ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Πρόγραμμα Προπτυχιακών Σπουδών
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αυτοματοποιημένο Σύστημα Συστάσεων για Χώρους Διασκέδασης

Μαντής Νικόλαος
Α.Μ.: 40568

Βασίλειος Ζγαντζούρης

Α.Μ.: 45408

Εισηγητής: Γεώργιος Πρεζεράκος, Καθηγητής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αυτοματοποιημένο Σύστημα Συστάσεων για Χώρους Διασκέδασης

Μαντής Νικόλαος
Α.Μ.: 40568

Βασίλειος Ζγαντζούρης

Α.Μ.: 45408

Εισηγητής:

Γεώργιος Πρεζεράκος, Καθηγητής

Georgios Prezerakos
Digitally signed by Georgios Prezerakos
Date: 2021.07.28 15:04:57 +03'00'

Εξεταστική Επιτροπή:

Παναγιώτης Γιαννακόπουλος, Καθηγητής
Νικόλαος Ζάχαρης, Καθηγητής

Panagiotis Giannakopoulos
Digitally signed by Panagiotis Giannakopoulos
Date: 2021.07.28 20:10:42 +03'00'

Digitally signed by Nikolaos Zacharis
Date: 2021.07.28 18:11:39 +03'00'

Ημερομηνία εξέτασης: 21/7/2021

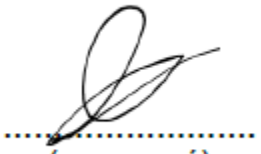
ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι **Μαντής Νικόλαος του Κωνσταντίνου με αριθμό μητρώου 40568 και Βασίλειος Ζγαντζούρης του Νικολάου με αριθμό μητρώου 45408** φοιτητές του Προγράμματος Προπτυχιακών Σπουδών **Μηχανικών Πληροφορικής και Υπολογιστών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών** της Σχολής **Μηχανικών** του Πανεπιστημίου Δυτικής Αττικής, δηλώνουμε ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

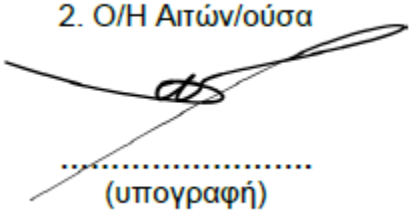
Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι **1/9/21** και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

1.Ο/Η Αιτών/ούσα



.....
(υπογραφή)

2. Ο/Η Αιτών/ούσα



.....
(υπογραφή)

ΠΕΡΙΛΗΨΗ

Στόχος της πτυχιακής εργασίας είναι η μελέτη συστημάτων συστάσεων χρησιμοποιώντας νέες, διαδεδομένες τεχνολογίες ανάπτυξης ιστοσελίδων. Στα πλαίσια της εργασίας έχει αναπτυχθεί μια εφαρμογή με τίτλο Magellan με δομικά στοιχεία τις γλώσσες προγραμματισμού Angular (JavaScript Framework), Spring (Java Framework), Flask (Python Framework). Το Magellan είναι μια πλατφόρμα που με βάση το προφίλ κάθε χρήστη, το οποίο δημιουργείται με στοιχεία από το ιστορικό αναζητήσεων του, προσπαθεί να προβλέψει ποια σημεία ψυχαγωγίας/διασκέδασης βρίσκονται στο φάσμα της αρεσκείας του και να προτείνει τα πιο πιθανά.

Παρακάτω θα αναλυθεί η δομή της εφαρμογής καθώς και οι αλγόριθμοι επεξεργασίας δεδομένων του χρήστη που μας δίνουν τα αποτελέσματα των συστάσεων.

ABSTRACT

The present thesis concerns the study of recommender systems using new, widespread web development technologies. As part of the study, an application called Magellan has been developed using development tools such as Angular (JavaScript Framework), Spring (Java Framework), Flask (Python Framework). Magellan is a platform based on each user's profile, created with data from their search history, trying to predict which entertainment places are in the range of their liking and recommending the most likely ones.

In the sections below, we describe the structure of the application as well as the recommendation algorithms used for processing the user’s data that create the recommendations.

Περιεχόμενα

_

1. Εισαγωγή....12
1.1 Ορισμός Συστημάτων Συστάσεων....13
1.2 Κατηγορίες Συστημάτων Σύστασης....15
1.2.1 Φιλτράρισμα με βάση το περιεχόμενο – Content Based....16
1.2.2 Συνεργατικό Φιλτράρισμα – Collaborative Filtering....17
1.2.3 Υποκατηγορίες Συνεργατικού Φιλτραρίσματος....18
1.2.4 Συστήματα που βασίζονται στην Γνώση - Knowledge Based.22
1.2.5 Υβριδικά Συστήματα – Hybrid Systems....23
1.3 Προβλήματα και Προκλήσεις Συστημάτων Συστάσεων....25
2. Η εφαρμογή Magellan....27
2.1 Δομικά Στοιχεία....27
2.2 Περιγραφή....31
3. Ανάλυση Συστήματος Συστάσεων....38
3.1 Ανάλυση Συνεργατικού Φιλτραρίσματος....38
3.2 Ανάλυση Φιλτραρίσματος με βάση το Περιεχόμενο....47
4. Επίλογος....54
Παράρτημα Α’....56
Βιβλιογραφία....57

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1: Συσχέτιση Pearson 19
Εικόνα 1. 2: Παράδειγμα Pearson..... 19
Εικόνα 1. 3: Παράδειγμα Συνημιτονικής Ομοιότητας..... 20
Εικόνα 2.1.1: Δομή Εφαρμογής……..29
Εικόνα 2.1. 2: Διάγραμμα Ροής Σύνδεσης 30
Εικόνα 2.1. 3: Διάγραμμα Ροής Αναζήτησης..... 31
Εικόνα 2.2. 1: Σελίδα Εισόδου……32
Εικόνα 2.2. 2: Σελίδα Εγγραφής..... 33
Εικόνα 2.2. 3: Σελίδα Προτιμήσεων Χρήστη 34
Εικόνα 2.2. 4: Αρχική Σελίδα..... 35
Εικόνα 2.2. 5: Σελίδα Σύνθετης Αναζήτησης..... 36
Εικόνα 2.2. 6: Σελίδα Συστάσεων 37
Εικόνα 3.1. 1: Εισαγωγή Δεδομένων σε Python (1)……38
Εικόνα 3.1. 2: Εισαγωγή Δεδομένων σε Python (2)..... 38
Εικόνα 3.1. 3: Συγχώνευση πινάκων 40
Εικόνα 3.1. 4: Αποτελέσματα Συγχώνευσης Πινάκων..... 40
Εικόνα 3.1. 5: Ταξινόμηση Αποτελεσμάτων..... 40
Εικόνα 3.1. 6: Ιστόγραμμα (Πλήθος Χώρων – Αξιολογήσεις)..... 41
Εικόνα 3.1. 7: Ιστόγραμμα (Πλήθος Χώρων – Πλήθος Αξιολογήσεων)..... 42
Εικόνα 3.1. 8: Διάγραμμα Διασποράς (Πλήθος Αξιολογήσεων – Μέσος όρος Αξιολογήσεων)..... 43
Εικόνα 3.1. 9: Εντολή εμφάνισης πίνακα χρηστών / χώρων..... 44
Εικόνα 3.1. 10: Αξιολογήσεις από «Melia Athens» 44
Εικόνα 3.1. 11: Υπολογισμός Βαθμού Συσχέτισης 45
Εικόνα 3.1. 12*:* Προσθήκη πλήθους αξιολογήσεων 45
Εικόνα 3.2. 1: Πιο χρησιμοποιημένες λέξεις (1)……48
Εικόνα 3.2. 2: Πιο χρησιμοποιημένες λέξεις (2) 48
Εικόνα 3.2. 3: Αφαίρεση Συνδετικών Λέξεων..... 49
Εικόνα 3.2. 4: Πιο συχνά ζεύγη λέξεων πριν την αφαίρεση των stop-words..... 49
Εικόνα 3.2. 5: : Πιο συχνά ζεύγη λέξεων μετά την αφαίρεση των stop-words.... 50

Εικόνα 3.2. 6: Συνάρτηση clean_text() 51
Εικόνα 3.2. 7: Δημιουργία πίνακα TF-IDF 52
Εικόνα 3.2. 8: Δημιουργία Συστάσεων (Content-Based) 53

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 3. 1: Χώροι Διασκέδασης..... 39
Πίνακας 3. 2: Πίνακας Αξιολογήσεων..... 39
Πίνακας 3. 3: Ταξινομημένοι χώροι διασκέδασης..... 41
Πίνακας 3. 4: Πίνακας Χρηστών / Χώρων Διασκέδασης 44
Πίνακας 3. 5: : Αξιολογήσεις από «Melia Athens»..... 44
Πίνακας 3. 6: Συσχέτιση χώρων διασκέδασης με «Melia Athens» 45
Πίνακας 3. 7: Τελική συσχέτιση αποτελεσμάτων..... 46
Πίνακας 3. 8: Πίνακας Συσχέτισης Content-Based..... 53

1. Εισαγωγή

Πλέον ο όγκος των διαθέσιμων ψηφιακών πληροφοριών στο διαδίκτυο είναι πελώριος και ο ρυθμός αύξησης του αυξάνεται καθημερινά. Ταυτόχρονα ο αριθμός των χρηστών δημιουργεί πιθανές προκλήσεις στην διαχείριση πόρων καθώς και στην αποτελεσματικότητα του Διαδικτύου όσον αναφορά την διαδικασία της αναζήτησης. Έτσι αυξήθηκε η ζήτηση για τα συστήματα συστάσεων - ΣΣ (Recommender Systems).

Τα συστήματα συστάσεων παρέχουν εξατομικευμένες προτάσεις στον χρήστη, σχετικά με το αντικείμενο ενδιαφέροντος. Είναι εφαρμογές σχεδιασμένες, με ειδικούς αλγορίθμους, που στοχεύουν στην πρόβλεψη των προτιμήσεων του εκάστοτε χρήστη. Καταφέρνουν έτσι να φιλτράρουν αυτόν τον όγκο πληροφοριών με αποτέλεσμα την καλύτερη ποιότητα εμπειρίας του χρήστη αλλά και την μείωση του κόστους συναλλαγής με τον πάροχο υπηρεσιών. Προφανώς τα πλεονεκτήματα των ΣΣ δεν σταματάνε εκεί αλλά θα αναλυθούν στην συνέχεια.

Η εφαρμογή Magellan ασχολείται με τα συστήματα συστάσεων - ΣΣ (Recommender Systems) και την ανάλυση αυτών. Πρακτικά σκοπός είναι να εμφανίζει στον εκάστοτε χρήστη εξατομικευμένες πληροφορίες ανάλογα με τις προτιμήσεις του σχετικά με τα μέρη ψυχαγωγίας που θα ήθελε να επισκεφτεί. Ο κάθε χρήστης έχει την δυνατότητα αναζήτησης τοποθεσίας με βάση τα επιθυμητά κριτήρια όπως απόσταση, ωράριο, εύρος τιμών κτλ. καθώς και να επιλέξει από τις αγαπημένες του τοποθεσίες με σκοπό το σύστημα να διαθέτει όσο δυνατόν περισσότερα δεδομένα .

Στη συνέχεια, αφού αποκλείσουμε τα δεδομένα που δε πληρούν τις προϋποθέσεις του χρήστη, χαρτογραφούμε (μεταφράζουμε) τα υπόλοιπα με αριθμητικές τιμές χρησιμοποιώντας όπου χρειάζεται κάποιον συντελεστή βαρύτητας και τα ομαδοποιούμε. Τα αποτελέσματα της ομαδοποίησης περνάνε έναν δεύτερο συγκριτικό έλεγχο σύμφωνα με τις προβλέψεις αρέσκειας του χρήστη και τέλος εμφανίζονται στην οθόνη του.

Στο πρώτο κεφάλαιο ορίζονται τα συστήματα σύστασης και κάνουμε μια εισαγωγή στις μεθόδους σύστασης που χρησιμοποιούνται αλλά και στα προβλήματα που αντιμετωπίζουν.

Στο δεύτερο κεφάλαιο πραγματοποιούμε μια εισαγωγή στην αρχιτεκτονική της εφαρμογής για να καταλάβει ο αναγνώστης πως δουλεύει αυτό που φτιάξαμε και τις τεχνολογίες που αποφασίσαμε να χρησιμοποιήσουμε.

Στο τρίτο κεφάλαιο περιγράφεται αναλυτικά η διαδικασία με την οποία θα γίνει η εκπαίδευση του συστήματος αλλά και πως δημιουργείται η "σύσταση".

1.1 Ορισμός Συστημάτων Συστάσεων

Με την δημιουργία του διαδικτύου καθώς και το πλεονέκτημα της ελεύθερης πρόσβασης σε αυτό δεν άργησε να γίνει μια πηγή ενός πελώριου όγκου πληροφοριών. Οι απλοί χρήστες για να αποφύγουν τον καταιγισμό πληροφοριών είχαν (και έχουν ακόμα) ως κύριο εργαλείο της μηχανές αναζήτησης (Google, AltaVista, Yahoo κτλ.), οι οποίες λύνουν εν μέρει αυτό το πρόβλημα. Αυτό που συνέχιζε να λείπει ακόμα ήταν η εξατομίκευση της πληροφορίας. Τα συστήματα συστάσεων αναδύθηκαν ως ανεξάρτητος ερευνητικός χώρος στα μέσα της δεκαετίας του 1990 όταν ξεκίνησαν οι ερευνητές να εστιάζουν σε προβλήματα προτάσεων που βασίζονται ρητά στη δομή των αξιολογήσεων. Από τότε, υπάρχουν πολλές ερευνητικές εργασίες που διεξήχθησαν τόσο στη βιομηχανία, όσο και στον ακαδημαϊκό χώρο για την ανάπτυξη νέων προσεγγίσεων στα συστήματα συστάσεων. Το ενδιαφέρον παραμένει υψηλό γιατί αποτελεί έναν πλούσιο σε προβλήματα ερευνητικό χώρο και λόγω της ευρέως αποδεκτής πρακτικής σε εφαρμογές που έχουν αναπτυχθεί για να βοηθήσουν τους χρήστες να αντιμετωπίσουν υπερφόρτωση πληροφοριών [Papagelis et al., 2005].Τα αποτελέσματα της έρευνας Digital 2019 από την HootSuite, αποκαλύπτουν ότι ένας άνθρωπος χρησιμοποιεί κατά μέσω όρο 6 ώρες και 42 λεπτά την ημέρα το διαδίκτυο. Ο χρόνος αυτός που ο χρήστης είναι συνδεδεμένος δημιουργεί το ψηφιακό του αποτύπωμα στο οποίο εμπεριέχονται δεδομένα της συμπεριφορικής δραστηριότητάς του που για αρκετό καιρό έμεναν ανεκμετάλλευτα. Από αυτά τα δεδομένα μπορούμε να εξάγουμε την πληροφορία για τις προτιμήσεις του χρήστη και την συμπεριφορά του προς κάποιο Α αντικείμενο, καθώς και πληροφορίες για ένα

αντικείμενο με βάση τις αλληλεπιδράσεις άλλων χρηστών με αυτό. Τα Συστήματα Συστάσεων λειτουργούν σαν φίλτρα πληροφορίας, τροφοδοτούμενα από τα δεδομένα του χρήστη αφού πρώτα έχουν δημιουργήσει το προφίλ του.

Τα πλεονεκτήματα των Συστημάτων Σύστασης όμως, δεν περιορίζονται μόνο στον χρήστη. Επιπρόσθετα, με την βελτίωση της ποιότητας εμπειρίας του χρήστη σε πολύπλοκα περιβάλλοντα πληροφοριών πρέπει να λάβουμε υπόψιν και το κόστος συναλλαγής με τον πάροχο υπηρεσιών. Αυτό επιτυγχάνεται με βάση την αποδοτικότητα του συστήματος. Οι εύστοχες προβλέψεις συνεπάγονται σε μειωμένο χρόνο αναζήτησης δεδομένων και πλοήγησης του χρήστη μειώνοντας έτσι την κίνηση συναλλαγών. Ο τομέας όμως που επωφελήθηκε περισσότερο και ταυτόχρονα είναι και αυτός ο οποίος ώθησε τα ΣΣ σε αυτήν την εκθετική ανάπτυξη τα τελευταία χρόνια είναι ο τομέας του ηλεκτρονικού εμπορίου (e-commerce).

Η ραγδαία ανάπτυξη της τεχνολογίας ανάγκασε την πληθώρα των επιχειρήσεων να προσαρμοστούν σε αυτή. Σαν αποτέλεσμα οι επιχειρήσεις που τα κατάφεραν, εξελίχθηκαν και συνεχίζουν να εξελίσσονται, δίνοντας μια νέα πνοή στον ανταγωνισμό. Με την πάροδο του χρόνου οι καταναλωτές εξοικειώθηκαν περισσότερο με τα νέα συστήματα, δημιουργώντας έτσι την ανάγκη ύπαρξης ηλεκτρονικών καταστημάτων (e-shop) από τις εμπορικές επιχειρήσεις. Πλέον, ειδικά με την πανδημία του Covid-19 και τα μέτρα, όπως την απαγόρευση κυκλοφορίας και όχι μόνο, σε πολλά σημεία του πλανήτη, οι επιχειρήσεις που ενεργούν μέσω του ηλεκτρονικού εμπορίου άρχισαν να εντάσσονται σε κυρίαρχη θέση. Πρωταρχικός σκοπός τους βέβαια, δεν είναι η επιβίωση αλλά η ανάπτυξη, η οποία με την σειρά της θα φέρει την βιωσιμότητα. Αυτό για να επιτευχθεί δεν αρκεί μόνο να προβούν στη προβολή των προϊόντων τους αλλά και οι επιχειρήσεις που επικεντρώνονται στο online κομμάτι τους, πέραν του φυσικού καταστήματος, χρειάζεται να απορροφήσουν τις νέες τεχνολογίες για την αύξηση των πωλήσεων, σε βαθμό που δεν ήταν δυνατόν πριν.

Οι καταναλωτές (χρήστες των ηλεκτρονικών καταστημάτων) από την άλλη, έρχονται σε επαφή με μια υπερμεγέθης βάση καταστημάτων και προϊόντων που ναι μεν διευκολύνει σε μεγάλο βαθμό την προσβασιμότητά προς την επιθυμητή αγορά, όμως δημιουργεί την ανάγκη της ευχρηστίας και της εύκολης αναζήτησης. Ο χρήστης που θα

επισκεφτεί ένα ηλεκτρονικό κατάστημα κατατάσσεται χονδρικά σε δυο κατηγορίες. Στην κατηγορία των χρηστών που επιθυμούν ένα συγκεκριμένο προϊόν και είναι έτοιμοι για την αγορά του και στην κατηγορία των χρηστών που κάνουν μια απλή περιήγηση που ίσως αποσκοπεί σε έρευνα αγοράς. Και στις δυο κατηγορίες, αλλά ειδικά στη δεύτερη, εάν είναι γνωστές οι προτιμήσεις του χρήστη με βάση το ιστορικό του, το ηλεκτρονικό κατάστημα έχει την δυνατότητα να προωθήσει προϊόντα με μεγάλη πιθανότητα αγοράς. Σε ένα ηλεκτρονικό κατάστημα, εμφανίζονται συχνά τα προϊόντα με την υψηλότερη δημοτικότητα. Με αυτόν τον τρόπο τα πιο δημοφιλή προϊόντα παραμένουν στην κορυφή των πωλήσεων και τα υπόλοιπα παραμένουν σχεδόν στάσιμα. Σε αρκετές περιπτώσεις όμως, το κατάλληλο προϊόν για τον εκάστοτε χρήστη δεν ανήκει στα πιο δημοφιλή. Οι στοχευμένες συστάσεις εξαφανίζουν αυτόν τον φαύλο κύκλο βασισμένο στην δημοτικότητα και εξασφαλίζουν έναν ικανοποιημένο επισκέπτη και περισσότερα κέρδη.

