Ατομική Διπλωματική Εργασία

ΚΑΤΑΣΚΕΥΗ ΣΥΣΤΗΜΑΤΟΣ ΣΥΣΤΑΣΕΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΜΕ ΤΗ ΧΡΗΣΗ ΑΙΣΘΗΤΗΡΩΝ BLUETOOTH

Μάριος Πιερής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Κατασκευή συστήματος συστάσεων σε πραγματικό χρόνο με τη χρήση αισθητήρων Bluetooth.

Μάριος Πιερής

Επιβλέπων Καθηγητής
Δρ. Γιώργος Πάλλης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2016

Ευχαριστίες

Αρχικά, θα ήθελα να απευθύνω θερμές ευχαριστίες στον επιβλέπων καθηγητή μου Δρ. Γιώργο Πάλλη ο οποίος μου εμπιστεύτηκε την έρευνα και ανάπτυξη του θέματος της διπλωματικής εργασίας. Τον ευχαριστώ για την άριστη συνεργασία που είχαμε καθ' όλη τη διάρκεια της διπλωματικής εργασίας και για την ανεκτίμητη βοήθεια που μου έχει προσφέρει. Επιπλέον, με έχει βοηθήσει να κατανοήσω τα καλά που θα μου προσφέρει αυτή η διπλωματική στο παρόν, στο μέλλον και ιδιαίτερα στην επαγγελματική μου κατάρτιση και ότι με κόπο, μόχθο και υπομονή θα δούμε ένα πιο λαμπρό μέλλον.

Στην συνέχεια, να ευχαριστήσω το διδακτορικό φοιτητή και φίλο Στέφανο Αντάρη για την καθοδήγηση, την υποστήριξη του σε στιγμές αδιεξόδου και τις πολύτιμες γνώσεις και συμβουλές που μου έχει δώσει κατά τη διάρκεια της περιόδου της διπλωματικής εργασίας.

Τέλος, σημαντικό στήριγμα αποτελούν οι φίλοι και η οικογένεια οι οποίοι μου έχουν συμπαρασταθεί σε όλη τη διάρκεια, εμψυχώνοντας με στις δύσκολες στιγμές.

Περίληψη

Στις μέρες φαίνεται ότι υπάρχει μεγάλο ενδιαφέρον για δημιουργία συστημάτων συστάσεων. Η ανάλυση των δεδομένων σε πληροφορίες και μετέπειτα σε γνώση αποτελεί πολύ σημαντική τεχνική, καθώς με αυτό τον τρόπο μπορούμε να παρατηρήσουμε διάφορες συμπεριφορές και να καταλήξουμε σε βοηθητικά συμπεράσματα. Μετέπειτα μπορούμε να εκμεταλλευτούμε αυτά τη γνώση για να προβλέψουμε τι είναι πιθανό να ενδιαφέρει ένα άτομο στο μέλλον με βάση το παρόν και τη σύγκριση της συμπεριφοράς του με άλλα άτομα, όπου συμπεριφορά εννοούμαι το προφίλ που το αντικατοπτρίζει.

Η εποχή της πληροφορίας και της τεχνολογικής ανάπτυξης μπορεί να αξιοποιηθεί με αποδοτικό τρόπο ώστε να μελετηθεί και να υλοποιηθεί σε ικανοποιητικό βαθμό ένα σύστημα συστάσεων του οποίου στόχος θα είναι να προτείνει διάφορα αντικείμενα σε διάφορους χρήστες. Ένα καλό σύστημα εξαρτάται από την απόδοση του ως προς την ορθότητα, το χρόνο εκτέλεσης και γενικά από το απλό γραφικό περιβάλλον με το οποίο οι χρήστες θα μπορούν να αλληλεπιδρούν με αυτό.

Μέσα από τη περιγραφή της διπλωματικής εργασίας αναγράφουμε τους αλγορίθμους που έχουν χρησιμοποιηθεί για να πετύχουμε το στόχο μας καθώς και τις τεχνικές και τις προσεγγίσεις τις οποίες ακολουθήσαμε. Επίσης, μελετήθηκε η σύγκριση μεταξύ διαφόρων τεχνικών, οι λόγοι επιλογής αυτών που χρησιμοποιούμαι και πως θα τις αξιοποιήσουμε για το σκοπό που θέλουμε.

Στη συνέχεια κατασκευάσαμε τον αλγόριθμο συστάσεων του οποίο έχουμε επικυρώσει την ορθότητα του, τη προσαρμοστικότητα του σε διάφορα σενάρια και την απόδοση του σε χρόνο. Ο τρόπος με τον οποίο έγινε αυτό είναι με την διεξαγωγή πειραμάτων τα οποία αφορούσαν δεδομένα τα οποία έχουν συλλεχθεί σε πραγματικές συνθήκες.

Τέλος, αναγράφουμε τα τελικά συμπεράσματα, αλλά και τις μελλοντικές εργασίες που μπορούν να λάβουν χώρα ώστε το σύστημα να καταστεί πλήρως λειτουργήσιμο και παρέχοντας επιπλέον ευκολίες οι οποίες μπορούν να προστεθούν.

Περιεχόμενα

Κεφάλαιο **1** **Εισαγωγή Διπλωματικής Εργασίας**……..…..……**1**
1.1 Γενική εισαγωγή 1
1.2 Κίνητρο διπλωματικής εργασίας 2
1.3 Στόχος διπλωματικής εργασίας 4
1.4 Συνεισφορά διπλωματικής εργασίας 6
1.5 Δομή διπλωματικής εργασίας 7

Κεφάλαιο **2** **Περιγραφή προβλήματος και προτεινόμενη προσέγγιση**……...……**10**
2.1 Περιγραφή προβλήματος 11
2.2 Προτεινόμενη προσέγγιση λύσης του προβλήματος 12
2.2.1 Περιεκτικές φάσεις αρχιτεκτονικής Συστήματος 14
2.3 Σύστημα Συστάσεων 14
2.3.1 Λόγοι επιλογής Apache Spark 15
2.3.2 Λόγοι επιλογής Apache Kafka 15
2.3.3 Απαιτήσεις του υπό κατασκευή Αλγορίθμου 16
2.3.4 Αλγόριθμοι Συστάσεων 17
2.3.5 Λόγοι επιλογής των Αλγορίθμων Συστάσεων 18
2.3.6 Αλγόριθμοι Συστάσεων «Collaborative Filtering» 18
2.3.6.1 «Collaborative Filtering» με βάση μοντέλο 19
2.3.6.2 Λόγοι επιλογής Αλγορίθμων Συστάσεων με βάση μοντέλο 19
2.3.7 Εναλλακτικοί τύποι για Αλγορίθμους Συστάσεων 19
2.3.7.1 Αλγόριθμοι Συστάσεων «Content Based» 19
2.3.8 Μηχανική Μάθηση 20
2.3.8.1 Μείωση διαστάσεων προβλήματος 20
2.3.8.2 «MLlib» βιβλιοθήκη στο Apache Spark 20
2.3.8.3 «Alternative Least-Square» προσέγγιση 21
2.4 Χαρακτηριστικά Αλγορίθμου Συστάσεων 22
2.4.1 Μοντέλο Αλγορίθμου Συστάσεων 22
2.4.2 Εκπαίδευση μοντέλου του Αλγορίθμου Συστάσεων 23
2.4.2.1 Παράμετροι της «Alternative Least-Square» προσέγγισης 25
2.4.2.2 Τεχνική εκπαίδευσης μοντέλου 25

2.4.3 «Cold Start» πρόβλημα στους Αλγορίθμους Συστάσεων 21
2.4.3.1 Τρόπος αντιμετώπισης «Cold Start» προβλήματος 22

Κεφάλαιο **3** **Απαιτούμενη Γνώση και Τεχνολογίες**.....**26**
3.1 Λογισμικό ανάπτυξης 26
3.1.1 Java 26
3.1.2 Spark 27
3.1.2.1 Spark Streaming 28
3.1.3 Kafka 28
3.2 Απαιτούμενες τεχνολογίες 29
3.2.1 MySQL 29
3.2.2 PHP 30
3.2.3 JSON 31
3.3 Περιβάλλον Ανάπτυξης 31
3.3.1 Eclipse 31
3.3.2 Android Studio 31
3.4 Διεπαφές προγραμματισμού εφαρμογών(API) 32
3.4.1 Facebook API 32
3.4.2 Android API 33
3.4.3 Estimote Beacon API 34
3.5 Hadoop 34
3.5.1 Αρχιτεκτονική Map-Reduce 35
3.5.2 Αρχιτεκτονική HDFS 36

Κεφάλαιο **4** **Αρχιτεκτονική Συστήματος** **37**
4.1 Ορισμός αρχιτεκτονικής συστήματος: 37
4.2 Βασικά συστατικά συστήματος 37
4.3 Σχήμα αρχιτεκτονικής συστήματος 38
4.4 Επεξήγηση αρχιτεκτονικής συστήματος 40
4.4.1 Επεξήγηση των αισθητήρων που χρησιμοποιήθηκαν 40
4.4.2 Επεξήγηση των οντοτήτων 41
4.4.3 Επεξήγηση των φάσεων / λειτουργιών 44
4.4.4 Βάση δεδομένων 47

Κεφάλαιο **5** **Υλοποίηση αλγορίθμου συστάσεων και Android εφαρμογής**……...**49**
5.1 Υλοποίηση Αλγορίθμου Συστάσεων 49
5.1.1 Εισαγωγή Αλγορίθμου Συστάσεων 49
5.1.2 Περιγραφή υλοποίησης σε φάσεις 50
5.2 Υλοποίηση Android εφαρμογής 53
5.2.1 Εισαγωγή Android εφαρμογής 53
5.2.2 Λειτουργίες Android εφαρμογής 54
5.2.3 Επεξήγηση λειτουργιών της Android εφαρμογής 54
5.2.3.1 Διαδικασία εγγραφής χρήστη και Διαχείριση του Cold Start 54
5.2.3.2 Κεντρικό μενού Android εφαρμογής 56
5.2.3.3 Ρυθμίσεις προφίλ και παραλαβής ειδοποιήσεων 58
5.2.3.4 Διαδικασία παραλαβής ειδοποιήσεων 59

Κεφάλαιο **6** **Πειραματική ανάλυση και αποτελέσματα**……..……...…...**61**
6.1 Εισαγωγή 61
6.1.1 Πειραματική ανάλυση 61
6.1.2 Πλατφόρμα δοκιμών 61
6.1.3 Μετρικές αξιολόγησης 62
6.1.4 Σύντομες περιγραφές των σετ δεδομένων και λόγοι επιλογής τους 63
6.2 1^{ο} Σετ Δεδομένων 65
6.2.1 Περιγραφή δεδομένων 65
6.2.2 Περιγραφή πειραμάτων 66
6.2.3 1^{ο} Πείραμα 66
6.2.3.1 Περιγραφή 66
6.2.3.2 Αποτελέσματα 67
6.2.4 2^{ο} Πείραμα 68
6.2.4.1 Περιγραφή 68
6.2.4.2 Αποτελέσματα 72
6.2.5 3^{ο} Πείραμα 73
6.2.5.1 Περιγραφή 73
6.2.5.2 Αποτελέσματα 74
6.3 2^{ο} Σετ Δεδομένων 74

6.3.1 Περιγραφή δεδομένων 74
6.3.2 Περιγραφή πειραμάτων 79
6.3.3 2ο Πείραμα 79
6.3.3.1 Περιγραφή 79
6.3.3.2 Αποτελέσματα 80
6.3.4 2ο Πείραμα 80
6.3.4.1 Περιγραφή 81
6.3.4.2 Αποτελέσματα 85
6.3.5 3ο Πείραμα 86
6.3.5.1 Περιγραφή 86
6.3.5.2 Αποτελέσματα 87
6.3.6 4ο Πείραμα 88
6.3.6.1 Περιγραφή 88
6.3.6.2 Αποτελέσματα 89
6.4 3ο Σετ Δεδομένων 90
6.4.1 Περιγραφή δεδομένων 90
6.4.2 Περιγραφή πειραμάτων 91
6.4.3 1ο Πείραμα 92
6.4.3.1 Περιγραφή 92
6.4.3.2 Αποτελέσματα 92
6.4.4 2ο Πείραμα 93
6.4.4.1 Περιγραφή 93
6.4.4.2 Αποτελέσματα 96
6.4.5 3ο Πείραμα 97
6.4.5.1 Περιγραφή 97
6.4.5.2 Αποτελέσματα 99
6.5 Πείραμα Επεκτασιμότητας αλγορίθμου συστάσεων 100

Κεφάλαιο **7** **Συμπεράσματα και μελλοντική εργασία**.....**103**
7.1 Γενικά Συμπεράσματα 103
7.2 Μελλοντική εργασία 104

Β ι β λ ι ο γ ρ α φ ί α.....106

Κεφάλαιο 1

Εισαγωγή Διπλωματικής Εργασίας

1.1 Γενική εισαγωγή 1
1.2 Κίνητρο διπλωματικής εργασίας 2
1.3 Στόχος διπλωματικής εργασίας 4
1.4 Συνεισφορά διπλωματικής εργασίας 6
1.5 Δομή διπλωματικής εργασίας 7

1.1 Γενική εισαγωγή

Πριν από λίγα χρόνια έκαναν την εμφάνιση τους οι έξυπνες κινητές συσκευές οι οποίες, όπως όλοι μας γνωρίζουμε, μας προσφέρουν μια μεγάλη γκάμα από υπηρεσίες που, με τη σωστή διαχείριση, κάνουν πιο άνετη τη ζωή μας. Ή για παράδειγμα τα έξυπνα ρολόγια ή οι έξυπνες τηλεοράσεις και οτιδήποτε άλλο συνδυάζει την τεχνολογική ανάπτυξη με διάφορα προϊόντα στη σύγχρονη εποχή. Αν ομαδοποιήσουμε όλα τα προϊόντα τεχνολογικής ανάπτυξης σε μία κοινή οργανωμένη οντότητα τότε θα μιλάμε για μία έξυπνη πόλη.

Μία έξυπνη πόλη είναι άρτια τεχνολογικά εξοπλισμένη, δηλαδή ενσωματωμένη με πολλαπλές τεχνολογίες πληροφοριών και επικοινωνίας. Στόχος της είναι η βελτίωση της ποιότητας ζωής των ανθρώπων, η ορθή και αποτελεσματική εκτέλεση των υπηρεσιών και γενικά η κάλυψη των αναγκών των κατοίκων της με τρόπο εύκολο και γρήγορο ως προς το χρόνο και τον κόπο.

Επίσης, μία έξυπνη πόλη πρέπει να είναι σε θέση να επιτρέπει στους κατοίκους της να έχουν άμεση αλληλεπίδραση με τη κοινότητα και την υποδομή της πόλης, να τους ενημερώνει για διάφορα νέα που συμβαίνουν σε αυτή και πως μπορούν να συμβάλουν σε μία καλύτερη ποιοτικά ζωή.

Όλα τα πιο πάνω που αναφέραμε σχετικά με τις υπηρεσίες που μας προσφέρει μία έξυπνη πόλη μπορούν να επιτευχθούν με την εξόρυξη μεγάλου όγκου δεδομένων από τους πολίτες και τις διάφορες οντότητες ή άλλα σημαντικά στοιχεία τα οποία απαρτίζουν την πόλη. Όλα αυτά τα δεδομένα μπορούν να συλλεχθούν μέσα από τη χρήση συστημάτων καταγραφής διαφόρων δεδομένων πραγματικού χρόνου όπως για παράδειγμα είναι οι αισθητήρες. Όλη αυτή η γνώση και η πληροφορία θα αναλυθεί κάτω από έξυπνους αλγορίθμους οι οποίοι θα αποτελούν το βασικό συστατικό που θα μας βοηθήσει να εξάγουμε αποτελέσματα και συμπεράσματα.

Η καταγραφή όγκου δεδομένων μέσα από την παρατήρηση της συμπεριφοράς των ανθρώπων μπορεί να γίνει μέσα από πολλούς τρόπους. Κάποιοι από αυτούς είναι η καταγραφή των συντεταγμένων με τη χρήση GPS για κάθε άνθρωπο ξεχωριστά χρησιμοποιώντας για παράδειγμα τη κινητή του συσκευή. Ένας άλλος τρόπος είναι η ενσωμάτωση αισθητήρων που χρησιμοποιούν τεχνολογία Bluetooth τους οποίους τοποθετούμε σε συγκεκριμένα σημεία και καταγράφουν αλληλεπιδράσεις που έγιναν από τις κινητές συσκευές κάποιων ατόμων που εκπέμπουν εξίσου Bluetooth.

Στο τέλος της μέρας, αυτό που μας ενδιαφέρει εμάς είναι η εξόρυξη συμπερασμάτων και κανόνων από αυτά τα δεδομένα, δηλαδή πως να συνδυάσουμε χρήστες βρίσκοντας όμοιους μεταξύ τους και χρήστες οι οποίοι βρέθηκαν να έχουν παρόμοια συμπεριφορά ως προς τις επιλογές τους.

Αναλύοντας τα δεδομένα που έχουμε στη διάθεση μας φτάνουμε στη κεντρική ιδέα της διπλωματικής εργασίας που είναι η χρήση δεδομένων τα οποία έχουν συλλεχθεί σε πραγματικές συνθήκες και στη συνέχεια, έχοντας στη διάθεση μας τα δεδομένα αυτά, έχουμε τη κατασκευή και τη χρήση αλγορίθμου ο οποίος προτείνει αντικείμενα σε ένα χρήστη διαμέσου της ανάλυσης των χρηστών, των αντικειμένων και των καταστάσεων.

1.2 Κίνητρο διπλωματικής εργασίας

Μέρα με τη μέρα ο αριθμός και ο όγκος των πληροφοριών που μας περιτριγυρίζουν αυξάνονται με δραματικό τρόπο. Από τη μία πλευρά τα προϊόντα και τα διάφορα υλικά

αγαθά αυξάνονται και νέες ιδέες κάνουν την εμφάνιση τους. Ενώ από την άλλη πλευρά, τα καταστήματα, οι καφετέριες και τα εστιατόρια πληθαίνουν με γοργούς ρυθμούς.

Ας πάρουμε το απλό παράδειγμα της Λευκωσίας για να είμαστε πιο σαφείς και συγκεκριμένα τις οδούς Λήδρας και Ονασαγόρου. Αυτές οι 2 οδοί σφύζουν, πλέον, από ζωή ενώ κάθε μέρα κάνουν την εμφάνιση τους καινούργια καταστήματα ή εστιατόρια. Σε αυτό το σημείο θα κάνουμε την εξής απλή σκέψη η οποία μας φέρει αντιμέτωπους με ένα δίλημμα: «Που θα καθίσουμε για ποτό ή για φαγητό ή ποιο κατάστημα θα εμπιστευτούμε για τα ψώνια μας;»

Ως απάντηση του πιο πάνω ερωτήματος μπορούμε να διακρίνουμε δύο κόσμους, δύο δηλαδή ομάδες ατόμων τις οποίες ξεχωρίζουμε ανάλογα με τον τρόπο σκέψης της κάθε ομάδας ή τον τρόπο προσέγγισης της σε αυτό το κομμάτι. Το κομμάτι στο οποίο αναφέρουμε είναι φυσικά η επιλογή. Η επιλογή ανάμεσα από μια μεγάλη γκάμα εστιατορίων, καφετεριών και καταστημάτων αποτελεί πλέον ταμπού για τους ανθρώπους.

Οι δύο μεγάλες ομάδες ατόμων στις οποίες αναφέρομαι είναι οι εξής: Η πρώτη είναι αυτή που περιλαμβάνει τα άτομα τους οποίους τους αρέσει η εξερεύνηση, η αναζήτηση καινούργιων εμπειριών και γενικά τα άτομα τα οποία δεν φοβούνται να έρθουν αντιμέτωποι με κάτι καινούργιο. Μπορούμε να τους παρομοιάσουμε με τον όρο εξωστρεφής άτομα, με απλά λόγια, χωρίς να θέτουμε αυστηρά αυτό τον όρο. Η δεύτερη ομάδα περιλαμβάνει τα πιο δύσπιστα άτομα, τα άτομα που γενικά δεν αναζητούν το καινούργιο και αρκούνται σε ένα συνηθισμένο τρόπο τον οποίο δύσκολα θα αλλάξουν.

Και οι 2 αυτές ομάδες όπως τις έχω περιγράψει πιο πάνω, αντιμετωπίζουν ένα κοινό πρόβλημα ή με διαφορετικά λόγια μια κοινή πρόκληση. Η πρόκληση αυτή συνδέεται άμεσα με το αρχικό ερώτημα-δίλημμα που θέσαμε το οποίο απορρέει από την επιλογή. Το κατά πόσο σωστή είναι αυτή η επιλογή που θα κάνουμε θα αποτελέσει σημαντικό παράγοντα ως προς την ευχαρίστηση, την ικανοποίηση και γενικά το ψυχολογικό τομέα σε ένα άνθρωπο.

Πως μπορούμε να κάνουμε τη σωστή επιλογή όμως; Ο πρώτος τρόπος ο οποίος μας έρχεται στο κεφάλι είναι να ρωτήσουμε τη γνώμη όλων των ανθρώπων οι οποίοι έχουν επισκεφθεί ένα εστιατόριο ή έχουν αγοράσει κάποιο συγκεκριμένο προϊόν. Να πάρουμε όλες αυτές τις απόψεις, να τις αναλύσουμε, να ζυγίσουμε τα δεδομένα και στο τέλος να αποφασίσουμε. Αυτός ο τρόπος βέβαια είναι πολύ δύσκολος να υλοποιηθεί στη πραγματικότητα και εξίσου χρονοβόρος και κουραστικός. Επίσης εδώ μπορούμε να επισημάνουμε και το γεγονός ότι οι κατάλογοι για παράδειγμα των εστιατορίων οι οποίοι περιλαμβάνουν τις τιμές είναι δύσκολα προσπελάσιμοι.

Όλα τα πιο πάνω μας δυσκολεύουν στο να πάρουμε τις σωστές αποφάσεις και ενέργειες. Η τεχνολογία αποτελεί ανάρπαστο κομμάτι της ζωής μας, μπορεί να μας προσφέρει πολλές υπηρεσίες και ευκολίες αν την αξιοποιήσουμε με σωστό τρόπο, αλλά και αποδοτικό.

Δεν θα ήταν ωραίο καθώς περπατάμε να λαμβάνουμε ειδοποιήσεις στη κινητή μας συσκευή με διάφορα προϊόντα, υπηρεσίες, φαγητά ή προσφορές; Και όλα αυτά δεν θα είναι τυχαία βεβαίως. Θέλουμε να λαμβάνουμε αυτά που πιθανόν να μας ενδιαφέρουν, να μας αρέσουν και να μας ικανοποιούν. Οι πιο πάνω συνειρμοί μας έφεραν αντιμέτωπους με μία πολύ ενδιαφέρον πρόκληση και έτσι σκεφτήκαμε να υλοποιήσουμε μία android εφαρμογή η οποία θα μας προτείνει, ορθά και γρήγορα, διάφορα προϊόντα.

1.3 Στόχος διπλωματικής εργασίας

Ζούμε σε μία κοινωνία όπου καθημερινά ερχόμαστε αντιμέτωποι με μεγάλου όγκου δεδομένων πληροφορίας και ενημέρωσης. Η εξέλιξη της τεχνολογίας μας προσφέρει πολλές ευκολίες στη καθημερινότητα μας, αλλά αν την δούμε από μία άλλη οπτική γωνιά μας έχει κινητοποιήσει σε μία κοινή απαίτηση ή με άλλα λόγια μας έχει δημιουργήσει μία ανάγκη η οποία σχετίζεται άμεσα με αυτή και γενικά τη ρουτίνα της ζωής.

Αυτή η απαίτηση είναι απλή ως ιδέα, αλλά δύσκολη στην υλοποίηση. Ας θέσουμε ένα ερώτημα: «Ποιοι παράγοντες οδηγούν μία υπηρεσία σε επιτυχία και ποιες ανάγκες των

χρηστών παίζουν το σημαντικότερο ρόλο σε αυτό;» Η απάντηση στο ερώτημα αυτό είναι το απλό, το γρήγορο και παράλληλα αυτό που κάνει σωστά τη δουλειά που θέλουμε. Αυτές οι τρεις λέξεις είναι το κλειδί της επιτυχίας και αποτελούν την πιο πάνω απαίτηση που θέσαμε. Επομένως, αυτός είναι ο στόχος της εργασίας που υλοποιήθηκε στα πλαίσια της διπλωματικής. Αναλύοντας τη κάθε έννοια ξεχωριστά μπορούμε να καταγράψουμε τα εξής:

Απλό:

Η Android εφαρμογή αποτελεί το γραφικό περιβάλλον με το οποίο οι χρήστες θα αλληλεπιδρούν. Θέλουμε η εφαρμογή να είναι εύκολη και απλή σε χρήση, χωρίς επιπλέον λεπτομέρειες ή πληροφορίες οι οποίες πιθανόν να ταλαιπωρούν ένα χρήστη. Επίσης, θέλουμε η εγγραφή ενός ατόμου στην εφαρμογή να είναι απλή όπως και η παραλαβή ειδοποιήσεων στη κινητή συσκευή. Η παρουσίαση μηνυμάτων και προσφορών ή με άλλα λόγια τα αντικείμενα που θα προτείνουμε σε ένα χρήστη να είναι ευκολοδιάβαστη και εύκολα προσπελάσιμη. Τέλος, να αναφέρουμε ότι γίνεται χρήση φίλτρων για τη καλύτερη εμπειρία των χρηστών τα οποία θα αποτρέπουν καταστάσεις οι οποίες πιθανόν να ενοχλήσουν το χρήστη όπως για παράδειγμα η συνεχής παραλαβή ειδοποιήσεων.

