πληροφορίες για εμάς [2]. Συχνά συναντάμε την παρακάτω εικόνα μετά από αγορές μας.



Σε αυτήν την περίπτωση, μπορούμε να προσδιορίσουμε την ομοιότητα κοιτάζοντας ποιος έκανε καλή κριτική σε ένα συγκεκριμένο αντικείμενο και βλέποντας και τα άλλα πράγματα που του άρεσαν. Αυτή είναι στην πραγματικότητα η ίδια μέθοδος που χρησιμοποιήσαμε νωρίτερα για να προσδιορίσουμε την ομοιότητα στις προτιμήσεις μεταξύ των χρηστών. Παρακάτω φαίνεται η υλοποίηση της συνάρτησης [2] καθώς και η εφαρμογή της στο τερματικό.

Εικόνα 16. Συνάρτηση που επιστρέφει τις βαθμολογίες που δέχθηκαν τίτλοι βιβλίων από τους χρήστες με Python.

```

150 | # Transform the recommendations into a mapping where persons are described
151 | # with interest scores for a given title e.g. {title: person} instead of
152 | # {person: title}.
153 | def transformPrefs(prefs):
154 |
155 |     result = {}
156 |     for person in prefs:
157 |         for item in prefs[person]:
158 |             result.setdefault(item, {})
159 |             # Flip item and person
160 |             result[item][person] = prefs[person][item]
161 |     return result
162 |

```

Εικόνα 17. Εφαρμογή της συνάρτησης transformPrefs στο τερματικό.

```

[>>> recommendations.transformPrefs(recommendations.critics)
{'Fundamentals of Discrete Math for Computer Science': {'Griselda Regi': 3.5, 'Keven Bearn': 4.0, 'Kelly Navarro': 3.5, 'Ferdinand Lewsey': 3.5, 'Kelvin Challenger': 4.0, 'Olympia Farr': 4.5, 'Caterina Farr': 3.0}, 'Database Systems': {'Keven Bearn': 2.0, 'Kelly Navarro': 3.0, 'Ferdinand Lewsey': 3.0, 'Griselda Regi': 1.5}, 'Foundation Mathematics for Computer Science': {'Keven Bearn': 3.0, 'Caterina Farr': 2.5, 'Ferdinand Lewsey': 2.5, 'Griselda Regi': 3.0, 'Kelvin Challenger': 3.0}, 'Computer Networks': {'Griselda Regi': 3.5, 'Keven Bearn': 2.0, 'Kelly Navarro': 2.5, 'Ferdinand Lewsey': 2.5, 'Kelvin Challenger': 3.5, 'Olympia Farr': 1.0}, 'Java-Object Oriented Programming': {'Griselda Regi': 5.0, 'Keven Bearn': 3.0, 'Kelly Navarro': 4.0, 'Ferdinand Lewsey': 3.5, 'Kelvin Challenger': 5.0, 'Olympia Farr': 4.0, 'Caterina Farr': 3.5}, 'Domain-Driven DESIGN': {'Griselda Regi': 3.0, 'Keven Bearn': 3.0, 'Kelly Navarro': 4.5, 'Ferdinand Lewsey': 3.0, 'Kelvin Challenger': 3.0, 'Caterina Farr': 4.0}}
]>>>

```

Εικόνα 18. Συνάρτηση που επιστρέφει μια ταξινομημένη λίστα ατόμων με παρόμοιες προτιμήσεις με αυτές του χρήστη.

```

104 | def topMatches(prefs, person, n=5, similarity=sim_pearson):
105 |
106 |     # Returns the best matches for person from the prefs dictionary.
107 |     # Number of results and similarity function are optional params.
108 |
109 |
110 |     scores = [(similarity(prefs, person, other), other) for other in prefs
111 |               | if other != person]
112 |     scores.sort()
113 |     scores.reverse()
114 |     return scores[0:n]
115 |

```

Εικόνα 19. Εφαρμογή συνάρτησης που επιστρέφει μια ταξινομημένη λίστα βιβλίων παρόμοιων προτιμήσεων με αυτές του χρήστη.

```

[>>> books=recommendations.transformPrefs(recommendations.critics)
]>>> recommendations.topMatches(books, 'Java-Object Oriented Programming')
[(0.6579516949597695, 'Computer Networks'), (0.4879500364742689, 'Foundation Mathematics for Computer Science'), (0.11180339887498941, 'Fundamentals of Discrete Math for Computer Science'), (-0.1798471947990544, 'Domain-Driven DESIGN'), (-0.42289003161103106, 'Database Systems')]
]>>>

```

Παρακάτω βλέπουμε τις υλοποιήσεις των συναρτήσεων [8] με PHP καθώς και στιγμιότυπο από την εφαρμογή της στο ηλεκτρονικό κατάστημα.

Εικόνα 20. Συνάρτηση που επιστρέφει τις βαθμολογίες που δέχθηκαν τίτλοι βιβλίων από τους χρήστες με PHP.

```

53 function transformPreferences($matrix)
54 {
55     $result = array();
56
57     foreach($matrix as $otherperson => $values)
58     {
59         foreach($values as $key => $value)
60         {
61             $result[$key][$otherperson] = $value;
62         }
63     }
64
65     return $result;
66 }
67

```

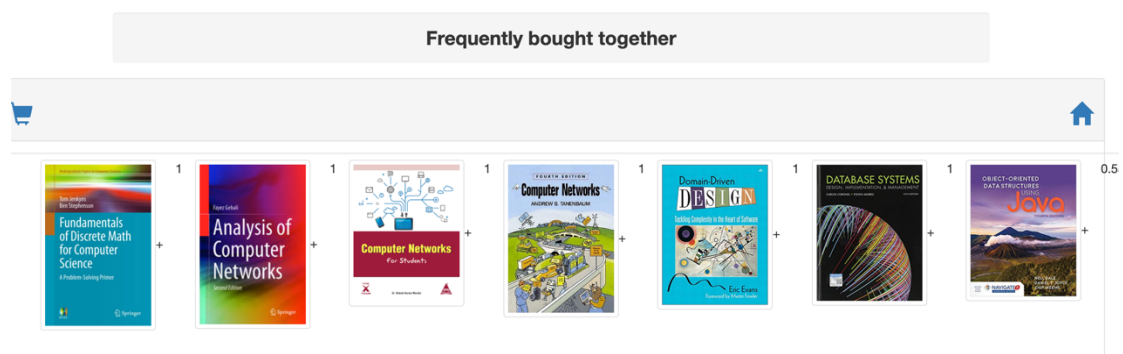
Εικόνα 21. Συνάρτηση που επιστρέφει μια ταξινομημένη λίστα βιβλίων παρόμοιων προτιμήσεων με αυτές του χρήστη με PHP.

```

31 function matchItems($matrix, $person)
32 {
33     $score = array();
34
35     foreach($matrix as $otherperson=>$values)
36     {
37         if($otherperson !== $person)
38         {
39
40             $sim=similarity_distance($matrix,$person,$otherperson);
41
42             if($sim > 0)
43                 $score[$otherperson] = $sim;
44         }
45     }
46
47     array_multisort($score, SORT_DESC);
48     return $score;
49

```

Εικόνα 22. Στιγμιότυπο από την εφαρμογή της συνάρτησης στο ηλεκτρονικό βιβλιοπωλείο.



Η αντικατάσταση των χρηστών με τα προϊόντα, όπως φαίνεται στη συνάρτηση της παρακάτω εικόνας, θα επιτρέψει να αναζητηθούν άτομα που ενδέχεται να αγοράσουν συγκεκριμένα προϊόντα. Αυτό μπορεί να είναι πολύ χρήσιμο για τον προγραμματισμό μιας προσπάθειας του τμήματος μάρκετινγκ για την παραγγελία συγκεκριμένων τίτλων βιβλίων.