Η βασική λογική που ακολουθούν τα συστήματα συστάσεων είναι η σύγκριση μεταξύ χρηστών ή προϊόντων με σκοπό τον εντοπισμό ομοιοτήτων. Η σύγκριση μεταξύ χρηστών πραγματοποιείται, όπως είπαμε, με βάση τις προτιμήσεις τους και μεταξύ αντικειμένων (προϊόντων) μεταξύ των ιδιοτήτων τους (κατηγορία, εύρος τιμών κτλ.). Τα στοιχεία τους χαρτογραφούνται σε αντίστοιχες αριθμητικές μονάδες χωρισμένες σε γραμμές ή στήλες ενός πίνακα. Αρκετές φορές αν υπάρχει μεγάλο ποσοστό ομοιοτήτων μεταξύ δύο ή παραπάνω χρηστών (ή προϊόντων) οπότε το σύστημα τους συγχωνεύει και τους συμπεριφέρεται σαν να είναι μία οντότητα. Οι πίνακες τέτοιας μορφής τείνουν να συμπεριλαμβάνουν, τιμές οι οποίες είναι αρκετά υψηλές η χαμηλές σχετικά με το μέσο όρο. Αυτές οι τιμές αποκαλούνται έκτροπα δεδομένα και έχουν την δυνατότητα να αποπροσανατολίσουν το αποτέλεσμα της πρόβλεψης αν δεν ληφθούν υπόψη. Η αποδοτικότητα ενός συστήματος σύστασης εξαρτάται από τον διαχωρισμό του σχετικού δεδομένου από το άσχετο.

1.2 Κατηγορίες Συστημάτων Σύστασης

Τα συστήματα συστάσεων χρησιμοποιούν ειδικούς αλγόριθμους για συστάσεις και με βάση το επίκεντρο της λογικής του αλγόριθμου, χωρίζονται σε κατηγορίες. Οι τέσσερις πιο βασικές από αυτές είναι οι παρακάτω:

- **Content Based**. Τα συστήματα που βασίζονται στο **περιεχόμενο**. Αυτά τα συστήματα χρησιμοποιούν τα χαρακτηριστικά των αντικειμένων για να προτείνουν άλλα αντικείμενα παρόμοια με αυτά που ο χρήστης έδειξε ενδιαφέρον, βάσει το ιστορικό του.
- **Collaborative**. Τα συστήματα με «**συνεργατικό**» φιλτράρισμα. Το σύστημα εντοπίζει, μέσα σε έναν πίνακα χρηστών, ομοιότητες μεταξύ τους. Στην ουσία οι συστάσεις γίνονται με βάση τις προτιμήσεις άλλων χρηστών που εμφανίζουν παρόμοιες προτιμήσεις και συμπεριφορά με τον χρήστη για τον οποίο προορίζεται η σύσταση.
- **Hybrid.** Υβριδικά συστήματα συστάσεων. Η πιο διαδεδομένη κατηγορία. Αυτά τα συστήματα χρησιμοποιούν έναν συνδυασμό των δυο παραπάνω κατηγοριών (ή και περισσότερων). Τα δεδομένα περνάνε από πολλαπλά φίλτρα κάνοντας το αποτέλεσμα πιο εύστοχο
- **Knowledge Based.** Συστήματα που φιλτράρουν δεδομένα με βάση την **γνώση**. Τα συστήματα που χρησιμοποιούν αυτήν την λογική αναγνωρίζουν τι ανάγκες καλύπτει κάθε αντικείμενο και αν αυτές ανταποκρίνονται στις ανάγκες του χρήστη.

1.2.1 Φιλτράρισμα με βάση το περιεχόμενο – Content Based

Στην συγκεκριμένη προσέγγιση, ο αλγόριθμος φιλτραρίσματος του συστήματος συστάσεων επικεντρώνεται στην ανάλυση των χαρακτηριστικών των αντικειμένων. Προσπαθεί να βρει ομοιότητες και διαφορές μεταξύ των αντικειμένων έχοντας ως πρότυπο τα χαρακτηριστικά των αντικειμένων που ο χρήστης έχει προτιμήσει στο παρελθόν. Αγνοεί πλήρως το πλήθος των χρηστών που έχουν επιλέξει αυτά τα αντικείμενα ή τις ομοιότητες του χρήστη με άλλους. Συνεπώς, για τη βέλτιστη αποτελεσματικότητα του συστήματος, δηλαδή για πιο έγκυρη πρόβλεψη, απαιτείται λεπτομερής περιγραφή των αντικειμένων, καθώς και πλήθος δεδομένων στο προφίλ του χρήστη, σχετικά με τις προτιμήσεις του.

Για παράδειγμα ένα αντικείμενο μπορεί να είναι ένα παιχνίδι για υπολογιστή. Τα χαρακτηριστικά του είναι ο τίτλος, το είδος (στυλ παιχνιδιού), οι κατηγορίες στο οποίο ανήκει, ο χαρακτηρισμός καταλληλόλητας η εταιρία που το δημιούργησε κ.α. Σε αυτά τα

χαρακτηριστικά αποδίδονται αριθμητικές τιμές και παίρνουν την θέση τους σε έναν πίνακα δεδομένων για να συγκριθούν με τις προτιμήσεις του χρήστη, για παράδειγμα ο χαρακτηρισμός καταλληλόλητάς μετατρέπεται σε έναν αριθμό Χ€[1,5]. Προφανώς μερικά χαρακτηριστικά είναι πιο σημαντικά από άλλα, οπότε υπάρχει η δυνατότητα να εφαρμοστεί ένας δείκτης βαρύτητας W όπου είναι επιθυμητό. Τα μη αριθμητικά δεδομένα ή οι μηδενικές τιμές εξαλείφονται από τον πίνακα με διάφορες τεχνικές. Αυτά μπορεί να προκύψουν από έλλειψη στοιχείων ή από την επιλογή να μην ληφθούν υπ' όψη αν επρόκειτο να αλλοιώσουν το τελικό αποτέλεσμα. Αφού χαρτογραφηθούν τα δεδομένα των αντικειμένων, το μόνο μέτρο σύγκρισής τους είναι τα αντικείμενα που εμπεριέχονται στο προφίλ του χρήστη. Επιλέγονται αυτά τα οποία έχουν μεγαλύτερη ομοιότητα με τις προτιμήσεις του, δηλαδή αυτά που έχουν την υψηλότερη βαθμολογία.

Το φιλτράρισμα με βάση το περιεχόμενο, χωρίς να λαμβάνει υπόψιν προτιμήσεις άλλων χρηστών, καταφέρνει να φέρει στην επιφάνεια προϊόντα με χαμηλότερη δημοτικότητα που ικανοποιούν τις ανάγκες του χρήστη. Επιπλέον, παρέχει οφέλη στην επεκτασιμότητα του συστήματος, διότι μπορεί να υποστηρίξει μεγάλο αριθμό χρηστών χωρίς να μειωθεί η απόδοση του συστήματος.

Από την άλλη, το μοντέλο μπορεί να κάνει προτάσεις μόνο με βάση τα υπάρχοντα ενδιαφέροντα του χρήστη. Με άλλα λόγια, το μοντέλο έχει περιορισμένη ικανότητα επέκτασης των ενδιαφερόντων των χρηστών. Ταυτόχρονα, από τη στιγμή που ο χαρακτηρισμός των αντικειμένων επηρεάζεται από τον ανθρώπινο παράγοντα, η ποιότητα των προβλέψεων είναι ανάλογη των χαρακτηριστικών που έχουν εισαχθεί.

1.2.2 Συνεργατικό Φιλτράρισμα – Collaborative Filtering

Το συνεργατικό φιλτράρισμα αποτελεί την πιο δημοφιλή μέθοδο φιλτραρίσματος στα συστήματα συστάσεων. Η μέθοδος αυτή επικεντρώνεται στην εύρεση ομοιοτήτων μεταξύ χρηστών. Πιο συγκεκριμένα, προσπαθεί να εντοπίσει χρήστες που έχουν παρόμοια συμπεριφορά, όσον αφορά τις προτιμήσεις τους. Συνήθως, οι προτιμήσεις αυτές είναι οι αξιολογήσεις των χρηστών σε διάφορα αντικείμενα. Αν οι προτιμήσεις των χρηστών συμπίπτουν θεωρούμε ότι έχουν κοινό γούστο οπότε τα αντικείμενα που αρέσουν σε

κάποιους είναι πολύ πιθανό να αρέσουν και στους υπόλοιπους. Σε αντίθεση με το φιλτράρισμα με βάση το περιεχόμενο, το συνεργατικό φιλτράρισμα δεν απαιτεί τον όγκο της πληροφορίας που χρειάζεται για τις λεπτομερείς περιγραφές των αντικειμένων και εξαφανίζει την ανάγκη ύπαρξης ενός οργανωμένου προφίλ χρήστη. Κατά συνέπεια απασχολεί λιγότερους υπολογιστικούς πόρους και η έλλειψη προτιμήσεων του χρήστη δεν αλλάζει την ακρίβεια του τελικού αποτελέσματος.

1.2.3 Υποκατηγορίες Συνεργατικού Φιλτραρίσματος

Στα συστήματα που εφαρμόζεται το συνεργατικό φιλτράρισμα χωρίζονται σε δύο κατηγορίες με βάση τον αλγόριθμο εύρεσης και υπολογισμού ομοιότητας που χρησιμοποιούν. Οι δύο κατηγορίες αλγοριθμικών προσεγγίσεων είναι:

- Βασιζόμενοι στην μνήμη (**memory-based**)
- Βασιζόμενοι στο μοντέλο (**model-based**)

Οι αλγόριθμοι που βασίζονται στην **μνήμη** ορίζονται ως **ευρετικοί** και εκμεταλλεύονται όλα τα αντικείμενα που έχει αξιολογήσει ο κάθε χρήστης. Προσπαθούν να βρούνε χρήστες που είναι όμοιοι με τον ενεργό χρήστη και χρησιμοποιούν τις προτιμήσεις τους για να προβλέψουν βαθμολογίες για αυτόν. Πρακτικά εξάγουν τις πληροφορίες που απαιτούνται για την πρόβλεψη από ολόκληρη την βάση δεδομένων των χρηστών.

Για να μετρήσουμε την ομοιότητα, πρέπει να βρούμε τη συσχέτιση μεταξύ δύο χρηστών με την χρήση των μεθόδων εύρεσης ομοιότητας που θα χρησιμοποιήσουμε. Αυτό μας δίνει μια τιμή από -1 έως 1 που καθορίζει πόσο μοιάζουν οι δύο χρήστες. Η τιμή 1 σημαίνει ότι και οι δύο βαθμολογούν με τον ίδιο ακριβώς τρόπο, ενώ η τιμή -1 σημαίνει ότι οι χρήστες είναι τελείως διαφορετικοί και βαθμολογούν τα πράγματα ακριβώς αντίθετα. Οι δύο πιο βασικές μέθοδοι εύρεσης ομοιότητας που χρησιμοποιούνται είναι ο **συντελεστής συσχέτισης Pearson** (Pearson correlation coefficient) και το μέτρο συσχέτισης με βάση την **συνημιτονική απόσταση** (cosine-similarity).

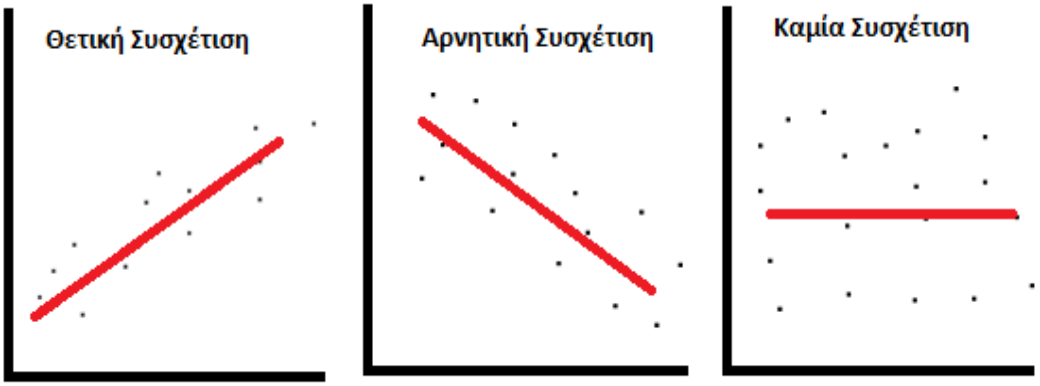
Το πιο συνηθισμένο μέτρο συσχέτισης στην στατιστική είναι η συσχέτιση Pearson. Το πλήρες όνομα είναι Moment Product Pearson Correlation (PPMC). Δείχνει τη γραμμική σχέση μεταξύ δύο συνόλων δεδομένων. Η τιμή της ομοιότητας μεταξύ δύο χρηστών u, u’ εξαρτάται από τις αξιολογήσεις R(u, i) και R(u’, i), που έχουν δώσει οι

χρήστες στο παρελθόν, για κάθε αντικείμενο i που ανήκει στο σύνολο των κοινών αντικειμένων I(u,u'). Το σύνολο I(u, u') αναφέρεται σε όλα τα αντικείμενα που έχουν βαθμολογήσει και οι δύο χρήστες [Πρατικάκης Εμμανουήλ 2017]. Η παρακάτω εξίσωση μας δίνει την τιμή συσχέτισης Pearson.

$$\text{simil}(u,u') = \frac{\sum_{i \in I(u,u')} (R(u,i) - \bar{R}(u)) \cdot (R(u',i) - \bar{R}(u'))}{\sqrt{\sum_{i \in I(u,u')} (R(u,i) - \bar{R}(u))^2} \cdot \sqrt{\sum_{i \in I(u,u')} (R(u',i) - \bar{R}(u'))^2}}$$

Εικόνα 1.1: Συσχέτιση Pearson

Πρακτικά, προσπαθεί να σχεδιάσει ένα γράφημα ευθείας γραμμής που αναπαριστά την συσχέτιση των δεδομένων και να εξάγει από αυτό συμπεράσματα.

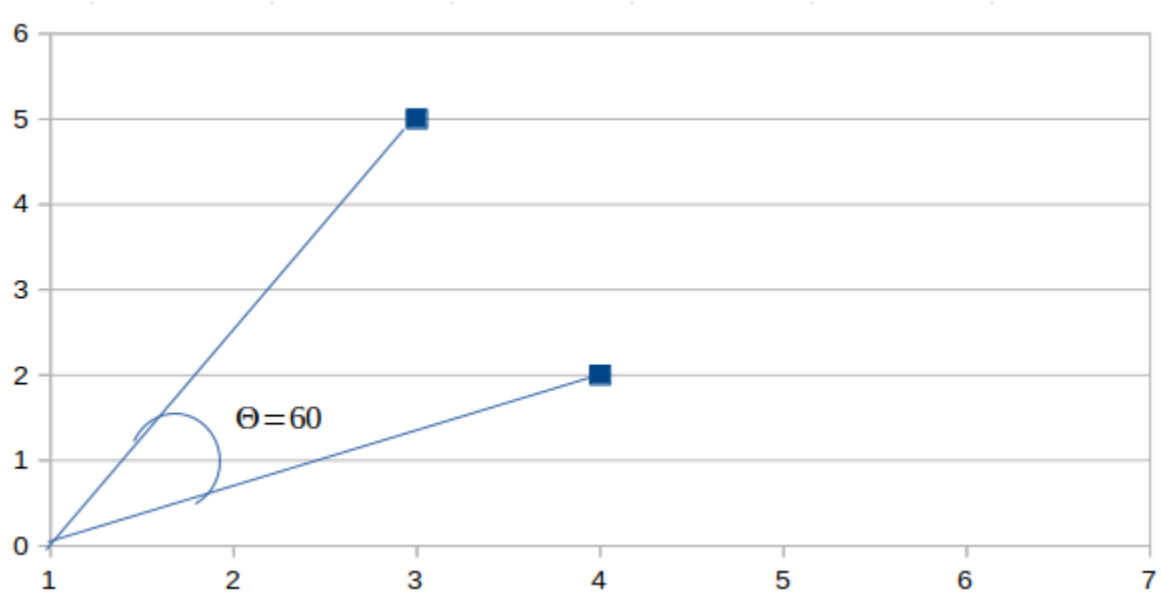


Εικόνα 1. 2: Παράδειγμα Pearson

Στις παραπάνω γραφικές παραστάσεις μπορούμε να δούμε ότι ο δείκτης συσχέτισης r είναι στην πρώτη r > 0, που σημαίνει ότι οι χρήστες έχουν ομοιότητες, στην δεύτερη r <

0, δηλαδή οι χρήστες διαφέρουν αρκετά και στην τελευταία r = 0, διότι υπάρχει μεγάλος βαθμός διασποράς και δεν γίνεται να εξάγουμε κάποιο έγκυρο αποτέλεσμα.

Οι αλγόριθμοι που επικεντρώνονται στις αποστάσεις που έχουν μεταξύ τους τα σημεία της γραφικής παράστασης, δηλαδή οι προτιμήσεις των χρηστών, χρησιμοποιούν διάφορες μεθόδους υπολογισμού απόστασης, όπως η Ευκλείδεια, η απόσταση Manhattan κτλ. Η πιο συνήθης όμως, στα συστήματα συστάσεων είναι η συνημιτονική απόσταση. Υπολογίζοντας την γωνία που σχηματίζουν μεταξύ τους δύο σημεία μπορούμε να εξάγουμε πληροφορίες σχετικά με την ομοιότητά τους. Το συνημίτονο μίας γωνίας 0°- 180° μας δίνει τιμές στο εύρος θ€[-1,1]. Χρησιμοποιώντας την συνημιτονική ομοιότητα, δύο σημεία επιτυγχάνουν τον μέγιστο βαθμό ομοιότητας όταν η γωνία μεταξύ τους είναι 0 ° (είναι προσανατολισμένα προς την ίδια κατεύθυνση), δεν συσχετίζονται όταν η γωνία μεταξύ τους είναι 90 ° (είναι ορθογώνια μεταξύ τους) και έχουν ομοιότητα -1 όταν η γωνία μεταξύ τους είναι 180° (προσανατολίζονται σε διαμετρικά αντίθετες κατευθύνσεις). Ακολουθεί ένα παράδειγμα σημείων με γωνία 60°.



Εικόνα 1. 3: Παράδειγμα Συνημιτονικής Ομοιότητας

Αφού θ = 60° η συνημιτονική ομοιότητα των δύο σημείων είναι 0.5 (συν60° = 0.5) που σημαίνει έχουν αρκετές ομοιότητες.