Γρήγορο:

Η Android εφαρμογή όπως επίσης και ο αλγόριθμος που θα τρέχει πίσω από αυτή και γενικά η αλληλοεπικοινωνία με τη βάση δεδομένων θέλουμε να γίνονται όσο το δυνατό πιο γρήγορα κάτι το οποίο το πετυχαίνουμε με τη χρήση τεχνολογιών γρήγορης ανάλυσης δεδομένων κάτι το οποίο το περιγράφουμε στη συνέχεια της διπλωματικής εργασίας.

Ορθό:

Η ορθότητα του αλγορίθμου αποτελεί πολύ σημαντικό κομμάτι. Το να είναι γρήγορος ο αλγόριθμος αλλά να δίνει μη-ορθά αποτελέσματα δεν μας ικανοποιεί βεβαίως, ενώ το ίδιο ισχύει και για το αντίθετο. Οι χρήστες επιθυμούν να λαμβάνουν ειδοποιήσεις με ορθό περιεχόμενο, δηλαδή θέλουν να τους προτείνουμε αντικείμενα τα οποία να τους ενδιαφέρουν πραγματικά, αλλιώς θα αποτύχουμε το στόχο μας. Αυτό το κομμάτι το ελέγχουμε στο κεφάλαιο όπου γίνεται η ανάλυση των αποτελεσμάτων που λαμβάνουμε.

Όλα τα πιο πάνω αποτελούν προκλήσεις στις οποίες ήρθαμε αντιμέτωποι στα πλαίσια της διπλωματικής εργασίας. Οι προκλήσεις είναι αυτές που μας δημιουργούν τους στόχους που θέτουμε, καθώς αποτελούν ίσως τις πιο σημαντικές παραμέτρους για να φτάσουμε σε επιτυχία.

1.4 Συνεισφορά διπλωματικής εργασίας

Η παρουσίαση της διπλωματικής εργασίας αναγράφει τα εξής προβλήματα τα οποία προσπαθεί να λύσει:

Έχουμε στη διάθεση μας ένα σύνολο από χρήστες, ένα σύνολο από αντικείμενα και για κάθε χρήστη έχουμε στη διάθεση μας το ιστορικό του, το οποίο αφορά δεδομένα τα οποία έχουμε στη διάθεση μας και αντικατοπτρίζουν το προφίλ ενός χρήστη. Οι χρήστες του συστήματος θα πρέπει να αλληλεπικοινωνούν με αυτό και να προσαρμόζουν τις δικές τους απαιτήσεις ανάλογα με τις προτιμήσεις τους.

Σε πρώτη φάση, η πρόκληση που έχουμε να αντιμετωπίσουμε έχει να κάνει με το χρόνο και την ορθότητα. Θέλουμε να προτείνουμε ένα αντικείμενο σε ένα χρήστη όσο πιο έγκαιρα γίνεται και επίσης αυτό που θα προτείνουμε να είναι ορθό και να ανταποκρίνεται στα ενδιαφέροντα και το προφίλ ενός χρήστη.

Σε δεύτερη φάση, θέλουμε να κατασκευάσουμε μια android εφαρμογή η οποία θα αποτελεί το γραφικό περιβάλλον αλληλεπίδρασης των χρηστών με το σύστημα. Η εφαρμογή θα είναι υπεύθυνη για την εγγραφή των χρηστών σε αυτή, την αλληλοεπικοινωνία χρήστη με το διαδίκτυο και τους αισθητήρες, θα δίνει το σήμα έναρξης του αλγορίθμου και τέλος θα παραλαμβάνει τα αποτελέσματα αυτής της εκτέλεσης, παρουσιάζοντας το στο χρήστη.

Σε τρίτη φάση, έχουμε την σωστή λειτουργία του συστήματος. Η ορθότητα αυτή πηγάζει, κυρίως, από την διαχείριση της σειράς των επικοινωνιών που λαμβάνουν χώρα στο σύστημα, την αποστολή αιτήσεων και λήψης αποτελεσμάτων από αυτό, αλλά και τη προσαρμογή του συστήματος σε κάθε χρήστη ξεχωριστά ανάλογα με τις απαιτήσεις του.

Σε τέταρτη και τελευταία φάση, έχουμε την επικύρωση της ορθότητας του αλγορίθμου. Με απλά λόγια, ζητούμε να δούμε αν ο αλγόριθμος δουλεύει και προτείνει αντικείμενα σε χρήστες ορθά. Η επικύρωση αυτή θέλουμε να γίνει στα πλαίσια πραγματικής δειγματοληψίας.

Οι μεθοδολογίες επίλυσης των πιο πάνω περιεκτικών περιγραφών του προβλήματος συνοψίζονται ως εξής:

Για τις ανάγκες επίλυσης του προβλήματος το οποίο περιγράφεται στη πρώτη φάση έχουμε μελετήσει, περιγράψει και υλοποιήσει ένα αλγόριθμο συστάσεων. Ο αλγόριθμος αυτός είναι ενσωματωμένος με βιβλιοθήκες μηχανικής μάθησης και υλοποιείται από εργαλεία τα οποία επιταχύνουν τον ρυθμό ανάλυσης των δεδομένων.

Η δεύτερη και η τρίτη φάση αφορούν τη λειτουργία του συστήματος το οποίο περιλαμβάνει και αφορά τα συστατικά όπως τα αναφέραμε πιο πάνω. Για το σύστημα έχουν χρησιμοποιηθεί τεχνολογίες οι οποίες είναι υπεύθυνες για την ορθή σε σειρά εκτέλεση του αλγορίθμου, δηλαδή τη παραλαβή πληροφοριών στον αλγόριθμο, την έναρξη του και αντίστοιχα την παράδοση του αποτελέσματος. Ο μεσολαβητής που θα χρησιμοποιείται για την επικοινωνία αυτή είναι αισθητήρες οι οποίοι έχουν την δυνατότητα να αλληλεπιδρούν με τις κινητές συσκευές των χρηστών. Επίσης, έχει αναπτυχθεί βάση δεδομένων η οποία περιλαμβάνει τις απαραίτητες πληροφορίες και η android εφαρμογή δίνει τη δυνατότητα στο χρήστη με κάποιες λειτουργίες να προσαρμόζει τη κατασκευή του προφίλ του.

Σχετικά με τη τέταρτη φάση έχει γίνει επιλογή δειγματοληψίας για αξιολόγηση της επίδοσης του αλγορίθμου συστάσεων. Έχουν επιλεγεί 3 σετ δεδομένων για την διεξαγωγή αυτών των πειραμάτων, τα οποία έχουν συλλεχθεί σε πραγματικές συνθήκες και το κάθε ένα έχει σκοπό τη παρατήρηση της συμπεριφοράς του αλγορίθμου συστάσεων κάτω από συγκεκριμένες προϋποθέσεις.

1.5 Δομή διπλωματική εργασίας

Το 2^{ο} κεφάλαιο αφορά τη περιγραφή του προβλήματος, τη προσεγγιστική λύση που δίνουμε σε αυτό διαμέσου μίας γενικής αρχιτεκτονικής του συστήματος η οποία θα αποτελεί το βασικό διάγραμμα που θα ακολουθήσουμε στη δομή της διπλωματικής εργασίας. Επίσης, σε αυτό το κεφάλαιο αναλύουμε το σύστημα συστάσεων που πρόκειται να υλοποιηθεί.

Το 3^{ο} κεφάλαιο περιέχει την αναλυτική περιγραφή της αρχιτεκτονικής του συστήματος, ποια πορεία ακολουθούν οι διεργασίες οι οποίες εκτελούνται και την αλληλοεπικοινωνία που γίνονται στα διάφορα συστατικά.

Το 4^{ο} κεφάλαιο αφορά την απαιτούμενη γνώση και το υπόβαθρο του συστήματος σχετικά με το λογισμικό ανάπτυξης και τις τεχνολογίες που έχουν χρησιμοποιηθεί για την ανάπτυξη και υλοποίηση του.

Το 5^{ο} κεφάλαιο έχει να κάνει με την υλοποίηση του Αλγορίθμου Συστάσεων και της Android εφαρμογής που θα χρησιμοποιούν οι χρήστες έχοντας την εγκατεστημένη στη κινητή τους συσκευή.

Το 6^{ο} κεφάλαιο περιέχει τη πειραματική ανάλυση που έχει γίνει στο σύστημα συστάσεων με τη χρήση διαφόρων τεχνικών για τη καταμέτρηση της απόδοσης του χρησιμοποιώντας συλλογές από πραγματικά δεδομένα.

Κεφάλαιο 2

Περιγραφή προβλήματος και Προτεινόμενη προσέγγιση

2.1 Περιγραφή προβλήματος 10
2.2 Προτεινόμενη προσέγγιση λύσης του προβλήματος 11
2.2.1 Περιεκτικές φάσεις αρχιτεκτονικής Συστήματος 12
2.3 Σύστημα Συστάσεων 14
2.3.1 Λόγοι επιλογής Apache Spark 14
2.3.2 Λόγοι επιλογής Apache Kafka 15
2.3.3 Απαιτήσεις του υπό κατασκευή Αλγορίθμου 15
2.3.4 Αλγόριθμοι Συστάσεων 16
2.3.5 Λόγοι επιλογής των Αλγορίθμων Συστάσεων 16
2.3.6 Αλγόριθμοι Συστάσεων «Collaborative Filtering» 17
2.3.6.1 «Collaborative Filtering» με βάση μοντέλο 18
2.3.6.2 Λόγοι επιλογής Αλγορίθμων Συστάσεων με βάση μοντέλο 18
2.3.7 Εναλλακτικοί τύποι για Αλγορίθμους Συστάσεων 19
2.3.7.1 Αλγόριθμοι Συστάσεων «Content Based» 19
2.3.8 Μηχανική Μάθηση 19
2.3.8.1 Μείωση διαστάσεων προβλήματος 19
2.3.8.2 «MLlib» βιβλιοθήκη στο Apache Spark 20
2.3.8.3 «Alternative Least-Square» προσέγγιση 20
2.4 Χαρακτηριστικά Αλγορίθμου Συστάσεων 20
2.4.1 Μοντέλο Αλγορίθμου Συστάσεων 21
2.4.2 Εκπαίδευση μοντέλου του Αλγορίθμου Συστάσεων 22
2.4.2.1 Παράμετροι της «Alternative Least-Square» προσέγγισης 22
2.4.2.2 Τεχνική εκπαίδευσης μοντέλου 23
2.4.3 «Cold Start» πρόβλημα στους Αλγορίθμους Συστάσεων 25
2.4.3.1 Τρόπος αντιμετώπισης «Cold Start» προβλήματος 25

2.1 Περιγραφή προβλήματος

Κάποιες βασικές έννοιες πριν τη περιγραφή τους προβλήματος:

- **Χρήστης:** Ο χρήστης είναι ένα οποιοδήποτε άτομο το οποίο έχει εγκαταστημένη την Android εφαρμογή στη κινητή του συσκευή.
- **Αντικείμενο:** Το αντικείμενο μπορεί να αποτελεί μία τοποθεσία, ένα υλικό αγαθό ή γενικά οτιδήποτε μπορεί να συνδεθεί με ένα χρήστη.
- **Αισθητήρας:** Ο αισθητήρας θα βρίσκεται τοποθετημένος σε συγκεκριμένα σημεία στο αστικό περιβάλλον και ο κάθε ένας θα αντιπροσωπεύει ένα αντικείμενο ή μια ομάδα αντικειμένων.
- **Προφίλ/Ιστορικό Χρήστη:** Το προφίλ ενός χρήστη αποτελείται από το πόσες φορές βρέθηκε ένας χρήστης σε μία τοποθεσία, σε ένα κατάστημα/εστιατόριο ή κοντά σε ένα αντικείμενο. Επίσης στο προφίλ του κρατάμε βαθμολογίες που έχει κάνει σε διάφορα αντικείμενα ή δεδομένα τα οποία αφορούν αγορές αντικειμένων που έχει κάνει.

Σύντομη περιγραφή προβλήματος:

Έχουμε στη διάθεση μας ένα σύνολο από χρήστες, ένα σύνολο από αντικείμενα και για κάθε χρήστη έχουμε στη διάθεση μας το προφίλ του. Το πρόβλημα που έχουμε να αντιμετωπίσουμε είναι η αποδοτική και σε πραγματικό χρόνο σύσταση διαφόρων αντικειμένων σε διάφορους χρήστες.

Διατύπωση προβλήματος:

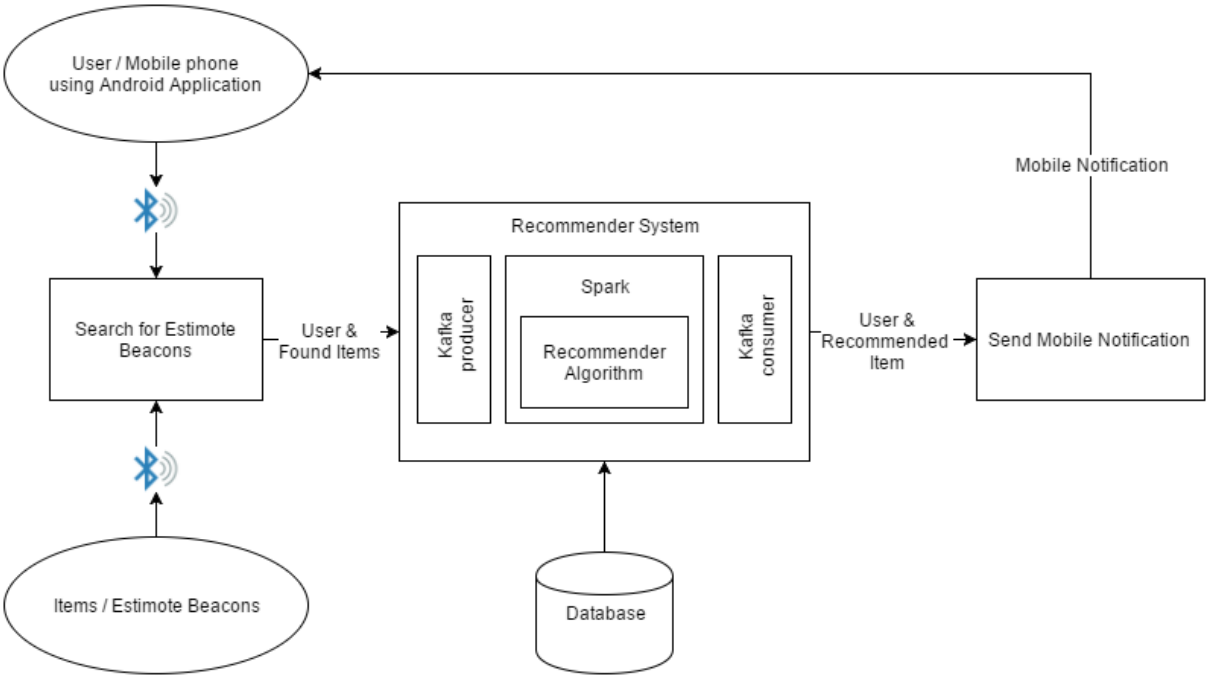
Όπως αναφέραμε και στη περιγραφή του προβλήματος επιθυμούμε να προτείνουμε αντικείμενα σε χρήστες με τρόπο αποδοτικό και γρήγορο σχετικά με την εκτέλεση και την εξαγωγή αποτελεσμάτων, ενώ αυτή η διαδικασία θα γίνεται σε πραγματικό χρόνο. Θέλουμε η προσομοίωση αυτής της διαδικασίας να μπορεί να αντεπεξέλθει σε διάφορα σενάρια όπου με τον όρο σενάριο εννοούμαι την προσαρμοστικότητα ως προς εγκατάσταση του συστήματος που πρόκειται να υλοποιηθεί.

Η απόδοση του συστήματος ως προς το χρόνο είναι σημαντική λόγω του ότι το σύστημα θα τρέχει σε πραγματικό χρόνο, ενώ ως προς την ορθότητα είναι εξίσου στοιχειώδης λόγω της φύσεως του προβλήματος που έχουμε να αντιμετωπίσουμε το οποίο μέσω της χρήσης διαφόρων τεχνικών και τεχνολογιών πρέπει να μελετήσει, να συνδυάσει και να προβλέψει τι θα ενδιέφερε ένα χρήστη. Αξίζει να σημειωθεί ότι οι χρήστες αποτελούν τη κριτική επιτροπή και το θεμελιώδη λίθο της επιτυχίας, καθώς από αυτούς εξαρτάται η δημοτικότητα του συστήματος, δηλαδή κατά πόσο το σύστημα κάνει καλά τη δουλειά του.

Άλλα σημαντικά στοιχεία τα οποία αντικατοπτρίζουν το πρόβλημα είναι η επικοινωνία που θα γίνεται μεταξύ χρηστών και συστήματος και αντίθετα. Δηλαδή πως θα δίνεται το σήμα έναρξης στο οποίο θα προτείνουμε ένα αντικείμενο σε ένα χρήστη, πως θα απαντάμε με το αποτέλεσμα που έχει υπολογιστεί. Επίσης, θα πρέπει να ληφθεί υπόψη ότι οι χρήστες πρέπει να έχουν τη δυνατότητα να προσαρμόζουν κάποιες ρυθμίσεις στα μέτρα τους, όπως για παράδειγμα το χρόνο που θέλουν να λαμβάνουν ειδοποιήσεις (προτεινόμενα αντικείμενα) στη κινητή τους συσκευή.

2.2 Προτεινόμενη προσέγγιση λύσης του προβλήματος

Η προσεγγιστική λύση που δίνουμε στο πρόβλημα μας ακολουθά την εξής περιεκτική αρχιτεκτονική η οποία περιγράφει, σε ένα γενικό πλαίσιο, πως θα προχωρήσουμε ως προς τη δομή της λύσης που θα προτείνουμε και θα αναλύσουμε σε αυτό και στα επόμενα κεφάλαια.



Εικόνα 2.1 – Αρχιτεκτονική συστήματος

Περιγραφή της προσεγγιστικής λύσης με βάση την εικόνα της αρχιτεκτονικής:

Θα χωρίσουμε την απλή αρχιτεκτονική του συστήματος σε 3 φάσεις τις οποίες θα ακολουθήσουμε ως ένα χάρτη πορείας για τη διπλωματική εργασία.

2.2.1 Περιεκτικές φάσεις αρχιτεκτονικής Συστήματος:

Πρώτη φάση: Η πρώτη φάση περιέχει την αλληλεπίδραση μεταξύ των χρηστών, του συστήματος συστάσεων και των αισθητήρων.

Επιλέγουμε τη χρήση αισθητήρων Bluetooth λόγω της μεγάλης γκάμας συσκευών που έχουν βγει στην αγορά οι οποίες επικοινωνούν με τις κινητές συσκευές διαμέσου της τεχνολογίας Bluetooth όπως για παράδειγμα τα έξυπνα ρολόγια.

Οι διεργασίες της πρώτης φάσης θα εκτελούνται κάτω από μία Android εφαρμογή η οποία θα είναι υπεύθυνη να επικοινωνεί με τους αισθητήρες οι οποίοι θα βρίσκονται στο αστικό περιβάλλον του χρήστη και τη βάση δεδομένων. Θα αποτελεί το γραφικό

περιβάλλον στο ο οποίο ο χρήστης θα μπορεί να ρυθμίζει την εκκίνηση του συστήματος συστάσεων.

Δεύτερη φάση: Η δεύτερη φάση περιέχει το σύστημα συστάσεων το οποίο αποτελείται από δύο ουρές προτεραιότητας και τον αλγόριθμο συστάσεων ανάμεσα από αυτές.

Η χρήση των δύο ουρών γίνεται για την είσοδο των δεδομένων και την έξοδο των αποτελεσμάτων αντίστοιχα όπου ο αλγόριθμος συστάσεων παραλαμβάνει τα δεδομένα από την πρώτη ουρά και παραδίδει το αποτέλεσμα στη δεύτερη.

Ο αλγόριθμος συστάσεων έχει υλοποιηθεί σε Apache Spark κάτω από την αρχιτεκτονική Map-Reduce η οποία μας προσφέρει ταχύτητα στην ανάλυση των δεδομένων. Άλλες εναλλακτικές λύσεις που είχαμε στη διάθεση μας είναι το Apache Hadoop. Οι δύο ουρές δεν έχουν υλοποιηθεί με άμεσο τρόπο καθώς μας δίνεται η δυνατότητα χρήσης έτοιμου εργαλείου Apache Kafka το οποίο μας παρέχει ορθή προτεραιότητα στις αιτήσεις που φθάνουν από την πλευρά της εισόδου (Kafka producers) και ανάλογα λαμβάνουν το αποτέλεσμα από τον αλγόριθμο συστάσεων και τροφοδοτούνται στην έξοδο (Kafka consumer).

Σημείωση: Σε αυτό το κεφάλαιο θα αναλύσουμε γιατί αποφασίσαμε να χρησιμοποιήσουμε Apache Spark και Kafka, τους λόγους που χρησιμοποιούμαι τον αλγόριθμο συστάσεων και τι μας οδήγησε σε συγκεκριμένες επιλογές χαρακτηριστικών του αλγορίθμου που καταλήξαμε.

Τρίτη φάση: Η τρίτη φάση περιλαμβάνει τη λήψη του αποτελέσματος που δίνει το σύστημα συστάσεων και την αποστολή του αποτελέσματος στο χρήστη.

Το αποτέλεσμα που δίνει το σύστημα συστάσεων περνά από μία σειρά από φίλτρα τα οποία σχετίζονται με τις ρυθμίσεις που έχει θέσει ο χρήστης με σκοπό να απομονώσουμε κάποιες επιλογές και να καταλήξουμε σε μία. Το τελικό αποτέλεσμα το οποίο εξάγεται από αυτά τα φίλτρα στέλνεται στη κινητή συσκευή του χρήστη και εμφανίζεται ως ειδοποίηση η οποία ανήκει στην Android εφαρμογή την οποία αναφέραμε στη πρώτη φάση.

2.3 Σύστημα Συστάσεων

Όπως αναφέραμε και πιο πάνω, σε αυτό το κεφάλαιο θα επικεντρωθούμε στην ανάλυση των απαιτήσεων του συστήματος συστάσεων, τις λειτουργίες του και τις επιλογές που έχουν γίνει ως προς τη χρήση διαφόρων τεχνολογιών.

Σημείωση: Η αναλυτική αρχιτεκτονική, οι τεχνολογίες, η απαιτούμενη γνώση, η υλοποίηση του αλγορίθμου και της Android εφαρμογής μπορούν να εντοπιστούν στα επόμενα κεφάλαια ακολουθώντας τη δομή της διπλωματικής εργασίας στο υποκεφάλαιο 1.5.

2.3.1 Λόγοι επιλογής του Apache Spark

Το Apache Spark έχει αναπτυχθεί με σκοπό να αντιμετωπίσουμε τους περιορισμούς σε σχέση με την Map-Reduce αρχιτεκτονική όπως αυτή χρησιμοποιείται στο Hadoop η οποία μας αναγκάζει σε μια συγκεκριμένη δομή γραμμικής ροής δεδομένων όταν γράφουμε κατανεμημένα προγράμματα.

Με άλλα λόγια, το Apache Spark δεν μας αναγκάζει να ακολουθούμε ένα συγκεκριμένο πρότυπο εγγραφής κώδικα για να εκτελέσουμε λειτουργίες οι οποίες έχουν να κάνουν με το Map-Reduce. Αυτό το πλεονέκτημα μας παρέχεται διαμέσου των λειτουργιών των Spark RDDs τα οποία μας δίνουν την δυνατότητα ενός συνόλου εργασιών σε ένα υψηλότερο επίπεδο αφαιρετικότητας οι οποίες μπορούν να εκτελεσθούν σε κατανεμημένα προγράμματα.

Το Apache Spark έχει σχεδιαστεί με τέτοιο τρόπο ώστε να επιταχύνει το ρυθμό ανάλυσης των δεδομένων σε σχέση με το Hadoop. Αυτό το πετυχαίνει γιατί υποστηρίζει την επεξεργασία των δεδομένων στη κύρια μνήμη ή μπορεί να κάνει συμβατική επεξεργασία στο δίσκο σε περίπτωση που τα δεδομένα δεν χωράνε στη διαθέσιμη κύρια μνήμη. Με αυτό τον τρόπο ενισχύεται η απόδοση της ανάλυσης των δεδομένων.

Επίσης μας παρέχει βιβλιοθήκες μηχανικής μάθησης όπως για παράδειγμα η βιβλιοθήκη MLlib την οποία χρησιμοποιούμαι στα πλαίσια του αλγορίθμου συστάσεων, ενώ επίσης μας παρέχει τη δυνατότητα επεξεργασίας των δεδομένων σε πραγματικό χρόνο με τη χρήση του Spark Streaming.[1]

Σύμφωνα με στατιστικά αποτελέσματα που μας παρέχονται από την επίσημη σελίδα του Apache Spark, το Spark είναι μέχρι 100 φορές πιο γρήγορο σε σχέση με το Hadoop όταν η επεξεργασία των δεδομένων γίνεται στη κύρια μνήμη ή μέχρι 10 φορές όταν η επεξεργασία γίνεται στο δίσκο.[2]

2.3.2 Λόγοι επιλογής του Apache Kafka

Τα μηνύματα τα οποία αποστέλλονται στο Kafka, γίνονται διαθέσιμα προς ανάλυση σε πραγματικό χρόνο. Το Kafka μας παρέχει υψηλή απόδοση στο ρυθμό παραλαβής και εξαγωγής μηνυμάτων, ανοχή σε σφάλματα και, επίσης, διατηρεί τη σειρά προτεραιότητας στη λήψη αιτήσεων μεγάλου όγκου στο σύστημα.

2.3.3 Απαιτήσεις του υπό κατασκευή Αλγορίθμου

- **Απαίτηση:** Επιθυμούμε να κατασκευάσουμε ένα αλγόριθμος ο οποίος θα είναι σε θέση να αναλύσει τα δεδομένα που έχουμε στη διάθεση μας (χρήστες, αντικείμενα και ιστορικό χρηστών) και θα μας δίνει αποτελέσματα σύμφωνα με τα οποία θα προτείνουμε σε ένα χρήστη, ένα αντικείμενο.
- **Λύση:** Για τη συγκεκριμένη απαίτηση θα χρησιμοποιήσουμε αλγορίθμους τύπου Συστάσεων και συγκεκριμένα θα γίνει χρήση του «Collaborative Filtering» αλγορίθμου συστάσεων με βάση ένα μοντέλο.