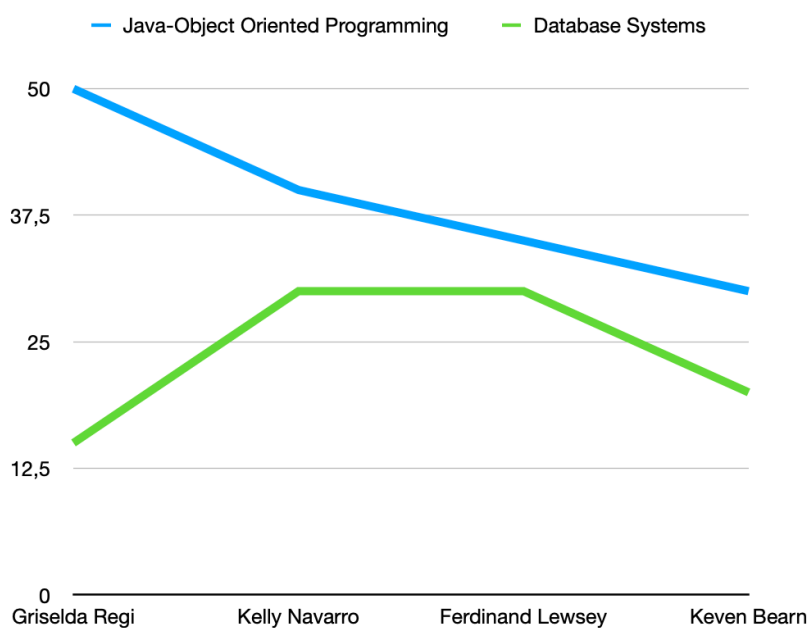
Βλέπουμε την υλοποίηση της συνάρτησης [2] που αναφέραμε παραπάνω κάνοντας αντικατάσταση τους χρήστες του ηλεκτρονικού βιβλιοπωλείου με βιβλία.

Εικόνα 23. Υλοποίηση συνάρτησης για εύρεση ατόμων που ενδέχεται να ενδιαφέρονται για τα συγκεκριμένα βιβλία.

```
[>>> recommendations.getRecommendations(books, 'Database Systems')
[(4.0, 'Caterina Farr'), (3.0, 'Kelvin Challenger')]
>>> █
```

Πίνακας 3

	Java-Object Oriented Programming	Database Systems
Griselda Regi	50	15
Kelly Navarro	40	30
Ferdinand Lewsey	35	30
Keven Bearn	30	20



4. Αντικειμενοστραφής Ανάπτυξη Λογισμικού με UML βασισμένη στη διαδικασία RUP

4.1 Στόχοι της εργασίας και ορισμός προβλήματος προς επίλυση

Στη συγκεκριμένη εργασία έγινε εφαρμογή των αρχών της τεχνολογίας λογισμικού σε ένα διαδικτυακό κατάστημα πώλησης βιβλίων. Μελετήθηκαν και εφαρμόστηκαν τα θέματα της τεχνολογίας λογισμικού (ανάλυση απαιτήσεων, σύνταξη διαγραμμάτων ροής δεδομένων, διαγραμμάτων μετάβασης κατάστασης, κ.λπ.) στο διαδικτυακό κατάστημα πώλησης βιβλίων.

Το πρόβλημα αποτελεί έννοια που απαντάται σε όλες τις επιστήμες και τους κλάδους τους, αλλά παράλληλα και στην καθημερινή μας ζωή. Τόσο η αντιμετώπιση, όσο και η διατύπωση ενός προβλήματος, αποτελούν διαδικασίες που απαιτούν ιδιαίτερες αναλυτικές και συνθετικές ικανότητες, ορθολογική σκέψη, αλλά και σωστό και εμπειριστατωμένο χειρισμό της φυσικής γλώσσας. Με τον όρο Πρόβλημα εννοείται μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της δεν είναι γνωστή, ούτε προφανής.[10] Στη συγκεκριμένη εργασία γίνεται ανάλυση ενός διαδικτυακού καταστήματος πώλησης βιβλίων. Περιλαμβάνει όλες τις ενέργειες που γίνονται σε μια επιχείρηση με φυσικό κατάστημα με μόνη διαφορά ότι αυτές γίνονται ηλεκτρονικά μειώνοντας αισθητά το χρόνο που απαιτείται για την ολοκλήρωσή τους. Τέτοιες ενέργειες είναι: η ανάπτυξη, η προώθηση, η παράδοση, η εξυπηρέτηση και πληρωμή προϊόντων και υπηρεσιών. Ο στόχος που πρέπει να επιτευχθεί είναι η γρήγορη και με απλές διαδικασίες εξυπηρέτηση των πελατών του ηλεκτρονικού καταστήματος. Τεχνικές απόκτησης γνώσης για την επίλυση του προβλήματος επιτυγχάνονται μετά από συνέντευξη με τον πελάτη, με την παρακολούθηση των εργασιών του προβλήματος από τον μηχανικό λογισμικού. Η RUP είναι μια Διαδικασία Τεχνολογίας Λογισμικού (Software Engineering Process). Ο στόχος της είναι να διασφαλίσει την παραγωγή λογισμικού υψηλής ποιότητας που ικανοποιεί τις ανάγκες των τελικών χρηστών μέσα σε ένα συγκεκριμένο χρονοδιάγραμμα και κόστος. Ο κύκλος ζωής του λογισμικού (software life-cycle) σκιαγραφεί τη ζωή του προγράμματος λογισμικού από τη στιγμή της γέννησης του μέχρι τη στιγμή της αντικατάστασης ή της εγκατάλειψής του. Ο κύκλος ζωής του λογισμικού στην RUP υποδιαιρείται σε τέσσερις συνεχόμενες φάσεις. Οι τέσσερις φάσεις είναι:

- Η Φάση Σύλληψης (Inception Phase),
- Η Φάση Επεξεργασίας (Elaboration Phase),
- Η Φάση Κατασκευής (Construction Phase) και
- Η Φάση Μετάβασης (Transition Phase).

4.2 Σύλληψη απαιτήσεων

Η σύλληψη απαιτήσεων περιλαμβάνει μια αφήγηση σχετικά με το τι πρέπει να κάνει το σύστημα. Στο συγκεκριμένο σύστημα, ο νέος χρήστης θα έχει την ευκαιρία αφού περιηγηθεί στο e-shop, να κάνει εγγραφή. Ο χρήστης, θα έχει τη δυνατότητα να προβάλει τις διάφορες κατηγορίες, καθώς και τους τίτλους των βιβλίων. Θα έχει τη δυνατότητα επίσης να προβάλει λεπτομέρειες για κάθε τίτλο ξεχωριστά, πχ Εκδότης, ISBN, Τίτλος, Έτος, Σελίδες. Ο χρήστης, αφού επιλέξει τον τίτλο ή τους τίτλους των βιβλίων που τον ενδιαφέρουν, θα έχει τη δυνατότητα να κάνει εισαγωγή στο καλάθι αγορών, να προβάλει τη λίστα παραγγελιών του

και να ολοκληρώσει την αγορά του. Η συγκεκριμένη ιστοσελίδα διαθέτει ένα Recommendation System το οποίο χρησιμοποιεί User-Based και Item-Based Filtering και προτείνει εξατομικευμένα στον επισκέπτη προτάσεις για αγορές βιβλίων με βάση τα ενδιαφέροντα του.