Οι αλγόριθμοι που βασίζονται στην μνήμη προσφέρουν εύστοχες και ποιοτικές προβλέψεις. Επιπροσθέτως, είναι απλοί και προσφέρουν την δυνατότητα εφαρμογής τους σε ποικίλες περιπτώσεις. Το μεγάλο τους μειονέκτημα ωστόσο, είναι η χρήση ολόκληρης της βάσης δεδομένων κάθε φορά που γίνεται η διαδικασία σύστασης. Συνεπώς, με το πέρας του χρόνου και καθώς τα δεδομένα αυξάνονται, οι επιδόσεις του συστήματος θα πέφτουν όλο και περισσότερο, ειδικά σε περιπτώσεις με μεγάλη διασπορά δεδομένων. Αυτό περιορίζει σημαντικά την δυνατότητα επέκτασης του συστήματος, γι' αυτό και οι μεγάλες εφαρμογές αποφεύγουν την συγκεκριμένη προσέγγιση.

Οι αλγόριθμοι που βασίζονται στο **μοντέλο**, κατασκευάζουν πρώτα ένα μοντέλο από το σύνολο των διαθέσιμων δεδομένων (αξιολογήσεις χρηστών). Αυτό το μοντέλο λειτουργεί ως «εικονικό» πρότυπο, το οποίο αντιπροσωπεύει τις προτιμήσεις του εκάστοτε χρήστη, χωρίς να χρησιμοποιεί κάθε φορά το πλήρες σύνολο των δεδομένων. Αυτή η προσέγγιση προσφέρει οφέλη τόσο στην ταχύτητα όσο και στην επεκτασιμότητα του συστήματος (δυνατότητα διαχείρισης μεγαλύτερου όγκου δεδομένων) και γενικότερα βελτιώνει την απόδοση του συνεργατικού φιλτραρίσματος.

Η διαδικασία δημιουργίας μοντέλων μπορεί να γίνει χρησιμοποιώντας μηχανική εκμάθησης ή τεχνικές εξόρυξης δεδομένων, όπως η τεχνική *Ομαδοποίησης* (Clustering). Ο αλγόριθμος ομαδοποίησης προσπαθεί να χωρίσει το σύνολο των δεδομένων σε υποσύνολα που αντιπροσωπεύουν ομάδες χρηστών με όμοιες προτιμήσεις. Το μέτρο ομοιότητας μεταξύ δύο χρηστών υπολογίζεται με βάση την απόσταση που έχουν μεταξύ τους τα αντίστοιχα διανύσματα με τα οποία αναπαρίστανται. Αυτή η τεχνική όμως, δεν αντιμετωπίζει το πρόβλημα αραιών δεδομένων, με αποτέλεσμα οι συστάσεις να μην είναι τόσο εύστοχες. Παραδείγματα άλλων αλγορίθμων περιλαμβάνουν την *τεχνική μείωσης διαστάσεων, Ταξινόμηση (Classification), Singular Value Decomposition* (SVD), *Matrix Factorization* κ.α.

Η προσέγγιση των βασιζόμενων στο μοντέλο αλγόριθμων, όπως αναφέρθηκε, προσφέρουν οφέλη επεκτασιμότητας και επιτυγχάνουν την αύξηση της ταχύτητας του συστήματος, τουλάχιστον συγκριτικά με τους αλγόριθμους που βασίζονται στην μνήμη γιατί η δημιουργία του μοντέλου συνήθως χρειάζεται αρκετά λιγότερο χρόνο από την

αναζήτηση σε ολόκληρη την βάση δεδομένων. Ωστόσο, η δημιουργία του μοντέλου απασχολεί πλήθος υπολογιστικών πόρων, οπότε η εισαγωγή νέων δεδομένων συνήθως είναι δύσκολη διαδικασία. Η ευστοχία των προβλέψεων είναι χαμηλότερη από τη στιγμή μου δεν χρησιμοποιούν όλες τις διαθέσιμες πληροφορίες εκτός από τις περιπτώσεις που τα δεδομένα είναι αρκετά αραιά και έχουν εφαρμοστεί οι σωστοί αλγόριθμοι.

1.2.4 Συστήματα που βασίζονται στην Γνώση - Knowledge Based

Σε αρκετές περιπτώσεις, στην εξόρυξη δεδομένων, τα αποτελέσματα πρέπει να περάσουν από κάποιο εξειδικευμένο αναλυτή. Τα συστήματα που βασίζονται στην γνώση προσπαθούν να εξαφανίσουν την ανάγκη του ανθρώπινου παράγοντα, χρησιμοποιώντας μια «βάση» γνώσεων σχετικά με τα αντικείμενα που εξετάζουν.

Στα συστήματα συστάσεων η μέθοδος αυτή συλλέγει πληροφορίες για τα χαρακτηριστικά των αντικειμένων αλλά και των χρηστών και στοχεύει στην κάλυψη των αναγκών που δημιουργούνται. Πιο συγκεκριμένα, αν κάποιο αντικείμενο καλύπτει μερικές από τις ανάγκες του χρήστη θα εμφανιστεί ως σύσταση. Για παράδειγμα, στην αγορά ενός φορητού ηλεκτρονικού υπολογιστή το σύστημα μπορεί να προτείνει στον χρήστη να αγοράσει μαζί και μια βάση με ανεμιστήρα ή μία τσάντα για την μεταφορά του.

Το φιλτράρισμα με βάση την γνώση χρησιμοποιείται μεμονωμένο σε αρκετές περιπτώσεις ειδικά σε βάσεις με μεγάλη πολυπλοκότητα χαρακτηριστικών, αλλά η πιο συχνή περίπτωση εφαρμογής του είναι σαν βραχυπρόθεσμη λύση του προβλήματος ψυχρής εκκίνησης (Cold start problem) σε συστήματα που χρησιμοποιούν τις προαναφερθέντες ή και άλλες τεχνικές φιλτραρίσματος. Τα συστήματα που χρησιμοποιούν συνεργατικό φιλτράρισμα πρέπει να αρχικοποιηθούν με μεγάλο αριθμό δεδομένων ειδάλλως οι προβλέψεις δεν θα είναι αρκετά εύστοχες. Από την άλλη οι ποιότητα των συστάσεων στα συστήματα με βάση το περιεχόμενο είναι βέλτιστη όταν ο χρήστης έχει αξιολογήσει μεγάλο αριθμό αντικειμένων. Ένα σύστημα που βασίζεται στη γνώση δεν απαιτεί να έχουν καταχωρηθεί οι αλληλεπιδράσεις των χρηστών με τα αντικείμενα, έτσι είναι ικανό να πάρει τα ινία μέχρι τα δεδομένα να είναι επαρκή και μετά να μεταβεί σε άλλη μέθοδο φιλτραρίσματος.

1.2.5 Υβριδικά Συστήματα – Hybrid Systems

Οι κατηγορίες συστημάτων που αναλύθηκαν παραπάνω προσφέρουν οφέλη σε διαφορετικούς τομείς αλλά κάθε μία έρχεται με τους δικούς της περιορισμούς. Τα υβριδικά συστήματα συστάσεων αποτελούν μια μεμονωμένη κατηγορία η οποία συνδυάζει διάφορες τεχνικές φιλτραρίσματος στη προσπάθεια να κρατήσει όσο το δυνατό περισσότερα από τα πλεονεκτήματα τους και αντίστοιχα, τα λιγότερα μειονεκτήματα. Ανάλογα με τον τρόπο που θα δομηθούν και θα συνδυαστούν οι τεχνικές, τα υβριδικά συστήματα γίνεται να χωριστούν σε επτά κατηγορίες.(Burke, R. (2000))

Σταθμισμένο (Weighted)

Η σύσταση ενός σταθμισμένου υβριδικού συστήματος προέρχεται από τα αποτελέσματα όλων των τεχνικών φιλτραρίσματος που εφαρμόζονται στο σύστημα. Σε κάθε τεχνική αναλογεί ένας συντελεστής βαρύτητας, ο οποίος μπορεί να προσαρμοστεί ανάλογα την περίπτωση, και τα αποτελέσματα κάθε αλγορίθμου ταξινομούνται με φθίνουσα σειρά συσχέτισης, η οποία βασίζεται σε αυτόν τον συντελεστή.

Εναλλασσόμενο (Switching)

Τα συστήματα αυτής της κατηγορίας προσφέρουν την δυνατότητα εναλλαγής τεχνικών βασισμένα σε κάποιο κριτήριο. Αν για παράδειγμα ένα σύστημα χρησιμοποιεί την μέθοδο φιλτραρίσματος με βάση το περιεχόμενο πρώτη και οι προβλέψεις της δεν ξεπερνάνε κάποιο κατώφλι που έχει τεθεί στον δείκτη συσχέτισης, τότε το σύστημα θα κάνει εναλλαγή στην επόμενη μέθοδο.

Ανάμεικτο (Mixed)

Τα ανάμεικτα υβριδικά εμφανίζουν ως αποτέλεσμα μια λίστα (ή περισσότερες) που περιέχει όλες τις συστάσεις από όλες τις μεθόδους που εκτελούνται. Χρησιμοποιούνται όπου είναι πρακτικό να εμφανιστεί μεγάλο πλήθος αποτελεσμάτων.

Με Συνδυασμό Χαρακτηριστικών (Feature Combination)

Ένα σύστημα που χρησιμοποιεί τις τεχνικές περιεχομένου και συνεργασίας, με αυτή την προσέγγιση, αντιμετωπίζει τις πληροφορίες της συνεργατικής μεθοδολογίας ως επιπρόσθετα δεδομένα χαρακτηριστικών αντικειμένων και χρησιμοποιεί τεχνικές φιλτραρίσματος με βάση το περιεχόμενο στο ανανεωμένο πλέον σύνολο δεδομένων. Αυτό επιτρέπει στο σύστημα να χρησιμοποιεί το συνεργατικό φιλτράρισμα χωρίς να βασίζεται εξ' ολοκλήρου σε αυτό και ταυτόχρονα να εξάγει δεδομένα σχετικά με τις αλληλεπιδράσεις χρηστών που δεν είναι ικανή να υπολογίσει η μέθοδος με βάση το περιεχόμενο.

Αλληλουχία (Cascade)

Σε αντίθεση με τις προηγούμενες μεθόδους, τα υβριδικά συστήματα αλληλουχίας περιλαμβάνουν μια σταδιακή διαδικασία. Ο συνδυασμός των εσωτερικών τεχνικών φιλτραρίσματος γίνεται σε σειρά, δηλαδή πιο απλά η είσοδος της μιας τεχνικής είναι η έξοδος της προηγούμενης. Ο σκοπός του επόμενου φιλτραρίσματος είναι η βελτιστοποίηση των προβλέψεων, όμως, αρκετές φορές οι συστάσεις της πρώτης τεχνικής μπορεί να είναι αρκετά εύστοχες, οπότε το σύστημα αποφεύγει την χρήση της επόμενης. Το δεύτερο φιλτράρισμα έχει χαμηλότερη προτεραιότητα και επικεντρώνεται μόνο στα στοιχεία για τα οποία απαιτείται πρόσθετη διάκριση. Τα συστήματα αυτής της κατηγορίας είναι πιο αποτελεσματικά από, για παράδειγμα, ένα σταθμισμένο υβριδικό σύστημα που εφαρμόζει όλες τις τεχνικές του σε όλα τα στοιχεία.

Βελτίωση Χαρακτηριστικών (Feature Augmentation)

Εδώ η μία τεχνική φιλτραρίσματος χρησιμοποιείται για να ομαδοποιήσει τα αντικείμενα ή απλά να τα βαθμολογήσει και στη συνέχεια αυτές οι πληροφορίες ενσωματώνονται στην εκτέλεση της επόμενης. Θυμίζει το σύστημα αλληλουχίας, με την διαφορά ότι εδώ το δεύτερο σε σειρά σύστημα δεν βελτιώνει τις προτάσεις του πρώτου αλλά χρησιμοποιεί τα βελτιωμένα επιπρόσθετα χαρακτηριστικά που έχουν παραχθεί.

Meta-level

Με μία παρόμοια λογική με την παραπάνω λειτουργούν και αυτού του είδους τα συστήματα. Η μία τεχνική δημιουργεί ένα μοντέλο και στη συνέχεια το περνάει σαν είσοδο στην επόμενη. Η διαφορά με τα συστήματα βελτίωσης χαρακτηριστικών είναι κυρίως ότι σε αυτά δεν χρησιμοποιούμε ένα εκπαιδευμένο μοντέλο για να δημιουργήσουμε χαρακτηριστικά για την επόμενη τεχνική αλλά κατασκευάζουμε και περνάμε ολόκληρο το μοντέλο. Ειδικά σε ένα υβριδικό σύστημα που χρησιμοποιεί τεχνικές βασιζόμενες στο περιεχόμενο και συνεργατικές τεχνικές. Το πλεονέκτημα αυτής της μεθόδου είναι ότι το μοντέλο που θα δημιουργηθεί θα αποτελεί μία συμπιεσμένη αναπαράσταση των προτιμήσεων του χρήστη και η ακρίβεια των συστάσεων του επακόλουθου συνεργατικού φιλτραρίσματος θα είναι αρκετά μεγαλύτερη.

1.3 Προβλήματα και Προκλήσεις Συστημάτων Συστάσεων

Ο σχεδιασμός ενός συστήματος συστάσεων μπορεί να γίνει μια αρκετά πολύπλοκη διαδικασία. Οι εφαρμογές πραγματικού κόσμου που απευθύνονται σε πλήθος χρηστών και επιχειρήσεων είναι αναγκαίο να δομούνται σωστά. Ταυτόχρονα με την αποδοτικότητα του συστήματος πρέπει να ληφθούν υπόψιν κι άλλοι τομείς, όπως η «ποιότητα ζωής» του λογισμικού, η επεκτασιμότητα κ.α. Σε αυτό το κεφάλαιο αναφέρονται μερικές προκλήσεις που αντιμετωπίζουν τα συστήματα συστάσεων που πρέπει να αντιμετωπιστούν σε πρώιμο στάδιο.

Πρόβλημα Ψυχρής Εκκίνησης (Cold Start problem)

Αυτό το πρόβλημα προκύπτει όταν προστίθενται νέοι χρήστες ή νέα αντικείμενα στο σύστημα. Σε τέτοιες περιπτώσεις, ούτε οι προτιμήσεις των νέων χρηστών μπορούν να προβλεφθούν ούτε τα νέα αντικείμενα μπορούν να αξιολογηθούν ή να αγοραστούν από τους χρήστες, με αποτέλεσμα την πτώση της ακρίβειας της πρόβλεψης. Η εισαγωγή νέων δεδομένων στο σύστημα και το πρόβλημα της ψυχρής εκκίνησης, αφορά και τις δυο προσεγγίσεις φιλτραρίσματος (με βάση το περιεχόμενο και το συνεργατικό φιλτράρισμα).

Το συγκεκριμένο πρόβλημα έχει αρκετές λύσεις όπως για παράδειγμα, η εμφάνιση συστάσεων με βάση την δημοτικότητα των αντικειμένων ή η χρήση ενός συστήματος που βασίζεται στη γνώση (όπως αναφέρεται παραπάνω) μέχρι τα δεδομένα να είναι επαρκής. Πολλές φορές χρησιμοποιείται και κάποιο είδος ερωτηματολογίου, που ζητάει από τους χρήστες να δηλώσουν ρητά τις προτιμήσεις τους με την έναρξη της πλοήγησης τους ή μετά το άνοιγμα του λογαριασμού τους όπως παράδειγμα στο Netflix.

Πρόβλημα Διασποράς Δεδομένων (Sparsity Problem)

Συμβαίνει πολλές φορές όταν οι περισσότεροι χρήστες δεν αξιολογούν αρκετά τα αντικείμενα που αγόρασαν ή επισκεφτήκαν και ως εκ τούτου το μοντέλο αξιολόγησης γίνεται πολύ αραιό, το οποίο μπορεί να οδηγήσει σε μεγάλη διασπορά των δεδομένων κατά τον υπολογισμό της σύστασης και έτσι μειώνει τις πιθανότητες εύρεσης ενός συνόλου χρηστών με παρόμοιες αξιολογήσεις ή ενδιαφέροντα. Είναι ένα πολύ σύνηθες πρόβλημα στα συστήματα συστάσεων και αντιμετωπίζεται κυρίως αλγοριθμικές λύσεις εξόρυξης δεδομένων, όπως αυτές που αναφέρθηκαν παραπάνω σχετικά με την ομαδοποίηση δεδομένων.

Πρόβλημα Κλιμάκωσης (Scalability Problem)

Με το πέρας του χρόνου μία εφαρμογή, αποκτά περισσότερους χρήστες και η βάση δεδομένων της μεγαλώνει εκθετικά. Όλες οι αλληλεπιδράσεις χρηστών και αντικειμένων που επεξεργάζεται ένα σύστημα συστάσεων αυξάνονται συνεχώς και δημιουργείται η ανάγκη προσαρμογής του συστήματος στις απαιτήσεις που συμπεριλαμβάνονται σε αυτόν τον όγκο πληροφορίας. Η απόδοση του συστήματος πρέπει να παραμείνει υψηλή καθώς το σύστημα μένει απαράλαχτο. Αυτό προϋποθέτει τον σωστό σχεδιασμό του συστήματος, με τη χρήση συνεργατικών αλγορίθμων που αναφέρθηκαν παραπάνω, οι οποίοι έχουν δημιουργηθεί για την απλοποίηση διαχείρισης δεδομένων σε μεγάλη κλίμακα

Πρόβλημα Υπερειδίκευσης (Overspecialization Problem)

Η υπερειδίκευση εμφανίζεται μόνο στα συστήματα που βασίζονται στο περιεχόμενο και αποτελεί το πιο συχνό πρόβλημα αυτής της τεχνικής. Από τη στιγμή που το σύστημα βασίζεται στο περιεχόμενο, δηλαδή στα χαρακτηριστικά των αντικειμένων, οι συστάσεις

θα δημιουργούνται με βάση τα αντικείμενα που αξιολόγησε ο χρήστης στο παρελθόν. Αυτό έχει σαν αποτέλεσμα την έλλειψη ποικιλίας από τις συστάσεις που δημιουργούνται. Εμποδίζει τους χρήστες να ανακαλύψουν κάτι νέο και διαφορετικό, διότι προτείνει στους χρήστες αντικείμενα με τα οποία είναι ήδη εξοικειωμένοι. Προκειμένου να προτείνει το σύστημα νέα αντικείμενα μαζί με τα συνήθη, πρέπει να εισάγουμε μια δόση ελεγχόμενης «τυχαιότητας» και αυτό επιτυγχάνεται με την χρήση γενετικών αλγορίθμων.

Πρόβλημα Καθυστέρησης (Latency Problem)

Όταν προστίθενται νέα αντικείμενα στο σύστημα, λόγω του ότι δεν έχουν αξιολογηθεί ακόμη, δεν θα ληφθούν υπ' όψη στις επόμενες προβλέψεις. Αυτή η καθυστέρηση που δημιουργείται από την στιγμή της εισαγωγής τους μέχρι την στιγμή που θα έχουν επαρκή δεδομένα για να προταθούν στον χρήστη, γίνεται να μειωθεί με την τεχνική φιλτραρίσματος με βάση το περιεχόμενο. Ωστόσο, υπάρχει ο κίνδυνος της υπερειδίκευσης που αναφέρεται παραπάνω. Στα συστήματα που χρησιμοποιούν συνεργατικό φιλτράρισμα όμως, η λύση είναι η κατηγοριοποίηση των αντικειμένων από την πρώτη στιγμή που θα εισαχθούν στο σύστημα.