- **Απαίτηση:** Θέλουμε ένα αλγόριθμο ο οποίος μέσα από την εκπαίδευση του να δημιουργεί ένα μοντέλο το οποίο θα χρησιμοποιούμαι για να προβλέψουμε ποιο αντικείμενο συνδέεται με τις ανάγκες και τα ενδιαφέροντα του χρήστη. Επίσης πρέπει να γίνεται μείωση των διαστάσεων του προβλήματος, δηλαδή των δεδομένων που έχουμε.

- **Λύση:** Για αυτή την απαίτηση θα γίνει χρήση μηχανικής μάθησης και συγκεκριμένα της βιβλιοθήκης MLlib και του αλγορίθμου ALS(Alternative Least-Square algorithm) ο οποίος εκτελεί μείωση των διαστάσεων. Ένας από τους λόγους επιλογής αυτής της βιβλιοθήκης έχει γίνει λόγω του ότι μας παρέχεται από το Apache Spark.

Οι λύσεις των πιο πάνω απαιτήσεων έχουν επιλεχθεί για τους λόγους τους οποίους εξηγούμε στη συνέχεια.

2.3.4 Αλγόριθμοι Συστάσεων

Στόχος των Αλγορίθμων Συστάσεων είναι η πρόβλεψη ή με άλλα λόγια η δημιουργία σωστών προτάσεων για προϊόντα ή αντικείμενα τα οποία πρόκειται να συσταθούν σε ένα σύνολο από χρήστες ανάλογα με τις προτιμήσεις τους ή το ιστορικό τους. Δηλαδή, χρησιμοποιώντας τα δύο σύνολα αντικειμένων και χρηστών αντίστοιχα, προσπαθούμε να προβλέψουμε ποια αντικείμενα θα προτιμούσε ένας χρήστη.

Για παράδειγμα μπορούμε να αναφερθούμε σε ένα σύνολο από βαθμολογίες χρηστών σε ταινίες όπου σκοπός μας είναι να ανακαλύψουμε ποια θα ήταν η πιθανή βαθμολογία που θα έδινε ένας χρήστης σε μία ταινία την οποία δεν έχει δει ή βαθμολογήσει ακόμη. Σε αυτό το σημείο υπάρχει ένας αριθμός από διαφορετικές προσεγγίσεις υλοποίησης ενός τέτοιου αλγορίθμου. Η κάθε προσέγγιση διαφέρει από την άλλη ως προς το τρόπο κατά τον οποίο γίνεται η ανάλυση των συνόλων από δεδομένα που έχουμε, πως συγκρίνουμε τα διαθέσιμα δεδομένα μεταξύ τους και πως προσεγγίζουμε τον όρο «όμοιοι χρήστες» ή «όμοια αντικείμενα». Για αυτό το σκοπό, έχουμε 2 κύρια είδη προσεγγίσεων: (1) «Collaborative Filtering» αλγόριθμοι συστάσεων και (2) «Content-based» αλγόριθμοι συστάσεων.[3]

2.3.5 Λόγοι επιλογής των αλγορίθμων συστάσεων

1. Βασίζονται στη πραγματική δραστηριότητα:

Οι αλγόριθμοι συστάσεων προτείνουν με βάση πραγματικά δεδομένα τα οποία έχουν συλλεχθεί και αντιπροσωπεύουν τη συμπεριφορά των χρηστών ή με άλλα λόγια το

ιστορικό τους. Οι προβλέψεις που δίνουν δεν βασίζονται σε εικασίες, αλλά στη πραγματικότητα. Οι αποφάσεις που λαμβάνονται σε αυτό το είδος αλγορίθμων βασίζονται στη παρατήρηση των χρηστών στο περιβάλλον τους και βέβαια δεν είναι τυχαίες.

2. Βασίζονται στην εξερεύνηση:

Ένα πολύ σημαντικό πρόβλημα που αντιμετωπίζουμε είναι οι διαφημίσεις ή διάφορα προϊόντα τα οποία μας προτείνουν όπου δεν συμβαδίζουν με τα ενδιαφέροντα μας. Ο αλγόριθμος συστάσεων γεφυρώνει το πρόβλημα αυτό διότι με την ανάλυση των δεδομένων μας επιτρέπει να εξερευνήσουμε αντικείμενα τα οποία είναι παρόμοια με αυτά που ήδη αρέσουν σε ένα χρήστη.

3. Τα δεδομένα είναι πάντα ενημερωμένα:

Ο αλγόριθμος συστάσεων ενημερώνει και αναβαθμίζει τα δεδομένα που έχει στη διάθεση του και επομένως είναι πάντα ενημερωμένο. Η δυνατότητα του αλγορίθμου συστάσεων να εξαρτάται από τη δραστηριότητα σε πραγματικό χρόνο είναι ένα μεγάλο πλεονέκτημα.[4]

2.3.6 «Collaborative Filtering» αλγόριθμοι συστάσεων

Το «Collaborative Filtering» είναι μια προσέγγιση των αλγορίθμων συστάσεων κατά την οποία συλλέγουμε πληροφορίες τις οποίες δίνουν οι χρήστες για διάφορα προϊόντα και αυτές τις πληροφορίες τις παρουσιάζουμε με τη μορφή αξιολόγησης ή διαφορετικά βαθμολογίας σε ένα συγκεκριμένο τομέα ο οποίος για παράδειγμα μπορεί να αφορά βαθμολογίες ταινιών. Στόχος της προσέγγισης αυτής είναι να εντοπίσει ομοιότητες ως προς τη συμπεριφορά κατά την οποία γίνονται οι αξιολογήσεις και με αυτό το τρόπο να μπορεί να προτείνει τα κατάλληλα προϊόντα στους κατάλληλους χρήστες.[5]

Διαισθητικά μπορούμε να παρουσιάσουμε το τρόπο κατά τον οποίο λειτουργεί αυτή η προσέγγιση ως εξής: Διαφορετικοί χρήστες οι οποίοι συμφωνούσαν σε διάφορες τους προτιμήσεις ως προς κάποια προϊόντα μέχρι τώρα, τότε είναι πιο πιθανό να συμφωνούν και στο μέλλον.

2.3.6.1 « Collaborative Filtering» αλγόριθμοι συστάσεων με βάση μοντέλο

Η βάση μοντέλου προσέγγιση είναι τεχνική η οποία χρησιμοποιείται στους αλγορίθμους πρόβλεψης του «Collaborative Filtering». Αυτή η τεχνική περιλαμβάνει την οικοδόμηση ενός μοντέλου το οποίο βασίζεται στο σύνολο των δεδομένων που έχουμε από τις αξιολογήσεις που έχουν κάνει διάφοροι χρήστες σε διάφορα προϊόντα. Με άλλα λόγια, εξάγουμε κάποιες πληροφορίες από ένα σύνολο από δεδομένα και τις χρησιμοποιούμαι για να αναπαραστήσουμε το μοντέλο μας και με αυτό το τρόπο μπορούμε να εξάγουμε προβλέψεις χωρίς να χρειάζεται κάθε φορά να χρησιμοποιήσουμε το πλήρες σύνολο δεδομένων. [6]

2.3.6.2 Λόγοι επιλογής του «Collaborative Filtering» με βάση μοντέλο

1. Δεν υπάρχει η ανάγκη να γνωρίζουμε το περιεχόμενο των αντικειμένων ή προϊόντων που προτείνουμε. Επίσης τα σημαντικά στοιχεία που απαρτίζουν ένα προϊόν είναι συνήθως δύσκολο να τα εντοπίσουμε, για παράδειγμα σε μια ταινία υπάρχουν πολλά συστατικά την οποία απαρτίζουν όπως είναι οι ηθοποιοί, οι παραγωγοί, η κατηγορία της ταινίας και τα λοιπά, κάτι το οποίο δεν είναι εύκολο να ξεχωρίσουμε τα πιο κύρια από αυτά.

2. Το «Cold Start» πρόβλημα περί των αντικειμένων ή προϊόντων γεφυρώνεται σε αυτή τη προσέγγιση με τη χρήση κάποιων τεχνικών. Μια τεχνική για παράδειγμα είναι να ζητήσουμε από το χρήστη να μας δώσει κάποιες βαθμολογίες σε ένα μικρό σύνολο από αντικείμενα για να μάθουμε κάποια πράγματα για το προφίλ του.

3. Με αυτή τη μέθοδο δίνουμε τη δυνατότητα να γνωρίζουμε τους λόγους για τους οποίους έχουμε προτείνει κάποιο αντικείμενο. Όταν δηλαδή ένα σύνολο από χρήστες έχουν για παράδειγμα αγοράσει τα X,Y και Z αντικείμενα και στη συνέχεια ένα υποσύνολο από αυτούς τους χρήστες έχει εξίσου αγοράσει το W αντικείμενο τότε αυτό σημαίνει ότι, πολύ πιθανό, οι υπόλοιποι χρήστες να αγοράσουν το W αντικείμενο. Άρα με αυτό τον επεξηγηματικό τρόπο μπορούμε να δικαιολογήσουμε τις επιλογές που δίνει ο αλγόριθμος.[5]

2.3.7 Εναλλακτικοί αλγόριθμοι συστάσεων

2.3.7.1 «Content-based» αλγόριθμος συστάσεων

Ο τρόπος με τον οποίο γίνεται η ανάλυση των δεδομένων σε αυτή τη προσέγγιση είναι με βάση το περιεχόμενο και ο αλγόριθμος προσπαθεί να προτείνει αντικείμενα τα οποία είναι παρόμοια με αυτά που ενδιέφεραν τον χρήστη από το παρελθόν μέχρι αυτή τη στιγμή. Με άλλα λόγια το μοντέλο χτίζεται με βάση κάποια χαρακτηριστικά που απαρτίζουν το χρήστη και τα αντικείμενα. Για παράδειγμα, τα χαρακτηριστικά ενός χρήστη μπορεί να είναι οι κατηγορίες ταινιών ή οι ηθοποιοί που του αρέσουν.[6]

Λόγοι για τους οποίους δεν επιλέχθηκε αυτή η προσέγγιση:

1. Είναι δύσκολο να βρούμε τα κατάλληλα χαρακτηριστικά τα οποία θα μας εγγυηθούν καλά αποτελέσματα. Για παράδειγμα, μία ταινία περιέχει πολλά χαρακτηριστικά όπως είναι οι κατηγορίες στις οποίες ανήκει (κωμωδία, δράμα κ.τ.λ.), οι ηθοποιοί, οι σκηνοθέτες και οτιδήποτε άλλο σχετίζεται με αυτήν. Το πιο σημαντικό ερώτημα είναι ποια από αυτά θα πρέπει να επιλέξουμε.
2. Αυτή η προσέγγιση είναι «πολύ ειδική» ως προς της προτιμήσεις ενός χρήστη. Με την έννοια «πολύ ειδική» εννοούμαι ότι δεν προτείνει, ποτέ, αντικείμενα τα οποία είναι έξω από τα όρια του προφίλ του χρήστη. Δηλαδή, αν ο χρήστης έχει πολλαπλά ενδιαφέροντα, αδυνατεί να τα εξερευνήσει.
3. Η κατασκευή ενός προφίλ για ένα χρήστη, ο οποίος είναι καινούργιος, είναι δύσκολη γιατί δεν έχουμε καθόλου πληροφορίες για τα ενδιαφέροντα και τις προτιμήσεις του.[7]

2.3.8 Μηχανική μάθηση

2.3.8.1 Μείωση διαστάσεων προβλήματος

Dimensionality reduction είναι η μέθοδος η οποία χρησιμοποιείται στη προσέγγιση του «model-based collaborative filtering». Σκοπός της μεθόδου αυτή είναι να μειώσουμε τις διαστάσεις του προβλήματος μας. Αυτό το πετυχαίνουμε μετατρέποντας δεδομένα από

υψηλές διαστάσεις σε δεδομένα πολύ χαμηλότερων διαστάσεων με τέτοιο τρόπο ώστε τα νέα δεδομένα να αντιπροσωπεύουν σε αρκετά μεγάλο βαθμό τα παλιά.[8]

Πλεονέκτημα της μείωσης διαστάσεων προβλήματος:
Τα πλεονεκτήματα αυτής της μεθόδου είναι η αισθητή μείωση του χρόνου και του χώρου που απαιτείται εξαρχής, βελτιώνεται η απόδοση του μοντέλου μηχανικής μάθησης και τέλος γίνεται ευκολότερη η απεικόνιση των δεδομένων όταν μειώνεται σε πολύ χαμηλές διαστάσεις όπως για παράδειγμα σε 2 ή 3 διαστάσεις.[8]

2.3.8.2 «MLlib» βιβλιοθήκη

Η MLlib είναι μία επεκτάσιμη βιβλιοθήκη μηχανικής μάθησης η οποία μπορεί να ενσωματωθεί διαμέσου προγράμματος το οποίο τρέχει με τη χρήση Spark αρχιτεκτονικής. Η βιβλιοθήκη αυτή αποτελείται από μια σειρά από έξυπνους αλγορίθμους ή προγράμματα εκμάθησης και γενίκευσης ενός συνόλου από γνώσεις ή με άλλα λόγια ενός συνόλου από κάποιου τύπου δεδομένα. Αυτό περιλαμβάνει την ομαδοποίηση («clustering»), η «Collaborative Filtering» προσέγγιση αλγορίθμων πρόβλεψης και τη μείωση των διαστάσεων ενός προβλήματος με τέτοιο τρόπο ώστε να μην αλλοιώνονται τα αρχικά δεδομένα και να μας παρέχονται εξίσου καλά αποτελέσματα.[9]

2.3.8.3 «Alternative Least-square» προσέγγιση

Η μέθοδος «Alternative least-square» χρησιμοποιείται στα μοντέλα μη γραμμικής παλινδρόμησης όπου υπάρχουν 2 εξαρτημένες μεταβλητές όπου στη περίπτωση του δικού μας αλγορίθμου είναι οι Χρήστες και τα Αντικείμενα. Η προσέγγιση την οποία ακολουθεί η μέθοδος αυτή βασίζεται στις 2 εξαρτημένες μεταβλητές που αναφέραμε πιο πάνω. Ο αλγόριθμος διορθώνει μία από τις παραμέτρους αυτές και ταυτόχρονα προσπαθεί να επιλύσει με το πιο βέλτιστο τρόπο την άλλη παράμετρο. Ο αλγόριθμος προσπαθεί να διορθώνει τη μεταβλητή των Χρηστών και να ενημερώνει αυτή των Αντικειμένων και αντίστροφα μέχρι να φτάσει σε μία ικανοποιητική σύγκλιση.[10]

2.4 Χαρακτηριστικά Αλγόριθμου Συστάσεων

2.4.1 Μοντέλο αλγορίθμου

Η προσέγγιση την οποία ακολουθά ο αλγόριθμος είναι η κατασκευή ενός μοντέλου το οποίο εξαρτάται από το ιστορικό ενός χρήστη, δηλαδή τη συμπεριφορά που παρατηρείται σε αυτό το χρήστη, από το παρελθόν του μέχρι τη δεδομένη στιγμή. Επιπλέον, το μοντέλο αυτό εξαρτάται από παρόμοιους χρήστες οι οποίο εντοπίστηκαν να ακολουθούν, σε αυξημένο ποσοστό, παρόμοια συμπεριφορά ως προς τα ενδιαφέροντα τους, τις αγορές τους, τις τοποθεσίες που εντοπίστηκαν ή ακόμη και τις αριθμητικές βαθμολογίες τις οποίες έχουν δώσει σε διάφορα προϊόντα. Στη συνέχεια το μοντέλο αυτό χρησιμοποιείται για να προβλέψουμε πιθανά αντικείμενα τα οποία ενδιαφέρουν ένα συγκεκριμένο χρήστη.

Ιστορικό ενός χρήστη:

Προηγούμενη συμπεριφορά ενός ατόμου η οποία περιλαμβάνει ένα αριθμό από παραμέτρους ανάλογα με τα δεδομένα που έχουμε στη διάθεση μας. Κάποια παραδείγματα τέτοιων παραμέτρων είναι τα εξής:

- Ένα άτομο πόσο συχνά επισκέπτεται ένα χώρο ή έχει εντοπιστεί σε μία συγκεκριμένη τοποθεσία (για παράδειγμα ένα άτομο επισκέπτεται ένα συγκεκριμένο εστιατόριο 2 φορές το μήνα).
- Ένα άτομο πόση χρονική διάρκεια αφιέρωσε στη παρατήρηση ενός αντικειμένου, δηλαδή το ενδιαφέρον που έδειξε για αυτό (για παράδειγμα κάποιος έχει εντοπιστεί κοντά σε ένα προϊόν ενός καταστήματος ηλεκτρικών ειδών για περισσότερο από μισή ώρα).
- Τι προϊόντα έχει αγοράσει στο παρελθόν (για παράδειγμα μέσα από τα δεδομένα που διαθέτουμε έχουμε εντοπίσει ότι συχνά αγοράζει ηλεκτρονικά παιχνίδια).
- Πιθανόν αριθμητικές βαθμολογίες τις οποίες έχει δώσει ένα άτομο σε κάποια αντικείμενα (για παράδειγμα ένα άτομο έχει βαθμολογήσει ένα προϊόν το οποίο έχει αγοράσει).

Όλες οι πιο πάνω παράμετροι αποτελούν το προφίλ ενός χρήστη και είναι αυτό που θα μας δώσει την εικόνα η οποία αντικατοπτρίζει το συγκεκριμένο άτομο. Παρόλα αυτά οι

πιο πάνω πληροφορίες δεν είναι πάντα όλες διαθέσιμες και εξαρτάται από τα δεδομένα τα οποία έχουμε στη διάθεση μας προς επεξεργασία.

2.4.2 Εκπαίδευση μοντέλου

Βασικά συστατικά εκπαίδευσης μοντέλου:

- Χρήστες, π.χ. χρήστης χ
- Αντικείμενα, π.χ. αντικείμενο ψ
- Αριθμητική αξιολόγηση ενός χρήστη για ένα αντικείμενο, π.χ. <x, ψ, αριθμητική αξιολόγηση ω>, όπου αυτό σημαίνει ότι η αριθμητική αξιολόγηση, για ένα χρήστη χ σε ένα αντικείμενο ψ, είναι ω. Οι τριάδες <χ, ψ, ω> αποτελούν τον τρόπο κατά τον οποίο αποθηκεύουμε τα δεδομένα ως ένα σετ για την εκπαίδευση του αλγορίθμου.
- Πίνακας δύο διαστάσεων του οποίου ο χ-άξονας αποτελεί τους χρήστες, ο ψ-άξονας αποτελεί τα αντικείμενα και οι τιμές στα κελιά αντιπροσωπεύουν την αριθμητική αξιολόγηση. Ο πίνακας, θεωρητικά, είναι αρκετά αραιός, δηλαδή οι τιμές που έχουμε στη διάθεση μας είναι πολύ λίγες.[11]

Σύμφωνα με τα πιο πάνω, σκοπός της προσέγγισης που ακολουθούμε είναι να συμπληρώσουμε τις κενές θέσεις του πίνακα με χρησιμοποιώντας Collaborative Filtering με βάση μοντέλου.

Στη τεχνική αυτή που χρησιμοποιούμαι μοντέλο οι χρήστες και τα αντικείμενα αναπαριστούνται με από ένα μικρό σετ από Latent Factors τα οποία θα τα χρησιμοποιήσουμε για να προβλέψουμε τις κενές θέσεις του πίνακα. Επομένως, σκοπός μας είναι να μάθουμε ή αλλιώς να εκπαιδεύσουμε αυτά τα Latent Factors ώστε να πάρουμε τα καλύτερα, δυνατό, αποτελέσματα. Για να το πετύχουμε αυτό χρησιμοποιούμαι την προσέγγιση ALS (=Alternating Least Squares) η οποία είναι μέρος της βιβλιοθήκης MLlib που μας παρέχει το Spark.

2.4.2.1 Παράμετροι «ALS» προσέγγισης για εκπαίδευση μοντέλου

- Το «*rank*» αναπαριστά τον αριθμό των «latent factors» τα οποία περιέχει το μοντέλο μας το οποίο θα επιλέξουμε ως το καλύτερο μετά την εκπαίδευση χρησιμοποιώντας τον ALS. Με άλλα λόγια, στόχος μας είναι να αναπαριστάνουμε

το κάθε χρήστη και το κάθε αντικείμενο σαν ένα διάνυσμα από «latent factors» όπου αυτή η αναπαράσταση θα μας εγγυάται καλά αποτελέσματα στη συνέχεια.

- Το «*lambda*» αναπαριστά το «regularization factor» του οποίου σκοπός είναι να μας αποτρέπει από την υπέρ-εκπαίδευση του αλγορίθμου.
- Τα «*iterations*» αναπαριστούν τον αριθμό επαναλήψεων για την εκπαίδευση του αλγορίθμου.[11]

2.4.2.2 Τεχνική εκπαίδευσης του μοντέλου μας

Έχουμε στη διάθεση μας το σετ από δεδομένα με τα οποία θα πρέπει να εκπαιδεύσουμε το μοντέλο μας με αποδοτικό τρόπο ούτως ώστε να μας εγγυηθεί καλά αποτελέσματα. Ο τρόπος που θα ακολουθήσουμε για να φτάσουμε στο στόχο μας είναι να διαμοιράσουμε τα δεδομένα που έχουμε στη διάθεση μας σε δεδομένα εκπαίδευσης και δεδομένα επαλήθευσης.

Χρησιμοποιώντας τις πιο πάνω παραμέτρους της «ALS» προσέγγισης, δηλαδή τα rank, lambda και Iterations θα εκπαιδεύσουμε το μοντέλο μας. Ο τρόπος που θα ακολουθήσουμε είναι ο εξής: Θα θέσουμε ένα σύνολο τιμών ξεχωριστά για κάθε παράμετρο και για κάθε συνδυασμό τιμών αυτών των τριών παραμέτρων θα χρησιμοποιούμαι τα δεδομένα εκπαίδευσης για να εκπαιδεύσουμε το μοντέλο μας και ακολούθως θα αξιολογούμε τα αποτελέσματα που παίρναμε χρησιμοποιώντας τα δεδομένα επαλήθευσης. Για την αξιολόγηση αυτή χρησιμοποιούσαμε τη μετρική ποσοστού επιτυχίας.

Τεχνική εκπαίδευσης μοντέλου σε ψευδό-κώδικα:

Θέτουμε ένα σύνολο τιμών για κάθε παράμετρο:
Rank = […]
Lambda = […]
Iterations = […]
Για όλες τις τιμές του Rank[]
  Για όλες τις τιμές του Lambda[]
    Για όλες τις τιμές του Iterations[]

μοντέλο = ALS εκπαίδευση(rank, lambda, iterations, δεδομένα εκπαίδευσης)

υπολόγισε το ποσοστό επιτυχίας (μοντέλο, δεδομένα επαλήθευσης)

Μετρική αξιολόγησης των αποτελεσμάτων του αλγορίθμου:

Χρησιμοποιούμαι το ποσοστό επιτυχίας που μας δίνουν τα δεδομένα επαλήθευσης με βάση το εκπαιδευμένο μας μοντέλο.

Ποσοστό επιτυχίας:

Το ποσοστό επιτυχίας εξαρτάται από τον αριθμό των αντικειμένων τα οποία συμπίπτουν με τα αντικείμενα τα οποία πραγματικά ενδιαφέρουν ένα χρήστη και ταυτόχρονα του τα έχουμε προτείνει, διαιρώντας τα με τα αντικείμενα τα οποία γνωρίζουμε ότι ενδιαφέρουν πραγματικά ένα χρήστη.

$$\text{Ποσοστό επιτυχίας χρήστη} = \frac{|\ \textit{αντικείμενα που τον ενδιαφέρουν}\ \cap\ \textit{αντικείμενα που του προτείναμε}\ |}{|\ \text{αντικείμενα που τον ενδιαφέρουν}\ |}$$

Παράδειγμα ποσοστού επιτυχίας:

Χρήστης χ:

Τον χρήστη χ τον ενδιαφέρουν τα εξής αντικείμενα:

- Αντικείμενο Α
- Αντικείμενο Β
- Αντικείμενο Γ
- Αντικείμενο Δ

Αντικείμενα τα οποία έχουμε προτείνει στο χρήστη χ:

- Αντικείμενο Α
- Αντικείμενο Β
- Αντικείμενο Ε
- Αντικείμενο Ζ

Υπολογισμός ποσοστού επιτυχίας:

$$\text{Ποσοστό επιτυχίας χρήστη } \chi = \frac{|\ A, B, \Gamma, \Delta\ \cap\ A, B, E, Z\ |}{|\ \text{A, B, Γ, Δ}|}$$

$$\text{Ποσοστό επιτυχίας χρήστη } \chi = \frac{|\ A, B\ |}{|\ \text{A, B, Γ, Δ}|}$$

$$\text{Ποσοστό επιτυχίας χρήστη } \chi = \frac{2}{4}$$

$$\text{Ποσοστό επιτυχίας χρήστη } \chi = 0.5$$

$$\text{Ποσοστό επιτυχίας χρήστη } \chi = 50\ \%$$

2.4.3 «Cold Start» πρόβλημα

Η περίπτωση στην οποία παρουσιάζεται το πιο πάνω πρόβλημα είναι όταν έχουμε ένα νέο χρήστη για τον οποίο δεν έχουμε στη διάθεση μας καμιά πληροφορία για αυτόν, δηλαδή δεν έχουμε το ιστορικό του με το οποίο θα μπορούμε να εισάγουμε στα δεδομένα μας όταν εκπαιδεύουμε το μοντέλο μας.[12]

2.4.3.1 Τρόπος αντιμετώπισης του «Cold Start» προβλήματος

Ο τρόπος με τον οποίο αντιμετωπίζουμε το πιο πάνω πρόβλημα είναι πολύ απλός και χρησιμοποιούμαι τη λογική με την οποία χρησιμοποιεί εξίσου και το Netflix όπου σου ζητά να βαθμολογήσεις κάποιες ταινίες κατά τη διαδικασία εγγραφής ενός νέου χρήστη. Στη δική μας περίπτωση όταν ένας νέος χρήστης κάνει εγγραφή στην Android εφαρμογή του ζητούμε να δώσει μία βαθμολογία κλίμακας από 1 μέχρι 5 για διάφορα αντικείμενα που έχουμε στη διάθεση μας. Επίσης δίνεται η δυνατότητα να αποφύγει την βαθμολογία κάποιων αντικειμένων τα οποία μπορεί να μην γνωρίζει ο ίδιος. Δηλαδή μία τέτοια περίπτωση είναι όταν ζητήσουμε σε ένα χρήστη να βαθμολογήσει ένα εστιατόριο το οποίο δεν έχει επισκεφθεί ποτέ.