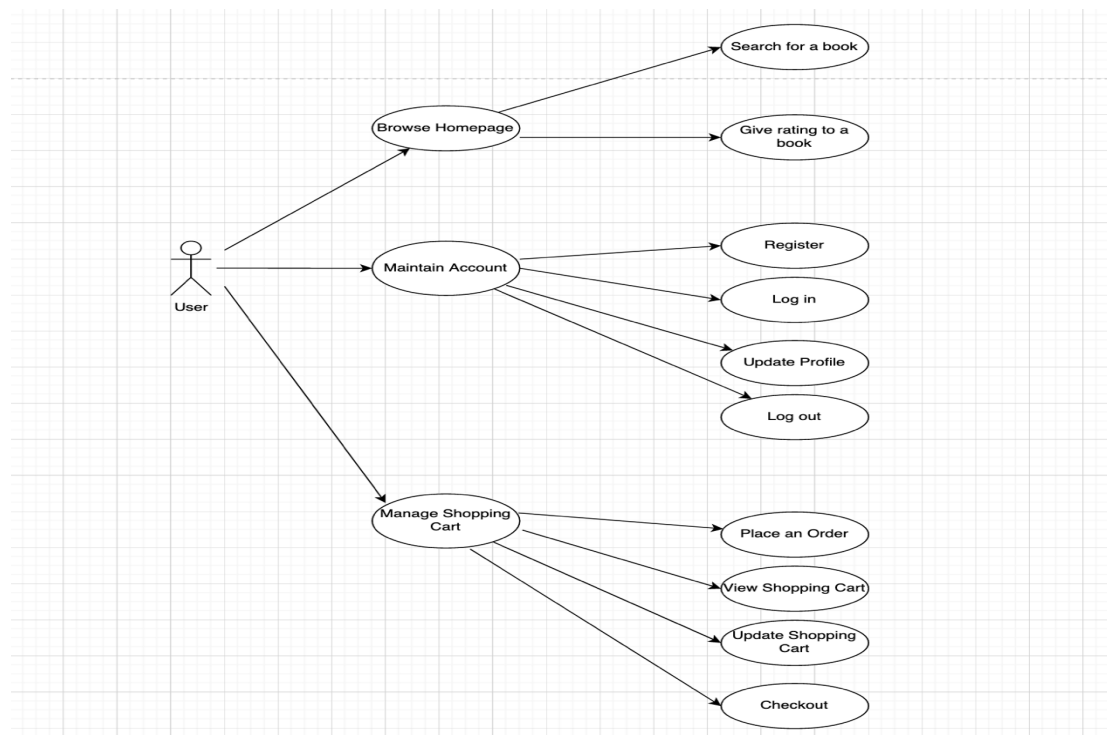
4.3 Ανάλυση-Σχεδιασμός

Γενικά ένα τέτοιο έγγραφο είναι αποτέλεσμα μιας λεπτομερούς διαδικασίας συλλογής των προδιαγραφών και μπορεί να περιλαμβάνει συνεντεύξεις με τους πιθανούς χρήστες του συστήματος και ειδικούς στον τομέα που σχετίζονται με το ηλεκτρονικό εμπόριο. Ο αναλυτής θα πρέπει να πάρει τις πληροφορίες που θα συγκεντρώσει, έτσι ώστε να δημιουργήσει μια λίστα προδιαγραφών του συστήματος. Στην ανάλυση και το σχεδιασμό, γίνεται μια περιγραφή του πως θα υλοποιηθεί το σύστημα. Το στάδιο ανάλυσης εστιάζει στον ορισμό του προβλήματος που πρέπει να επιλυθεί. Όταν σχεδιάζεται οποιοδήποτε σύστημα, κάποιος πρέπει να λύσει το πρόβλημα σωστά αλλά είναι εξίσου σημαντικό να επιλυθεί το σωστό πρόβλημα.

Διάγραμμα Περιπτώσεων Χρήσης

Δημιουργούμε ένα διάγραμμα περιπτώσεων χρήσης για να μοντελοποιήσουμε τις αλληλεπιδράσεις μεταξύ των πελατών ενός συστήματος (στη συγκεκριμένη περίπτωση τους πελάτες του ηλεκτρονικού βιβλιοπωλείου) και των περιπτώσεων χρήσης του όπως αναφέρεται στο βιβλίο των Paul Deitel – Harvey Deitel “Java How To Program (Early Objects) ”[1]. Ο στόχος είναι να δείξουμε τα ήδη των αλληλεπιδράσεων που εκτελούν οι χρήστες με το σύστημα χωρίς να παρέχονται λεπτομέρειες. Αυτές παρέχονται με άλλα UML διαγράμματα που παρουσιάζονται σε όλη αυτή τη μελέτη. Τα διαγράμματα περιπτώσεων χρήσης δημιουργούνται στο στάδιο της ανάλυσης του κύκλου ζωής του λογισμικού.

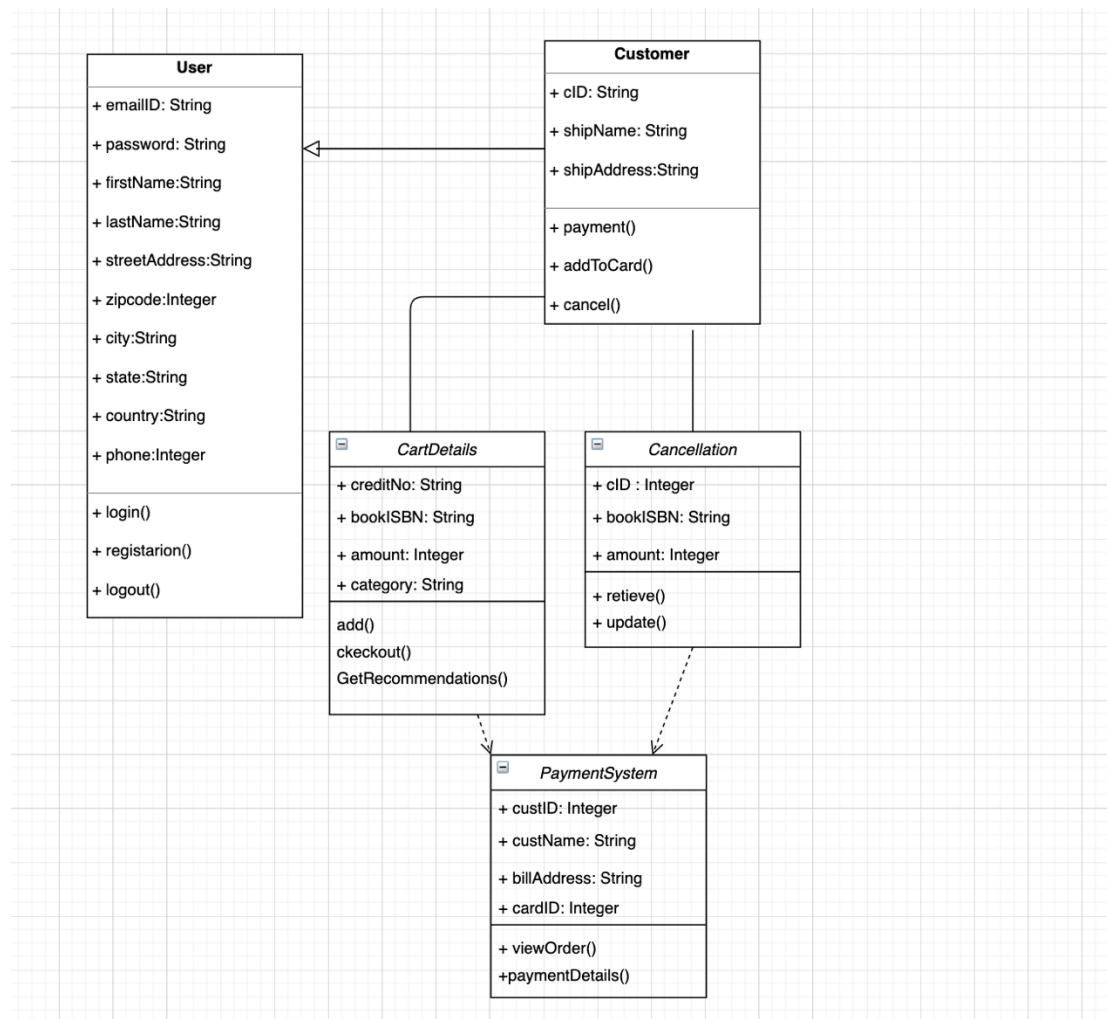
Εικόνα 24. Διάγραμμα Περιπτώσεων Χρήσης.



Διάγραμμα Τάξεων

Ένα ολοκληρωμένο διάγραμμα τάξεων, δείχνει όλους τους συσχετισμούς μεταξύ των τάξεων και όλες τις ιδιότητες και λειτουργίες κάθε τάξης. Στη συγκεκριμένη έκδοση ορίστηκαν απλά ο ρόλος του απλού user, αυτός του customer, όταν δηλαδή ο χρήστης αποφασίζει να προχωρήσει σε αγορά. Ένα ολοκληρωμένο διάγραμμα τάξεων, δείχνει όλους τους συσχετισμούς μεταξύ των τάξεων και όλες τις ιδιότητες και λειτουργίες κάθε τάξης παρέχοντας κάποιες λεπτομέρειες με τις διαθέσιμες κατηγορίες βιβλίων.