2. Η εφαρμογή Magellan

2.1 Δομικά Στοιχεία

Οι τεχνολογίες που χρησιμοποιήθηκαν για την κατασκευή της εφαρμογής χωρίζονται σε τρεις κατηγορίες. Το γραφικό περιβάλλον διεπαφής χρηστών (**front-end**), τον διακομιστή (**back-end**) και την υπηρεσία επεξεργασίας δεδομένων (**service**).

Για την δημιουργία του **front-end** έχουν χρησιμοποιηθεί οι γλώσσες προγραμματισμού **TypeScript (Angular) / HTML5 / CSS3.** Η Angular ανήκει στα 3 πιο δημοφιλή και μοντέρνα JavaScript Frameworks (Angular, React, Vue). Διευκολύνει την κατασκευή ιστοσελίδων και προσφέρει τη δυνατότητα κατασκευής εφαρμογών που προορίζονται για κινητές συσκευές ή ακόμα και για την επιφάνεια εργασίας. Αναπτύσσεται αποκλειστικά από την Google και συνδυάζει τεχνικές, όπως δρομολόγηση (routing), animation, λειτουργικές μονάδες (modules) που άλλα frameworks πρέπει να στηριχθούν σε εξωτερικές βιβλιοθήκες, με σκοπό την βέλτιστη ανάπτυξη λογισμικού.

Για την δημιουργία του **back-end** κατασκευάστηκε ένα REST-API χρησιμοποιώντας το Spring Boot (JAVA framework). Η Spring εμπεριέχει όλα τα πολύπλοκα κομμάτια της JAVA που χρησιμοποιούνται για δικτυακές εφαρμογές σε αρκετά απλουστευμένη μορφή. Έχει μεγάλη απήχηση στην κοινότητα των προγραμματιστών για την γρήγορη ταχύτητά της, την ασφάλεια αλλά και την ευελιξία της.

Για την υλοποίηση του συστήματος συστάσεων χρησιμοποιήθηκε η γλώσσα Python για τα πλεονεκτήματα που προσφέρει σε αυτό τον τομέα. Τα πλεονεκτήματα αυτά είναι :

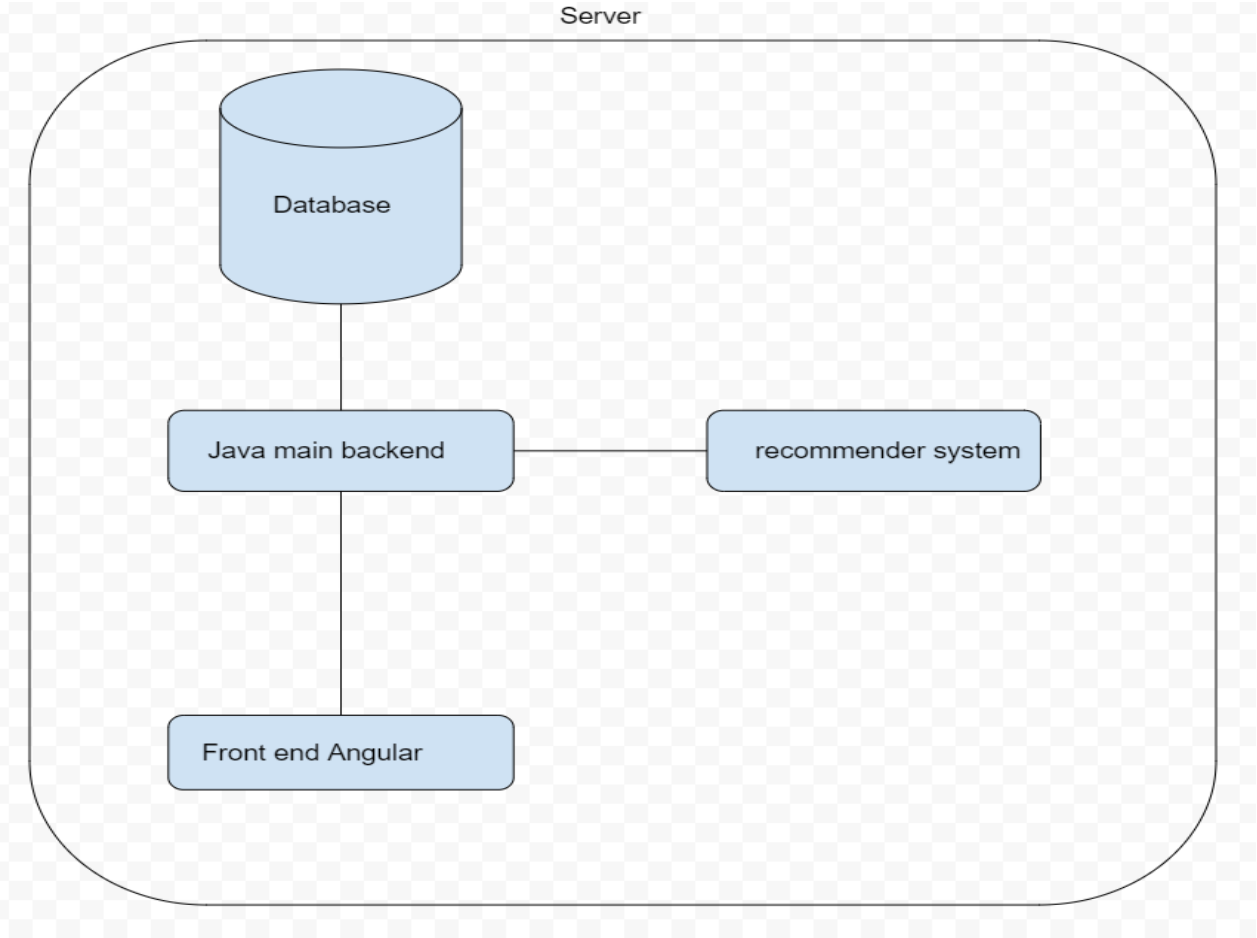
- **Ευχρηστία**. Η Python μας προσφέρει συνοπτικό και αναγνώσιμο κώδικα. Ενώ οι περίπλοκοι αλγόριθμοι και οι ευέλικτες ροές εργασίας υποστηρίζουν τη μηχανική εκμάθηση και το AI. Η απλότητα της Python επιτρέπει στους προγραμματιστές να γράφουν αξιόπιστα συστήματα και να καταβάλουν περισσότερο χρόνο στην επίλυση ενός προβλήματος μηχανικής μάθησης.
- **Πολλές επιλογές βιβλιοθηκών και frameworks**. Μας προσφέρει πολλές βιβλιοθήκες και διάφορα εργαλεία έτσι ώστε να μειώσουμε κατά πολύ τον χρόνο ανάπτυξης που χρειαζόμαστε. Οι βιβλιοθήκες αυτές είναι προ-γραμμένος κώδικας που χρησιμοποιούμε για επίλυση κοινών εργασιών προγραμματισμού. Η Python, με την πλούσια στοίβα της τεχνολογίας, διαθέτει ένα εκτεταμένο σύνολο βιβλιοθηκών για τεχνητή νοημοσύνη και μηχανική μάθηση, όπως Scikit-learn (machine learning), NumPy (high-performance scientific computing and data analysis), Pandas (general-purpose data analysis), Seaborn (data visualization), SciPy (advanced computing).

- **Ανεξαρτησία πλατφόρμας**. Η ανεξαρτησία της πλατφόρμας αναφέρεται σε μια γλώσσα προγραμματισμού ή ένα framework που επιτρέπει στους προγραμματιστές να φτιάξουν προγράμματα σε ένα μηχάνημα και να τα χρησιμοποιούν σε άλλο μηχάνημα χωρίς καμία (ή με ελάχιστες) αλλαγές. Ένα κλειδί της δημοτικότητα της Python είναι ότι είναι μια γλώσσα ανεξάρτητη από το σύστημα που φτιάχνεται. Η Python υποστηρίζεται από πολλές πλατφόρμες, όπως Linux, Windows και macOS. Ο κώδικας Python μπορεί να χρησιμοποιηθεί για τη

δημιουργία αυτόνομων εκτελέσιμων προγραμμάτων για τα περισσότερα κοινά λειτουργικά συστήματα, πράγμα που σημαίνει ότι το λογισμικό Python μπορεί εύκολα να διανεμηθεί και να χρησιμοποιηθεί σε αυτά τα λειτουργικά συστήματα χωρίς διερμηνέα Python.

Το Docker είναι ένα εργαλείο, το οποίο μας βοηθάει να τρέξουμε την εφαρμογή μας σε οποιοδήποτε λειτουργικό σύστημα και να είμαστε σίγουροι ότι τρέχει με τον ίδιο ακριβώς τρόπο όπως τρέχει στο δικό μας σύστημα. Ο λόγοι που χρησιμοποιούμε Docker είναι επειδή έχουμε μια λογική microservices στην εφαρμογή μας. Η ενορχήστρωση πολλών “μικροεφαρμογών” γίνεται αρκετά πιο εύκολη και διαχειρίσιμη με το Docker. Τα σημαντικά οφέλη του Docker είναι η αναπαραγωγιμότητα του περιβάλλοντος ανάπτυξης, η απομόνωση από το εκάστοτε λειτουργικό σύστημα και η ασφάλεια.

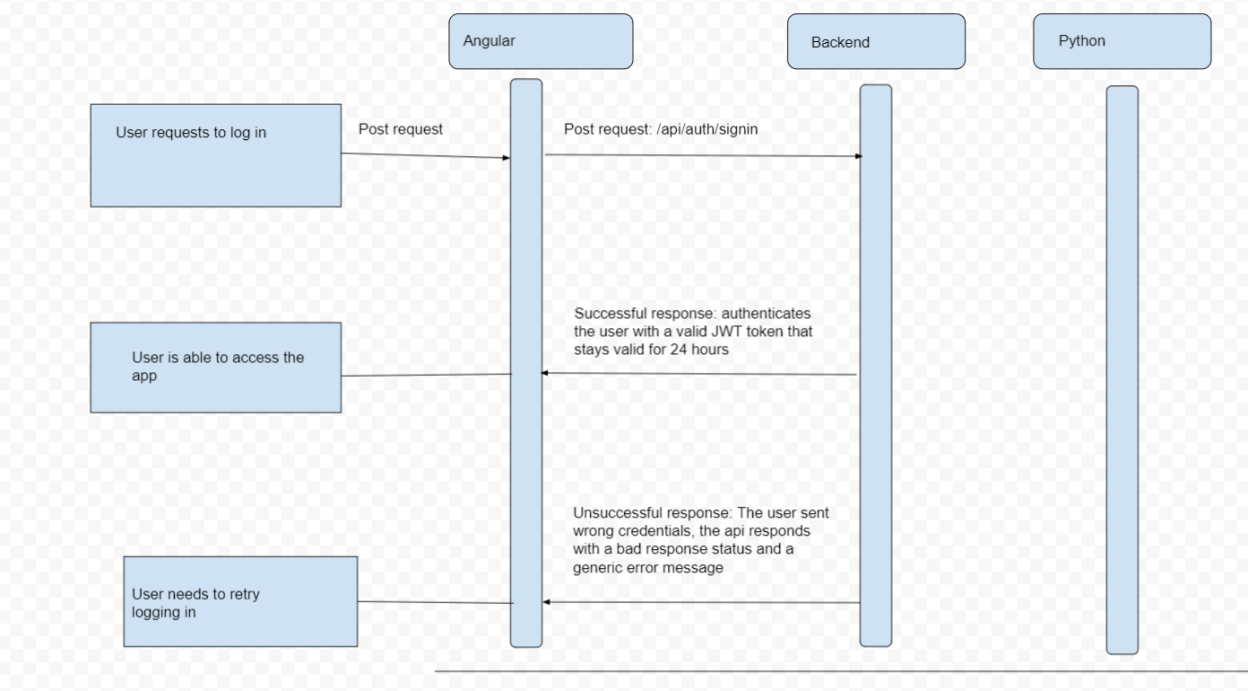
Η γενική εικόνα της αρχιτεκτονικής της εφαρμογής είναι η παρακάτω:



Εικόνα 2.1.1: Δομή Εφαρμογής

Από το σχήμα φαίνεται ότι το backend (JAVA API) διαχειρίζεται την επικοινωνία μεταξύ όλων των υπηρεσιών/συστημάτων (services). Επιλέξαμε αυτή την αρχιτεκτονική επειδή μέσω του εργαλείου Docker που χρησιμοποιούμε είναι πολύ εύκολο να ενθυλακώσουμε (encapsulate) το κάθε service στο δικό του περιβάλλον το οποίο προσφέρει οφέλη στην ασφάλεια του συστήματος διότι αποτρέπει την επικοινωνία από εξωτερικές πηγές.

Στην περίπτωση που ο χρήστης θέλει να κάνει login στην εφαρμογή το σχήμα μοιάζει κάπως έτσι:

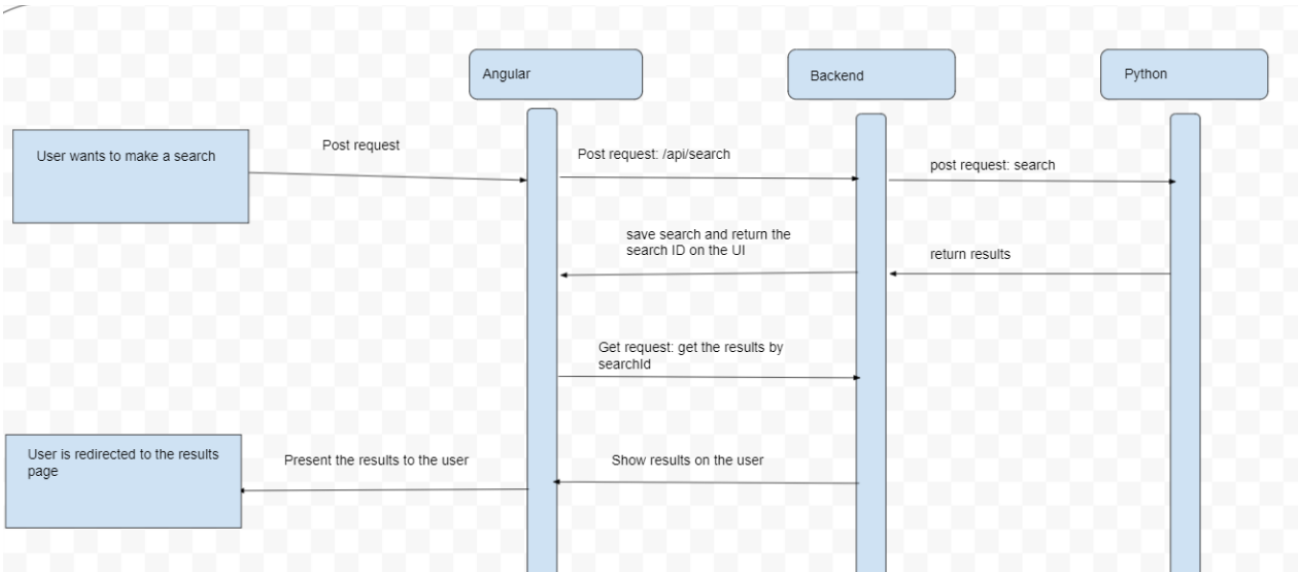


Εικόνα 2.1. 2: Διάγραμμα Ροής Σύνδεσης

Ο χρήστης μέσω της εφαρμογής βάζει τα στοιχεία του και πατάει «σύνδεση», έπειτα το frontend επαληθεύει τα στοιχεία που πληκτρολόγησε ο χρήστης και στέλνει ένα αίτημα στο backend. Το backend μιλάει με την βάση δεδομένων και βλέπει αν υπάρχει χρήστης με τα αντίστοιχα στοιχεία, εάν υπάρχει τότε φτιάχνει ένα κλειδί το οποίο έχει διάρκεια ζωής 24 ώρες και με αυτό ο χρήστης μπορεί να χρησιμοποιεί την εφαρμογή χωρίς να ξανακάνει login για τις επόμενες 24 ώρες εκτός αν αποφασίσει να κάνει

αποσύνδεση μέσω της εφαρμογής όπου τότε το κλειδί διαγράφεται. Σε περίπτωση όμως που ο χρήστης βάλει λάθος στοιχεία τότε του εμφανίζεται μήνυμα λάθους και πρέπει να προσπαθήσει ξανά.

Ας δούμε όμως πως είναι το σχήμα για μια πιο πολύπλοκη διαδικασία όπως να κάνουμε μια αναζήτηση.



Εικόνα 2.1. 3: Διάγραμμα Ροής Αναζήτησης

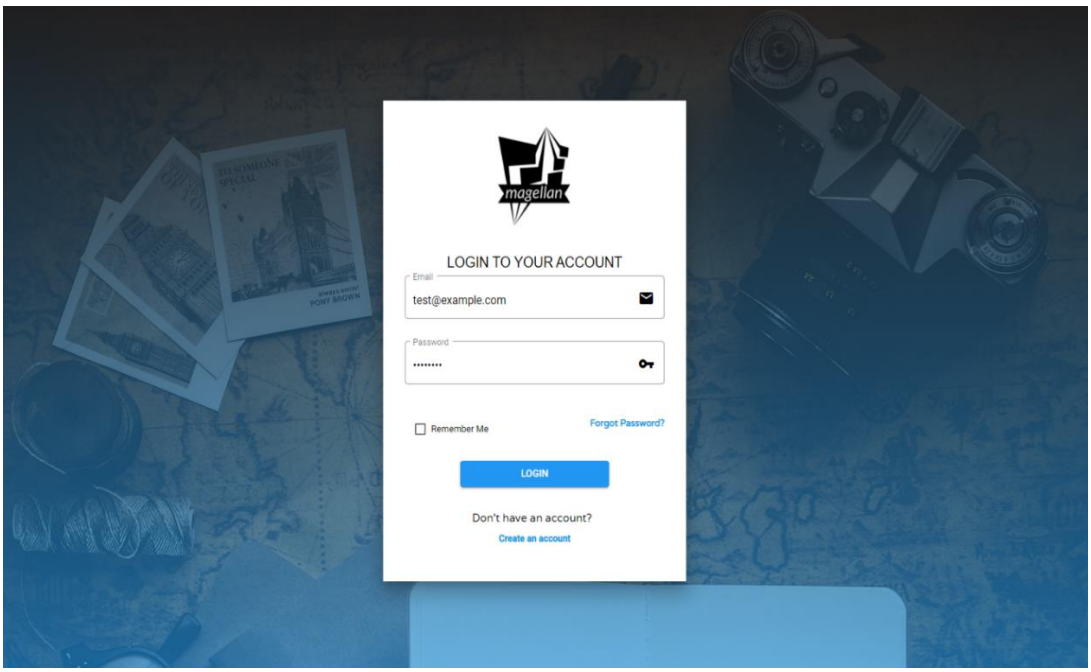
Αρχικά ο χρήστης πατάει το κουμπί αναζήτησης στην εφαρμογή. Το frontend κάνει ένα αίτημα στο backend και εφόσον μιλήσει με την βάση δεδομένων και πάρει τα δεδομένα του χρήστη τα στέλνει στο σύστημα συστάσεων (python). Το python service τρέχει τα δεδομένα του χρήστη ενάντια στον ευριστικό αλγόριθμο και αυτός επιστρέφει κάποιες προτάσεις. Το backend μετά αποθηκεύει τα αποτελέσματα στην βάση δεδομένων και επιστρέφει τα δεδομένα στο frontend. Τέλος εμφανίζονται στην οθόνη του χρήστη οι συστάσεις του συστήματος.