Κεφάλαιο 3

Απαιτούμενη Γνώση και Τεχνολογίες

3.1 Λογισμικό ανάπτυξης 26
3.1.1 Java 26
3.1.2 Spark 27
3.1.2.1 Spark Streaming 28
3.1.3 Kafka 28
3.2 Απαιτούμενες τεχνολογίες 29
3.2.1 MySQL 29
3.2.2 PHP 30
3.2.3 JSON 31
3.3 Περιβάλλον Ανάπτυξης 31
3.3.1 Eclipse 31
3.3.2 Android Studio 31
3.4 Διεπαφές προγραμματισμού εφαρμογών(API) 32
3.4.1 Facebook API 32
3.4.2 Android API 33
3.4.3 Estimote Beacon API 34
3.5 Hadoop 34
3.5.1 Αρχιτεκτονική Map-Reduce 35
3.5.2 Αρχιτεκτονική HDFS 36

3.1 Λογισμικό ανάπτυξης

3.1.1 Java

Η Java είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού η οποία έχει σχεδιαστεί με σκοπό να έχει όσο το δυνατό λιγότερες εξαρτήσεις από άλλες εφαρμογές

και για να μπορεί να χρησιμοποιηθεί στο κατανεμημένο περιβάλλον του διαδικτύου. Μπορεί να χρησιμοποιηθεί για την δημιουργία ολοκληρωμένων εφαρμογών και μπορεί να τρέξει είτε σε ένα υπολογιστή είτε κατανεμημένα σε μία συστοιχία από διακομιστές.[16]

Χρήση JAVA στα πλαίσια της διπλωματικής εργασίας: Η γλώσσα προγραμματισμού Java έχει χρησιμοποιηθεί για την ανάλυση των δεδομένων, για την γραφή κώδικα για Spark και για την ανάπτυξη της android εφαρμογής.

3.1.2 Spark

Το Apache Spark είναι ένα πλαίσιο ανοιχτού κώδικα το οποίο μας δίνει τη δυνατότητα παράλληλης επεξεργασία δεδομένων. Το Spark έχει κυρίως αναπτυχθεί για το χειρισμό μεγάλης κλίμακας επεξεργασίας και ανάλυσης δεδομένων. Μας προσφέρει τη δυνατότητα πρόσβασης στα δεδομένα χρησιμοποιώντας το κατανεμημένο σύστημα αρχείων (HDFS). Υποστηρίζει την επεξεργασία στη μνήμη ενισχύοντας με αυτό τον τρόπο την απόδοση εφαρμογών ανάλυσης δεδομένων, αλλά επίσης μπορεί να κάνει συμβατική επεξεργασία η οποία βασίζεται στο δίσκο όταν τα σύνολα δεδομένων είναι μεγάλα σε όγκο για να χωρέσουν στη διαθέσιμη κύρια μνήμη που μας παρέχει το σύστημα. Έχει σχεδιαστεί με τέτοιο τρόπο ώστε να επιταχύνει το ρυθμό ανάλυσης των δεδομένων στο Hadoop. Επίσης μας παρέχει μια πλήρως εξοπλισμένη βιβλιοθήκη μηχανικής μάθησης (MLlib).

Η διεπαφή προγραμματισμού στην οποία επικεντρώνεται το Spark ονομάζεται Resilient Distributed Dataset (RDD) η οποία αποτελεί ένα πολύ-σύνολο από δεδομένα τα οποία διανέμονται σε μια συστοιχία από μηχανές στα οποία έχουμε δικαιώματα μόνο να τα διαβάσουμε. Με αυτό τον τρόπο πετυχαίνουμε την ανοχή σε σφάλματα και επομένως εξασφαλίζουμε την αξιοπιστία.

Το Apache Spark έχει αναπτυχθεί με σκοπό να αντιμετωπίσουμε τους περιορισμούς σε σχέση με την Map-Reduce αρχιτεκτονική όπως αυτή χρησιμοποιείται στο Hadoop η οποία μας αναγκάζει σε μια συγκεκριμένη δομή γραμμικής ροής δεδομένων επί των κατανεμημένων προγραμμάτων. Αυτό επιτυγχάνεται από τις λειτουργίες που μας

παρέχονται από τα Spark RDDs δίνοντας μας την δυνατότητα ενός συνόλου εργασιών για κατανεμημένα προγράμματα.[17]

3.1.2.1 Spark Streaming

Το Spark Streaming αποτελεί μία επέκταση του Spark API το οποίο μας επιτρέπει την επεξεργασία δεδομένων τα οποία φτάνουν σε πραγματικό χρόνο, ενώ ταυτόχρονα μας εγγυάται υψηλή απόδοση σε αυτή την επεξεργασία πληροφοριών αλλά και την ανοχή σε διάφορα σφάλματα τα οποία μπορεί να εμφανιστούν. Τα δεδομένα μπορούν να φθάνουν από διάφορες πηγές όπως για παράδειγμα είναι το Kafka και στη συνέχεια να σταλούν για επεξεργασία. Όταν αυτή η διαδικασία τελειώσει τότε στη συνέχεια τα αποτελέσματα μπορούν να αποθηκευτούν σε κάποιο σύστημα αρχείων ή κάποια βάση δεδομένων.[18]

3.1.3 Kafka

Το Kafka είναι μία κατανεμημένη υπηρεσία καταγραφής, επικοινωνίας και ανταλλαγής μηνυμάτων διαμέσου της αρχιτεκτονικής ενός συστήματος. Τα βασικά συστατικά τα οποία αποτελείται το σύστημα που μας παρέχει το Kafka είναι τα εξής:

- Topics: Το Kafka είναι υπεύθυνο να διατηρεί τις τροφοδοσίες των μηνυμάτων σε διάφορες κατηγορίες οι οποίες ονομάζονται topics.
- Producers: Οι διαδικασίες οι οποίες παράγουν και περνούν μηνύματα προς ένα Kafka topic ονομάζονται producers.
- Consumers: Οι διαδικασίες οι οποίες επεξεργάζονται τη τροφοδοσία των μηνυμάτων τα οποία έχουν παραχθεί από τους producers ονομάζονται consumers.
- Broker: Το Kafka λειτουργά ως μία συστοιχία η οποία αποτελείται από ένα ή περισσότερους διακομιστές όπου ο κάθε ένας από αυτούς ονομάζεται broker.

Άρα, σέ ένα αφαιρετικό επίπεδο, οι producers στέλνουν μηνύματα στο δίκτυο και κατά συνέπεια στη συστοιχία από διακομιστές του Kafka και στη συνέχεια φτάνουν στους consumers. Η επικοινωνία μεταξύ των πελατών και των διακομιστών γίνεται διαμέσου

ενός απλού και υψηλής ποιότητας πρωτοκόλλου TCP. Υπάρχουν διαθέσιμοι πελάτες σε πολλαπλές γλώσσες προγραμματισμού όπως για παράδειγμα σε Java(Java Client).[19]

3.2 Απαιτούμενες τεχνολογίες

3.2.1 MySQL

Η MySQL είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης μίας βάσης δεδομένων. Οι πληροφορίες σε μία βάση δεδομένων τύπου MySQL αποθηκεύονται σε μορφή πινάκων όπου ο κάθε πίνακας αντιπροσωπεύει ένα αντικείμενο/οντότητα και μέσα σε αυτόν φυλάμε τα πεδία του με το όνομα και το τύπο τους. Οι βάσεις δεδομένων MySQL συνήθως χρησιμοποιούνται για την ανάπτυξη διαδικτυακών εφαρμογών και είναι απευθείας προσβάσιμες χρησιμοποιώντας γλώσσες προγραμματισμού όπως είναι η PHP, Java κ.τ.λ. Η αναζήτηση σε μία τέτοια βάση δεδομένων γίνεται χρησιμοποιώντας SQL εντολές.[20]

Χρήση MySQL στα πλαίσια της διπλωματικής εργασίας: Χρησιμοποιείται για να φυλάμε τα δεδομένα που απαρτίζουν την Android εφαρμογή και γενικά τα δεδομένα που χρειάζεται για να εκτελεσθεί ο αλγόριθμος συστάσεων. Η ανταλλαγή των δεδομένων από τη βάση δεδομένων και την android εφαρμογή γίνεται με τη χρήση JSON.

Πλεονεκτήματα χρήσης MySQL:

- Επεκτασιμότητα και ευελιξία:
  Ο διακομιστής της βάσης δεδομένων MySQL παρέχει επεκτασιμότητα και επίσης ευελιξία γιατί η ανοιχτού κώδικα μορφή του μας επιτρέπει να την προσαρμόζουμε στα μέτρα μας ή να προσθέσουμε δικές μας απαιτήσεις στο διακομιστή της βάσης δεδομένων.

- Υψηλή απόδοση:
  Η αρχιτεκτονική αποθήκευσης που χρησιμοποιεί η MySQL μας επιτρέπει να διαμορφώσουμε το διακομιστή της βάσης δεδομένων ειδικά για συγκεκριμένες

εφαρμογές με αποτέλεσμα την υψηλή απόδοση. Η MySQL έχει τη δυνατότητα να εξυπηρετεί μεγάλου όγκου αναζητήσεις και γενικά να καλύψει απαιτητικές προσδοκίες ενός συστήματος με τη χρήση μηχανισμών και τεχνικών που ενισχύουν την απόδοση.

- Υψηλή διαθεσιμότητα:

  Η MySQL μας προσφέρει αξιοπιστία και συνεχή διαθεσιμότητα, ενώ μας εγγυάται πλήρη εξυπηρέτηση όλο το 24ώρο. Αυτό το πετυχαίνει προσφέροντας μια ποικιλία από υψηλής διαθεσιμότητάς και επίδοσης διακομιστών οι οποίοι αντιμετωπίζουν επιτυχώς προβλήματα που μπορεί να εμφανιστούν όπως είναι αυτά του φόρτου εργασίας ή της διαχείρισης λαθών.

- Προστασία δεδομένων:

  Η φύλαξη των δεδομένων και των διαφόρων στοιχείων είναι το πιο σημαντικό για μία βάση δεδομένων. Η MySQL μας προσφέρει εξαιρετικές δυνατότητες ασφαλείας οι οποίες εξασφαλίζουν την απόλυτη προστασία των δεδομένων παρέχοντας μηχανισμούς διασφάλισης εξουσιοδότησης των χρηστών. Επίσης παρέχει βοηθητικά προγράμματα δημιουργίας αντιγράφων και ανάκαμψης μετά από εμφάνιση κάποιου προβλήματος.

- Εύκολη διαχείριση:

  Η MySQL προσφέρει τη δυνατότητα ταχείας εκκίνησης σχετικά με την ολοκληρωμένη εγκατάσταση της. Μας παρέχει μία πλήρη σειρά από γραφικά εργαλεία διαχείρισης, αντιμετώπισης προβλημάτων και ελέγχου των λειτουργιών, καθώς και εργαλεία τρίτων που χειρίζονται θέματα όπως είναι η παρακολούθηση των επιδόσεων.[21]

3.2.2 PHP

Η PHP με αρχικά τα οποία αντιστοιχούν σε Personal Home Page είναι μία script γλώσσα της οποία η επεξεργασία της γίνεται στον εξυπηρετητή. Σκοπός αυτής της γλώσσας είναι η χρήση της για ανάπτυξη ιστοσελίδας και γενικά μπορεί να χρησιμοποιηθεί και ως γλώσσα προγραμματισμού γενικού σκοπού. Ο κώδικας ο οποίος

είναι γραμμένος σε PHP μπορεί να ενσωματωθεί σε κώδικα HTML ή να χρησιμοποιηθεί σε συστήματα διαχείρισης περιεχομένου μίας ιστοσελίδας και σε διαδικτυακά πλαίσια(web framework).[22]

Χρήση PHP στα πλαίσια της διπλωματικής εργασίας: Χρησιμοποιείται στο σύστημα διαχείρισης των δεδομένων της Android εφαρμογής και για την εξυπηρέτηση αιτημάτων που στέλνει η εφαρμογή προς στη βάση δεδομένων.

3.2.3 JSON

Το JSON με αρχικά τα οποία αντιστοιχούν σε JavaScript Object Notation αποτελεί ένα οργανωμένο και εύκολο σε πρόσβαση τρόπο για να αποθηκεύουμε τις πληροφορίες. Μας δίνει μια ανθρώπινα αναγνώσιμη συλλογή των δεδομένων που μπορούμε να έχουμε πρόσβαση με τρόπο λογικό.[23]

Χρήση JSON στα πλαίσια της διπλωματικής εργασίας: Η χρήση του JSON γίνεται στα πλαίσια ανταλλαγής δεδομένων μεταξύ της βάσης δεδομένων και της Android εφαρμογή χρησιμοποιώντας MySQL.

3.3 Περιβάλλον Ανάπτυξης

3.3.1 Eclipse

Η Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης(IDE) το οποίο μας παρέχει ένα ολοκληρωμένο χώρο εργασίας γραφής κώδικα για την ανάπτυξη εφαρμογών κυρίως σε γλώσσα Java, αλλά και σε άλλες γλώσσες προγραμματισμού.[24] Επίσης είναι ενσωματωμένο με μηχανισμούς εντοπισμού λογικών και συντακτικών λαθών στο κώδικα, μας προτείνει λύσεις στα λάθη που μπορεί να εμφανιστούν και επίσης διαθέτει κονσόλα καταγραφής αποτελεσμάτων κατά την εκτέλεση προγραμμάτων.

3.3.2 Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης(IDE) για τη δημιουργία εφαρμογών Android. Αποτελεί ένα ισχυρό επεξεργαστή κώδικα και

εργαλείων ανάπτυξης και προσφέρει διάφορα χαρακτηριστικά τα οποία ενισχύουν τη κατασκευή μίας εφαρμογής. Επίσης μας παρέχει ολοκληρωμένο πρόγραμμα ανάπτυξης του γραφικού περιβάλλοντος της εφαρμογής.[25]

3.4 Διεπαφές προγραμματισμού εφαρμογών (API)

API είναι η διεπαφή η οποία μας παρέχεται από μία εφαρμογή ή βιβλιοθήκη, μέσω της οποίας μας δίνετε η δυνατότητα να ανταλλάξουμε δεδομένα με αυτή και να αιτηθούμε διάφορες δυνατότητες που μας προσφέρει.[26]

3.4.1 Facebook API

Το Facebook ονομάζει τη δυνατότητα αυτή ως Graph API. Μέσω του Graph API, μας παρέχετε η δυνατότητα διαφόρων ενεργειών που μπορούμε να πράξουμε για τη πλοήγηση μας σε διάφορα δεδομένα τα οποία αφορούν πληροφορίες λογαριασμών/σελίδων που υπάρχουν στο Facebook.

Μέρη του Graph API:

- Κόμβος: σημαντικές παράμετροι όπως είναι ο χρήστης, μια σελίδα, ένα σχόλιο ή μια φωτογραφία.
- Ακμή: οι διασυνδέσεις μεταξύ αυτών των σημαντικών παραμέτρων όπως είναι οι φωτογραφίες μιας σελίδας ή τα σχόλια σε μια φωτογραφία.
- Πεδία κόμβου: διάφορες πληροφορίες σχετικά με αυτή τη σημαντική παράμετρο όπως είναι η ημερομηνία γεννήσεως ενός χρήστη ή το όνομα μιας σελίδας.

Η δομή του Graph API μας δίνει τη δυνατότητα να διαπερνούμε δεδομένα σύμφωνα με τα δημόσια δικαιώματα που δίνουμε στο χρήστη. Δικαιώματα όπως να διατρέξουμε τις προσωπικές πληροφορίες ενός χρήστη για παράδειγμα.

Για να μπορούμε να προσπελάσουμε τα πιο πάνω δικαιώματα γίνεται χρήση των Access Tokens τα οποία παρέχονται από το Facebook.

Access Token:

Όταν ένας χρήστης εισέρχεται σε μια εφαρμογή μέσω του Facebook (όπως χρησιμοποιούν πολλές android εφαρμογές για παράδειγμα) τότε η εφαρμογή λαμβάνει ένα access token το οποίο της επιτρέπει την ασφαλή πλοήγηση της στο Facebook API. Υπάρχουν 4 είδη access token τα οποία είναι: User access token, App access token, Page access token και Client access token.

User access token:

Η χρήση αυτού του access token, όταν η εφαρμογή καλεί το API, μας δίνει τη δυνατότητα να διαβάσουμε, να γράψουμε ή να επεξεργαστούμε τα δεδομένα τα οποία αφορούν ένα συγκεκριμένο χρήστη του Facebook.[27]

3.4.2 Android API

Το Android μας παρέχει ένα πλαίσιο εφαρμογής (framework) στο οποίο μας επιτρέπει να κτίσουμε διάφορες εφαρμογές/παιχνίδια για κινητές συσκευές οι οποίες χρησιμοποιούν android λογισμικό.

Γλώσσα προγραμματισμού που χρησιμοποιείται: JAVA, xml για τη κατασκευή των γραφικών.

Σημεία εισόδου μιας Android εφαρμογής (entry points):

- Χρήση ενός activity για να παρέχουμε στο χρήστη μια διεπαφή στην οθόνη της κινητής συσκευής του, μέσω της οποίας εκτελείτε κάποια ενέργεια.
- Χρήση ενός service το οποίο εκτελείτε ανεξάρτητα, καθώς τρέχει στο παρασκήνιο της εφαρμογής. (background service)

Στοιχεία τα οποία συνάπτουν την ανάπτυξη μιας Android εφαρμογής:

- Manifest
- Builder (app)
- Layouts (xml)
- Java
- Styles [28]

Για να διατρέξουμε διάφορες υπηρεσίες μέσω της Android εφαρμογής οι οποίες υπηρεσίες χρησιμοποιούν χαρακτηριστικά μίας κινητής συσκευής όπως για παράδειγμα η χρήση της κάμερας πρέπει να χρησιμοποιήσουμε τη διάθεση δικαιωμάτων την οποία ο χρήστης θα μπορεί να εγκρίνει ούτως ώστε να έχουμε πρόσβαση. Η έγκριση των δικαιωμάτων που ζητά μία Android εφαρμογή γίνεται κατά τη λήψη και εγκατάσταση της στη κινητή συσκευή. Ένα πολύ απλό παράδειγμα είναι η χρήση της Facebook εφαρμογής μέσω του οποίου μπορούμε να παρατηρήσουμε μια μεγάλη λίστα από δικαιώματα στο οποίο μας ζητά να λάβει όταν εμείς αποδεχόμαστε να εγκαταστήσουμε την εφαρμογή στο κινητό μας.

3.4.3 Estimote Beacon API

Estimote βιβλιοθήκη (SDK): Η βιβλιοθήκη η οποία μας παρέχεται από την Estimote μέσω της οποίας μας δίνετε η δυνατότητα να λαμβάνουμε ή να στέλνουμε δεδομένα και αιτήσεις από και προς το beacon.

Δυνατότητες διασύνδεσης έξυπνης κινητής συσκευής και estimote beacon:
Μία κινητή συσκευή μπορεί να λαμβάνει ειδοποιήσεις από ένα ή περισσότερα beacons ταυτόχρονα τα οποία βρίσκονται σε προσιτές αποστάσεις μεταξύ τους.

Αναγνώριση beacon μέσω μοναδικών αριθμών: Το κάθε beacon αποτελείται από μια μοναδική φόρμουλα την οποία διαιρούμε σε 3 συμβολοσειρές.

- proximity UUID (16 bytes)
- major number (2 bytes)
- minor number (2 bytes) [29]

3.5 Hadoop

Η βιβλιοθήκη λογισμικού Apache Hadoop είναι ένα πλαίσιο το οποίο μας επιτρέπει τη κατανεμημένη επεξεργασία μεγάλων όγκων δεδομένων. Είναι ένας τρόπος κατά τον οποίο αποθηκεύουμε ένα τεράστιο σύνολο από δεδομένα διαμέσου της χρήσης μιας συστοιχίας από διακομιστές χρησιμοποιώντας προσεγγίσεις από απλά προγραμματιστικά μοντέλα. Στη συνέχεια αναλύουμε τα δεδομένα αυτά με

κατανεμημένο/παράλληλο τρόπο με σκοπό να εξάγουμε κάποια συμπεράσματα ή κάποιες χρήσιμες πληροφορίες.

Το Hadoop είναι μία υλοποίηση του Map-Reduce και προσφέρεται για δημόσια χρήση, ενώ η κύρια προγραμματιστική γλώσσα που χρησιμοποιείται είναι η Java. Σε γενικές γραμμές, ο πυρήνας του Hadoop αποτελείται από δύο βασικά μέρη, το πρώτο αποτελεί το πλαίσιο όπου γίνεται η επεξεργασία των δεδομένων και το δεύτερο αποτελεί ένα κατανεμημένο σύστημα αρχείων όπου χρησιμοποιείται για την αποθήκευση των δεδομένων.

Για το πρώτο μέρος χρησιμοποιείται το Map-Reduce το οποίο αποτελεί ένα προγραμματιστικό μοντέλο το οποίο έχει τη δυνατότητα να κατανέμει τα δεδομένα σε συστοιχίες οι οποίες αποτελούνται από ένα αριθμό από κόμβους και μας επιτρέπει τη παράλληλη λειτουργία και επεξεργασία αυτών των δεδομένων. Το περιβάλλον του Map-Reduce έχει το πλήρη έλεγχο και την ευθύνη για την διαμοίραση των δεδομένων, καθώς και το προγραμματισμό και την επικοινωνία μεταξύ των μηχανών για την σειρά εκτέλεσης του προγράμματος. Επίσης είναι υπεύθυνο για το χειρισμό διαφόρων αποτυχιών που μπορεί να προκύψουν κατά την εκτέλεση του προγράμματος και να δράσει με τον κατάλληλο τρόπο.[30]

3.5.1 Αρχιτεκτονική Map-Reduce

Ο μηχανισμός του Map-Reduce στο Hadoop ακολουθεί ένα συγκεκριμένο μοντέλο και συγκεκριμένα αυτό του master & slaves. Το μοντέλο αυτό αποτελείται από το master κόμβο ο οποίος είναι υπεύθυνος για το καταμερισμό των εργασιών και τους slave κόμβους οι οποίοι εκτελούν τις εργασίες και επιστέφουν τα διάφορα αποτελέσματα στο master κόμβο. Ο master κόμβος έχει το ρόλο του «job tracker», δηλαδή το καταμερισμό των εργασιών, ενώ οι slave κόμβοι έχουν το ρόλο των «task trackers», δηλαδή εκτελούν την εργασία η οποία τους έχει ανατεθεί από τον «job tracker».
Για το δεύτερο μέρος χρησιμοποιείται το κατανεμημένο σύστημα αρχείων, το Hadoop Distributed File System (HDFS). Είναι σχεδιασμένο με τέτοιο τόπο ώστε να μας επιτρέπει την αποθήκευση μεγάλου όγκου από δεδομένα, καθώς και να μας προσφέρει αξιοπιστία στα δεδομένα.[30]

3.5.2 Αρχιτεκτονική HDFS

Το HDFS μας παρέχει αξιοπιστία μέσω της αντιγραφής των δεδομένων του συστήματος σε περισσότερους από ένα κόμβους. Με αυτό τον τρόπο τα δεδομένα μπορούμε να τα ανακτήσουμε από διάφορους κόμβους οι οποίοι κρατούν αντίγραφα των δεδομένων αυτών. Οι λόγοι που γίνεται αυτό είναι σε περιπτώσεις όπου κάποιοι κόμβοι είναι είτε απασχολημένοι, είτε υπερφορτωμένοι ή μη-διαθέσιμοι τότε αυτό σημαίνει ότι θα αναζητήσουμε τα δεδομένα από άλλους. Ο master κόμβος στο HDFS έχει το ρόλο του «name κόμβου» ο οποίος έχει την ευθύνη να κρατά διάφορες μετά-πληροφορίες οι οποίες είναι χρήσιμες για την σωστή διαχείριση στο HDFS.[30]

Κεφάλαιο 4

Αρχιτεκτονική συστήματος

4.1 Ορισμός αρχιτεκτονικής συστήματος: 37
4.2 Βασικά συστατικά συστήματος 37
4.3 Σχήμα αρχιτεκτονικής συστήματος 38
4.4 Επεξήγηση αρχιτεκτονικής συστήματος 40
4.4.1 Επεξήγηση των αισθητήρων που χρησιμοποιήθηκαν 40
4.4.2 Επεξήγηση των οντοτήτων 41
4.4.3 Επεξήγηση των φάσεων / λειτουργιών 44
4.4.4 Βάση δεδομένων 47

4.1 Ορισμός αρχιτεκτονικής συστήματος

Η αρχιτεκτονική ενός συστήματος είναι ένα αρκετά σημαντικό κομμάτι το οποίο μας υποδηλώνει την οργάνωση του συστήματος ως μία σύνθεση από συστατικά. Τα συστατικά αυτά μας δείχνουν πως εκτελούνται οι λειτουργίες του συστήματος, πως γίνεται η ανταλλαγή και η παροχή δεδομένων και πως γίνονται διάφορες αναβαθμίσεις στο σύστημα όπως για παράδειγμα η αναβάθμιση της βάσης δεδομένων κάτω από συγκεκριμένες προϋποθέσεις. Επίσης περιγράφεται η σύνθεση των στοιχείων μεταξύ τους και η φυσική υλοποίηση του συστήματος.[13]