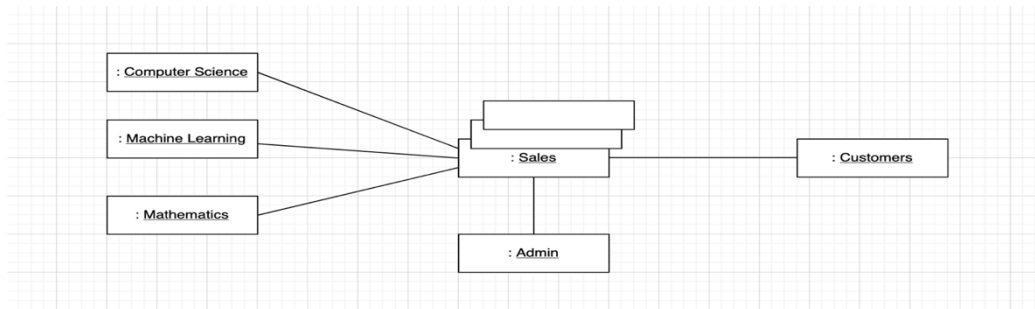
Εικόνα 25. Διάγραμμα τάξεων



Διάγραμμα Αντικειμένων

Στο παρακάτω σχήμα βλέπουμε τα αντικείμενα (object) των Customers του admin, των πωλήσεων καθώς και των διαθέσιμων κατηγοριών και πως αυτά συνδέονται μεταξύ τους.

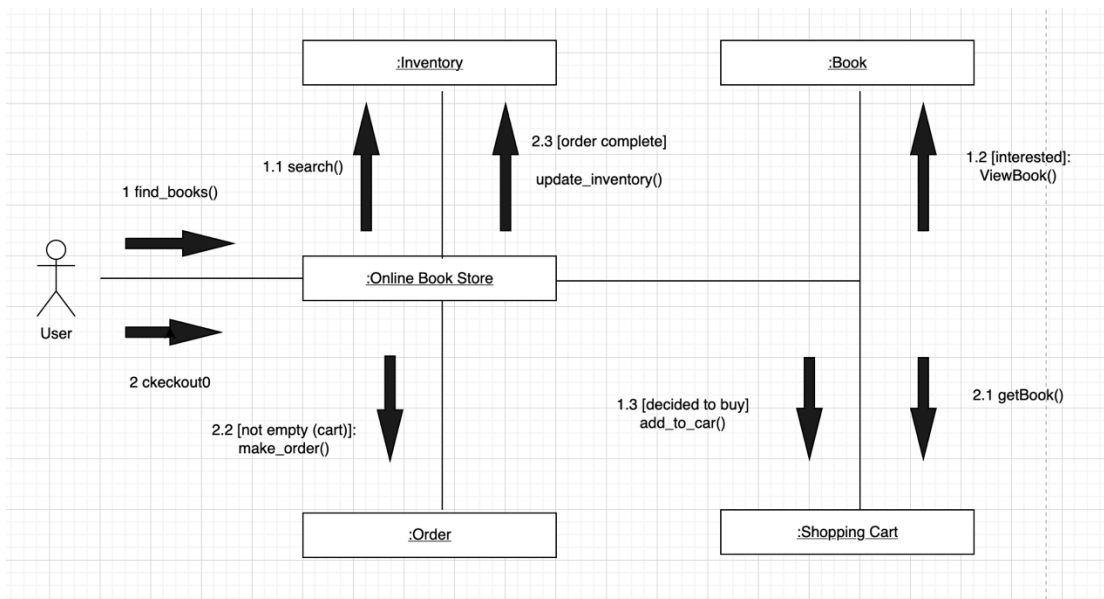
Εικόνα 26. Διάγραμμα αντικειμένων.



Διάγραμμα συνεργασίας

Στο διάγραμμα συνεργασίας (Collaboration Diagram) προσδιορίζουμε τις συνεργασίες στο σύστημα, διαβάζοντας προσεκτικά τις ενότητες του εγγράφου προδιαγραφών που καθορίζουν τί πρέπει να κάνει το σύστημα για να φτάσει μέχρι και τη διαδικασία της πληρωμής. Για κάθε ενέργεια ή βήμα που περιγράφεται, αποφασίζουμε ποια αντικείμενα στο σύστημα μας πρέπει να αλληλεπιδρούν για να πετύχουμε το σωστό αποτέλεσμα. Παρακάτω βλέπουμε το διάγραμμα συνεργασίας. Τα διαγράμματα αυτά υποδεικνύουν τη συνεργασία μεταξύ αντικειμένων. Όταν δυο αντικείμενα επικοινωνούν μεταξύ τους, προκειμένου να εκτελέσουν μια εργασία, λέμε ότι συνεργάζονται. Τα αντικείμενα το κάνουν αυτό καλώντας το ένα τις λειτουργίες του άλλου. Η συνεργασία αποτελείται από ένα αντικείμενο μιας κλάσης που στέλνει ένα μήνυμα σε ένα αντικείμενο μιας άλλης κλάσης, μέσω των μεθόδων. Για να προσδιορίσουμε τις συνεργασίες θα πρέπει να διαβάσουμε προσεκτικά τις ενότητες του εγγράφου προδιαγραφών.

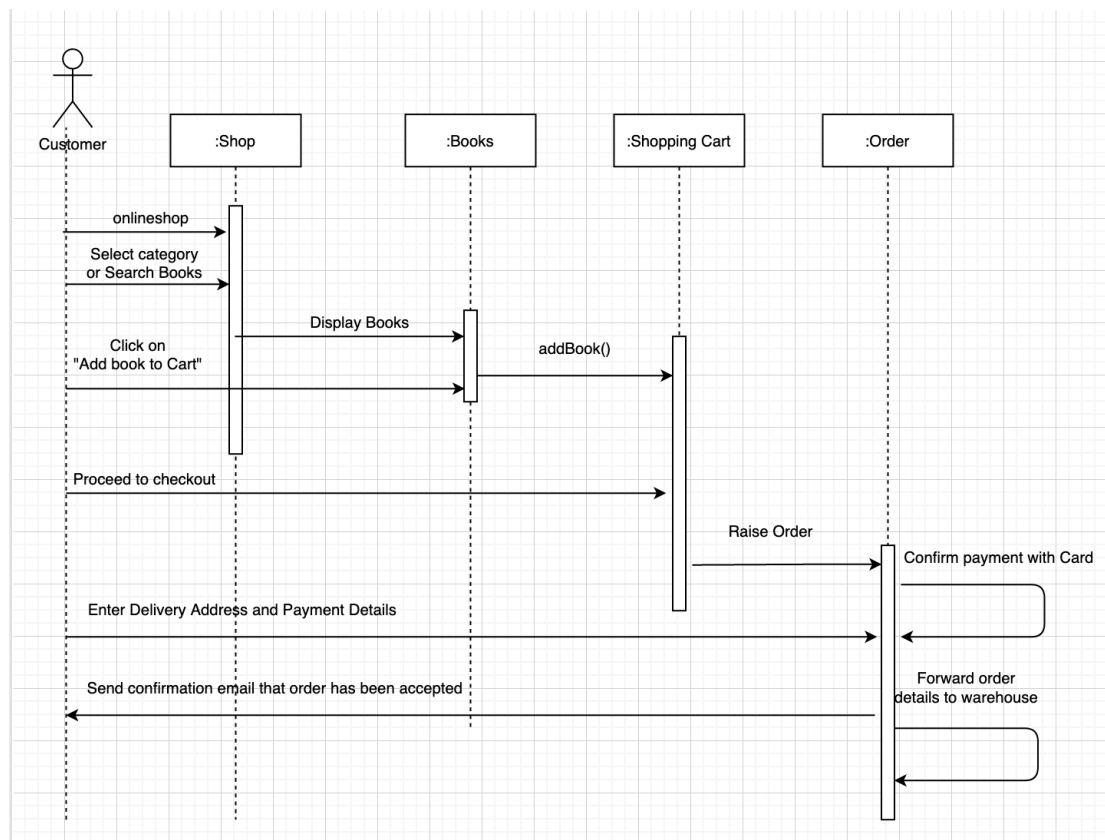
Εικόνα 27. Διάγραμμα Συνεργασίας.