2.2 Περιγραφή

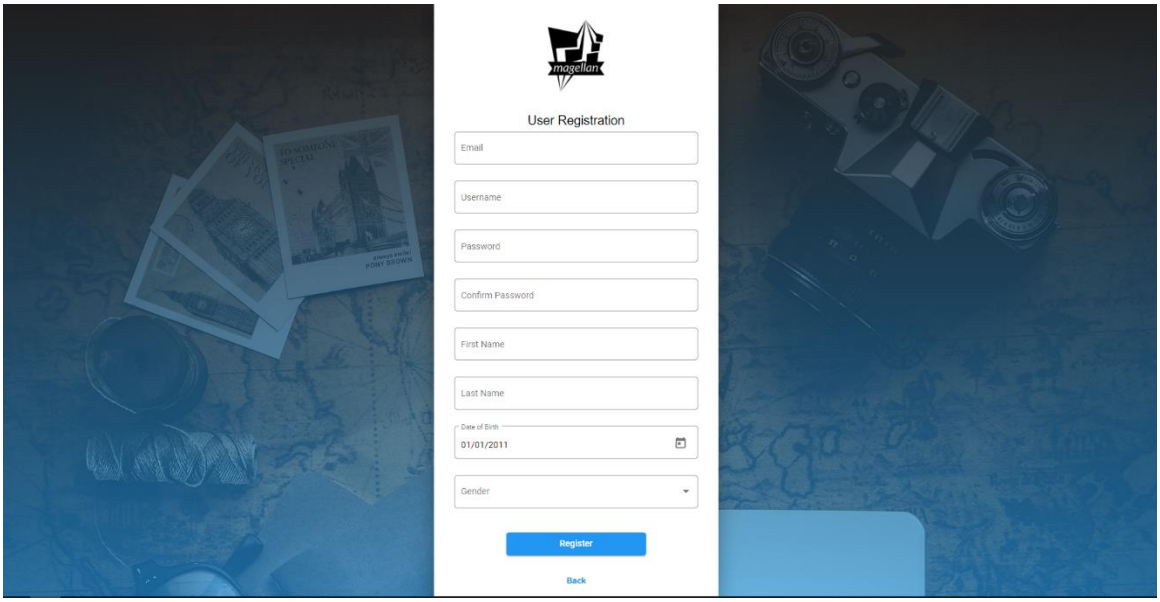
Η εφαρμογή **Magellan** (*Explore your options)* απευθύνεται σε χρήστες που αναζητούν μέρη διασκέδασης ταιριαστά με τις προτιμήσεις τους. Τα περισσότερα από αυτά τα μέρη/μαγαζιά απευθύνονται σε όλους, όμως, το καθένα έχει χαρακτηριστικά τα

οποία μπορούν να το διαφοροποιήσουν από τα υπόλοιπα ή τουλάχιστον να το κατατάξουν σε μια συστάδα (cluster). Παράδειγμα, σαν γενικές κατηγορίες, μπορούμε να πάρουμε τις καφετέριες και τα εστιατόρια αν και υπάρχουν πολλά εστιατόρια που σερβίρουν καφέ και πολλές καφετέριες που σερβίρουν φαγητό. Από την άλλη, υπάρχουν τα ακριβά και τα φθηνά εστιατόρια. Αν ένας χρήστης συχνάζει σε φθηνά εστιατόρια, δεν αποκλείεται κάποια στιγμή να θέλει να πάει και σε ένα ακριβό ή και το αντίστροφο. Επειδή λοιπόν, οι προτιμήσεις ψυχαγωγίας είναι ανάλογες των περιστάσεων και συνεχώς μεταβαλλόμενες, η εφαρμογή υποστηρίζει μια πιο εξατομικευμένη λειτουργία αναζήτησης την οποία θα δούμε στη συνέχεια.

Την πρώτη φορά που ο χρήστης επισκέπτεται την εφαρμογή έχει την δυνατότητα να κάνει εγγραφή παρέχοντας τα απαραίτητα προσωπικά στοιχεία του, όπως email, ημερομηνία γέννησης, γένος κτλ. από τα οποία ξεκινάμε να χτίζουμε σιγά σιγά το προφίλ του. Για πιο εξειδικευμένα συστήματα συστάσεων, στο σημείο της εγγραφής ή και σε επόμενο στάδιο, μπορούν να χρησιμοποιηθούν επιπλέον πληροφορίες όπως, αν ο χρήστης έχει αυτοκίνητο, περιοχές στις οποίες συχνάζει και άλλα. Αφού γίνει επιτυχής εγγραφή μπορεί πλέον να συνδεθεί στην εφαρμογή με τα διαπιστευτήριά του. Η επικύρωση των στοιχείων του γίνεται μέσω του API.

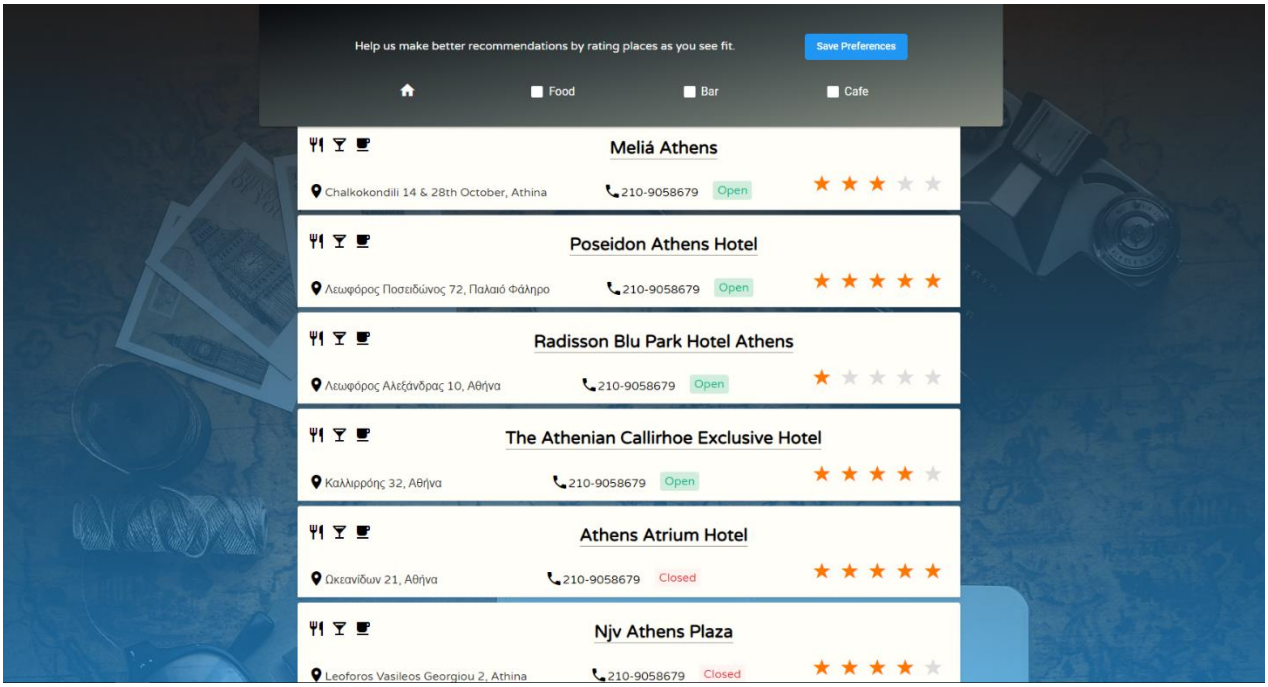


Εικόνα 2.2. 1: Σελίδα Εισόδου



Εικόνα 2.2. 2: Σελίδα Εγγραφής

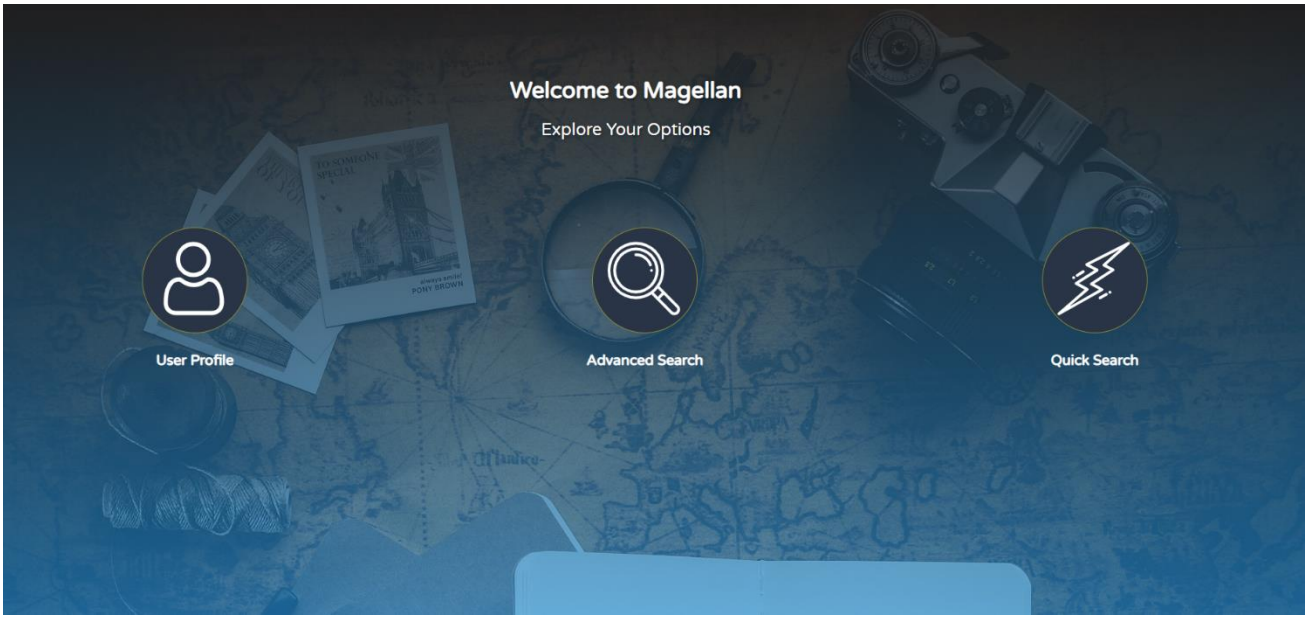
Σε αυτό το σημείο τα συστήματα συστάσεων συναντούν το πρώτο τους πρόβλημα. Το πρόβλημα της ψυχρής εκκίνησης (**Cold start problem**). Εισάγουμε έναν νέο χρήστη στο σύστημα για τον οποίον δεν γνωρίζουμε τίποτα. Δεν έχει δηλώσει καμία προτίμηση του και δεν είναι δυνατό να ξέρουμε ποια μέρη έχει επισκεφθεί. Τα αποτελέσματα των πρώτων αναζητήσεων, μέχρι να υπάρχουν επαρκή δεδομένα για ποιοτική σύσταση, μπορούν να είναι με βάση την δημοτικότητα ή καλύτερα να εμφανίζονται μέρη που άλλοι χρήστες έχουν βαθμολογήσει υψηλά και έχουν κι άλλα κοινά χαρακτηριστικά με τον χρήστη, για παράδειγμα, να ανήκουν στο ίδιο ηλικιακό εύρος. Αλλά προφανώς, αυτή η προσέγγιση δεν θα ήταν αρκετά εξατομικευμένη και η ακρίβεια των συστάσεων θα ήταν αρκετά χαμηλή και θα βασιζόταν σε μεγάλο βαθμό στην τύχη. Από επιχειρηματικής μεριάς, ένας χρήστης που λαμβάνει ακριβείς προβλέψεις είναι πολύ πιο πιθανό να συνεχίσει να χρησιμοποιεί την εφαρμογή και ταυτόχρονα να την συστήσει σε άλλους. Οπότε, τοποθετήσαμε ένα επιπλέον βήμα, μετά την πρώτη σύνδεση του χρήστη στην εφαρμογή, στο οποίο δύναται να αξιολογήσει μέρη ψυχαγωγίας όπως επιθυμεί και να τροφοδοτήσει έτσι το σύστημα με τις προτιμήσεις του.



Εικόνα 2.2. 3: Σελίδα Προτιμήσεων Χρήστη

Τα μέρη ψυχαγωγίας που εμφανίζονται, προέρχονται από το Places API της Google.

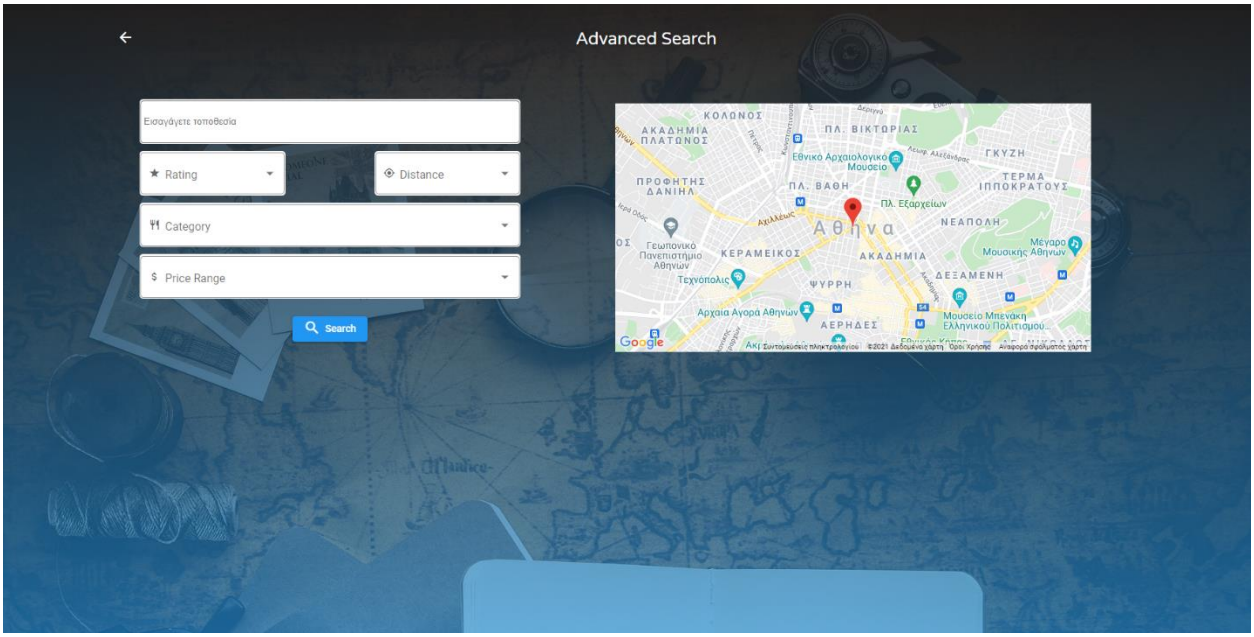
Στη συνέχεια, αφού ο χρήστης συνδεθεί στην εφαρμογή και υπάρχει ιστορικό αξιολογήσεων, εμφανίζεται η αρχική σελίδα της εφαρμογής. Εδώ εμφανίζονται τρείς επιλογές. Η πρώτη αφορά τις ενέργειες του χρήστη όσον αφορούν τα δεδομένα του. Αναλυτικότερα, ο χρήστης έχει την δυνατότητα να ενημερωθεί για τις προηγούμενες συστάσεις που έγιναν (τις 5 τελευταίες), να επαναξιολογήσει τα διαθέσιμα μέρη διασκέδασης και να αποσυνδεθεί από την εφαρμογή. Η σελίδα αξιολόγησης είναι ένα πολύ σημαντικό μέρος της εφαρμογής, διότι η πιο συχνή πρόκληση που αντιμετωπίζουν τα συστήματα συστάσεων είναι η έλλειψη των αξιολογήσεων. Για παράδειγμα σε ένα ηλεκτρονικό κατάστημα η αξιολόγηση ενός αντικειμένου μπορεί να γίνει μόνο μετά την αγορά και οι περιπτώσεις που ο χρήστης θα επανέρθει στην σελίδα για να το βαθμολογήσει αφού του έχει παραδοθεί σπανίζουν ή είναι συνήθως λόγω αρνητικών εντυπώσεων.



Εικόνα 2.2. 4: Αρχική Σελίδα

Στο δεξί μέρος υπάρχει η επιλογή, *Γρήγορης Αναζήτησης,* που επιτρέπει στον χρήστη να ενεργοποιήσει το σύστημα συστάσεων παράγοντας μια πρόβλεψη για τα μέρη διασκέδασης που μπορεί να τον ενδιαφέρουν με βάση το προφίλ του. Για την ακρίβεια στην γρήγορη αναζήτηση, ενεργοποιείται η υβριδική προσέγγιση φιλτραρίσματος συνδυάζοντας και την μέθοδο φιλτραρίσματος με βάση το περιεχόμενο αλλά και το συνεργατικό φιλτράρισμα. Η λειτουργεία του συστήματος συστάσεων που πραγματοποιείται στη μεριά της python θα αναλυθεί περεταίρω στο επόμενο κεφάλαιο.

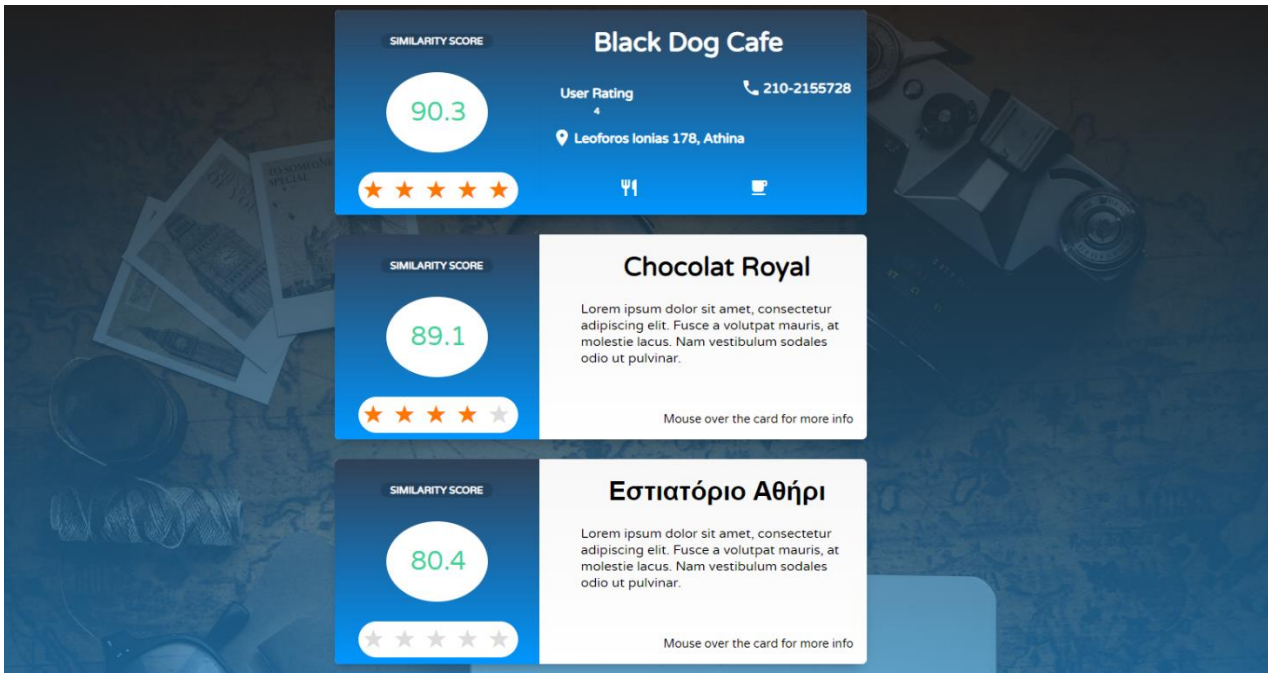
Στο κέντρο της αρχικής σελίδας βρίσκεται η *Προηγμένη Αναζήτηση.* Εδώ ο χρήστης μπορεί να εφαρμόσει ένα επιπλέον επίπεδο φιλτραρίσματος με ειδικά φίλτρα για τα μέρη διασκέδασης. Οι επιλογές που δίνονται στον χρήστη είναι η κατηγορία εύρους τιμών των καταστημάτων, η βαθμολογία άλλων χρηστών, ο τύπος του καταστήματος (Εστιατόρια, Καφετέριες ή Μπαρ), η γεωγραφική τους τοποθεσία τους καθώς ο ορισμός χιλιομέτρων της ακτίνας γύρω απ’ αυτά.