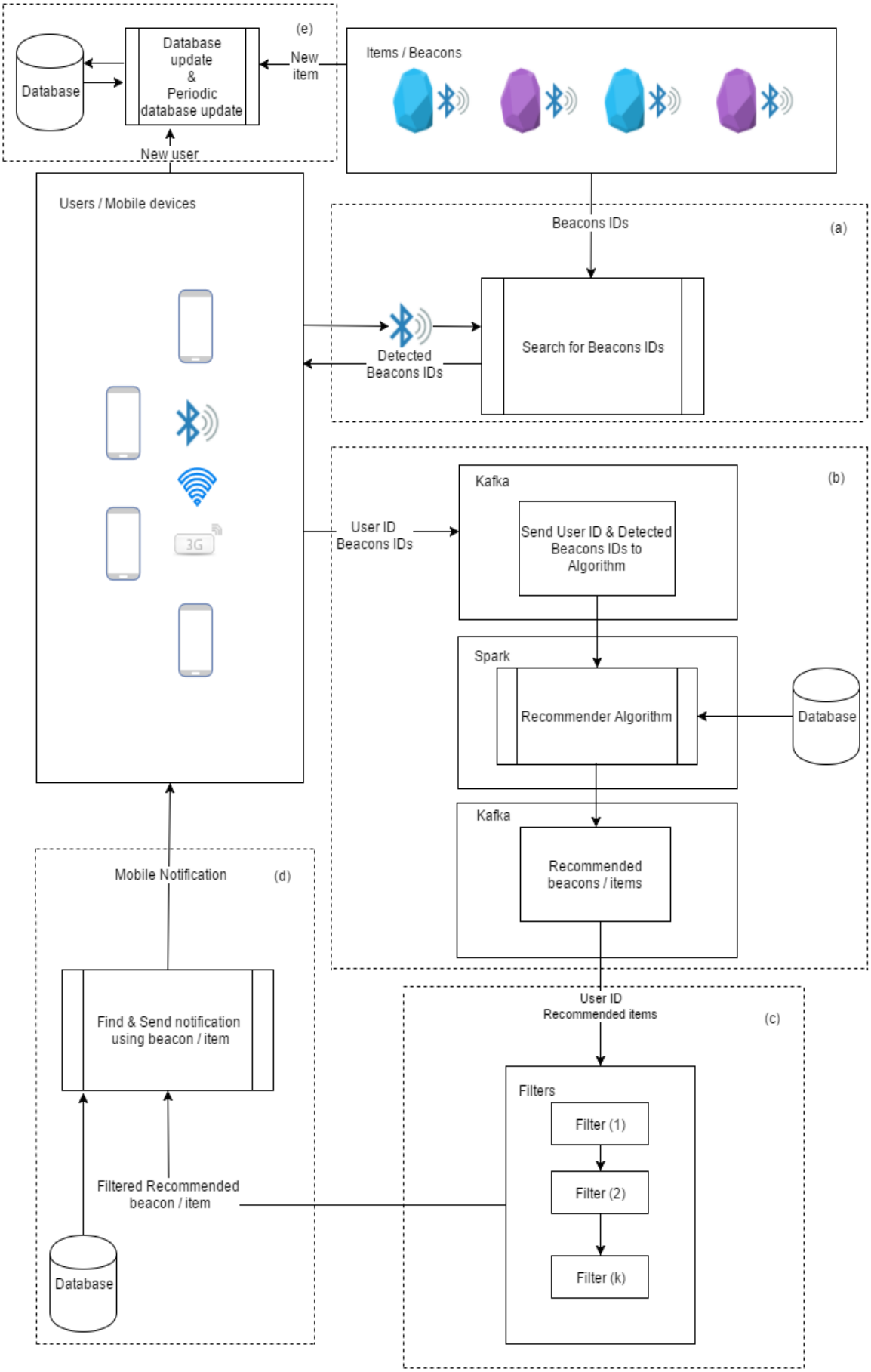
4.2 Βασικά συστατικά συστήματος

- Χρήστες τους οποίους αντιπροσωπεύουν οι κινητές συσκευές.
- Αντικείμενα τα οποία αντιπροσωπεύουν οι αισθητήρες και συγκεκριμένα τα Estimate Beacons.

- Λειτουργία ανταλλαγής δεδομένων μεταξύ Estimate Beacons και κινητών συσκευών.
- Λειτουργία ανταλλαγής δεδομένων μεταξύ Cloud και κινητών συσκευών.
- Λειτουργία αναβάθμισης βάσης δεδομένων.

Σε αφηρημένο επίπεδο, το σύστημα μας διαχειρίζεται χρήστες οι οποίοι αλληλεπιδρούν με διάφορους αισθητήρες που βρίσκονται τοποθετημένοι στο πραγματικό κόσμο, καθώς και με ένα Cloud. Στη συνέχεια οι χρήστες αιτούνται τη παροχή υπηρεσιών από αυτούς και γίνεται μία συγκεκριμένη ροή ανάλυσης δεδομένων και επιλογής ώστε να καταλήξουμε ποιο αντικείμενο θα προτείνουμε στο χρήστη.

4.3 Σχήμα αρχιτεκτονικής συστήματος



Σχήμα 4.1 Αρχιτεκτονική συστήματος

4.4 Επεξήγηση αρχιτεκτονικής συστήματος

Διευκρινήσεις στοιχείων σχήματος:

- Τα μη-διακεκομμένα ορθογώνια αναπαριστούν μία οντότητα.
- Τα διακεκομμένα ορθογώνια αναπαριστούν μία λειτουργία.
- Το αλφαβητικό γράμμα το οποίο βρίσκεται τοποθετημένο στη δεξιά γωνία των διακεκομμένων ορθογωνίων χρησιμοποιείται για να ορίσουμε τη μοναδικότητα των φάσεων όπου κάθε φάση αντιπροσωπεύει μία λειτουργία.

4.4.1 Επεξήγηση των αισθητήρων που χρησιμοποιήθηκαν

Estimote Beacon αισθητήρας: Είναι ένας μικρός υπολογιστής ο οποίος είναι εφοδιασμένος με τα εξής τεχνικά χαρακτηριστικά:

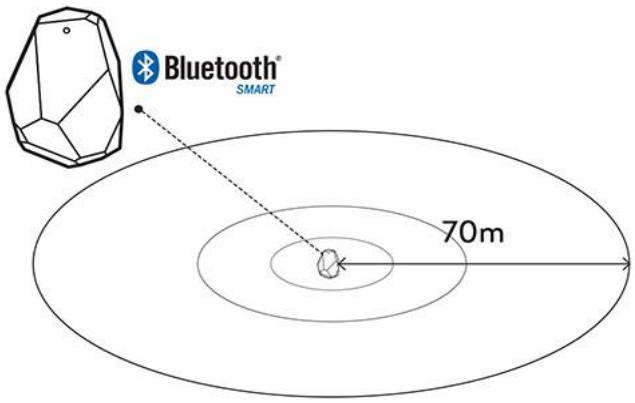
1. 32-bit ARM® Cortex M0 CPU
2. Επιταχυνσιόμετρο του οποίου σκοπός είναι να ανιχνεύει το προσανατολισμό της συσκευής στην οποία βρίσκεται εγκαταστημένο.
3. Αισθητήρας θερμοκρασίας
4. 2.4 GHz Bluetooth 4.0 Smart ή Bluetooth χαμηλής ενέργειας το οποίο δίνει τη δυνατότητα 3 χρόνων ζωής στον αισθητήρα.



Σχήμα 4.2 Estimote Beacon

Φάσμα σήματος που εκπέμπει το Estimote Beacon:

Το σήμα Bluetooth που εκπέμπει έχει εμβέλεια 70 μέτρων. Παρόλα αυτά, το σήμα μπορεί να διαθλάται και να εμποδίζεται από διάφορους παράγοντες όπως είναι τα κτίρια και τα ανθρώπινα σώματα. Για αυτό σε πραγματικές συνθήκες, θα πρέπει να αναμένουμε εύρος το οποίο φθάνει τα 50 μέτρα περίπου.[14]



Σχήμα 4.3 Εύρος σάρωσης του Estimote Beacon

4.4.2 Επεξήγηση των οντοτήτων

Οντότητες:

- Χρήστες / Κινητές συσκευές
- Αντικείμενα / Estimote beacons

Χρήστες / Κινητές συσκευές:

Ορισμός: Οι κινητές συσκευές τρέχουν στο λογισμικό τους την Android εφαρμογή με όνομα «Meet Nicosia». Σκοπός της εφαρμογής είναι η αλληλοεπικοινωνία της κινητής συσκευής με τα αντικείμενα (τα οποία αντιπροσωπεύουν τα Estimote Beacons) και με το δίκτυο στο οποίο θα τρέχει ο αλγόριθμος συστάσεων. Δηλαδή η εφαρμογή είναι υπεύθυνη να εντοπίζει τα Estimote Beacons, να ενημερώνει τη κινητή συσκευή για αυτά που εντοπίστηκαν, να δίνει το σήμα έναρξης της διαδικασίας εκτέλεσης του αλγορίθμου και τέλος να λαμβάνει τα αποτελέσματα από τον αλγόριθμο και να τα παρουσιάζει στη κινητή συσκευή του χρήστη.

Αλληλοεπικοινωνία των κινητών συσκευών με τα άλλα συστατικά τα οποία απαρτίζουν την αρχιτεκτονική:

Επικοινωνία με τα Estimote Beacons: Οι κινητές συσκευές επικοινωνούν με τα Estimote Beacons. Ο τρόπος με τον οποίο γίνεται η επικοινωνία είναι πολύ απλός. Δηλαδή, το κινητό τηλέφωνο, χρησιμοποιώντας τεχνολογία Bluetooth, σαρώνει τη γύρω περιοχή του. Η σάρωση αυτή γίνεται περιοδικά κάθε χ δευτερόλεπτα και διαρκεί ψ δευτερόλεπτα. Η κινητή συσκευή λαμβάνει μία σειρά από αριθμούς και γράμματα τα οποία απαρτίζουν τη μοναδικότητα του κάθε Estimote Beacon.
Η απόσταση σάρωσης που γίνεται εξαρτάται από το κινητό τηλέφωνο και τη τεχνολογία Bluetooth την οποία χρησιμοποιεί. Για παράδειγμα ένα γενικό στατιστικό αναφέρει ότι η κλασική Bluetooth τεχνολογία διαθέτει απόσταση σάρωσης μέχρι 100 μέτρα ενώ η Smart Bluetooth τεχνολογία μπορεί να ξεπεράσει αυτή την απόσταση των 100 μέτρων.Η απόσταση σάρωσης που διαθέτει το Estimote Beacon είναι 70 μέτρα και χρησιμοποιεί Smart Bluetooth τεχνολογία.[15]

Επικοινωνία με το Kafka:

Επικοινωνία από τη κινητή συσκευή προς το Kafka: Η Android εφαρμογή η οποία τρέχει στη κινητή συσκευή είναι υπεύθυνη να δώσει τις απαραίτητες πληροφορίες στους Kafka Receivers όπως φαίνεται στο σχήμα από τη μεταφορά δεδομένων από την οντότητα Χρήστες προς τη φάση (β). Οι πληροφορίες αυτές που μεταφέρονται είναι ο χρήστης τον οποίο παρουσιάζουμε με ένα μοναδικό αριθμό και τα αντικείμενα τα οποία επίσης τα παρουσιάζομαι με ένα μοναδικό αριθμό το κάθε ένα.

Επικοινωνία από το Kafka προς τη κινητή συσκευή: Η Android εφαρμογή έχει την ευθύνη να λάβει τα φιλτραρισμένα αποτελέσματα όπως φαίνονται και από το σχήμα όπου λαμβάνονται τα αποτελέσματα από τη φάση (δ) στην οντότητα Χρήστες. Τα αποτελέσματα αυτά περιλαμβάνουν το αντικείμενο, αν υπάρχει βέβαια, το οποίο θα προτείνουμε στο χρήστη μέσω της εφαρμογής.
Τεχνολογίες επικοινωνίας:

3G ή Wi-Fi: Οι υπηρεσίες διαδικτύου 3G ή Wi-Fi(αν υπάρχει διαθέσιμη) τις χρησιμοποιούμαι για αν δώσουμε τις αναγκαίες πληροφορίες στον αλγόριθμο και να λάβουμε τα αποτελέσματα από αυτόν.

Bluetooth: Η τεχνολογία Bluetooth χρησιμοποιείται για την σάρωση και τη λήψη πληροφορίας από το Estimote Beacon προς την Android κινητή συσκευή.

Παρουσίαση χρήστη: Όπως αναφέραμε και πιο πάνω, ο χρήστης παρουσιάζεται με τη χρήση ενός μοναδικού αριθμού.

Αντικείμενα / Estimote Beacons:

Τα αντικείμενα αντιπροσωπεύουν οτιδήποτε μπορεί να χρησιμοποιηθεί για να το προτείνουμε σε ένα χρήστη. Για παράδειγμα ένα αντικείμενο μπορεί να αποτελεί ένα προϊόν, ένα εστιατόριο ή μία τοποθεσία. Τα αντικείμενα είναι αυτά τα οποία βρίσκονται πίσω από τα Estimote Beacons. Δηλαδή, για κάθε Estimote Beacon έχουμε το αντίστοιχο αντικείμενο. Σκοπός των Estimote Beacons είναι η επικοινωνία με το χρήστη.

Αλληλοεπικοινωνία των Estimote Beacons με τα άλλα συστατικά τα οποία απαρτίζουν την αρχιτεκτονική:

Επικοινωνία με την Android εφαρμογή η οποία τρέχει σε μία κινητή συσκευή: Τα Estimote Beacons σαρώνουν τη γύρω περιοχή τους και όταν μία κινητή συσκευή σαρώσει στο πεδίο του Beacon τότε γίνεται αποστολή μίας σειράς από μοναδικούς αριθμούς και γράμματα η οποία αντιπροσωπεύει τη μοναδικότητα του.

Τεχνολογίες επικοινωνίας:

Bluetooth: Η τεχνολογία Bluetooth χρησιμοποιείται για την αποστολή πληροφορίας από το Estimote Beacon προς την Android κινητή συσκευή.
Παρουσίαση αντικειμένου: Κάθε αντικείμενο παρουσιάζεται με τη χρήση ενός μοναδικού αριθμού.

4.4.3 Επεξήγηση των φάσεων / λειτουργιών

Φάση / Λειτουργία (α): Η φάση αυτή αφορά την αλληλοεπικοινωνία μεταξύ των κινητών συσκευών και των Estimote Beacons η οποία έχει αναλυθεί πιο πάνω.

Φάση / Λειτουργία (β): Η φάση αυτή αφορά την ορθή εκτέλεση του αλγορίθμου και σε αυτή τη φάση έχουμε τρία βήματα για την ολοκλήρωση της.

1^{ο} Βήμα: Η εισαγωγή των δεδομένων:
Τα δεδομένα λαμβάνονται από τις κινητές συσκευές των χρηστών και αποτελούν το μοναδικό αριθμό του χρήστη και τους μοναδικούς αριθμούς των αντικειμένων τα οποία εντοπίστηκαν στη διαδικασία (α). Τα δεδομένα αυτά λαμβάνονται από τους Kafka producers οι οποίοι τα βάζουν σε μια σειρά ή με άλλα λογία σε μία ουρά.

2^{ο} Βήμα: Διαδικασία εκτέλεσης του αλγορίθμου:
Όταν τα δεδομένα τα οποία βρίσκονται στην ουρά και ελέγχονται από τους Kafka producers φτάσουν στη πρώτη θέση της ουράς τότε αυτό σημαίνει ότι είναι έτοιμα να μπουν ως δεδομένα εισαγωγής στον αλγόριθμο. Για την εκτέλεση του αλγορίθμου χρησιμοποιείται το Apache spark.

3^{ο} Βήμα: Εξαγωγή των αποτελεσμάτων:
Σε αυτό το βήμα σκοπός είναι η λήψη των αποτελεσμάτων από την εκτέλεση του αλγορίθμου. Για τη παραλαβή αυτών των αποτελεσμάτων γίνεται χρήση των kafka consumers οι οποίοι είναι υπεύθυνοι να κρατήσουν την ορθή σειρά των αποτελεσμάτων ανάλογα με τη ροή των δεδομένων που έρχονται από το 1^{ο} βήμα. Και σε αυτή τη περίπτωση οι consumers χρησιμοποιούν την λογική της ουράς για να μας εγγυηθούν τη σωστή σειρά ροής δεδομένων από τους Kafka Producers στους Kafka Consumers.

Γενικά σε αυτή τη διαδικασία έχουμε μία κύρια πρόκληση. Η πρόκληση αυτή είναι η γρήγορη εξυπηρέτηση των δεδομένων που φτάνουν στους kafka producers και ταυτόχρονα η ορθή λήψη των αποτελεσμάτων του αλγορίθμου από τους kafka consumers. Άρα, από αυτή τη διαδικασία έχουμε την εξαγωγή των αποτελεσμάτων του

αλγορίθμου τα οποία, για κάθε χρήστη, είναι τα αντικείμενα που πρόκειται να προτείνουμε σε αυτόν.

Φάση / Λειτουργία (γ): Διαδικασία φιλτραρίσματος των δεδομένων μας:

Σε αυτό το σημείο έχουμε στη διάθεση μας, για κάθε χρήστη, τα αποτελέσματα, δηλαδή μία λίστα από αντικείμενα. Τα αντικείμενα αυτά πρέπει να περάσουν μέσα από μια σειρά από φίλτρα για να μας δώσουν το τελικό αποτέλεσμα το οποίο θα είναι ένα μοναδικό αντικείμενο.

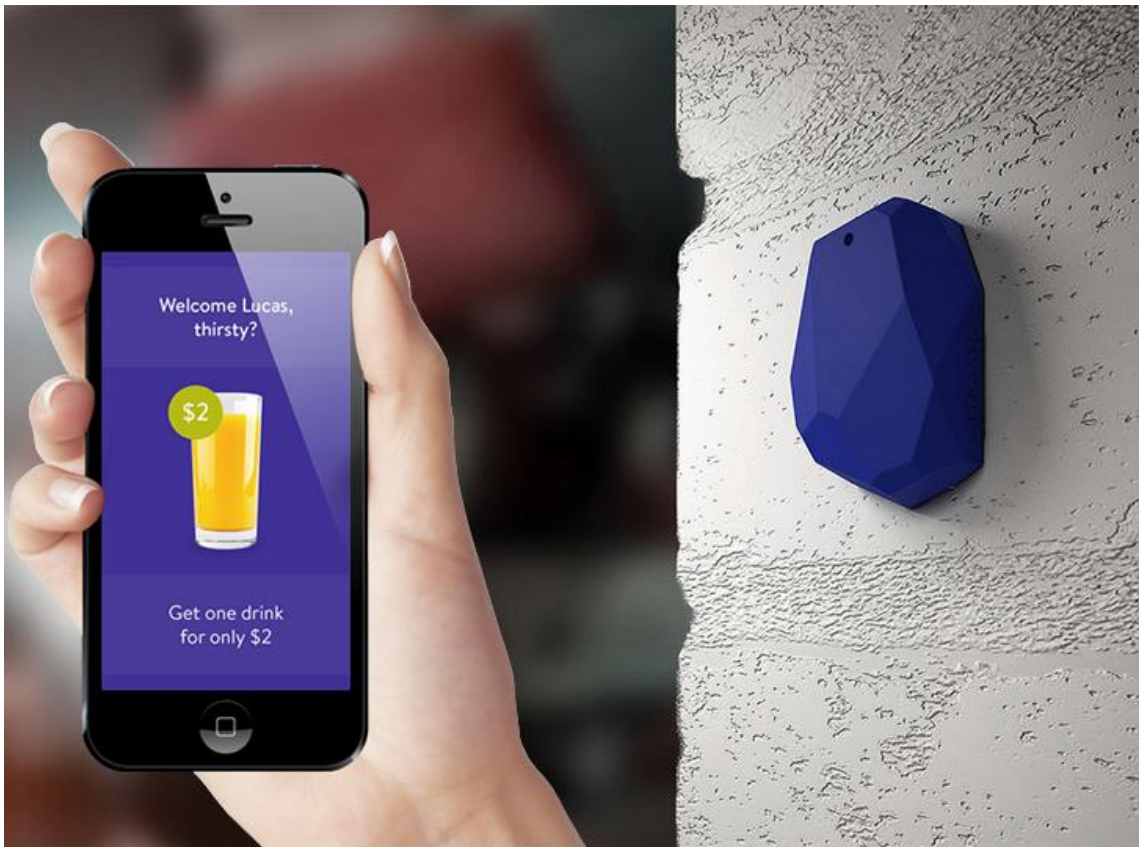
Φάση / Λειτουργία (δ): Διαδικασία εντοπισμού ειδοποίησης σύμφωνα με το αντικείμενο που θα προτείνουμε στο χρήστη.

Σε αυτό το κομμάτι σκοπός μας είναι να εμφανίσουμε σαν ειδοποίηση στη κινητή συσκευή του χρήστη το κατάλληλο μήνυμα. Για παράδειγμα αν πρόκειται να προτείνουμε σε ένα χρήστη να αγοράσει ένα ποδήλατο τότε η ειδοποίηση θα είναι διαφορετική από το να προτείνουμε σε ένα χρήστη να επισκεφτεί ένα εστιατόριο ή να λάβει μία προσφορά σε ένα εστιατόριο. Αξίζει να σημειωθεί ότι αυτή η διαδικασία μπορεί να μην εκτελεσθεί αν δεν λάβουμε κανένα αντικείμενο από τη διαδικασία φιλτραρίσματος των δεδομένων.
Για να πετύχουμε το πιο πάνω στόχο χρειαζόμαστε 2 δεδομένα. Το πρώτο είναι σε ποιο χρήστη θα σταλθεί η ειδοποίηση και το δεύτερο είναι ποιο αντικείμενο θα αφορά. Έχουμε στη διάθεση μας μία βάση δεδομένων στην οποία μπορούμε να καταχωρούμε τι πληροφορίες έχουμε στη διάθεση μας για κάθε αντικείμενο. Αυτές οι πληροφορίες είναι οι αυτές που περιγράφονται στη παράγραφο όπου απεικονίζουμε τη βάση δεδομένων.

Ειδοποίηση που λαμβάνει ο χρήστης στη κινητή του συσκευή:

- Μηνύματα: Τα μηνύματα είναι ειδοποιήσεις που λαμβάνει ένας χρήστης στη κινητή του συσκευή και τον ενημερώνουν για ένα αντικείμενο. Για παράδειγμα ενημερώνουν το χρήστη ότι σήμερα υπάρχει ειδικό μενού για νηστεία σε ένα εστιατόριο.

Παράδειγμα περιεχομένου μίας ειδοποίησης που στέλνεται στη κινητή συσκευή:



Σχήμα 4.4 Παραλαβή ειδοποίησης στη κινητή συσκευή

Ανάκτηση ειδοποίησης πριν την αποστολή της στη κινητή συσκευή:

Στη συνέχεια θα ανακτούμε από τη βάση τις αντίστοιχες πληροφορίες οι οποίες αφορούν το συγκεκριμένο αντικείμενο χρησιμοποιώντας το μοναδικό αριθμό που το χαρακτηρίζει και τέλος θα σταλεί η κατάλληλη ειδοποίηση στη συγκεκριμένη κινητή συσκευή.

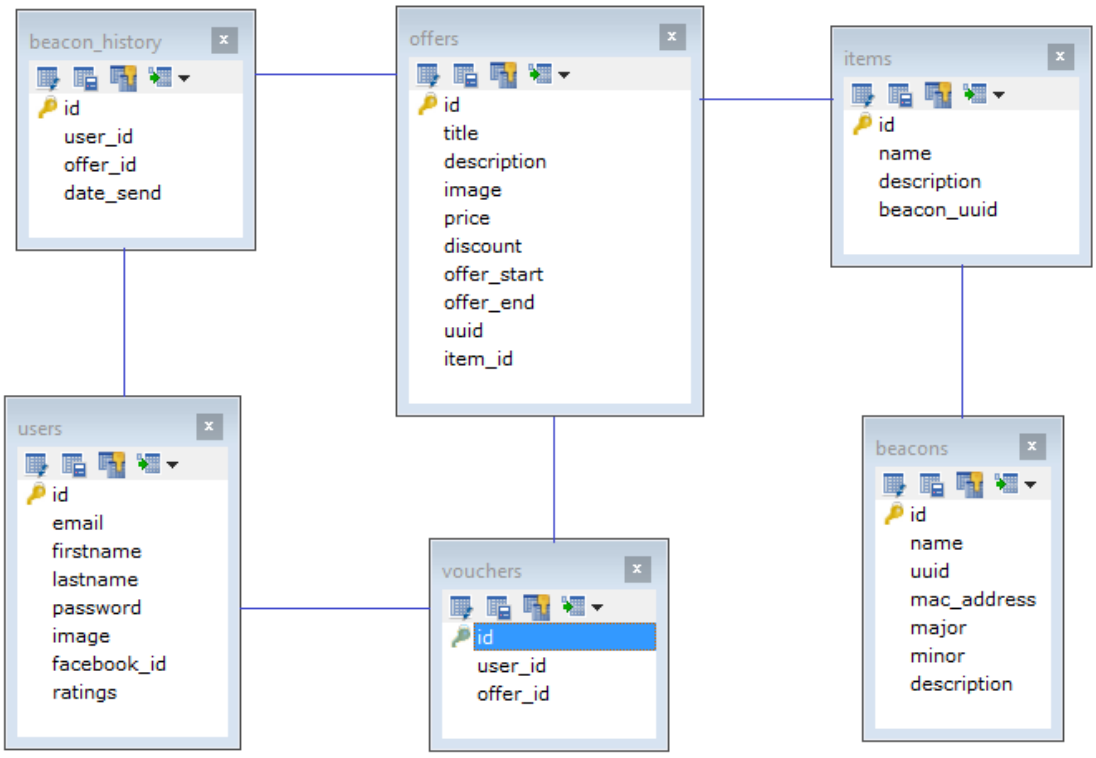
Φάση / Λειτουργία (ε): Διαδικασία αναβάθμισης βάσεων δεδομένων:

Περιπτώσεις αναβάθμισης βάσης δεδομένων:

- Όταν έχουμε εισαγωγή νέου χρήστη
- Όταν έχουμε εισαγωγή νέου αντικειμένου

4.4.4 Βάση δεδομένων

Σχήμα βάσης δεδομένων:



Σχήμα 4.5 Σχήμα βάσης δεδομένων

Εξήγηση πινάκων βάσης δεδομένων:

Πίνακες:

- Users:

  Σε αυτό το πίνακα βρίσκονται καταχωρημένοι οι εγγεγραμμένοι στο σύστημα χρήστες μαζί με τις απαραίτητες πληροφορίες που χρειαζόμαστε για αυτούς όπως φαίνονται πιο πάνω. Αξίζει να σημειωθεί ότι οι κωδικοί πρόσβασης στέλνονται και αποθηκεύονται σε κωδικοποιημένη μορφή.