Διάγραμμα σειράς

Ένα διάγραμμα σειράς βοηθάει στη μοντελοποίηση του χρονισμού των συνεργασιών με σαφή τρόπο. Η παρακάτω εικόνα δείχνει ένα διάγραμμα σειράς που μοντελοποιεί την ακολουθία αλληλεπιδράσεων που συμβαίνουν όταν ένας Customer επισκέπτεται τη σελίδα του ηλεκτρονικού καταστήματος. Η διάστικτη γραμμή, που εκτείνεται προς τα κάτω από το ορθογώνιο ενός αντικειμένου είναι η γραμμή ζωής του αντικειμένου, που αντιπροσωπεύει την πρόοδο στον χρόνο. Οι ενέργειες συμβαίνουν κατά μήκος της γραμμής ζωής του αντικειμένου, που αντιπροσωπεύει την πρόοδο στο χρόνο. Οι ενέργειες συμβαίνουν κατά μήκος της γραμμής ζωής του αντικειμένου σε χρονολογική σειρά από πάνω προς τα κάτω. Μια ενέργεια κοντά στην κορυφή συμβαίνει πριν από κάποια που είναι στο κάτω μέρος. Ένα συμπαγές βέλος με γεμισμένη αιχμή εκτείνεται από το αντικείμενο αποστολέα προς το αντικείμενο παραλήπτη και αντιπροσωπεύει ένα μήνυμα ανάμεσα στα δυο αντικείμενα. Η αιχμή του βέλους δείχνει σε μια δραστηριοποίηση στη γραμμή ζωής του αντικειμένου παραλήπτη. Μια δραστηριοποίηση που αντιπροσωπεύεται από ένα λεπτό κατακόρυφο ορθογώνιο υποδεικνύει ότι εκτελείται ένα αντικείμενο. Όταν ένα αντικείμενο επιστρέφει τον έλεγχο, ένα μήνυμα επιστροφής, που αντιπροσωπεύεται σαν μια διακεκομμένη γραμμή με απλή αιχμή βέλους. Παρακάτω βλέπουμε τη λεπτομερή έκδοση του διαγράμματος σειράς. Η ακολουθία των μηνυμάτων αρχίζει με τον Customer να επιλέγει μια κατηγορία βιβλίων να μεταβαίνει στη σελίδα με τους τίτλους των βιβλίων, να προσθέτει ένα βιβλίο στην shopping cart και να προχωράει στην παραγγελία. Επίσης ο χρήστης, θα μπορούσε κατευθείαν να μεταβεί στην shopping card ή και κατευθείαν να προσθέσει τα στοιχεία της παραγγελίας του.

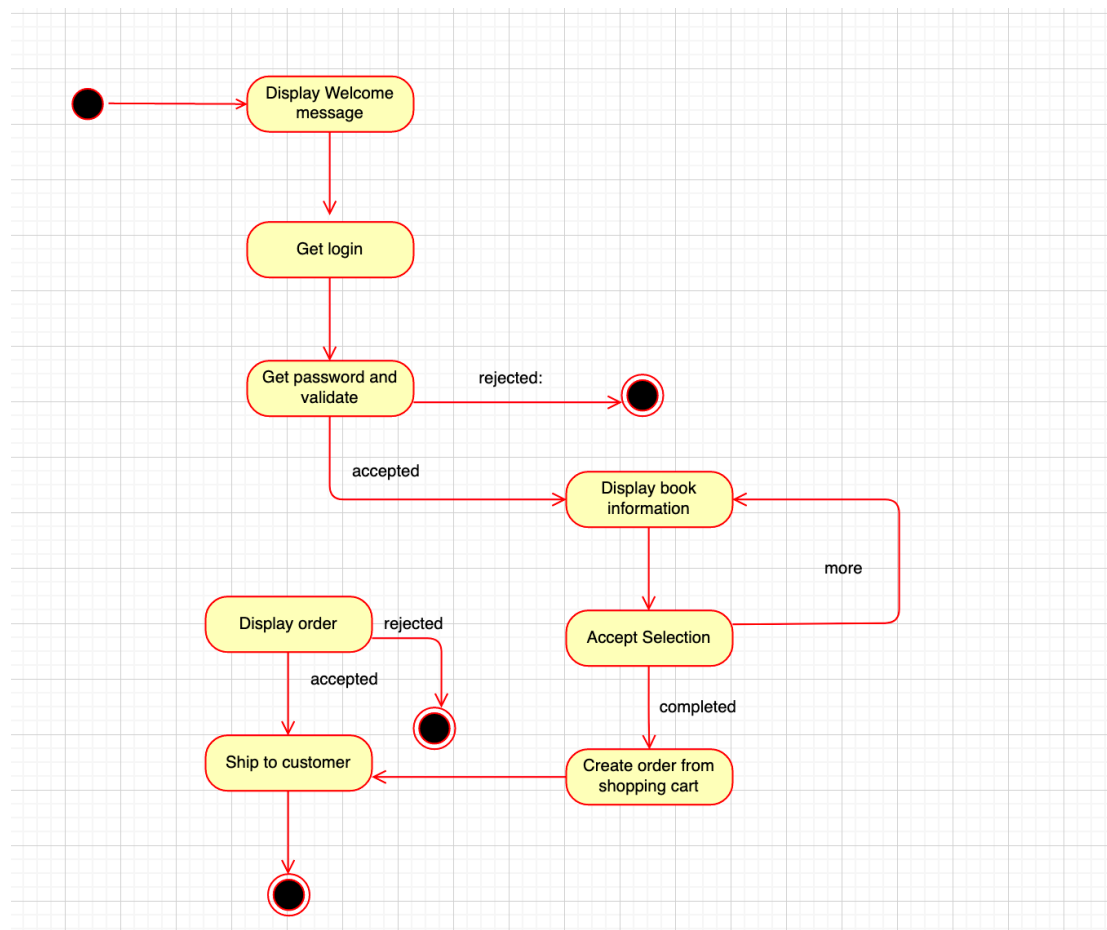
Εικόνα 28. Διάγραμμα Σειράς.



Διάγραμμα δραστηριοτήτων

Το διάγραμμα δραστηριοτήτων μοντελοποιεί τη δραστηριότητα ενός αντικειμένου, τη ροή λειτουργίας ενός αντικειμένου, δηλαδή την ακολουθία συμβάντων κατά την εκτέλεση ενός προγράμματος. Ένα διάγραμμα δραστηριότητας μοντελοποιεί τις ενέργειες που θα εκτελέσει το αντικείμενο και καθορίζει τη σειρά, με την οποία το αντικείμενο θα εκτελέσει αυτές τις ενέργειες. Στο συγκεκριμένο παράδειγμα, παρατηρούμε ότι ο χρήστης θα βρεθεί αρχικά σε μια Welcome Page, και στη συνέχεια για να προχωρήσει θα του ζητηθεί να συνδεθεί στο σύστημα. Αφού το σύστημα κάνει έλεγχο στα στοιχεία του, υπάρχουν δυο επιλογές: Αυτή του rejected και αυτή του accepted. Αφού συνδεθεί στο σύστημα, θα μπορεί να δει τους διαθέσιμους τίτλους, να προσθέσει κάποιο τίτλο στο καλάθι και να ξεκινήσει η διαδικασία της παραγγελίας.

Εικόνα 29. Διάγραμμα Δραστηριοτήτων

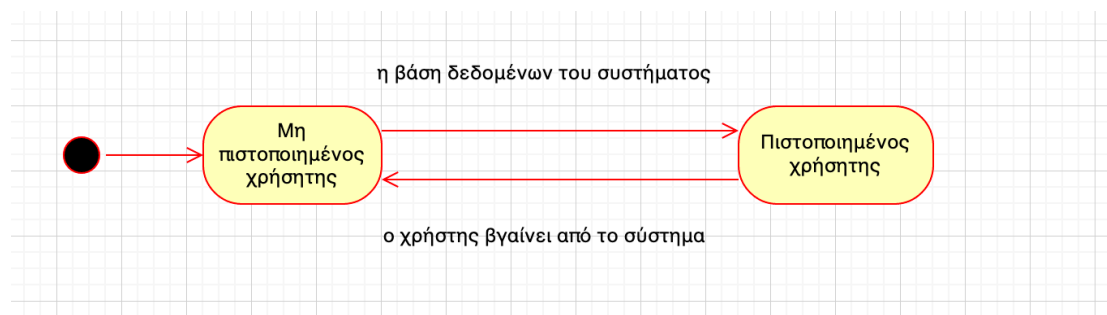


Διάγραμμα καταστάσεων

Στο βιβλίο “Java How To Program (Early Objects)”, [1] οι Paul Deitel – Harvey Deitel αναλύουν τα διαγράμματα καταστάσεων μηχανής τα οποία μοντελοποιούν τους τρόπους με τους οποίους ένα αντικείμενο αλλάζει κατάσταση. Η κατάσταση ενός αντικειμένου υποδεικνύεται από τις τιμές όλων των χαρακτηριστικών του σε μια δεδομένη στιγμή. Όταν ένα αντικείμενο αλλάζει κατάσταση, τότε μπορεί να συμπεριφέρεται διαφορετικά στο σύστημα.