Εικόνα 2.2. 5: Σελίδα Σύνθετης Αναζήτησης

Αυτή η σελίδα υπάρχει για τις περιπτώσεις που ο χρήστης είναι μεν αβέβαιος για τον προορισμό της εξόδου του αλλά πιο σίγουρος για τα χαρακτηριστικά του μέρους που θα ήθελε να επισκεφτεί. Με βάση τη λογική που ακολουθεί η εφαρμογή Magellan, δηλαδή την εύρεση του βέλτιστου μέρους διασκέδασης για τον εκάστοτε χρήστη οποιαδήποτε στιγμή, η συγκεκριμένη σελίδα θεωρείται απαραίτητη. Το μέρος που θα επισκεφτεί ο χρήστης είναι ανάλογο των περιστάσεων και υπάρχουν πολλαπλές μεταβλητές που ορίζουν την κάθε περίσταση.

Τέλος, μετά από κάθε αναζήτηση ο χρήστης μεταφέρεται στην σελίδα εμφάνισης των συστάσεων. Εδώ του δίνεται η δυνατότητα περιήγησης σε μία λίστα με μέρη διασκέδασης τα οποία είναι προϊόν του συστήματος συστάσεων. Πιο αναλυτικά, οι πληροφορίες που εμφανίζονται για κάθε μέρος είναι ο τίτλος του, η τοποθεσία του, οι κατηγορίες στις οποίες ανήκει, η βαθμολογία που έχουν δώσει άλλοι χρήστες σε αυτό το μέρος μέσω της Google, οι ώρες λειτουργίας του, το τηλέφωνο και η περιγραφή του αν είναι διαθέσιμα. Επιπλέον, εμφανίζεται και ο βαθμός ομοιότητάς του σύμφωνα με τις προτιμήσεις του χρήστη.



Εικόνα 2.2. 6: Σελίδα Συστάσεων

Το κυριότερο μέρος της σελίδας, σχετικά με το σύστημα συστάσεων, είναι η δυνατότητα βαθμολογίας του χρήστη στις συστάσεις που εμφανίζουμε. Κάθε φορά που ο χρήστης επεξεργάζεται τις αξιολογήσεις του οι προβλέψεις του συστήματος γίνονται πιο ακριβείς.

Η εφαρμογή είναι αρκετά απλή στην χρήση της, χωρίς επιπλέον πληροφορία που συνήθως μπερδεύει τους χρήστες. Η εμπιστοσύνη των χρηστών κερδίζεται αποκλειστικά από την ποιότητα των προβλέψεων. Βέβαια, η ψηφιακή απεικόνιση των καταστημάτων που υπάρχει στην βάση δεδομένων μερικές φορές μπορεί να μην αντικατοπτρίζει πιστά τα φυσικά χαρακτηριστικά του. Σε διαφορετικές περιπτώσεις, όπως για παράδειγμα μία ιστοσελίδα με ταινίες, βιβλία ή ένα ηλεκτρονικό κατάστημα στα αντικείμενα που εισάγονται οι ψευδής πληροφορίες εντοπίζονται πιο εύκολα. Οι πληροφορίες που υπάρχουν όμως για τα μέρη διασκέδασης εισάγονται συνήθως από τους ιδιοκτήτες, οπότε για σκοπούς προώθησης μπορεί να περιέχουν περισσότερες από τις απαραίτητες λεπτομέρειες.

3. Ανάλυση Συστήματος Συστάσεων

Το πρώτο βήμα σε κάθε επιστημονικό πρόβλημα στην εξόρυξη δεδομένων είναι η απεικόνιση και η προ-επεξεργασία των δεδομένων. Τα σύνολο δεδομένων (dataset) των χώρων διασκέδασης που χρησιμοποιεί η εφαρμογή Magellan προέρθει από το Places API της Google. Πρόκειται για πραγματικά δεδομένα φυσικών καταστημάτων και περιέχει μπαρ, εστιατόρια, καφετέριες, ξενοδοχεία και άλλους χώρους ψυχαγωγίας σε διάφορα μέρη της Αθήνας. Τα έτοιμα δεδομένα όμως δεν ήταν αρκετά για το φιλτράρισμα με βάση το περιεχόμενο διότι απαιτεί περιγραφή των αντικειμένων τις οποίες συμπληρώσαμε εμείς. Κατά συνέπεια, δεν ήταν δυνατό να πραγματοποιηθεί για όλους τους χώρους διασκέδασης στην Αθήνα οπότε το διαθέσιμο σύνολο δεδομένων αποτελείται από 127 εγγραφές.

Το σύστημα συστάσεων της εφαρμογής χρησιμοποιεί και φιλτράρισμα με βάση το περιεχόμενο και συνεργατικό φιλτράρισμα. Για τους σκοπούς της παρουσίασης αλλά και της περιγραφής τους, έχει γίνει διαχωρισμός των δύο τεχνικών και θα αναλυθούν μεμονωμένα παρακάτω.

3.1 Ανάλυση Συνεργατικού Φιλτραρίσματος

Για την απεικόνιση των δεδομένων χρησιμοποιούμε την εξωτερική βιβλιοθήκη της python, **pandas.** Πρώτα περνάμε τους χώρους διασκέδασης και στη συνέχεια τις αξιολογήσεις χρηστών. Το σύνολο των δεδομένων το εισάγουμε χρησιμοποιώντας την μέθοδο *read_csv().*

```
place_names = pd.read_csv("collaborative-recommender/data/actual_example_places/places.csv")
print(place_names.head())
```

Εικόνα 3.1. 1: Εισαγωγή Δεδομένων σε Python (1)

```
4  ratings_data = pd.read_csv("collaborative-recommender/data/actual_example_places/actual_ratings.csv")
5  print(ratings_data.head())
```

Εικόνα 3.1. 2: Εισαγωγή Δεδομένων σε Python (2)

Οι χώροι διασκέδασης εμφανίζονται σε αυτή την μορφή:

ID	Τίτλος	Κατηγορίες	Περιγραφή
0	Melia Athens	Restaurant | food | establishment	Savour traditional Greek flavours mingled with Mediterranean taste...
1	POSEIDON ATHENS HOTEL	Restaurant | food | establishment	Poseidon Athens Hotel provides the ultimate destination for holidays in..
2	Radisson Blu Park Hotel Athens	Bar | restaurant | food | gym | health	Our conveniently located Athens hotel sits at the edge of Pedion tou Areos one..
3	The Athenian Callirhoe Exclusive Hotel	Café | lodging | bar | restaurant | food	A unique 4 star superior Boutique Hotel in the heart of..
4	Athens Atrium	Restaurant | food | | establishment	Known for its comfort personal service and classic elegance Athens..

Πίνακας 3. 1: Χώροι Διασκέδασης

Κάθε χώρος αντιπροσωπεύεται από το όνομα του, τις κατηγορίες στις οποίες ανήκει, την περιγραφή του και το μοναδικό αναγνωριστικό αριθμό του (ID).

Οι αξιολογήσεις χρηστών εμφανίζονται σε αυτή την μορφή:

ID	userId	placeId	rating
0	605360cd4ddf96d0004d2e1b2	0	4.0
1	605360cd4ddf96d0004d2e1b2	1	5.0
2	605360cd4ddf96d0004d2e1b2	3	3.0
3	605360cd4ddf96d0004d2e1b2	4	5.0
4	605360cd4ddf96d0004d2e1b3	0	3.0

Πίνακας 3. 2: Πίνακας Αξιολογήσεων

Κάθε σειρά στο σύνολο δεδομένων αντιστοιχεί σε μία βαθμολογία. Η στήλη *userId* περιέχει το αναγνωριστικό του χρήστη που πραγματοποίησε την αξιολόγηση. Η στήλη *placeId* περιέχει το αναγνωριστικό του χώρου διασκέδασης και η στήλη *rating* περιέχει την βαθμολογία που άφησε ο χρήστης. Οι βαθμολογίες μπορούν να έχουν τιμές μεταξύ 1 και 5.

Το επόμενο βήμα είναι η συγχώνευση των δύο πινάκων. Όπως αναφέρθηκε παραπάνω το συνεργατικό φιλτράρισμα με την αλγοριθμική προσέγγιση που βασίζεται στη μνήμη εκμεταλλεύονται όλα τα αντικείμενα που έχει αξιολογήσει ο κάθε χρήστης και χρησιμοποιούν ολόκληρη την βάση δεδομένων. Η συγχώνευση πραγματοποιείται με την εντολή *merge()*.

```
12  # Merge place names and ratings
13  place_data = pd.merge(ratings_data, place_names, on='placeId')
14  place_data.head()
```

Εικόνα 3.1. 3: Συγχώνευση πινάκων

	userId	placeId	rating	title	tags	description
0	605360cd4df96d0004d2e1b2	0	4.0	Meliá Athens	lodging|restaurant|food|point_of_interest|esta...	Meliá Athens facilities cover all our guests' ...
1	605360cd4df96d0004d2e1b3	0	3.0	Meliá Athens	lodging|restaurant|food|point_of_interest|esta...	Meliá Athens facilities cover all our guests' ...
2	605360cd4df96d0004d2e1b6	0	5.0	Meliá Athens	lodging|restaurant|food|point_of_interest|esta...	Meliá Athens facilities cover all our guests' ...
3	605360cd4df96d0004d2e1b8	0	4.0	Meliá Athens	lodging|restaurant|food|point_of_interest|esta...	Meliá Athens facilities cover all our guests' ...
4	86	0	1.0	Meliá Athens	lodging|restaurant|food|point_of_interest|esta...	Meliá Athens facilities cover all our guests' ...

Εικόνα 3.1. 4: Αποτελέσματα Συγχώνευσης Πινάκων

Το αποτέλεσμα της συγχώνευσης είναι ένας ενιαίος πίνακας που συμπεριλαμβάνει τις στήλες του πίνακα 3.1 και 3.3. Αφού έχουμε τροφοδοτήσει το σύστημα με όλα τα απαραίτητα δεδομένα, είναι έτοιμο να ξεκινήσει την διαδικασία σύστασης.

Σε αυτό το σημείο, μπορούμε να εξάγουμε πληροφορίες από τα δεδομένα τα συστήματος, όπως για παράδειγμα το σύνολο τον αξιολογήσεων που έχει κάθε χώρος, τον μέσο όρο βαθμολογίας κ.α. Για να απεικονιστεί αυτή η πληροφορία πρέπει αρχικά να ομαδοποιήσουμε τους χώρους και στη συνέχει να ταξινομηθούν με βάση τις αξιολογήσεις τους.

```
15  place_data.groupby('title')['rating'].count().sort_values(ascending=False).head()
```

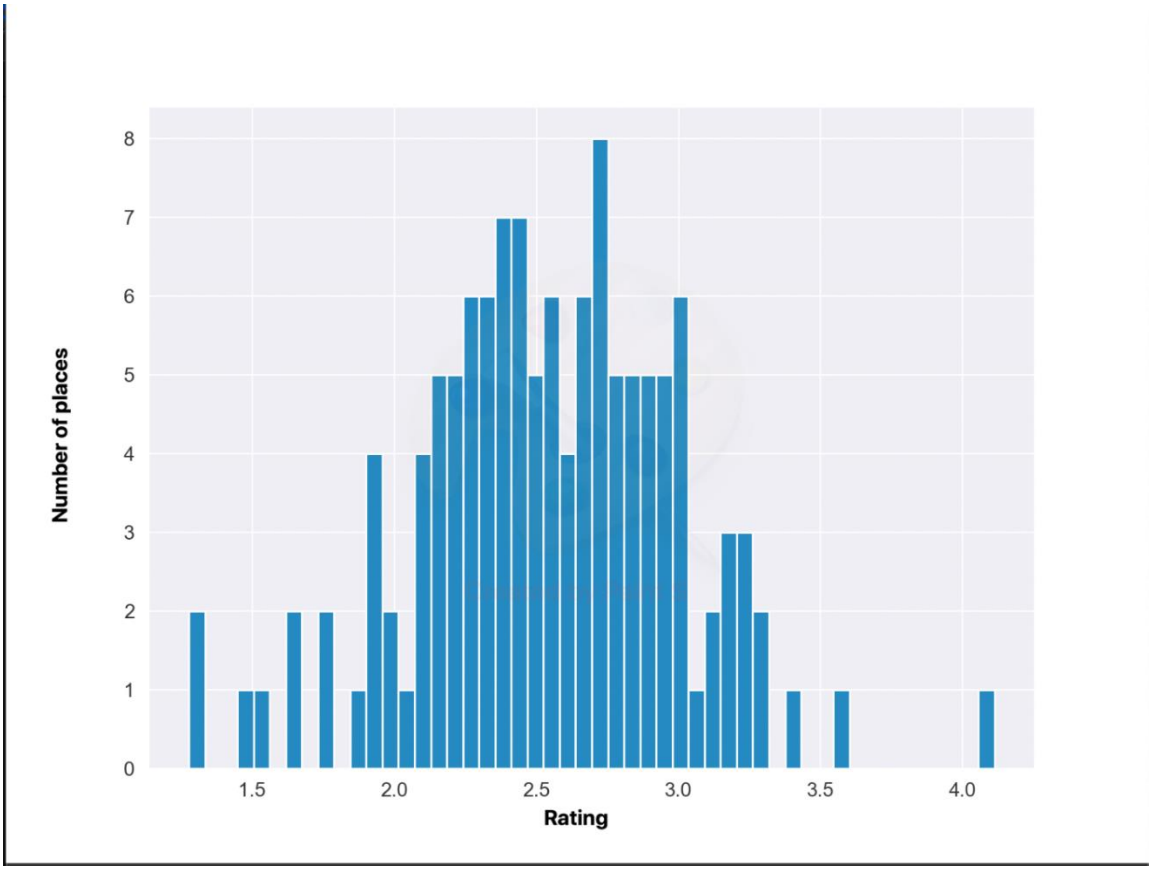
Εικόνα 3.1. 5: Ταξινόμηση Αποτελεσμάτων

Η εντολή της εικόνας 3.1.5 εμφανίζει τα παρακάτω:

Title	Number_of_ratings
TAR beer restaurant	28
Τα Πέντε Πιάτα	25
Moreno	25
FAROS HOTEL ΠΑΜΠΟΡΗΣ ΧΡΗΣΤΟΣ&ΣΙΑ Ε.Ε.	22
Radisson Blue Park Hotel Athens	22
The Party Bar	21
ROCKWOOD	21
Kibubu Music Bar	19
Στούντιο Καφέ	19
CENTRALE CAFE	19

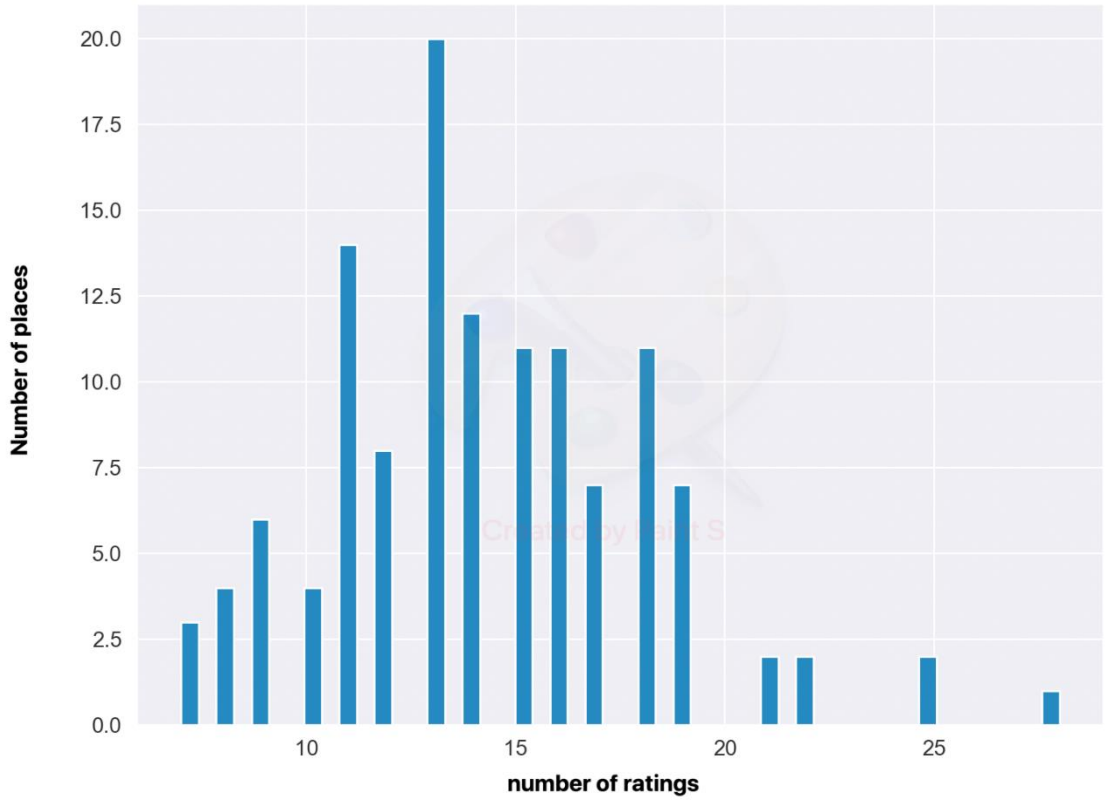
Πίνακας 3. 3: Ταξινομημένοι χώροι διασκέδασης

Έχοντας διαθέσιμες τις πληροφορίες για τις αξιολογήσεις των χώρων, μπορούμε να εμφανίσουμε την πρώτη γραφική απεικόνιση των δεδομένων χρησιμοποιώντας ένα ιστόγραμμα.