- Items:

Σε αυτό το πίνακα βρίσκονται καταχωρημένα τα διαθέσιμα αντικείμενα του συστήματος.

- Beacons:

  Αποθηκεύουμε τα διαθέσιμα beacons και τα συσχετίζουμε το κάθε ένα με ένα αντικείμενο το οποίο θα αντιπροσωπεύουν.

- Offers:

  Σε αυτό το πίνακα έχουμε τις ειδοποιήσεις τις οποίες θα λαμβάνουν οι χρήστες στις κινητές τους συσκευές.

- Beacon History

  Σε αυτό το πίνακα αποθηκεύουμε το ιστορικό παραλαβής ειδοποιήσεων που έχει ο κάθε χρήστης. Δηλαδή τι έχει παραλάβει στη κινητή του συσκευή ως ειδοποίηση.

Κεφάλαιο 5

Υλοποίηση αλγορίθμου συστάσεων και Android εφαρμογής

5.1 Υλοποίηση Αλγορίθμου Συστάσεων 49
5.1.1 Εισαγωγή Αλγορίθμου Συστάσεων 49
5.1.2 Περιγραφή υλοποίησης σε φάσεις 50
5.2 Υλοποίηση Android εφαρμογής 53
5.2.1 Εισαγωγή Android εφαρμογής 53
5.2.2 Λειτουργίες Android εφαρμογής 54
5.2.3 Επεξήγηση λειτουργιών της Android εφαρμογής 54
5.2.3.1 Διαδικασία εγγραφής χρήστη και Διαχείριση του Cold Start 54
5.2.3.2 Κεντρικό μενού Android εφαρμογής 56
5.2.3.3 Ρυθμίσεις προφίλ και παραλαβής ειδοποιήσεων 58
5.2.3.4 Διαδικασία παραλαβής ειδοποιήσεων 59

5.1 Υλοποίηση Αλγορίθμου Συστάσεων

5.1.1 Εισαγωγή Αλγορίθμου Συστάσεων

Στόχος Αλγορίθμου Συστάσεων: Ο στόχος του αλγορίθμου είναι να εξάγει ως αποτέλεσμα ποια αντικείμενα είναι πιο πιθανό να ενδιαφέρουν ένα χρήστη, δηλαδή να προβλέψει ορθά.

Περιπτώσεις όπου εκτελείται ο αλγόριθμος: Υπάρχουν 2 περιπτώσεις στις οποίες διαχωρίζουμε την εκτέλεση του αλγορίθμου. Η πρώτη περίπτωση έχει να κάνει με τη πρόβλεψη των αντικειμένων για ένα χρήστη, ενώ η δεύτερη περίπτωση έχει να κάνει με την εισαγωγή ενός νέου χρήστη κατά την οποία ο αλγόριθμος εκτελείται για να εισάγει στο μοντέλο το νέο χρήστη και να εκπαιδεύσει ξανά το μοντέλο ώστε στη συνέχεια να μπορεί να εκτελέσει τη πρώτη περίπτωση όταν του ζητηθεί.

Γλώσσα προγραμματισμού: Έχει αναπτυχθεί σε Java χρησιμοποιώντας τη Map-Reduce προσέγγιση που μας παρέχει το Apache Spark..

Πλατφόρμα ανάπτυξης: Eclipse

5.1.2 Περιγραφή υλοποίησης Αλγορίθμου Συστάσεων σε φάσεις:

Πρώτη φάση:

Τίτλος: Αρχικοποίηση του Spark.
Περιγραφή: Στη πρώτη φάση γίνεται η αρχικοποίηση το Spark όπου θέτουμε το όνομα της εφαρμογής το οποίο μπορεί να είναι οτιδήποτε και ποιος θα είναι ο Master κόμβος.

Δεύτερη φάση:

Τίτλος: Διάβασμα και εισαγωγή των δεδομένων με τη χρήση του Spark και υπολογισμός των μοναδικών χρηστών και αντικειμένων, καθώς και τον αριθμό των τριάδων <χρήστης, αντικείμενο, αριθμητική αξιολόγηση> που έχουμε στη διάθεση μας.
Περιγραφή: Στη δεύτερη φάση παίρνουμε τα δεδομένα που έχουμε στη διάθεση μας τα οποία είναι της εξής μορφής σε τριάδες:

<Αριθμός χρήστη, Αριθμός αντικειμένου, Αριθμητική αξιολόγηση>
Παράδειγμα:
<1, 5, 0.5>
<1, 7, 1>
<2, 7, 0.25>

Στη συνέχεια τοποθετούμε τα δεδομένα αυτά σε ένα java RDD όπου σε κάθε θέση του εισάγουμε αντικείμενο τύπου Rating το οποίο περιλαμβάνει τον αριθμό του χρήστη, τον αριθμό του αντικειμένου και την αντίστοιχη αριθμητική αξιολόγηση. Επίσης σε αυτή τη φάση μπορούμε να φυλάμε σε java RDDs επιπλέον πληροφορίες για τους χρήστες και τα αντικείμενα. Για παράδειγμα για κάθε αντικείμενο μπορούμε να φυλάμε το όνομα του και διάφορες άλλες πληροφορίες όπως είναι η κατηγορία στην οποία ανήκει. Το

ίδιο παράδειγμα ισχύει και για το κάθε χρήστη. Στη συνέχεια υπολογίζουμε τον αριθμό των μοναδικών χρηστών και μοναδικών αντικειμένων που έχουμε στη διάθεση μας και τον αριθμό από τριάδες που αντιπροσωπεύουν τις αριθμητικές αξιολογήσεις.

Τρίτη φάση:

Τίτλος: Διαμοίραση των δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και ελέγχου.
Περιγραφή: Διαμοιράζουμε τα δεδομένα που έχουμε στη διάθεση μας σε τρία σετ όπου το πρώτο αφορά τα δεδομένα τα οποία θα χρησιμοποιήσουμε για να εκπαιδεύσουμε το μοντέλο μας, το δεύτερο αφορά αυτά που θα χρησιμοποιήσουμε για να επαληθεύσουμε το μοντέλο μας και τέλος το τρίτο αφορά αυτά που θα ελέγξουμε το μοντέλο μας κατά πόσο μας δίνει καλά αποτελέσματα. Η διαμοίραση των δεδομένων που γίνεται σε αυτά τα τρία σετ είναι τυχαία.

Τέταρτη φάση:

Τίτλος: Εκπαίδευση μοντέλου
Περιγραφή: Το μοντέλο εκπαιδεύεται στις περιπτώσεις όπου έχουμε αναβάθμιση των δεδομένων μας ή όταν θέλουμε να προτείνουμε ένα αντικείμενο σε ένα χρήστη. Εκπαιδεύουμε το μοντέλο χρησιμοποιώντας τα δεδομένα εκπαίδευσης όπως τα έχουμε διαμοιράσει στη τρίτη φάση.

Πέμπτη φάση:

Τίτλος: Ανάκτηση των προτεινόμενων αντικειμένων για ένα συγκεκριμένο χρήστη χρησιμοποιώντας το εκπαιδευμένο μοντέλο που έχουμε στη διάθεση μας
Περιγραφή: Ανακτούμε τα προτεινόμενα αντικείμενα για ένα χρήστη όπου για κάθε διαθέσιμο αντικείμενο βρίσκουμε την αριθμητική αξιολόγηση που δίνει ως πρόβλεψη το μοντέλο και ακολούθως τα εισάγουμε σε μία σειρά από φίλτρα όπως περιγράφονται στη συνέχεια.

Έκτη φάση:

Τίτλος: Φιλτράρισμα αποτελεσμάτων

Περιγραφή: Έχουμε στη διάθεση μας μία σειρά από φίλτρα από τα οποία περνούν τα αποτελέσματα μας για να μας δοθεί το τελικό αποτέλεσμα το οποίο αποτελείται από ένα μοναδικό αντικείμενο το οποίο θα προτείνουμε στο χρήστη.

Φίλτρα τα οποία μπορούν να χρησιμοποιηθούν: (Σε αυτό το σημείο αξίζει να σημειωθεί ότι τα φίλτρα μπορούν να αυξηθούν σε αριθμό ανάλογα με τις ανάγκες και τα προβλήματα τα οποία πιθανό θα εμφανιστούν κατά τη διάρκεια της λειτουργίας του συστήματος):

1. Πότε ήταν η τελευταία φορά που προτείναμε το συγκεκριμένο αντικείμενο σε αυτό το συγκεκριμένο χρήστη. Έχουμε μία μεταβλητή έστω χ την οποία προσαρμόζουμε ανάλογα και η οποία αντιπροσωπεύει τη περίοδο (σε μέρες) την οποία θέτουμε σαν ελάχιστο όριο. Για παράδειγμα αν αυτή η μεταβλητή χ έχει τιμή ίση με 10 τότε αυτό σημαίνει ότι οποιοδήποτε αντικείμενο το οποίο έχουμε στη διάθεση μας και επίσης το έχουμε προτείνει από 0 μέχρι 10 μέρες πριν, θα το αφαιρέσουμε από τη λίστα των διαθέσιμων αντικειμένων.

2. Πόσα αντικείμενα έχουμε προτείνει σήμερα στο χρήστη. Σε αυτό το κομμάτι μας ενδιαφέρει να μην γινόμαστε ενοχλητικοί στο χρήστη. Δεν θέλουμε ο αλγόριθμος να προτείνει συνέχεια στο χρήστη. Αυτό αφορά τη ποσότητα των αντικειμένων που θα προτείνουμε σε μία μέρα σε ένα χρήστη. Για παράδειγμα αν αυτή η ποσότητα των αντικειμένων είναι ίση με 2 τότε ο μέγιστος αριθμός από ειδοποιήσεις που θα λάβει θα είναι 2.

3. Σκοπός του συγκεκριμένου φίλτρου είναι να απομονώσει το αντικείμενο με το υψηλότερο σκορ όπως έχει ληφθεί από τον αλγόριθμο. Αυτό το φίλτρο είναι αναγκαίο αν και μόνο αν τα αντικείμενα τα οποία έχουν περάσει από τα πιο πάνω φίλτρα είναι περισσότερα από 1. Αν δηλαδή έχει απομείνει μόνο ένα αντικείμενο τότε αυτό σημαίνει ότι θα είναι και το αποτέλεσμα που δίνουν τα φίλτρα. Αν δεν έχει απομείνει κανένα αντικείμενο τότε απλά δεν θα δώσουμε κάτι στην έξοδο και κατά συνέπεια δεν θα προτείνουμε κάτι. Για παράδειγμα αν έχουμε στη διάθεση μας

3 εναπομείναντα αντικείμενα έστω Α, Β και Γ με σκορ 2.5, 1.5 και 4 αντίστοιχα τότε η έξοδος θα είναι το αντικείμενο Γ.

5.2 Υλοποίηση Android εφαρμογής:

5.2.1 Εισαγωγή Android εφαρμογής

Όνομα Android εφαρμογής: Meet Nicosia

Λογότυπο:



Η Android εφαρμογή αποτελεί το γραφικό περιβάλλον με το οποίο οι χρήστες θα αλληλεπιδρούν με το σύστημα και θα λαμβάνουν ειδοποιήσεις στη κινητή του συσκευή οι οποίες αφορούν αντικείμενα στα οποία θα τους προτείνουμε.

Πλατφόρμα ανάπτυξης της εφαρμογής: Android Studio
Πραγματικό σενάριο εγκατάστασης του συστήματος:

Η Android εφαρμογή αποτελεί την προσομοίωση του συστήματος (αλγόριθμος συστάσεων) σε πραγματικές συνθήκες. Με άλλα λόγια, αποτελεί ένα συγκεκριμένο σενάριο εγκατάστασης του συστήματος σε χώρους της Λευκωσίας στη Κύπρο. Το σύστημα, βέβαια, μπορεί να εγκατασταθεί κάτω από οποιοδήποτε σενάριο, όπως για παράδειγμα σε ένα εμπορικό κέντρο.

Στόχος της Android εφαρμογής:

Η Android εφαρμογή έχει ως στόχο να είναι εύκολη, απλή σε χρήση, φιλική προς το χρήστη και χωρίς επιπλέον λεπτομέρειες ή πληροφορίες οι οποίες, πιθανόν, να επιβαρύνουν τη λειτουργία της ή να ενοχλούν το χρήστη.

5.2.2 Λειτουργίες της Android εφαρμογής:

Αλληλοεπικοινωνία Android εφαρμογής με αισθητήρες και αλγόριθμο συστάσεων: Η Android εφαρμογή είναι υπεύθυνη να εντοπίζει τους αισθητήρες και συγκεκριμένα τα Estimote Beacons, να ενημερώνει τη βάση δεδομένων για αυτά που εντοπίστηκαν, να δίνει το σήμα έναρξης της διαδικασίας εκτέλεσης του αλγορίθμου συστάσεων και τέλος να λαμβάνει τα αποτελέσματα από τον αλγόριθμο και να τα παρουσιάζει στη κινητή συσκευή του χρήστη ως ειδοποίηση.

Άλλες λειτουργίες Android εφαρμογής:

- Εγγραφή ενός νέου χρήστη στο σύστημα και διαχείριση cold start προβλήματος.
- Κεντρικό μενού επιλογών.
- Προφίλ ενός χρήστη και διαχείριση του.
- Ρυθμίσεις για ενημέρωση εκκίνησης παραλαβής ειδοποιήσεων.

5.2.3 Επεξήγηση λειτουργιών Android εφαρμογής:

5.2.3.1 Διαδικασίας εγγραφής ενός νέου χρήστη στο σύστημα και διαχείριση Cold start προβλήματος:

Η Android εφαρμογή μας δίνει τη δυνατότητα εγγραφής στο σύστημα είτε με το κλασικό τρόπο είτε μέσω του Facebook. Η εγγραφή μέσω του Facebook επιταχύνει τη διαδικασία εγγραφής καθώς ζητούμε πρόσβαση στα δημόσια στοιχεία ενός χρήστη και συγκεκριμένα στο ονοματεπώνυμο, τη προφίλ φωτογραφία και το ηλεκτρονικό ταχυδρομείο. Με αυτό τον τρόπο συμπληρώνουμε αυτόματα τα πεδία αυτά, ενώ ο χρήστης μπορεί να τα τροποποιήσει αν επιθυμεί.

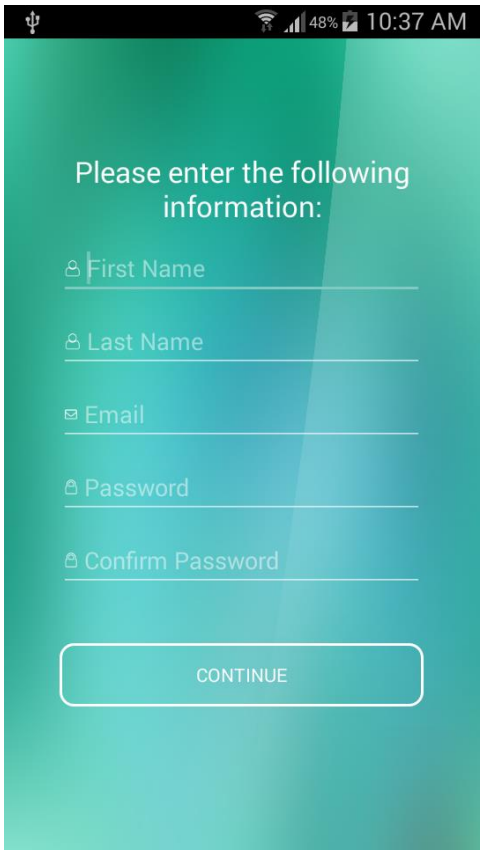
Τα απαιτούμενα στοιχεία που χρειαζόμαστε για την εγγραφή είναι τα εξής:

- Όνομα
- Επίθετο
- Ηλεκτρονικό ταχυδρομείο
- Κωδικός Πρόσβασης

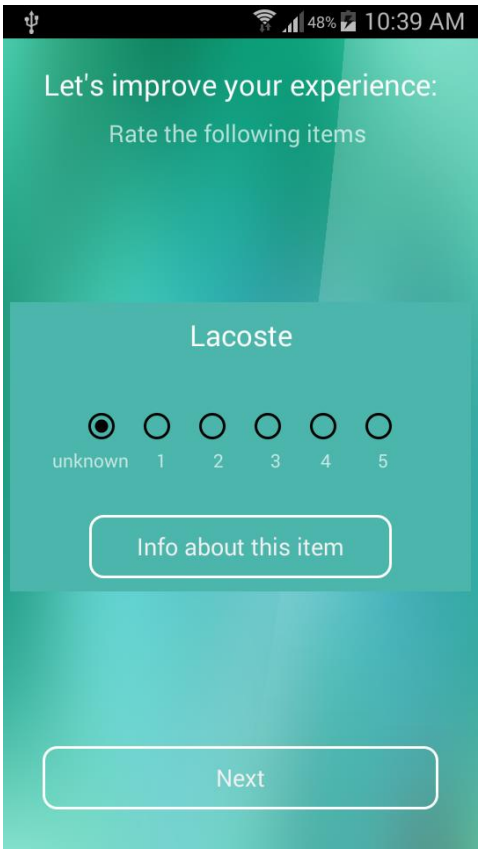
Στη συνέχεια, αφού ο χρήστης επιβεβαιώσει τα στοιχεία του, γίνεται η αλληλοεπικοινωνία με τη βάση δεδομένων για την εισαγωγή των στοιχείων του σε αυτή, ενώ γίνονται και κάποιοι έλεγχοι όπως για παράδειγμα η μοναδικότητα του ηλεκτρονικού ταχυδρομείου και ενημερώνεται κατάλληλα ο χρήστης για τυχόν λάθη στην εισαγωγή των στοιχείων του.

Αν ο χρήστης περάσει με επιτυχία το πιο πάνω βήμα τότε προχωράμε στο επόμενο στάδιο της εγγραφής του όπου είναι η διαχείριση του Cold start προβλήματος. Δηλαδή διαλέγουμε τυχαία x-αριθμό από αντικείμενα από τη βάση δεδομένων και καλούμε το χρήστη τα να βαθμολογήσει (με βάση τη προσωπική του εμπειρία ή οποιοδήποτε άλλο παράγοντα επιθυμεί ο ίδιος) σε κλίμακα 1 μέχρι 5 αστέρια. Επίσης δίνουμε τη δυνατότητα στο χρήστη να μην βαθμολογήσει ένα αντικείμενο αν δεν τον γνωρίζει, ενώ ο χρήστης μπορεί να δει περισσότερες πληροφορίες για αυτό το αντικείμενο όπως δείχνουμε στο σχήμα 5.2.

- Στο σχήμα 5.1 δείχνουμε τη διαδικασία εγγραφής του χρήστης.
- Στο σχήμα 5.2 δείχνουμε τη διαδικασία διαχείρισης του Cold start προβλήματος. Το κουμπί «Info about item» είναι αυτό που μας παρέχει κάποιες περεταίρω πληροφορίες για αυτό για να βοηθήσουμε το χρήστη στις επιλογές βαθμολογιών που θα κάνει.



Σχήμα 5.1



Σχήμα 5.2

Στη συνέχεια, γίνεται καταχώρηση των στοιχείων του χρήστη στη βάση δεδομένων.

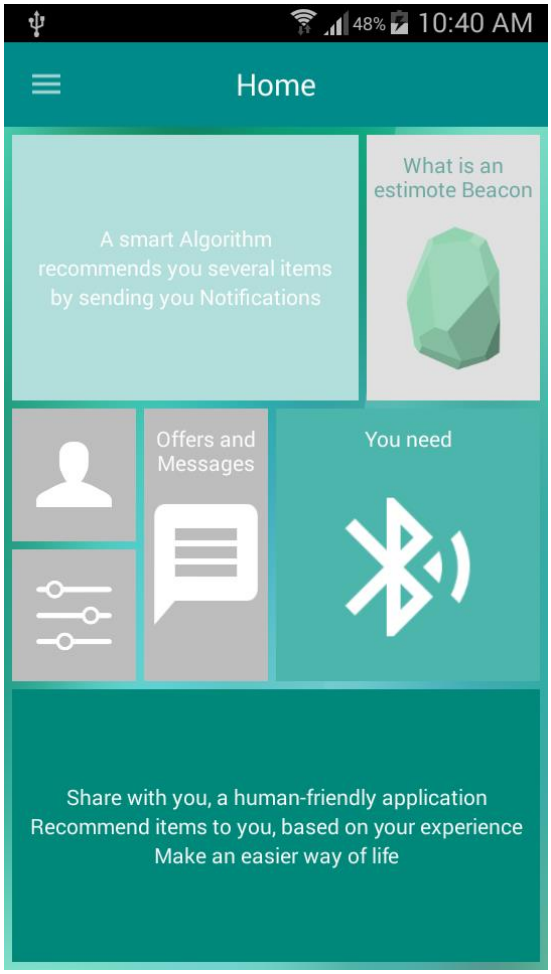
Σημείωση: Η διαδικασία εισόδου του χρήστη στην Android εφαρμογή γίνεται με τη καταχώρηση του ηλεκτρονικού του ταχυδρομείου και του κωδικού πρόσβασης που έχει εισάγει κατά την εγγραφή του.

5.2.3.2 Κεντρικό μενού εφαρμογής:

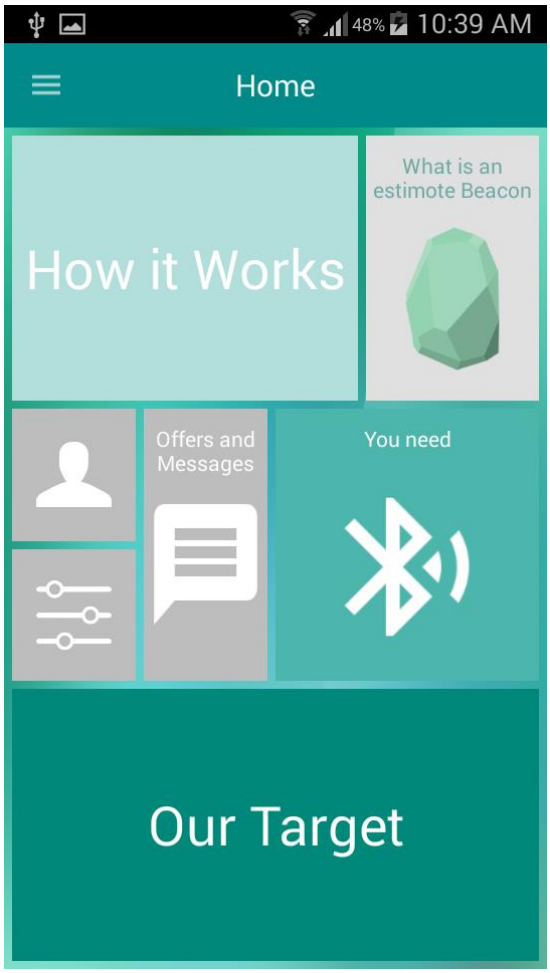
Το κεντρικό μενού αποτελεί την οθόνη που βλέπει ο χρήστης μόλις εγγραφεί ή κάνει είσοδο στην εφαρμογή. Το μενού αυτό δίνει μία πρώτη γεύση στο χρήστη περί τίνος πρόκειται με κάποιες απλές πληροφορίες όπως είναι «Το πώς δουλεύει», «Ποιος είναι στόχος μας», «Τι είναι το Estimote Beacon». Στα αριστερά του κεντρικού μενού υπάρχει και το μενού επιλογών όπως φαίνεται στο σχήμα 5.4.