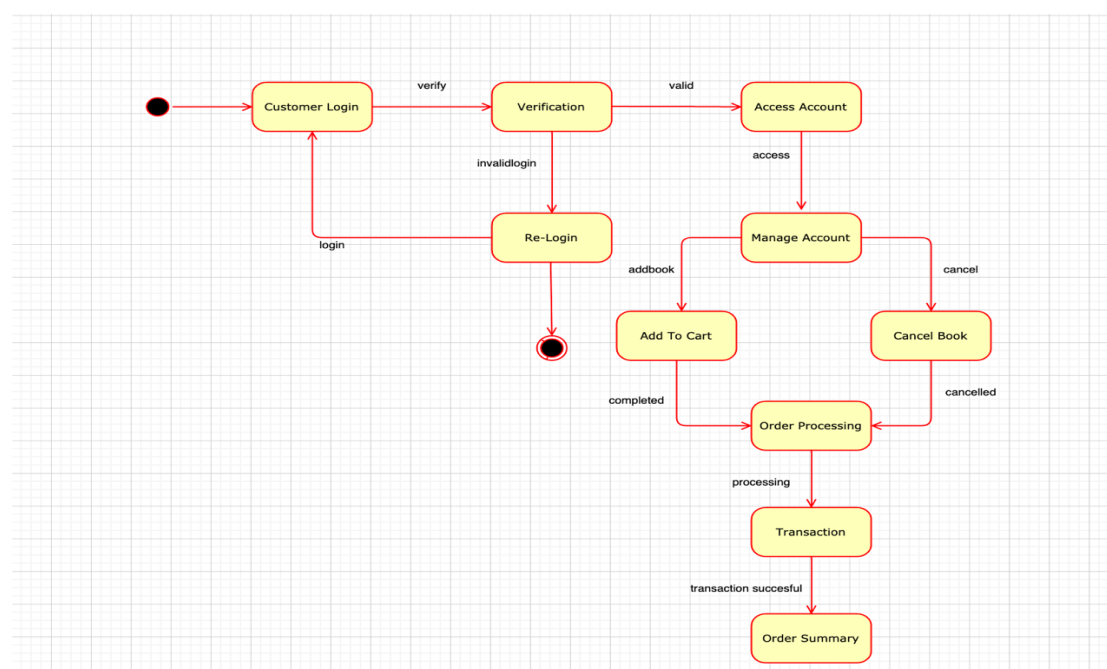
Για παράδειγμα, μετά την εισαγωγή των στοιχείων του χρήστη κατά τη διάρκεια της σύνδεσης, το σύστημα μεταπηδά από την κατάσταση «μη πιστοποιημένος χρήστης στην κατάσταση «πιστοποιημένος χρήστης», οπότε το σύστημα επιτρέπει στο χρήστη να συνεχίσει, πραγματοποιώντας τις αγορές του. Από το έγγραφο προδιαγραφών, η βάση δεδομένων πιστοποιεί ένα χρήστη συγκρίνοντας το username και το password που εισήγαγε με αυτά που είναι ήδη καταχωρημένα στη βάση. Εάν ο χρήστης εισάγει ένα έγκυρο username και password το σύστημα πηγαίνει στην κατάσταση «Πιστοποιημένος χρήστης». Όταν ο χρήστης εγκαταλείψει το σύστημα, το αντικείμενο επιστρέφει στην κατάσταση «Μη πιστοποιημένος χρήστης».

Εικόνα 30. Διάγραμμα Καταστάσεων



Στο παρακάτω σχήμα βλέπουμε το διάγραμμα καταστάσεων μηχανής. Τα βέλη δηλώνουν μεταβιβάσεις μεταξύ καταστάσεων. Ένα αντικείμενο μπορεί να πάει από μια κατάσταση σε μια άλλη αποκρινόμενο σε διάφορα συμβάντα που συμβαίνουν στο σύστημα. Το όνομα ή η περιγραφή του συμβάντος που προκαλεί μια μετάβαση γράφεται δίπλα στη γραμμή που αντιστοιχεί σε αυτή τη μετάβαση. Υπενθυμίζουμε ότι από το έγγραφο προδιαγραφών ότι η βάση δεδομένων πιστοποιεί ένα χρήστη συγκρίνοντας το username και το password που εισήγαγε με αυτά που είναι ήδη καταχωρημένα στη βάση.

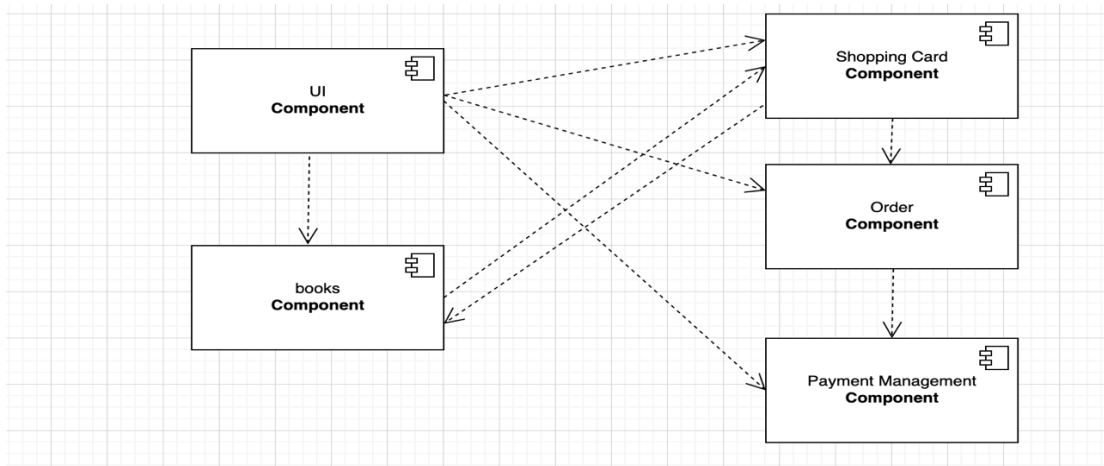
Εικόνα 31. Διάγραμμα Καταστάσεων.



Διάγραμμα Εξαρτημάτων

Το διάγραμμα εξαρτημάτων είναι ένα διάγραμμα δομής υλοποίησης που χρησιμοποιείται για να μοντελοποιήσει πηγαίο κώδικα, βάσεις δεδομένων, δυναμικά προσαρμοζόμενα συστήματα κ.α. Περιλαμβάνει κυρίως εξαρτήματα και διεπαφές (interfaces).

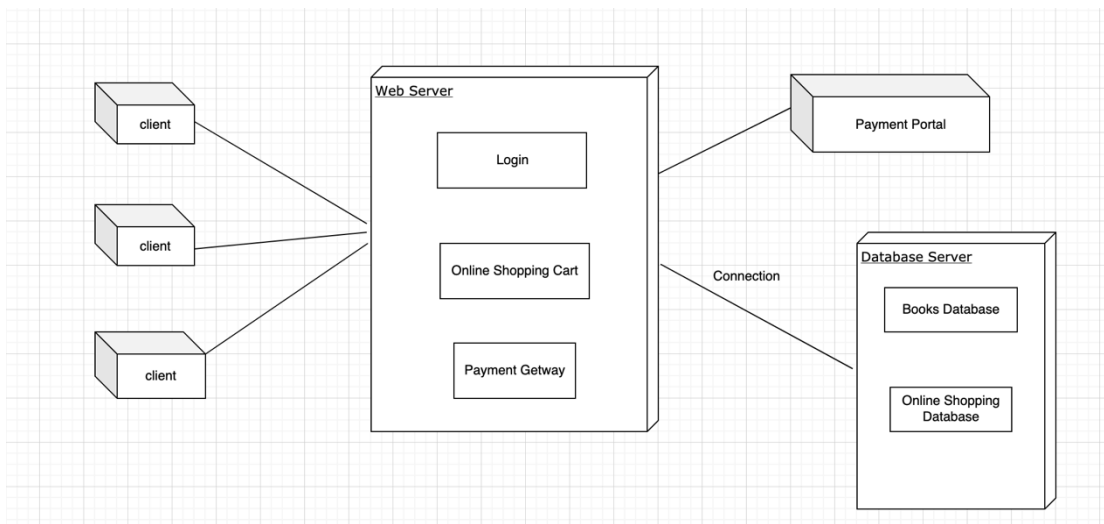
Εικόνα 32. Διάγραμμα Εξαρτημάτων.