Εικόνα 3.1. 6: Ιστόγραμμα (Πλήθος Χώρων – Αξιολογήσεις)

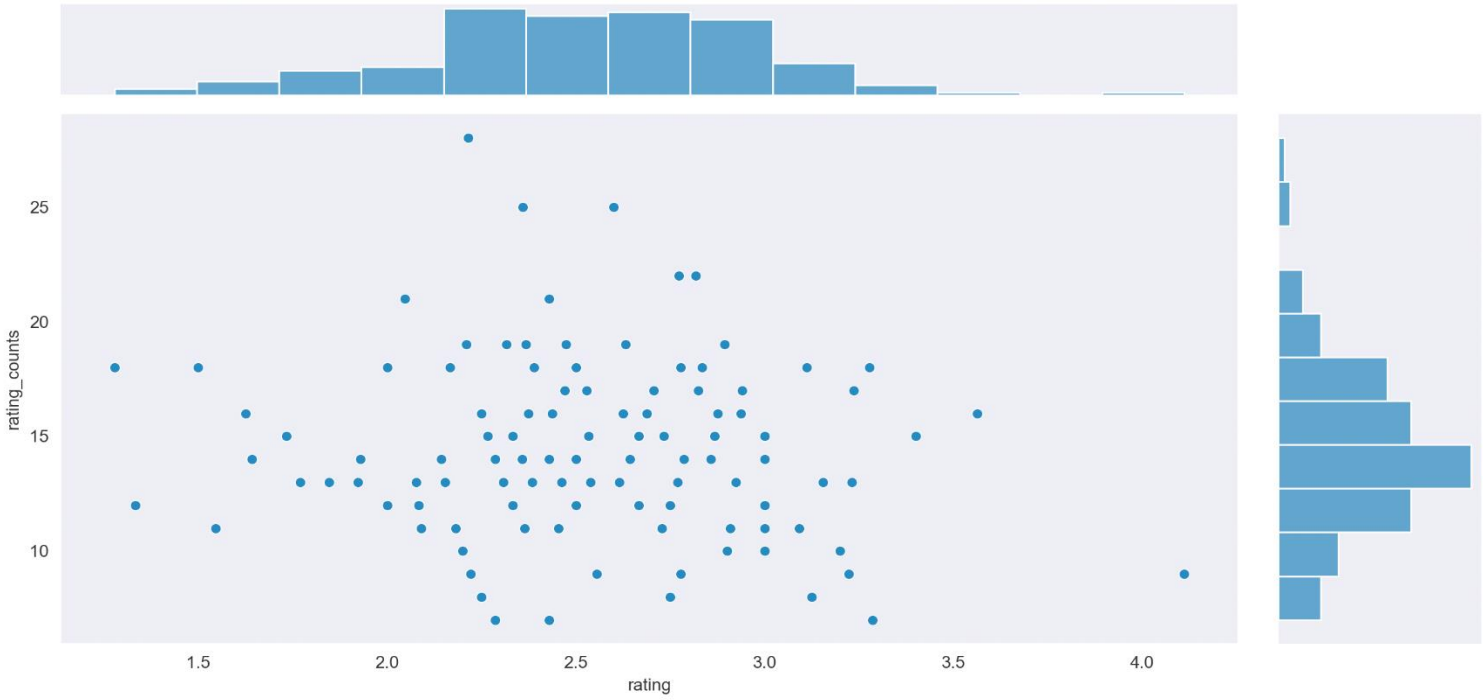
Από το ιστόγραμμα της εικόνας 3.7, συμπεραίνουμε ότι οι αξιολογήσεις έχουν μέσο όρο περίπου το 2.5 ενώ είναι κατανεμημένες κανονικά με εξαίρεση τα ακραία σημεία του γραφήματος. Συνήθως, τα δημοφιλή αντικείμενα έχουν τις υψηλότερες βαθμολογίες επειδή είναι δημοφιλή, και αξιολογούνται πιο συχνά. Το σύνολο δεδομένων που χρησιμοποιούμε όμως, είναι αρκετά μικρό για να αποδείξουμε αυτή την υπόθεση. Για να δούμε τι συμβαίνει στη δική μας περίπτωση και να εξάγουμε συμπεράσματα, πρέπει να εξετάσουμε πιο αναλυτικά τα δεδομένα των αξιολογήσεων.



Εικόνα 3.1. 7: Ιστόγραμμα (Πλήθος Χώρων – Πλήθος Αξιολογήσεων)

Το παραπάνω ιστόγραμμα απεικονίζει στον άξονα χ το πλήθος των αξιολογήσεων και στον άξονα t το πλήθος των χώρων διασκέδασης. Βλέπουμε πως τα περισσότερα καταστήματα έχουν βαθμολογηθεί 11-18 φορές. Στη συνέχεια, μένει να ελεγχθούν οι

βαθμολογίες που αντιστοιχούν σε αυτές τις αξιολογήσεις. Αυτό μπορεί να επιτευχθεί με ένα διάγραμμα διασποράς.



Εικόνα 3.1. 8: Διάγραμμα Διασποράς (Πλήθος Αξιολογήσεων – Μέσος όρος Αξιολογήσεων)

Στο διάγραμμα διασποράς της εικόνας 3.8 εμφανίζονται στον άξονα y, το πλήθος των αξιολογήσεων, ενώ στον άξονα χ οι βαθμολογίες. Παρατηρούμε λοιπόν πως όλες οι βαθμολογίες είναι αρκετά χαμηλές, πράγμα που πολύ πιθανό να μη συνέβαινε με πραγματικά δεδομένα. Το συνεργατικό φιλτράρισμα όμως, δεν χρειάζεται υψηλές βαθμολογίες. Αρκούν οι ομοιότητες που παρουσιάζουν οι χρήστες για να δημιουργηθεί η σύσταση.

Η τελική μορφή στην οποία πρέπει να φέρουμε τα δεδομένα είναι ένας πίνακας που κάθε στήλη του αντιπροσωπεύει έναν χώρο διασκέδασης και οι γραμμές του περιέχουν τις αξιολογήσεις των χρηστών. Δηλαδή για N χώρους ψυχαγωγίας και M χρήστες, χρειαζόμαστε έναν πίνακα αξιολογήσεων NxM. Αυτό γίνεται εφικτό με την παρακάτω εντολή.

```
# convert to userId x place1, place2, place3, ....., placeN
user_place_rating = place_data.pivot_table(index='userId', columns='title', values='rating')
print(user_place_rating.head(5))
```

Εικόνα 3.1. 9: Εντολή εμφάνισης πίνακα χρηστών / χώρων

	21 Restaurant	ALTO	Akrotiri	Ariston	Crawl
0	NaN	0.66667	NaN	NaN	2.5
1	NaN	NaN	NaN	NaN	2.0
10	NaN	NaN	NaN	NaN	NaN
100	NaN	NaN	NaN	NaN	NaN
101	NaN	NaN	NaN	4.0	NaN

Πίνακας 3. 4: Πίνακας Χρηστών / Χώρων Διασκέδασης

Ο πίνακας 3.4 απεικονίζει ένα μέρος του πραγματικού πίνακα δεδομένων. Παρατηρούμε ότι πολλές εγγραφές είναι κενές από το δείγμα, δηλαδή οι χρήστες δεν έχουν αξιολογήσει αυτά τα μέρη. Αν υπάρχει μεγάλη έλλειψη αξιολογήσεων σε ένα σύνολο δεδομένων, δημιουργείτε το πρόβλημα διασποράς δεδομένων (sparsity problem) που αναφέρθηκε στο κεφάλαιο 1.3.

Για να αναλυθεί καλύτερα ο αλγόριθμος θα πάρουμε για παράδειγμα το ξενοδοχείο «Melia Athens» επειδή έχει τον μεγαλύτερο αριθμό αξιολογήσεων. Αρχικά εμφανίζουμε ένα δείγμα από τις αξιολογήσεις που του αντιστοιχούνε.

```
Melia_ratings = user_place_rating['Meliá Athens']
```

Εικόνα 3.1. 10: Αξιολογήσεις από «Melia Athens»

userId	Melia Athens
0	NaN
1	5.0
10	NaN
100	5.0
101	NaN

Πίνακας 3. 5: : Αξιολογήσεις από «Melia Athens»

Στόχος μας, είναι η εύρεση χώρων διασκέδασης που παρουσιάζουν μεγάλο βαθμό ομοιότητας με το ξενοδοχείο Melia. Στον πίνακα 3.4 εφαρμόζουμε την συνάρτηση *corrWith()* η οποία συγκρίνει όλα τα μέρη ψυχαγωγίας που υπάρχουν στον πίνακα με το επιλεγμένο και επιστρέφει την συσχέτιση τους.

```
places_like_Melia = user_place_rating.corrwith(Melia_ratings)
corr_Melia_athens = pd.DataFrame(places_like_Melia, columns=['Correlation'])
corr_Melia_athens.dropna(inplace=True)
corr_Melia_athens = corr_Melia_athens.sort_values('Correlation', ascending=False)
```

Εικόνα 3.1. 11: Υπολογισμός Βαθμού Συσχέτισης

Τίτλος	Συσχέτιση
Malvazia	1.0
Τα Πέντε Πιάτα	1.0
Saloon Piano Restaurant	1.0
Elaea Mezedadiko	1.0
Moreno	1.0

Πίνακας 3. 6: Συσχέτιση χώρων διασκέδασης με «Melia Athens»

Στον παραπάνω πίνακα παρατηρούμε πως η συσχέτιση όλων των συγκρινόμενων χώρων διασκέδασης έχει τιμή 1 που σημαίνει ότι θεωρητικά αντικατοπτρίζουν τις προτιμήσεις του χρήστη. Όμως, στην πραγματικότητα τα αποτελέσματα του πίνακα 3.6 είναι λανθασμένα διότι δεν έχουμε υπολογίσει το πλήθος των αξιολογήσεων του κάθε χώρου. Όταν προσθέσουμε και αυτό στην συνάρτηση εμφανίζονται οι σωστές προβλέψεις.

```
corr_Melia_athens[corr_Melia_athens['rating_counts'] > 18].sort_values('Correlation', ascending=False).head()
```

Εικόνα 3.1. 12*:* Προσθήκη πλήθους αξιολογήσεων

Με την πάνω γραμμή κώδικα, ορίζουμε το κατώτατο όριο των αξιολογήσεων που θέλουμε να κρατήσουμε (18) και εμφανίζουμε σε φθίνουσα σειρά ως προς τον βαθμό συσχέτισης, τον κάθε χώρο ψυχαγωγίας.

Τίτλος	**Συσχέτιση**	**Αριθμός Αξιολογήσεων**
Τα Πέντε Πιάτα	1	25
Moreno	1	25
TAR beer restaurant	1	28
POSEIDON ATHENS HOTEL	0.94	18
Radisson Blu Park Hotel Athens	0.88	22

Πίνακας 3. 7: Τελική συσχέτιση αποτελεσμάτων

Παρατηρούμε πως τα αποτελέσματα έχουν τροποποιηθεί. Θέσαμε το κατώτερο όριο αξιολογήσεων να είναι 18 έχοντας λάβει υπ' όψη τα γραφήματα 3.1.7 & 3.1.8 που δείχνουν ότι ο μέγιστος αριθμός αξιολογήσεων είναι το 28. Η εικόνα που μας δίνει ο μέσος όρος βαθμολογίας του κάθε χώρου ψυχαγωγίας είναι ανάλογη με τον αριθμό των αξιολογήσεων που διαθέτει.

Τα παραπάνω αποτελούν ένα παράδειγμα που αντιπροσωπεύει ένα μέρος του αλγόριθμου συνεργατικού φιλτραρίσματος που χρησιμοποιεί το σύστημα της εφαρμογής Magellan. Συγκεκριμένα ο πίνακας 3.7 θα μπορούσε να εμφανιστεί ως σύσταση της εφαρμογής αν ο χρήστης είχε αξιολογήσει μόνο το ξενοδοχείο Melia Athens. Αυτή η διαδικασία γίνεται για κάθε χώρο διασκέδασης που έχει αξιολογήσει ο χρήστης, και οι πληροφορίες που εξάγονται συγκεντρώνονται σε μία ενιαία σύσταση που με την σειρά της εμφανίζεται στο γραφικό περιβάλλον της εφαρμογής.

3.2 Ανάλυση Φιλτραρίσματος με βάση το Περιεχόμενο

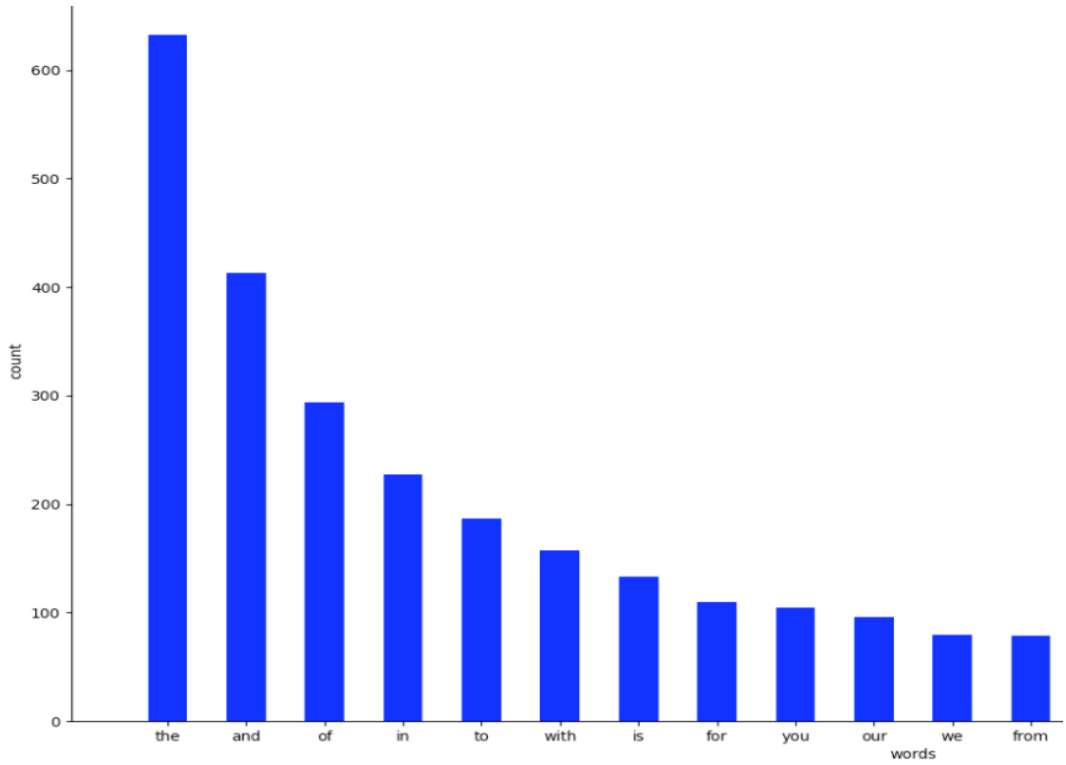
Στο φιλτράρισμα με βάση το περιεχόμενο, είναι απαραίτητη η λεπτομερής περιγραφή των αντικειμένων. Το σύστημά μας εξαρτάται σε μεγάλο βαθμό από τον καθορισμό ενός κατάλληλου μέτρου ομοιότητας οπότε επιλέξαμε την συνημιτονική ομοιότητα. Η λογική που ακολουθούμε είναι η τροποποίηση της περιγραφής με διάφορους αλγόριθμους εξόρυξης κειμένου, που επιλέγουν μόνο τις λέξεις κλειδιά και στη συνέχεια η εύρεση ομοιοτήτων μεταξύ των χώρων που έχει αξιολογήσει ο χρήστης και των υπόλοιπων καταστημάτων.

Για αρχή, όπως και στο συνεργατικό φιλτράρισμα εισάγουμε το σύνολο δεδομένων στο σύστημα. (εικόνα 3.1, πίνακας 3.1). Μια πλήρης περιγραφή ενός χώρου διασκέδασης έχει την εξής μορφή:

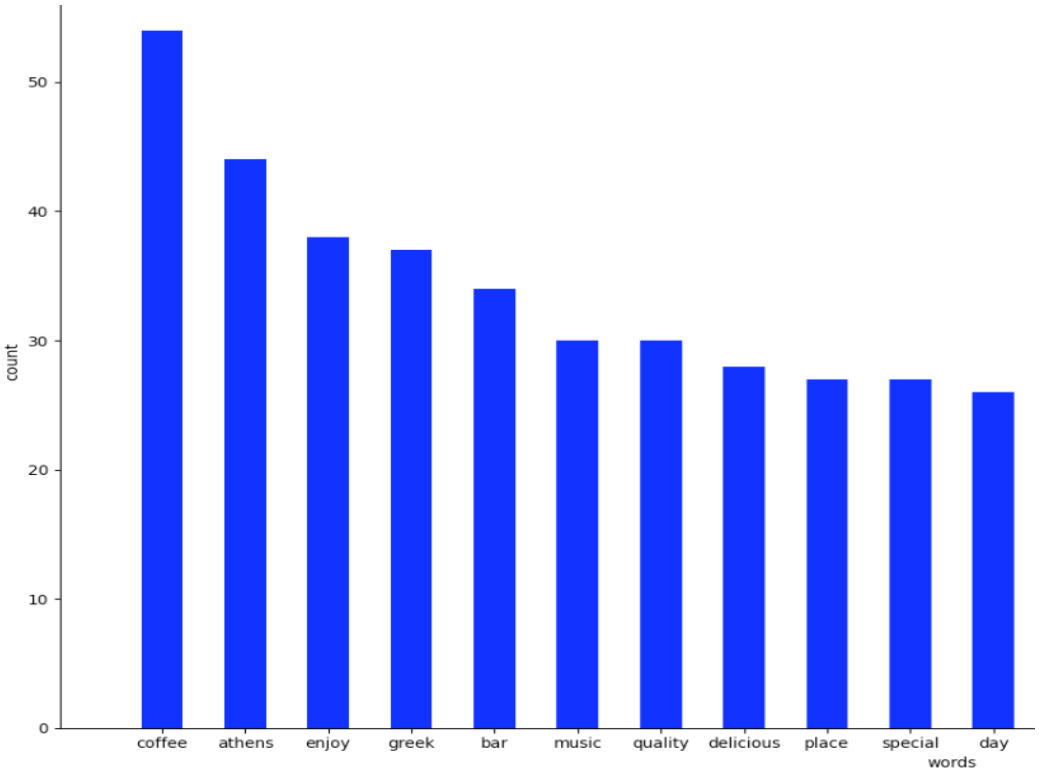
This unpretentious hotel is 2 km from the Port of Piraeus a 3-minute walk from the Archaeological Museum of Piraeus and 1 km from the Maritime Museum of Greece. The simple rooms have traditional décor free Wi-Fi flat-screen TVs and minibars and some have balconies. Suites have extra spa baths with tea and coffee making facilities. Room service is available. Free amenities include parking and a buffet breakfast. There is also a lobby bar and a quaint restaurant as well as a meeting room.

Name: Savoy Hotel

Για να καταλήξουμε στις λέξεις κλειδιά που χαρακτηρίζουν το κάθε μαγαζί, η κάθε περιγραφή πρέπει να περάσει από μερικά φίλτρα. Το πρώτο από αυτά είναι η εύρεση και αφαίρεση των συνδετικών λέξεων (stop-words) όπως άρθρα, σύνδεσμοι κτλ.