Επιλογές μενού:

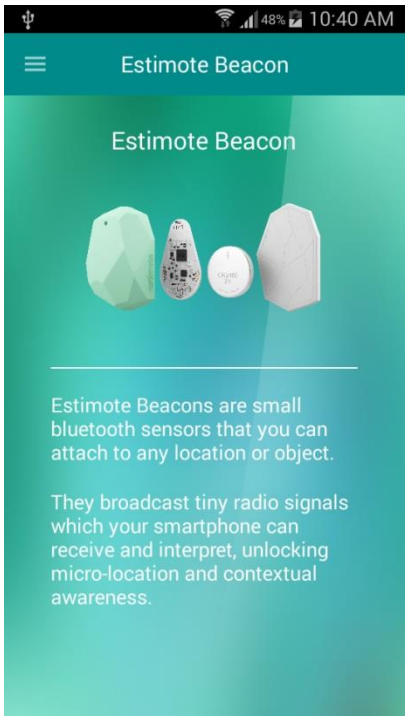
- Προφίλ χρήστη
- Ειδοποιήσεις που έχει λάβει από τον αλγόριθμο συστάσεων
- Ρυθμίσεις
- Αποσύνδεση χρήστη



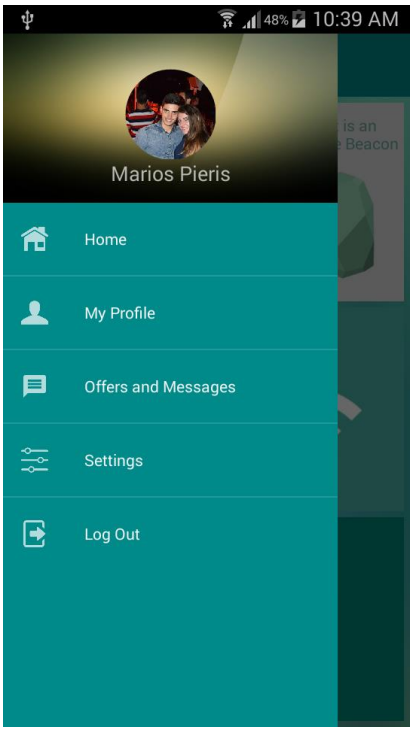
Σχήμα 5.3



Σχήμα 5.4



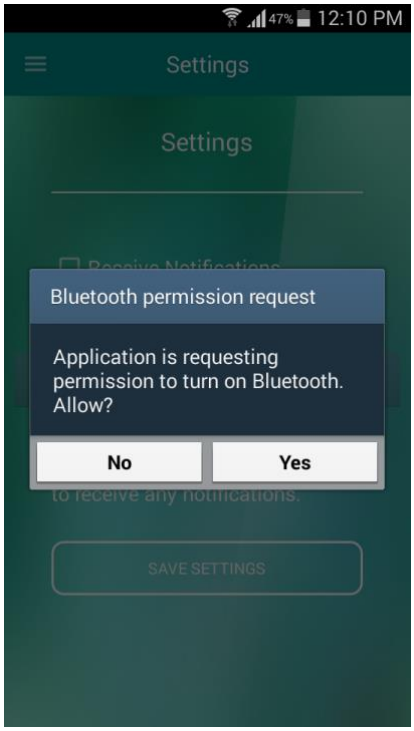
Σχήμα 5.5



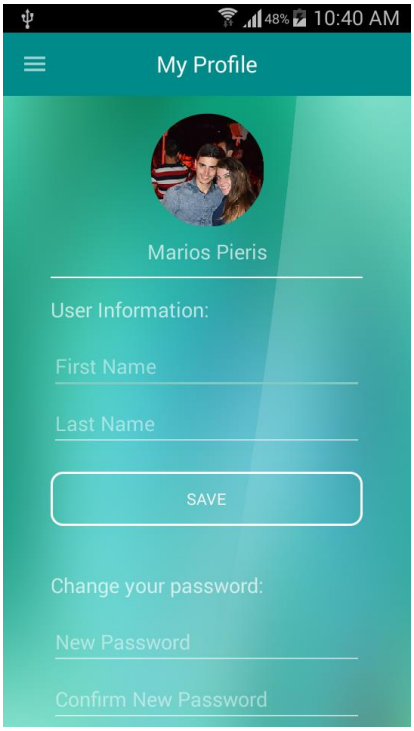
Σχήμα 5.6

5.2.3.3 Ρυθμίσεις προφίλ χρήστη και παραλαβής ειδοποιήσεων:

Ο χρήστης έχει τη δυνατότητα να αλλάξει το όνομα και το επίθετο που έχει καταχωρήσει στην εγγραφή του ή να αλλάξει το κωδικό πρόσβασης του και να εκκινήσει τη διαδικασία παραλαβής ειδοποιήσεων από τον αλγόριθμο συστάσεων.



Σχήμα 5.7



Σχήμα 5.8

5.2.3.4 Διαδικασία παραλαβής ειδοποιήσεων:

Περιεκτικά βήματα διαδικασίας:

1. Ενεργοποίηση ρυθμίσεων παραλαβής ειδοποιήσεων όπου γίνεται ενεργοποίηση του Bluetooth το οποίο εκπέμπει η κινητή συσκευή.
2. Σάρωση της γύρω περιοχής για εντοπισμό των Estimote Beacons
    - Η σάρωση γίνεται κάθε x δευτερόλεπτα
    - Η σάρωση διαρκεί y δευτερόλεπτα
3. Εντοπισμός Estimote Beacons – Αποστολή των μοναδικών αριθμών τους στο σύστημα συστάσεων.
4. Παραλαβή αποτελεσμάτων από το σύστημα συστάσεων αν υπάρχει αποτέλεσμα και εμφάνιση του ως ειδοποίηση στη κινητή συσκευή του χρήστη.

Κεφάλαιο 6

Πειραματική ανάλυση και αποτελέσματα

6.1 Εισαγωγή 61

6.1.1 Πειραματική ανάλυση 61

6.1.2 Πλατφόρμα δοκιμών 61

6.1.3 Μετρικές αξιολόγησης 62

6.1.4 Σύντομες περιγραφές των σετ δεδομένων και λόγοι επιλογής τους 63

6.2 1^{ο} Σετ Δεδομένων 65

6.2.1 Περιγραφή δεδομένων 65

6.2.2 Περιγραφή πειραμάτων 66

6.2.3 1^{ο} Πείραμα 66

6.2.3.1 Περιγραφή 66

6.2.3.2 Αποτελέσματα 67

6.2.4 2^{ο} Πείραμα 68

6.2.4.1 Περιγραφή 68

6.2.4.2 Αποτελέσματα 72

6.2.5 3^{ο} Πείραμα 73

6.2.5.1 Περιγραφή 73

6.2.5.2 Αποτελέσματα 74

6.3 2^{ο} Σετ Δεδομένων 74

6.3.1 Περιγραφή δεδομένων 74

6.3.2 Περιγραφή πειραμάτων 79

6.3.3 2^{ο} Πείραμα 79

6.3.3.1 Περιγραφή 79

6.3.3.2 Αποτελέσματα 80

6.3.4 2^{ο} Πείραμα 80

6.3.4.1 Περιγραφή 81

6.3.4.2 Αποτελέσματα 85

6.3.5 3^{ο} Πείραμα 86

6.3.5.1 Περιγραφή 86
6.3.5.2 Αποτελέσματα 87
6.3.6 4ο Πείραμα 88
6.3.6.1 Περιγραφή 88
6.3.6.2 Αποτελέσματα 89
6.4 3ο Σετ Δεδομένων 90
6.4.1 Περιγραφή δεδομένων 90
6.4.2 Περιγραφή πειραμάτων 91
6.4.3 1ο Πείραμα 92
6.4.3.1 Περιγραφή 92
6.4.3.2 Αποτελέσματα 93
6.4.4 2ο Πείραμα 93
6.4.4.1 Περιγραφή 93
6.4.4.2 Αποτελέσματα 96
6.4.5 3ο Πείραμα 97
6.4.5.1 Περιγραφή 97
6.4.5.2 Αποτελέσματα 99
6.5 Πείραμα Επεκτασιμότητας αλγορίθμου συστάσεων και απόδοσης σε χρόνο 100

6.1 Εισαγωγή

6.1.1 Πειραματική ανάλυση

Αυτό το κεφάλαιο αφορά την επικύρωση της ορθότητας του αλγορίθμου. Με απλά λόγια, χρησιμοποιούμαι κάποια σετ δεδομένων ως η δειγματοληψία μας, η οποία έχει συλλεχθεί σε πραγματικές συνθήκες, με σκοπό να δούμε αν ο αλγόριθμος δουλεύει και προτείνει αντικείμενα σε χρήστες ορθά σε διάφορες περιπτώσεις ανάλογα με τη φύση των δεδομένων που έχουμε στη διάθεση μας. Το κάθε σετ δεδομένων έχει περάσει από προ-επεξεργασία όπως αναλύεται σε κάθε υποκεφάλαιο των πειραμάτων που έχουν διεξαχθεί.

6.1.2 Πλατφόρμα δοκιμών

Η πλατφόρμα που έχει χρησιμοποιηθεί στα πλαίσια της εκτέλεσης των πειραμάτων είναι μία συστοιχία από διακομιστές η οποία βρίσκεται τοποθετημένη στο πανεπιστήμιο Κύπρου. Επίσης, το περιβάλλον ανάπτυξης που έχει χρησιμοποιηθεί είναι το Eclipse, ενώ για την γρήγορη προ-επεξεργασία και ανάλυση των δεδομένων, για το κάθε σετ δεδομένων ξεχωριστά, έχει χρησιμοποιηθεί το Spark.

6.1.3 Μετρικές αξιολόγησης

Οι μετρικές αξιολόγησης οι οποίες έχουν χρησιμοποιηθεί κατά τη διεξαγωγή των πειραμάτων είναι οι εξής:

- Ποσοστό επιτυχίας: Το ποσοστό επιτυχίας εξαρτάται από τον αριθμό των αντικειμένων τα οποία συμπίπτουν με τα αντικείμενα τα οποία πραγματικά ενδιαφέρουν ένα χρήστη και ταυτόχρονα του τα έχει προτείνει ο αλγόριθμος, διαιρώντας τα με τα αντικείμενα τα οποία ενδιαφέρουν ένα χρήστη.

  Ποσοστό επιτυχίας για το χρήστη (x) =

  $$\frac{|\ \textit{αντικείμενα που τον ενδιαφέρουν}\ \cap\ \textit{αντικείμενα που του προτείναμε}\ |}{|\ \text{αντικείμενα που τον ενδιαφέρουν}\ |}$$

- Interest: μετρική από το 0.0 μέχρι το 1.0 με την οποία μετράμε κατά πόσο δείχνει ενδιαφέρον για ένα κατάστημα ή μία τοποθεσία κάποιος χρήστης το οποίο εξαρτάται από το πόσες φορές εντοπίσθηκε σε αυτό το κατάστημα ή την τοποθεσία αντίστοιχα.

  Interest για το χρήστη (x) στο κατάστημα/τοποθεσία (y) =

  $$\frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης x}\ -\ \text{ελάχιστη τιμή})}{(\text{μέση τιμή}\ -\ \text{ελάχιστη τιμή})}$$

  ,αν οι φορές που εντοπίστηκε ένας χρήστης είναι μικρότερη της μέσης τιμής.

- BuySomething: μετρική η οποία παίρνει τιμή 0 ή 1 και αντιπροσωπεύει το αν κάποιος χρήστης έχει αγοράσει κάτι = 1 ή δεν έχει αγοράσει κάτι = 0 από το κατάστημα.

BuySomething(x,y) = 1, αν ο χρήστης x έχει αγοράσει κάτι από το y κατάστημα.
BuySomething(x,y) = 0, αν ο χρήστης x δεν έχει αγοράσει κάτι από το y κατάστημα

- Combine(Interest, Buy Something): η μετρική κατά την οποία συνδυάζουμε τις μετρικές Interest και BuySomething με το πιο κάτω τρόπο.

  $Combine(Interes, BuySomething) = 0.9 * \text{BuySomething} + 0.1 * \text{Interest}$,
  αν BuySomething = 1

  $Combine(Interes, BuySomething) = \text{Interest}$,
  αν BuySomething = 0

Σημείωση: Η επιλογή διαμοίρασης σε 0.9 βαρύτητας στη μετρική Interest και 0.1 βαρύτητας στη μετρική BuySomething έχει γίνει μετά την εκτέλεση στο υποκεφάλαιο 6.3.6.1.

Χρήση των πιο πάνω μετρικών αξιολόγησης στα σετ δεδομένων:

- Το ποσοστό επιτυχίας έχει χρησιμοποιηθεί και στο $1^{ο}$, $2^{ο}$ και $3^{ο}$ σετ δεδομένων.
- Η μετρική Interest στο $2^{ο}$ και $3^{ο}$ σετ δεδομένων αντίστοιχα.
- Η μετρική BuySomething στο $2^{ο}$ σετ δεδομένων.
- Η μετρική Combine(Interest, BuySomething) στο $2^{ο}$ σετ δεδομένων.

Σημείωση: Για καλύτερη κατανόηση της κάθε μετρικής είναι σκόπιμο να διαβάσετε τις περιγραφές του κάθε σετ δεδομένων στα παρακάτω υποκεφάλαια.

6.1.4 Σύντομες περιγραφές των σετ δεδομένων και οι λόγοι που επιλογής τους:

$1^{ο}$ σετ δεδομένων:

Σύντομη περιγραφή: Αυτό το σετ αφορά δεδομένα τα οποία έχουν συλλεχθεί από τη σελίδα Movielens και αφορούν βαθμολογίες που έγιναν από χρήστες σε διάφορες ταινίες.

Λόγοι επιλογής του 1ου σετ:

- Οι λόγοι επιλογής αυτού του σετ δεδομένων αφορούν την πρώτη εκτέλεση του αλγορίθμου σε πραγματικά δεδομένα στα οποία έχουμε βαθμολογίες χρηστών σε ένα είδος προϊόντος (ταινίες) για να παρατηρήσουμε τη συμπεριφορά του αλγορίθμου και να αναλύσουμε τα αποτελέσματα που εξάγαμε.
- Αυτό το σετ δεδομένων αποτελεί τον έναυσμα για το πόσο καλός σε απόδοση είναι ο αλγόριθμος συστάσεων και κατά πόσο θα συνεχίσουμε να ακολουθούμε τον αλγόριθμο αυτό και στα επόμενα σετ δεδομένων. Με άλλα λόγια, το σετ αυτό είναι η βάση της πυραμίδας η οποία θα μας επιτρέψει να ανεβούμε τα επόμενα σκαλιά και να δοκιμάσουμε τον αλγόριθμο συστάσεων σε πιο περίπλοκα δεδομένα τα οποία θα πρέπει να αναλυθούν προτού τα δώσουμε σαν είσοδο.

2ο σετ δεδομένων:

Σύντομη περιγραφή: Τα δεδομένα έχουν συλλεχθεί από το εμπορικό κέντρο στο Leece. Ο τρόπος με τον οποίο έγινε η συλλογή αυτή είναι η καταγραφή των κινητών συσκευών που κρατούν οι επισκέπτες ή οι πελάτες του σε διάφορους χώρους στους οποίους έχουν εντοπιστεί από κάποιες συσκευές που χρησιμοποιούν τεχνολογία Bluetooth και βρίσκονται σε διάφορα σημεία τοποθετημένες στο εμπορικό κέντρο.

Λόγοι επιλογής του 2ου σετ:

- Ο λόγος επιλογής αυτού του σετ δεδομένων αφορά τη προσομοίωση της λειτουργίας του αλγορίθμου συστάσεων σε δεδομένα στα οποία μπορούμε να εγκαταστήσουμε τη λειτουργία του συστήματος σε πραγματικό σενάριο. Δηλαδή, δεν έχουμε, όπως, στο πρώτο σετ δεδομένων τη λογική του <χρήστη, αντικείμενο, βαθμολογία>. Πρέπει εμείς να βρούμε ένα αποδοτικό τρόπο να κανονικοποιήσουμε τα δεδομένα μας σε αυτή τη μορφή και να τα δώσουμε σαν είσοδο στον αλγόριθμο συστάσεων.
- Σκοπός αυτού του σετ δεδομένων είναι η παρατήρηση της απόδοσης του αλγορίθμου σε κλειστούς χώρους όπως είναι, δηλαδή, ένα εμπορικό κέντρο. Επίσης, θέλουμε να δούμε κατά πόσο λαμβάνουμε καλά αποτελέσματα στη περίπτωση όπου έχουμε στη διάθεση μας επιπλέον χαρακτηριστικά και πώς να τα αξιοποιήσουμε σωστά, συνδυάζοντας τα με το καλύτερο δυνατό τρόπο με άλλα χαρακτηριστικά.

Το επιπλέον χαρακτηριστικό που έχουμε σε αυτή τη περίπτωση είναι οι αγορές που έχουν γίνει από κάποιο χρήστη σε ένα κατάστημα.

3^{ο} σετ δεδομένων:

Σύντομη περιγραφή: Τα δεδομένα έχουν συλλεχθεί μέσα από τη καταγραφή των GPS συντεταγμένων κάποιων ανθρώπων στο KAIST της Αμερικής μέσω της κινητής συσκευής τους.

Λόγοι επιλογής του 3^{ου} σετ:

- Ο λόγος επιλογής αυτού του σετ δεδομένων αφορά τη προσομοίωση της λειτουργίας του αλγορίθμου συστάσεων σε δεδομένα στα οποία μπορούμε να εγκαταστήσουμε τη λειτουργία του συστήματος σε πραγματικό σενάριο όπως και στη περίπτωση του 2^{ου} σετ δεδομένων. Σκοπός είναι να βρούμε με ποιο τρόπο θα κανονικοιήσουμε τα δεδομένα καταγραφής των GPS συντεταγμένων και να τα αναπαραστήσουμε ως αντικείμενα τα οποία στη συγκεκριμένη περίπτωση θα αποτελούν τις τοποθεσίες στις οποίες έχει βρεθεί ένας χρήστης.
- Επίσης, όπως και στο δεύτερο σετ δεδομένων, σε αυτή τη περίπτωση θέλουμε να παρατηρήσουμε την απόδοση του αλγορίθμου συστάσεων σε ανοιχτούς χώρους και το πώς θα αντεπεξέλθει στην αναπαράσταση μίας τοποθεσίας ως ένα αντικείμενο το οποίο θα προτείνει σε ένα χρήστη. Ακόμη, μια πρόκληση που έχουμε να αντιμετωπίσουμε εδώ είναι ποιες τοποθεσίες θεωρούνται σημαντικές, ώστε να θεωρηθούν αντικείμενα.

6.2 1^{ο} Σετ δεδομένων

6.2.1 Περιγραφή 1^{ου} Σετ δεδομένων

Έχουμε στη διάθεση μας δεδομένα τα οποία έχουν συλλεχθεί από τη σελίδα Movielens και αφορούν βαθμολογίες που έγιναν από χρήστες σε διάφορες ταινίες. Έχουμε συνολικά 100 χιλιάδες βαθμολογίες με εύρος τιμών από 0 μέχρι 5. Ο αριθμός των χρηστών σε αυτό το σετ ανέρχεται στους 943, ενώ οι ταινίες είναι 1682 και αντιπροσωπεύουν τα αντικείμενα στη περίπτωση αυτού του σετ δεδομένων.
Μορφή δεδομένων: <Χρήστης, Ταινία, Βαθμολογία>

Μετρικές που χρησιμοποιήθηκαν στο 3^{ο} Σετ δεδομένων:

- Ποσοστό επιτυχίας

Μέγεθος του σετ δεδομένων:

- 100 χιλιάδες βαθμολογίες του τύπου: <χρήστης, αντικείμενο, βαθμολογία>

6.2.2 Έχουν γίνει 3 είδη πειραμάτων στο 1^{ο} Σετ δεδομένων

- Το **1^{ο}** πείραμα αφορά τη διαδικασία δίκαιης και ορθής διαμοίρασης των δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και ελέγχου.
- Το **2^{ο}** πείραμα αφορά τη διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank, Lambda και Iterations μέσω των οποίων ο αλγόριθμος χρησιμοποιώντας ALS προσέγγιση μαθαίνει τα δεδομένα με τρόπο του οποίου του επιτρέπει να γενικεύει, έτσι ώστε να δίνει τα κατάλληλα αποτελέσματα που αφορούν τι αντικείμενα (ταινίες στη συγκεκριμένη περίπτωση) θα προτείνουμε και σε ποιους χρήστες.
- Το **3^{ο}** πείραμα αφορά ένα επιπρόσθετο πείραμα που έχει γίνει με σκοπό τη παρατήρηση των αποτελεσμάτων για πολύ μικρές τιμές του Lambda σε αντίθεση με αυτές που ελέγχουμε στο 2^{ο} πείραμα.

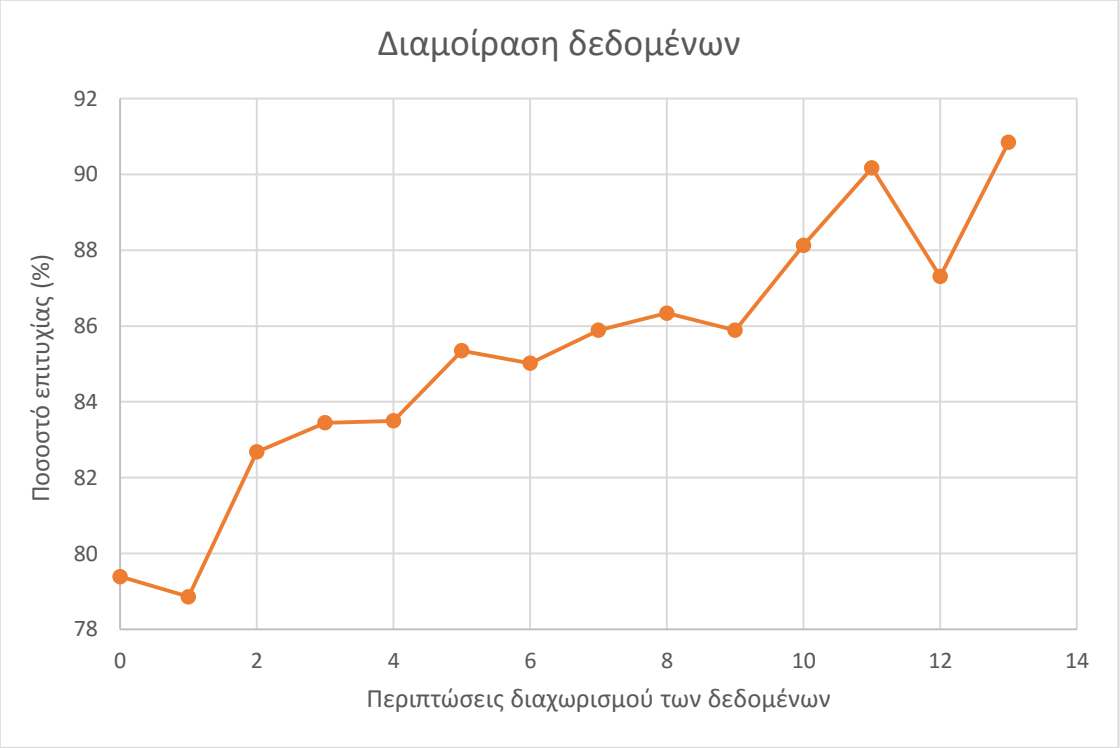
6.2.3 1^{ο} πείραμα: Δίκαιος διαχωρισμός των δεδομένων μας σε σύνολα εκπαίδευσης, επαλήθευσης και ελέγχου

6.2.3.1 Περιγραφή πειράματος: Ο *πίνακας 6.1.1* όπως φαίνεται πιο κάτω, αναγράφει 13 συνδυασμούς διαχωρισμού των δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και ελέγχου, ενώ στην αριστερότερη στήλη αναγράφεται το ποσοστό επιτυχίας το οποίο αναλογεί σε κάθε περίπτωση.

Ποσοστό επιτυχίας (%)	Δεδ. εκπαίδευσης (%)	Δεδ. επαλήθευσης (%)	Δεδ. ελέγχου(%)
79,39	20	20	60
78,86	20	10	70
82,68	30	20	50
83,45	30	10	60
83,50	40	10	50
85,35	40	20	40
85,02	50	20	30

85,89	50	10	40
86,34	60	20	20
85,89	60	10	30
88,13	70	20	10
90,17	70	10	20
87,31	70	15	15
90,85	80	10	10

Πίνακας 6.1.1: Πίνακας καταγραφής περιπτώσεων οι οποίες έχουν εκτελεστεί στα πλαίσια του 1ου πειράματος



Γραφική παράσταση 6.2.1: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.1

6.2.3.2 Συμπεράσματα: Ο διαχωρισμός 80% - 10% -10% μας δίνει το καλύτερο, με πολύ μικρή διαφορά από το δεύτερο 70% - 10% - 20%, ποσοστό επιτυχίας. Αυτό μας οδηγεί στην επιλογή να διαχωρίσουμε τα δεδομένα σε **70% για την εκπαίδευση, 10% για την επαλήθευση και 20% για τον έλεγχο.** Ο λόγος για τον οποίο δεν επιλέγουμε το πρώτο είναι η αρκετά μικρή διαφορά τα οποία χωρίζει αυτά τα δύο, αλλά και επίσης για λόγους ορθότητας και επίδοσης του αλγορίθμου καθώς η εκπαίδευση η οποία αγγίζει το 80% των συνολικών δεδομένων μπορεί να θεωρηθεί υπερβολική.

6.2.4 $2^{ο}$ πείραμα: Η διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank, Lambda και Iterations για την εκπαίδευση και την επικύρωση του αλγορίθμου.

6.2.4.1 Περιγραφή πειράματος: Σε αυτό το πείραμα σκοπός μας είναι να βρούμε ποιος συνδυασμός των τριών παραμέτρων Rank, Lambda και Iterations θα μας δώσει το μεγαλύτερο ποσοστό επιτυχίας.

Σημείωση σχετικά με τους πίνακες:

- Ο άξονας χ αντιπροσωπεύει τη παράμετρο Lambda.
- Ο άξονας ψ αντιπροσωπεύει τη παράμετρο Rank.
- Κάθε κελί του πίνακα αντιστοιχεί στο ποσοστό επιτυχίας που λήφθηκε σε κάθε συνδυασμό Rank και Lambda.

Σημείωση σχετικά με τις γραφικές παραστάσεις:

- Η κάθε θέση στον άξονα χ της γραφικής παραστάσεις αντιστοιχεί σε κάθε θέση του άξονα ψ του πίνακα. Δηλαδή, το 0 στην γραφική παράσταση αντιπροσωπεύει το Lambda = 0.05, το 1 αντιπροσωπεύει το Lambda = 0.1, το 2 αντιπροσωπεύει το Lambda = 0.5 και ούτω κάθε εξής.
- Το χρώμα της γραμμής όπως φαίνεται και στη γραφική παράσταση αντιπροσωπεύει τη τιμή του Rank.
- Ο άξονας ψ της γραφικής παράστασης αντιπροσωπεύει το ποσοστό επιτυχίας.