Διάγραμμα διανομής

Τέλος, το διάγραμμα διανομής αναπαριστά τα φυσικά εξαρτήματα μια εφαρμογής.

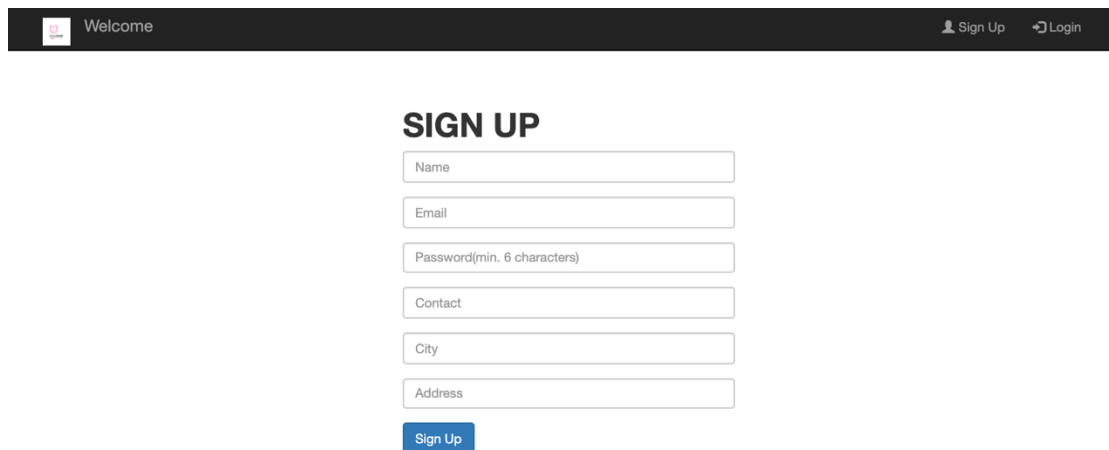
Εικόνα 33. Διάγραμμα Διανομής.



Ηλεκτρονικό Βιβλιοπωλείο

Ενδεικτικά βλέπουμε κάποιες εικόνες από το ηλεκτρονικό βιβλιοπωλείο. Στην παρακάτω εικόνα ο χρήστης έχει τη δυνατότητα να πραγματοποιήσει την εγγραφή του.

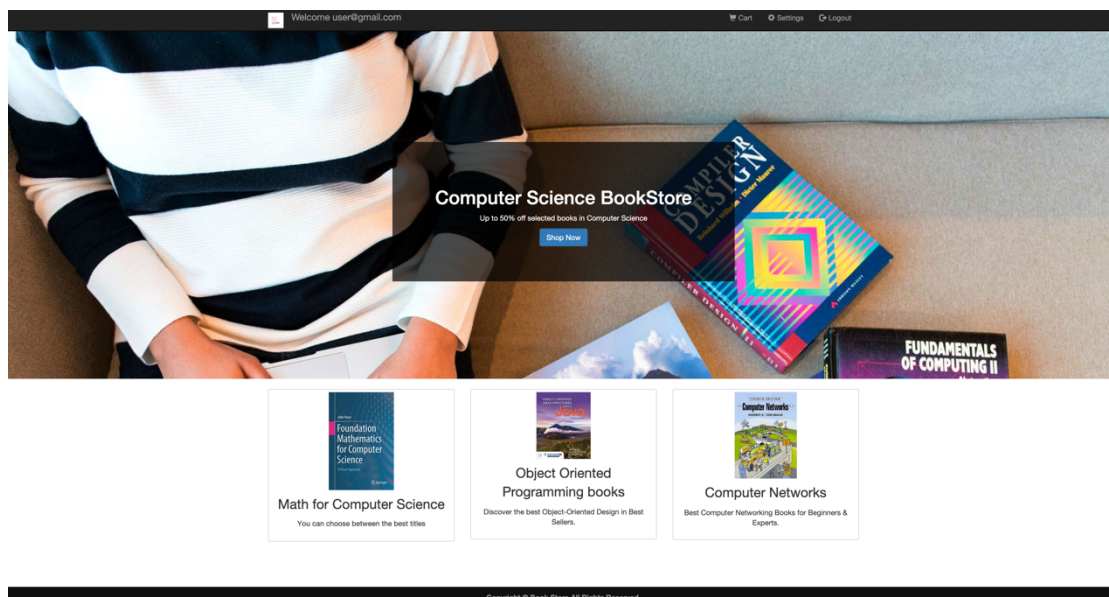
Εικόνα 34. Εγγραφή χρήστη στο ηλεκτρονικό βιβλιοπωλείο.



The screenshot shows a 'SIGN UP' form on a website. The form is centered on a white background with a dark header. The header contains a 'Welcome' message and 'Sign Up' and 'Login' links. The form consists of several input fields: 'Name', 'Email', 'Password (min. 6 characters)', 'Contact', 'City', and 'Address'. Below the fields is a blue 'Sign Up' button.

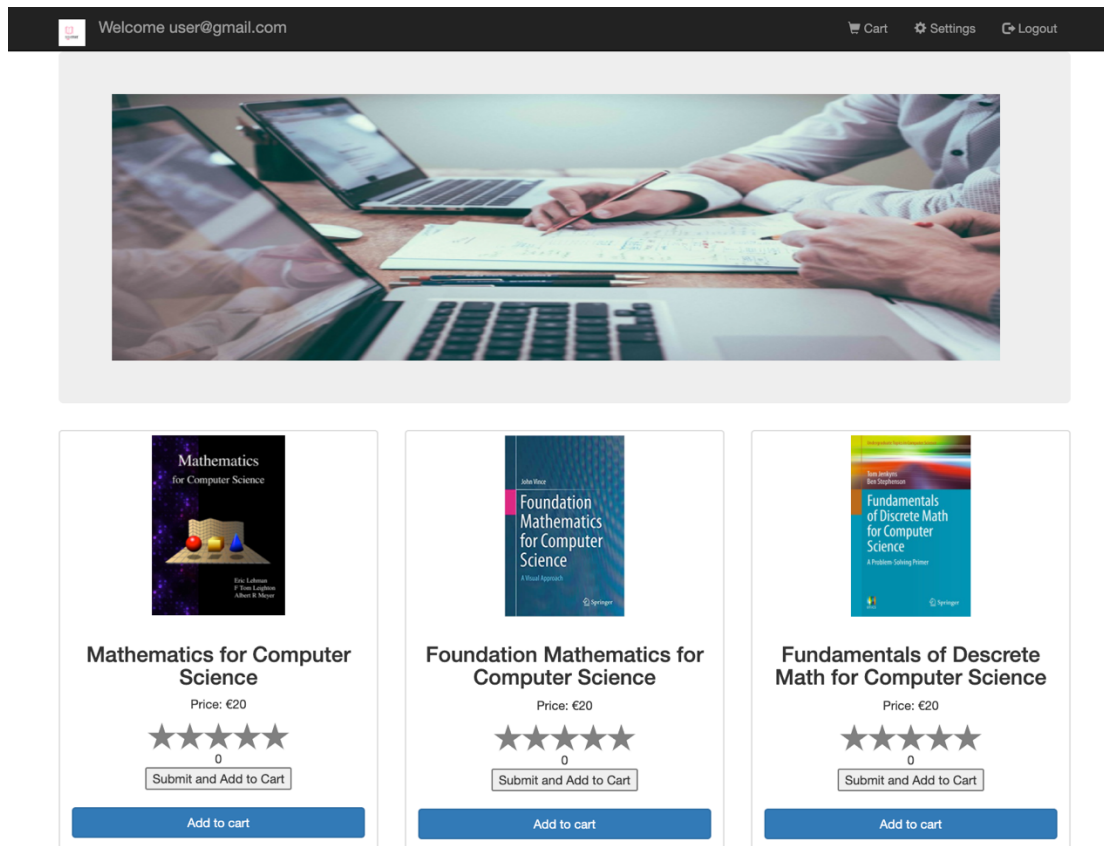
Παρακάτω βλέπουμε στιγμιότυπο από την αρχική σελίδα του ηλεκτρονικού βιβλιοπωλείου και αφού ο χρήστης έχει πραγματοποιήσει την εγγραφή του.