Εικόνα 3.2. 1: Πιο χρησιμοποιημένες λέξεις (1)



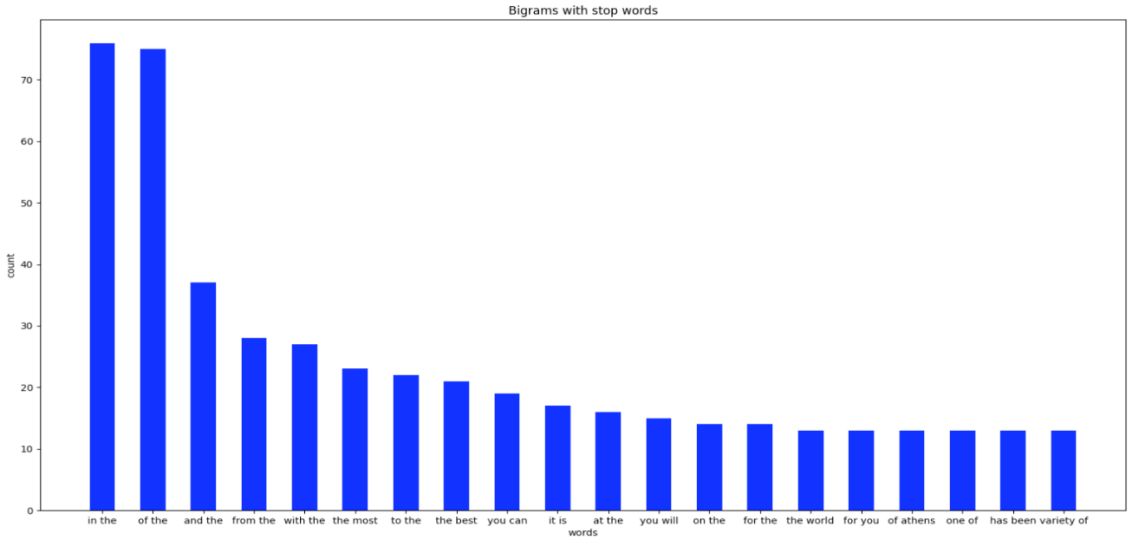
Εικόνα 3.2. 2: Πιο χρησιμοποιημένες λέξεις (2)

Στα παραπάνω γραφήματα (εικόνα 3.10 & 3.11) εμφανίζονται οι περισσότερο χρησιμοποιημένες λέξεις και η συχνότητα στην οποία εμφανίζονται στις περιγραφές των χώρων ψυχαγωγίας πριν αφαιρέσουμε τις συνδετικές λέξεις και μετά. Για την αφαίρεση τους χρησιμοποιήθηκε ο παρακάτω κώδικας.

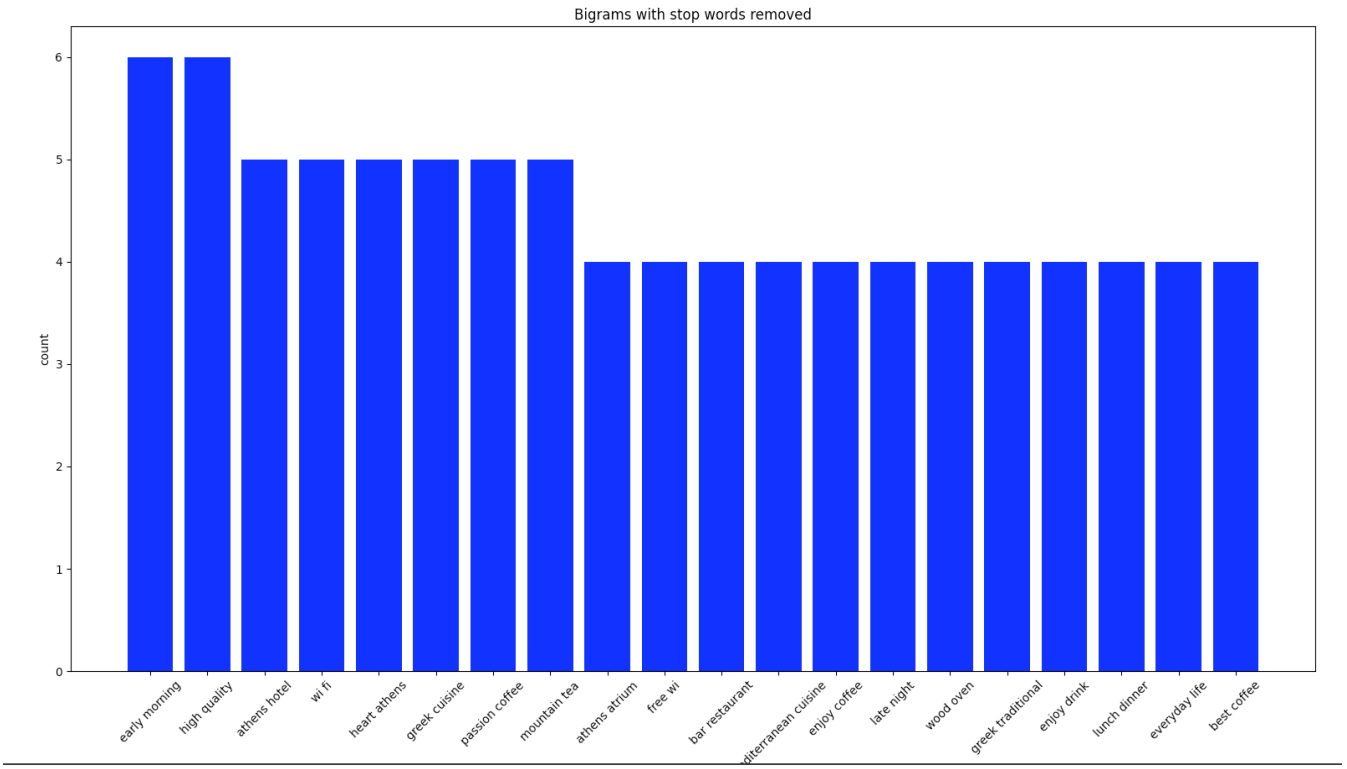
```
59  def get_top_n_words_no_stops(corpus, n=None):
60      vec = CountVectorizer(stop_words='english').fit(corpus)
61      bag_of_words = vec.transform(corpus)
62      sum_words = bag_of_words.sum(axis=0)
63      words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
64      words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
65      return words_freq[:n]
```

Εικόνα 3.2. 3: Αφαίρεση Συνδετικών Λέξεων

Οι συνδετικές λέξεις (stop-words) και η αφαίρεση τους, είναι πολύ συχνό φαινόμενο σε εφαρμογές που χρησιμοποιούν ανάλυση κειμένου. Ένα κείμενο χωρίς αυτές, διατηρεί το σκοπό του, όπως για παράδειγμα εάν θέλαμε να αναζητήσουμε το εξής: «Πώς να βρω πληροφορίες για τα συστήματα συστάσεων» η μηχανή αναζήτησης θα πρέπει να αφαιρέσει το «πως να» και το «για τα» και να επικεντρωθεί στις υπόλοιπες λέξεις. Κατά συνέπεια, τα αποτελέσματα είναι αρκετά πιο εύστοχα απ' ότι θα ήταν αν αναζητούσε και τις συνδετικές λέξεις. Σε αυτό το σημείο μπορούμε να ελέγξουμε τι πληροφορίες μας παρέχουν τα πιο συχνά ζεύγη λέξεων (bigrams) πριν και μετά την αφαίρεση των συνδετικών λέξεων.



Εικόνα 3.2. 4: Πιο συχνά ζεύγη λέξεων πριν την αφαίρεση των stop-words



Εικόνα 3.2. 5: : Πιο συχνά ζεύγη λέξεων μετά την αφαίρεση των stop-words

Στο πρώτο γράφημα (εικόνα 3.12) τα ζεύγη λέξεων με την μεγαλύτερη συχνότητα εμφάνισης, με πάνω από 70 εμφανίσεις έκαστος, ήταν τα “in the” και “of the” που δε μας προσφέρουν καμία πληροφορία. Αφού αφαιρέσαμε τις συνδετικές λέξεις (εικόνα 3.13) τα πιο πολυσύχναστα ζεύγη λέξεων έγιναν τα “early morning” και “high quality”. Αυτά τα ζεύγη είναι πολύ πιο πιθανό να χαρακτηρίσουν ένα μέρος σωστά. Εάν το σύνολο δεδομένων στην διάθεση μας ήταν μεγαλύτερο θα μπορούσαμε να εξάγουμε περισσότερες πληροφορίες, διότι η περιγραφή του κάθε χώρου στοχεύει στην προώθηση του και συχνά χρησιμοποιούνται παρόμοιες φράσεις-λέξεις κλειδιά. Για παράδειγμα ένα ζεύγος λέξεων που στοχεύει στον τουριστικό πληθυσμό είναι το «κέντρο Αθήνας».

Στο τελικό βήμα προ-επεξεργασίας δεδομένων, πριν την πρόβλεψη, αφαιρούμε από το κείμενο όλους τους ειδικούς χαρακτήρες και οτιδήποτε άλλο μπορεί να επηρεάσει τα αποτελέσματα χρησιμοποιώντας την παρακάτω συνάρτηση. Οι περιγραφές οι οποίες έχουμε είναι αρκετά καθαρές ήδη αλλά δε βλάπτει ένας επιπλέον έλεγχος.

```
REPLACE_BY_SPACE_RE = re.compile('[/(){}\[\]\|@,;]')
BAD_SYMBOLS_RE = re.compile('[^0-9a-z #+_]')
STOPWORDS = set(stopwords.words('english'))

def clean_text(text):
    """
        text: a string

        return: modified initial string
    """
    # lowercase text
    text = text.lower()

    # replace REPLACE_BY_SPACE_RE symbols by space in text.
    # substitute the matched string in REPLACE_BY_SPACE_RE with space.
    text = REPLACE_BY_SPACE_RE.sub(' ', text)

    # remove symbols which are in BAD_SYMBOLS_RE from text.
    # substitute the matched string in BAD_SYMBOLS_RE with nothing.
    text = BAD_SYMBOLS_RE.sub('', text)

    # remove stopwords from text
    text = ' '.join(word for word in text.split() if word not in STOPWORDS)

    return text
```

Εικόνα 3.2. 6: Συνάρτηση clean_text()

Η κατανομή λέξεων για τους χώρους ψυχαγωγίας που διαθέτει η εφαρμογή είναι η εξής:

- Σύνολο περιγραφών: **117**
- Μέσος όρος λέξεων: **81.8**
- Μέγιστος αριθμός λέξεων: **323**
- Ελάχιστος αριθμός λέξεων: **0**

Αφού έχουν τροποποιηθεί οι περιγραφές των αντικειμένων (στην περίπτωση αυτή των χώρων διασκέδασης) το σύστημα είναι σε θέση να ξεκινήσει την διαδικασία σύστασης.

Στόχος του συστήματος είναι για ακόμη μία φορά η εύρεση ομοιοτήτων μεταξύ των αντικειμένων. Από την στιγμή που χρησιμοποιούμε ανάλυση κειμένου εφαρμόζουμε την μέθοδο **TF-IDF** (term frequency-inverse document frequency)**.** Η TF-IDF είναι μια αριθμητική στατιστική που αποσκοπεί να αντικατοπτρίζει πόσο σημαντική είναι μια λέξη σε ένα έγγραφο ή σε μια συλλογή. Συχνά χρησιμοποιείται ως παράγοντας σταθμίσεως στις αναζητήσεις ανάκτησης πληροφοριών, εξόρυξης κειμένου και μοντελοποίησης χρηστών [Δ. Χρυσινά 2018]. Αναλυτικότερα, η μέθοδος αυτή βοηθάει στη μετατροπή των κειμένων σε αριθμητικές τιμές.

$$\text{Term Frequency (TF)} = \frac{\text{Frequency of a term in the document}}{\text{Total number of terms in documents}}$$

$$\text{Inverse Document Frequency(IDF)} = log\frac{\text{total number of documents}}{\text{number of documents with term t}}$$

```
df.set_index('title', inplace=True)
tf = TfidfVectorizer(analyzer='word', ngram_range=(1, 3),
                     min_df=0, stop_words='english')
tfidf_matrix = tf.fit_transform(df['description_clean'])
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

Εικόνα 3.2. 7: Δημιουργία πίνακα TF-IDF

Με την χρήση του παραπάνω αποσπάσματος κώδικα (εικόνα 3.2.7), δημιουργούμε έναν πίνακα TF-IDF από τις λέξεις της περιγραφής μεμονωμένες, σε ζεύγη και σε τριάδες. Κατόπιν, χρησιμοποιούμε την συνάρτηση *linear_kernel()* της βιβλιοθήκης sklearn η οποία στην περίπτωσή μας, υπολογίζει την συνημιτονική ομοιότητα των χώρων διασκέδασης. Τέλος, έχουμε δημιουργήσει μια συνάρτηση η οποία δέχεται ως είσοδο της, τον τίτλο του χώρου διασκέδασης καθώς και τον πίνακα ομοιοτήτων και εμφανίζει σαν αποτέλεσμα τα 10 μέρη με τον υψηλότερο βαθμό συσχέτισης.

```python
def predict(name, cosine_similarities=cosine_similarities):
    recommended_places = []

    # gettin the index of the hotel that matches the name
    try:
        idx = indices[indices == name].index[0]

        # creating a Series with the similarity scores in descending order
        score_series = pd.Series(
            cosine_similarities[idx]).sort_values(ascending=False)

        # getting the indexes of the 10 most similar hotels except itself
        top_10_indexes = list(score_series.iloc[1:11].index)
        # populating the list with the names of the top 10 matching hotels
        for (i, score) in zip(top_10_indexes, score_series):
            recommended_places.append({
                "placeId": list(df.placeId)[i],
                "title": list(df.index)[i],
                "correlation": score
            })
    except:
        print('Something went wrong')
    finally:
        return recommended_places
```

Εικόνα 3.2. 8: Δημιουργία Συστάσεων (Content-Based)

Ας πάρουμε για παράδειγμα το μπαρ “Silly Wizards Pub”. Όπως και στον αλγόριθμο συνεργατικού φιλτραρίσματος τα αποτελέσματα είναι της μορφής:

Τίτλος	Συσχέτιση
KingSize Beer House	0.99
Litharia	0.22
Tender Bar	0.18
Attalos	0.17
BEER GARDEN RITTERBURG	0.14
Nice n easy	0.13

Πίνακας 3. 8: Πίνακας Συσχέτισης Content-Based

Παρατηρούμε πως ο βαθμός συσχέτισης κυμαίνεται σε χαμηλούς αριθμούς, δε σημαίνει όμως ότι το σύστημα δε λειτουργεί σωστά. Όπως είδαμε στα γραφήματα της προηγούμενης ενότητας ο μέσος όρος των αξιολογήσεων είναι χαμηλός και σε συνδυασμό με περιορισμένο σύνολο δεδομένων τα αποτελέσματα της πρόβλεψης είναι αρκετά ικανοποιητικά. Επιπλέον, όπως αναφέρθηκε παραπάνω, τα συστήματα συστάσεων παράγουν πιο ακριβή αποτελέσματα όταν υπάρχουν πολλές αξιολογήσεις.

Όπως και στο προηγούμενο παράδειγμα του συνεργατικού φιλτραρίσματος, έτσι και σε αυτό, η διαδικασία που αναλύθηκε πραγματοποιείται για κάθε χώρο διασκέδασης που έχει αξιολογήσει ο χρήστης.

4. Επίλογος

Ο στόχος της παρούσας πτυχιακής εργασίας, είναι η μελέτη και ανάλυση των συστημάτων σύστασης αλλά και η υλοποίηση της εφαρμογής Magellan χρησιμοποιώντας πραγματικά δεδομένα. Δυστυχώς, η συλλογή αυτών των δεδομένων και συγκεκριμένα η περιγραφή τους, είναι μια αρκετά δύσκολη και χρονοβόρα διαδικασία. Ο όγκος του συνόλου δεδομένων, στη διάθεση μας ήταν αρκετά μικρός για να χρησιμοποιηθούν πιο εξειδικευμένες τεχνικές όπως, αλγόριθμοι βασιζόμενοι στο μοντέλο. Σε μία πραγματική εφαρμογή που χρησιμοποιεί συστήματα συστάσεων, το πλήθος το δεδομένων αυξάνεται συνεχώς και ταυτόχρονα η ποιότητα των συστάσεων.

Σχετικά με τα παραδείγματα των τεχνικών φιλτραρίσματος που παρουσιάστηκαν στο τελευταίο κεφάλαιο, θεωρούμε πως το φιλτράρισμα με βάση το περιεχόμενο μας επιστρέφει πιο εύστοχα αποτελέσματα. Στην προσέγγιση συνεργατικού φιλτραρίσματος που εφαρμόστηκε, υπάρχει μεγάλος βαθμός διασποράς δεδομένων, όχι αρκετός για να μην είμαστε σε θέση να εξάγουμε προβλέψεις αλλά αρκετός για ρίξει την ακρίβεια των συστάσεων.

Τέλος, η εφαρμογή Magellan εάν ήταν μέρος της αγοράς, πιστεύουμε πως οι καλύτερη προσέγγιση για το σύστημα συστάσεων της θα ήταν ένα υβριδικό σύστημα, που θα χρησιμοποιούσε και φιλτράρισμα με βάση το περιεχόμενο αλλά και συνεργατικό, σε έναν συνδυασμό αλληλουχίας ή meta-level. Αρχικά για να εξαλειφθούν μερικά από τα

μειονεκτήματα των μεμονωμένων μεθόδων, αλλά και με την λογική ότι τα μέρη που θα προτείνονται θα είναι μεν όμοια με τις προτιμήσεις του χρήστη αλλά ταυτόχρονα δε θα υπάρχει κορεσμός, και θα προτείνονται γνωστά και άγνωστα μέρη.

Παράρτημα Α’

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

Frontend

https://github.com/NMantis/magellan-ui

Backend

https://github.com/Billzg13/magellan-api

Service

https://github.com/Billzg13/python-recommender

Βιβλιογραφία

[1] J.S. Breese, D.Heckerman, and C.Kadie. *Empirical analysis of predictive algorithms for collaborative filtering*.

[2] M.Deshpande and G. Karypis. *Item-based top-n recommendation algorithms*.

[3] B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Reidl. *Item-based collaborative filtering recommendation algorithms*.

[4] Burke, R. (2000). *Hybrid Recommender Systems: Survey and Experiments*.

[5] Mohamed, Marwa & Khafagy, Mohamed & Ibrahim, Mohamed. (2019). Recommender Systems Challenges and Solutions Survey.

[6] Khusro, Shah & Ali, Zafar & Ullah, Irfan. (2016). Recommender Systems: Issues, Challenges, and Research Opportunities.

[7] Spring Framework Documentation at https://docs.spring.io/spring-framework/docs/current/reference/html/

[8] Πρατικάκης Εμμανουήλ – Ανάπτυξη ενός συστήματος εξατομικευμένων συστάσεων για ηλεκτρονικές κρατήσεις ξενοδοχείων βασισμένο στη πολυκριτήρια ανάλυση αποφάσεων (2017)

[9] Manos Papagelis – Crawling the algorithmic foundations of recommendation technologies (2005)

[10] Burke, Robin. (2000). Knowledge-Based Recommender Systems. Encyclopedia of library and information systems.

[11] Pagare, Reena & Shinde, Anita. (2012). A Study of Recommender System Techniques. International Journal of Computer Applications.

[12] Raghuwanshi, Sandeep & Pateriya, R.. (2019). Recommendation Systems: Techniques, Challenges, Application, and Evaluation: SocProS 2017, Volume 2