Εικόνα 35. Αρχική σελίδα ηλεκτρονικού βιβλιοπωλείου.



Ακολουθεί στιγμιότυπο του ηλεκτρονικού βιβλιοπωλείου που φαίνονται ενδεικτικά τίτλοι βιβλίων στον τομέα Computer Science.

Εικόνα 36. Ενδεικτικοί τίτλοι βιβλίων για προσθήκη στο καλάθι του χρήστη.



Στην εικόνα 36 βλέπουμε τη δυνατότητα που έχει ο χρήστης να μάθει λεπτομέρειες για το βιβλίο της επιλογής του κάνοντας κλικ πάνω σε αυτό.

Εικόνα 37. Λεπτομέρειες για το συγκεκριμένο βιβλίο.



Ακολουθεί εικόνα από το καλάθι του Customer.

Εικόνα 38. Καλάθι του χρήστη (Shopping Cart).

Welcome user@gmail.com Cart Settings Logout			
Item Number	Item Name	Price	
1	Fundamentals of Discrete Math for Computer Science	20	Remove
2	Java-Object Oriented Programming	20	Remove
Total		€ 40/-	Confirm Order

[Recommending Books](#)

Παρακάτω βλέπουμε εξατομικευμένες προτάσεις για αγορές βιβλίων.

Εικόνα 39. Εξατομικευμένες συστάσεις βιβλίων.

What other items do customers buy after viewing this item?

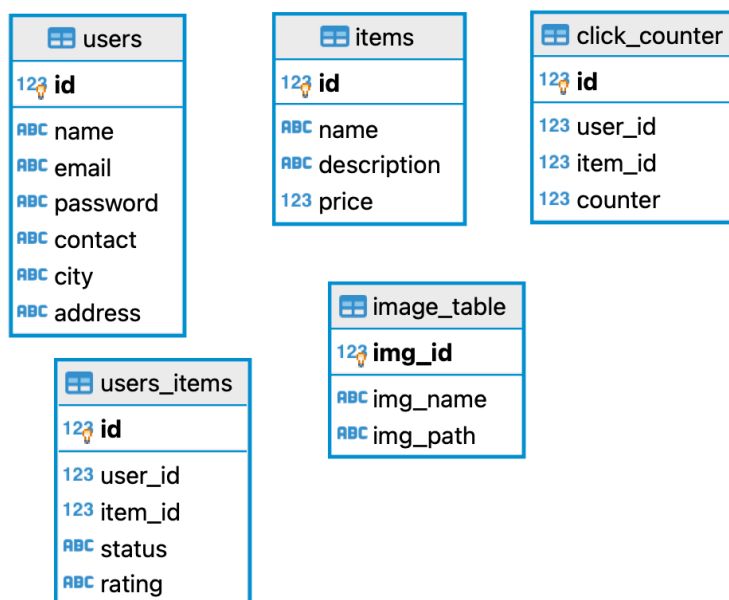
Ακολουθεί εικόνα από εξατομικευμένες συστάσεις για αγορές βιβλίων όταν το σύστημα ακόμη δεν έχει συλλέξει αρκετές πληροφορίες για το χρήστη.

Εικόνα 40. Εξατομικευμένες συστάσεις βιβλίων.

Frequently bought together

Στο ηλεκτρονικό κατάστημα έγινε χρήση του συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων MySQL. Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα τρέχει έναν εξυπηρετητή παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Παρακάτω φαίνεται η δομή των πινάκων της βάσης δεδομένων.

Εικόνα 41. Πίνακες βάσης δεδομένων e-bookstore.



4.4 Εγχειρίδιο Χρήστη

Ένα τοπικό βιβλιοπωλείο προτίθεται να δημιουργήσει ένα ηλεκτρονικό κατάστημα πώλησης βιβλίων, που θα επιτρέπει στους χρήστες να κάνουν τις αγορές τους ηλεκτρονικά. Κάθε χρήστης θα μπορεί να δημιουργεί ένα λογαριασμό στο διαδικτυακό κατάστημα. Οι χρήστες θα μπορούν να επιλέξουν κατηγορίες βιβλίων που τους ενδιαφέρουν. Στο συγκεκριμένο σύστημα, ο νέος χρήστης θα έχει την ευκαιρία αφού περιηγηθεί στο e-shop, να κάνει εγγραφή. Ο customer, θα έχει τη δυνατότητα να προβάλλει λεπτομέρειες για κάθε τίτλο ξεχωριστά, πχ Εκδότης, ISBN, Τίτλος, Έτος, Σελίδες. Ο customer, αφού επιλέξει τον τίτλο ή τους τίτλους των βιβλίων που τον ενδιαφέρουν, θα έχει τη δυνατότητα να κάνει εισαγωγή στο καλάθι αγορών, να προβάλλει τη λίστα παραγγελιών του και να ολοκληρώσει την αγορά του. Το ηλεκτρονικό κατάστημα διαθέτει ένα σύστημα συστάσεων (Recommendation System) και παρέχει στους χρήστες εξατομικευμένες προτάσεις με βάση τις προτιμήσεις τους και το ιστορικό περιήγησής τους.

5 Συμπεράσματα

Η εργασία αυτή μελετά αλγορίθμους για τη δημιουργία συστήματος που θα παρέχει στους χρήστες εξατομικευμένες προτάσεις για αγορά βιβλίων. Αυτοί οι αλγόριθμοι βρίσκουν εφαρμογή σε ένα ηλεκτρονικό βιβλιοπωλείο (e-bookstore). Για το σκοπό αυτό αρχικά έγινε αναφορά στο Collaborative Filtering, μια τεχνική που χρησιμοποιείται από συστήματα προτάσεων, στη συνέχεια μελετήσαμε τον υπολογισμό της Ευκλείδειας απόστασης η οποία μας δίνει το βαθμό ομοιότητας (similarity cost), είτε αυτός αφορά τους χρήστες είτε τους τίτλους βιβλίων. Οι δύο γενικές προσεγγίσεις για την επίλυση του προβλήματος αφορούσαν τις συστάσεις βιβλίων και τη μελέτη αλγορίθμων σχετικά με ποια προϊόντα είναι παρόμοια μεταξύ τους.

Βιβλιογραφία - Αναφορές

- [1] Deitel Harvey M., Deitel Paul J. (2015), Java How To Program (Early Objects), 10th Edition, ISBN: 9789605126810.
- [2] Toby, Segaran (2007), Programming Collective Intelligence: Building Smart Web 2.0 Applications, ISBN-10: 0-596-52932-5.
- [3] Πανεπιστήμιο Πειραιά - Τμήμα Πληροφορικής - Πρόγραμμα Μεταπτυχιακών Σπουδών «Πληροφορική» - Σημειώσεις μαθήματος «Τεχνολογίας Λογισμικού».
- [4] Machine learning | artificial intelligence. Encyclopedia Britannica.
- [5] Wernick, Yang, Brankov, Yourganov and Strother, Machine Learning in Medical Imaging, IEEE Signal Processing Magazine, vol.27, no4, July 2010, pp.25-38.
- [6] Ron Kohavi; Foster Provost(1998). Glossary of Terms. Machine Learning 30:271-274.
- [7] Smith, Karl (2013), Precalculus: A Functional Approach to Graphing and Problem Solving, Jones & Bartlett Publishers, p.8, ISBN 978-0-7637-5177-7.
- [8] KIT405 Week 10 University of Tasmania - Course "Programming for Intelligent Web Services and Applications (KIT405) Academic Year 2020/21 Uploaded by Hai Le.
- [9] https://en.wikipedia.org/wiki/Euclidean_distance#/media/File:Euclidean_distance_2d.svg
- [10] Α. Βακάλη· Η. Γιαννόπουλος· Ν. Ιωαννίδης· Χ. Κοίλιας· Κ. Μαλάμας· Ι. Μανωλόπουλος· Π. Πολίτης (2011). Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (1B έκδοση). Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων - Παιδαγωγικό Ινστιτούτο. σελίδες 3–5. ISBN 960-06-1408-3.