Σημείωση σχετικά με τη παράμετρο Iterations:

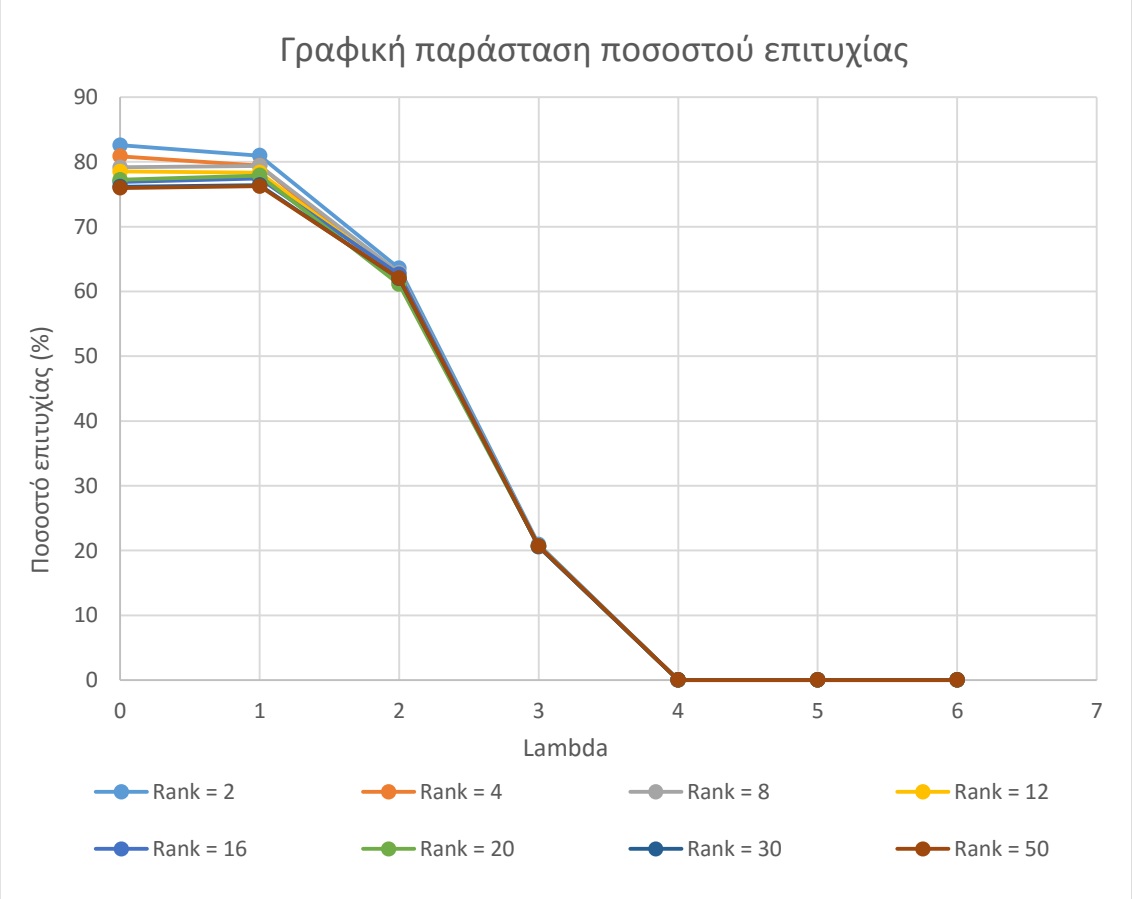
- Ο πίνακας 6.1.2 και η γραφική παράσταση 6.2.2 έχουν ληφθεί με Iterations = 5.
- Ο πίνακας 6.1.3 και η γραφική παράσταση 6.2.3 έχουν ληφθεί με Iterations = 10.
- Ο πίνακας 6.1.4 και η γραφική παράσταση 6.2.4 έχουν ληφθεί με Iterations = 20.

Οι πιο κάτω *πίνακες 6.1.2 , 6.1.3, 6.1.4 και 6.1.5* και αντίστοιχα *οι γραφικές παραστάσεις 6.2.2, 6.2.3, 6.2.4 και 6.2.5* παρουσιάζουν τα αποτελέσματα όπως αυτά έχουν ληφθεί από τα πειράματα.

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 5:

	2	4	8	12	16	20	30	50
0.05	82.59	80.88	79.17	78.53	76.91	77.24	76.17	75.97
0.1	80.96	79.45	79.4	78.36	77.43	77.9	76.42	76.25
0.5	63.58	61.64	62.9	62.31	62.64	61.11	61.98	62.03
1.0	20.92	20.69	20.6	20.59	20.57	20.67	20.62	20.67
5.0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	0
20.0	0	0	0	0	0	0	0	0

Πίνακας 6.1.2: Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 5.

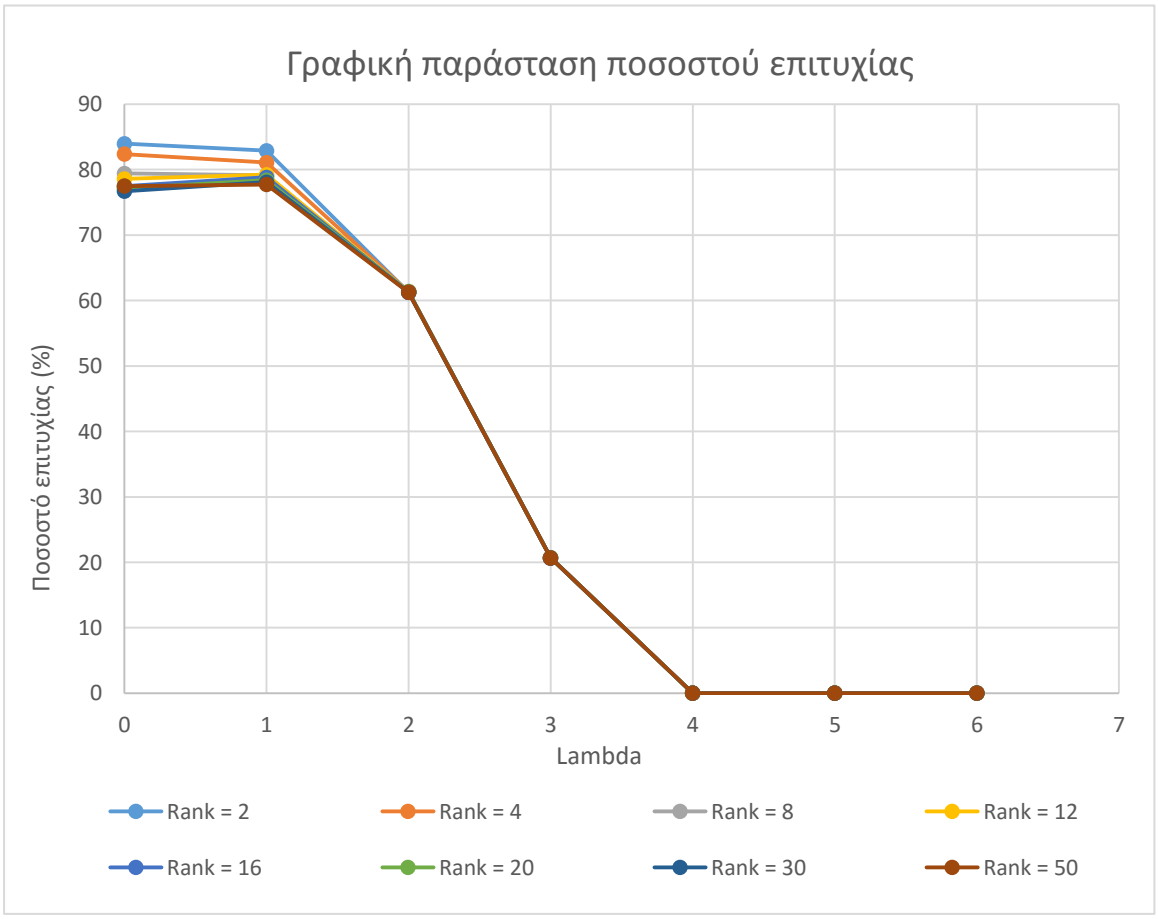


Γραφική παράσταση 6.2.2.: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.2

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 10:

	2	4	8	12	16	20	30	50
0.05	83.97	82.39	79.42	78.6	77.46	77.12	76.72	77.5
0.1	82.89	81.08	79.14	79.24	78.83	78.4	78.1	77.73
0.5	61.25	61.34	61.39	61.35	61.27	61.32	61.28	61.25
1.0	20.68	20.69	20.69	20.69	20.68	20.69	20.68	20.68
5.0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	0
20.0	0	0	0	0	0	0	0	0

Πίνακας 6.1.3: Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 10.

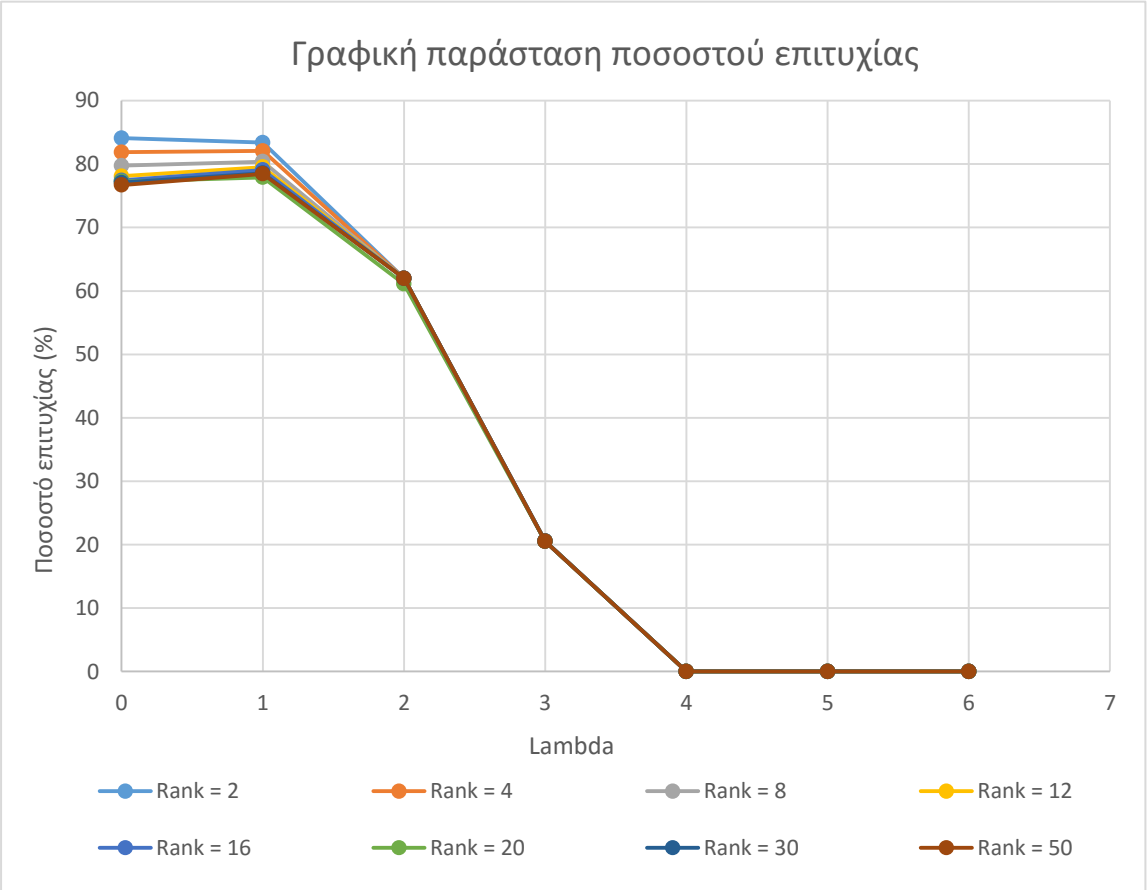


Γραφική παράσταση 6.2.3: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.3

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 20:

	2	4	8	12	16	20	30	50
0.05	84.09	81.87	79.74	78.08	77.42	77.24	77.07	76.7
0.1	83.39	82.08	80.37	79.52	79.07	77.9	78.56	78.46
0.5	61.99	61.99	61.99	61.99	61.99	61.11	61.99	61.99
1.0	20.53	20.53	20.53	20.53	20.57	20.53	20.53	20.53
5.0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	0
20.0	0	0	0	0	0	0	0	0

Πίνακας 6.1.4: Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 20.



Γραφική παράσταση 6.2.4: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.4

6.2.4.2 Αποτελέσματα & παρατηρήσεις 2ου πειράματος

1. Σε όλες τις περιπτώσεις, ανεξαιρέτως της τιμής του *rank*, όταν το *lambda* ξεπεράσει μία τιμή, ένα κατώφλι δηλαδή τότε το *ποσοστό επιτυχίας* πέφτει στη τιμή 0. Σε αυτό το σημείο μπορώ να επισημάνω το γεγονός ότι το **Root mean square error (RMSE)**, το οποίο συγκρίνει τις πραγματικές βαθμολογίες με αυτές που προβλέπει ο αλγόριθμος, παίρνει ψηλή τιμή σε σχέση με το πεδίο τιμών του = [0, 5].
   RMSE: Το εύρος των τιμών που μπορεί να πάρει δεν μπορεί να ξεπερνά το 5 (μέγιστη βαθμολογία) και βέβαια να μην είναι αρνητική. Επομένως, αν αυτό ξεπεράσει για παράδειγμα τη τιμή του 3 τότε αυτό σημαίνει ότι οι βαθμολογίες που προβλέπει ο αλγόριθμος είναι κατά μέσο όρο διαφορετικές κατά 3 αστέρια (αστέρια = βαθμολογία από 1 μέχρι 5 αστέρια). Επομένως το ποσοστό επιτυχίας παίρνει αυτή την φθίνουσα πορεία αφού υπάρχει μεγάλη διαφορά μεταξύ των βαθμολογιών που έχει προβλέψει ο αλγόριθμος σε σχέση με τις πραγματικές.

2. Κρατώντας σταθερή τη τιμή του **lambda**, τότε όσο περισσότερο αυξάνουμε τη τιμή του **rank**, τόσο περισσότερο μειώνεται η τιμή του ποσοστού επιτυχίας που λαμβάνουμε. Αυτή η μείωση στο ποσοστό επιτυχίας όμως, παρατηρείται σε μικρές τιμές του lambda. Δίνοντας δηλαδή στο lambda τιμή κυρίως μεγαλύτερη του 0.1 (>0.1) τότε αυτή η μείωση γίνεται πολύ μικρή μέχρι τελικά να μην επηρεάζει καθόλου τα αποτελέσματα που λαμβάνουμε σε σχέση με άλλες τιμές του rank. Αυτή τη παρατήρηση μπορούμε να τη δούμε επίσης από τις πιο πάνω γραφικές παραστάσεις καθώς οι 8 γραμμές, οι οποίες αντιπροσωπεύουν κάποια τιμή του rank, φαίνεται να γίνονται σαν μία κοινή γραμμή, δηλαδή ακολουθούν την ίδια πορεία.

3. Ο αριθμός των επαναλήψεων, δηλαδή τα **iterations** φαίνεται ότι βοηθά ως ένα βαθμό να εκλάβουμε καλύτερο ποσοστό επιτυχίας κάτι το οποίο μπορούμε να παρατηρήσουμε από τις πιο πάνω γραφικές παραστάσεις 6.2.2, 6.2.3 και 6.2.4. Βέβαια ο αριθμός των επαναλήψεων επιβαρύνει το χρόνο εκπαίδευσης του αλγορίθμου. Ελέγχοντας μεμονωμένο παράδειγμα παρατήρησα ότι το ποσοστό επιτυχίας αυξάνεται όσο αυξάνουμε τα iterations αλλά σε ένα σημείο και μετέπειτα τότε μπορεί να παρατηρηθεί το φαινόμενο της υπέρ-εκπαίδευσης καθώς το ποσοστό επιτυχίας, πλέον, δεν επηρεάζεται, δηλαδή η αλλαγή είναι πολύ μικρή. Η δική μου

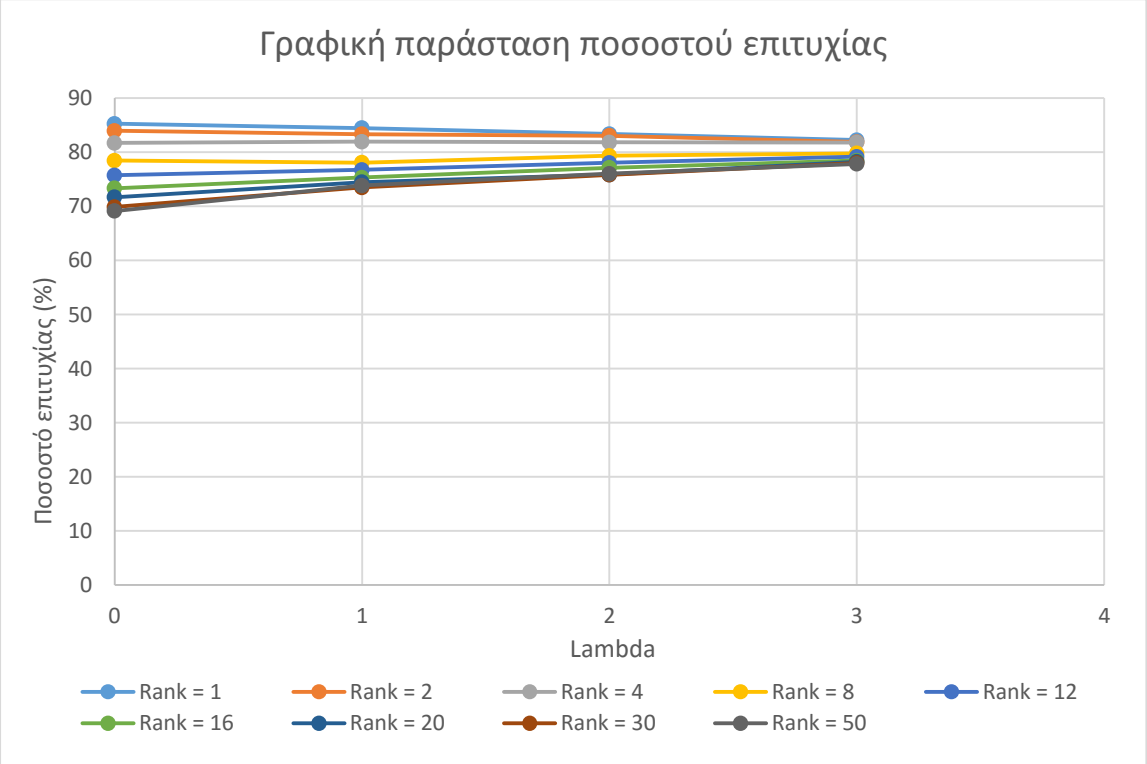
γνώμη λέει ότι ο αριθμός των *20 iterations* είναι μια δίκαια επιλογή σύμφωνα με τα πιο πάνω.

6.2.5 3ο πείραμα: Δοκιμές σε μικρότερες του 0.1 (<= 0.1) τιμές του lambda.

6.2.5.1 Περιγραφή πειράματος: Ο λόγος επιλογής αυτού του πειράματος είναι να παρατηρήσουμε πως το ποσοστό επιτυχίας επηρεάζεται και κυρίως σε ποιο βαθμό σε κάθε περίπτωση όπως φαίνεται στο πιο κάτω πίνακα 6.1.5. Σε αυτό το πείραμα χρησιμοποιήθηκε αριθμός από Iterations = 10.

	2	4	8	12	16	20	30	50
0.01	85.25	83.95	81.72	78.46	75.73	73.34	71.68	69.86
0.025	84.45	83.34	81.95	78.08	76.73	75.32	74.45	73.48
0.05	83.36	83.05	81.81	79.34	78.04	77.13	75.86	75.81
0.1	82.25	81.87	81.79	79.77	79.17	78.44	78.23	78.02

Πίνακας 6.1.5: Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 10



Γραφική παράσταση 6.2.5: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.5

6.2.5.2 Αποτελέσματα & παρατηρήσεις 3ου πειράματος

Όπως μπορούμε να παρατηρήσουμε και στη πιο πάνω γραφική παράσταση οι 3 πρώτες γραμμές οι οποίες αντιπροσωπεύουν τα Rank = 1, Rank = 2 και Rank = 4 ακολουθούν φθίνουσα πορεία ως προς το ποσοστό επιτυχίας καθώς το Lambda αυξάνεται από 0.01 σε 0.1 . Αντιθέτως οι υπόλοιπες γραμμές οι οποίες αντιπροσωπεύουν τα Rank = [8, 12, 16, 20, 30, 50] ακολουθούν ανοδική πορεία. Δηλαδή, για Lambda μικρότερο του 0.1 αρχίζει και δίνει καλύτερα αποτελέσματα αλλά όσο αυξάνουμε το Rank πέφτει.

6.3 2ο Σετ Δεδομένων

6.3.1 Περιγραφή 2ου Σετ δεδομένων

Τα δεδομένα έχουν συλλεχθεί από το εμπορικό κέντρο στο Leece. Ο τρόπος με τον οποίο έγινε η συλλογή αυτή είναι ο εξής:

Χρησιμοποιήθηκαν 25 κινητές συσκευές οι οποίες χρησιμοποιούσαν τεχνολογία Bluetooth για να εντοπίζουν άλλες συσκευές και να τις καταγράφουν. Οι 18 από αυτές τις συσκευές κρατούσαν οι καταστηματάρχες ή οι υπάλληλοι τους και οι υπόλοιπες 7 ήταν στατικές σε κάποια σημεία στα καταστήματα του εμπορικού κέντρου.
Για το πείραμα αυτό χρησιμοποιήθηκαν 11 καταστήματα, 1 μπαρ και ένα 1 store. Άρα σύνολο έχουμε 13 καταστήματα τα οποία αντιπροσωπεύουν τα αντικείμενα μας με άλλα λόγια.

Τους χρήστες αντιπροσωπεύουν οι πελάτες ή οι επισκέπτες του εμπορικού καταστήματος οι οποίοι εντοπίστηκαν από τις 25 αυτές κινητές συσκευές.

Η κάθε κινητή συσκευή έκανε σάρωση γύρω από αυτήν κάθε 120 δευτερόλεπτα (2 λεπτά) και αυτή η σάρωση είχε διάρκεια 14 δευτερολέπτων. Μετά από αυτή τη σάρωση αποθηκεύονται σε Log αρχεία ποιες εξωτερικές συσκευές έχουν εντοπισθεί και αυτή η αποθήκευση γίνεται με τον εξής τρόπο:
<χρόνος, MAC διεύθυνση του σαρωτή, [MAC διεύθυνση της συσκευής που εντοπίστηκε]>

Άρα, τα δεδομένα που έχουμε στη διάθεση μας είναι σε Log αρχεία. Κάθε τέτοιο αρχείο αντιπροσωπεύει τη συλλογή πληροφοριών για κάθε κατάστημα ξεχωριστά. Ενώ, επίσης, υπάρχουν αρχεία τα οποία αντιπροσωπεύουν ποιοι χρήστες έχουν περάσει από τις ταμειακές μηχανές, δηλαδή έχουν αγοράσει κάτι από ένα συγκεκριμένο κατάστημα.

Μετρικές που έχουν χρησιμοποιηθεί στο 2^{ο} Σετ δεδομένων:

- Ποσοστό επιτυχίας
- Interest σε αντικείμενα
- BuySomething
- Combine(Interest, BuySomething)

Μέγεθος του σετ δεδομένων:

- Έχουμε στη διάθεση μας περίπου 35 χιλιάδες γραμμές από δεδομένα του τύπου όπως φαίνονται στη σελίδα 77. Μετά τη κανονικοποίηση και το συνδυασμό που έγινε στα δεδομένα έχουμε περίπου 10 χιλιάδες γραμμές του τύπου <χρήστης, αντικείμενο, Combine(Interest, BuySomething)>

Περιγραφή προβλήματος 2^{ου} Σετ δεδομένων:

Θέλουμε να βρούμε ένα τρόπο ώστε να αξιολογήσουμε το κάθε χρήστη με το κάθε κατάστημα. Δηλαδή θέλουμε να εισάγουμε στα δεδομένα μας, μία μετρική αξιολόγησης η οποία θα μας επιτρέπει στη συνέχεια να εκπαιδεύσουμε τα δεδομένα που έχουμε στη διάθεση μας. Αυτό που θα κάνουμε στο τέλος της μέρας είναι τριάδες οι οποίες έχουν την εξής μορφή:

<χρήστης, κατάστημα, μετρική αξιολόγησης>

π.χ. <23, 11, 0.5> σημαίνει η μετρική αξιολόγησης του χρήστη 23 ως προς το κατάστημα 11 είναι 0.5

Εμείς στη διάθεση μας έχουμε ποιες χρονικές στιγμές εντοπίστηκε ένας χρήστης σε ένα συγκεκριμένο κατάστημα. Επομένως, ο στόχος μας είναι να κανονικοποίησουμε τα δεδομένα αυτά με ένα ορθό τρόπο ο οποίος θα μας εγγυάται σωστά αποτελέσματα και εξίσου καλό ποσοστό επιτυχίας, όπου ποσοστό επιτυχίας εννοούμαι κατά πόσο ο

αλγόριθμος μας προτείνει σωστά. Η κανονικοποίηση αυτή γίνεται με τον τρόπο όπως περιγράφεται πιο κάτω.

Επιλογή κανονικοποίησης των δεδομένων:

1η δοκιμή:

Max-min κανονικοποίηση: Βρίσκουμε ποια ήταν η μέγιστη τιμή των φορών που εντοπίστηκε κάποιος χρήστης και ποια η ελάχιστη σε ένα συγκεκριμένο κατάστημα.

Λόγος αποτυχίας: Αν για παράδειγμα στο αρχείο μας είχαμε κάποια ακραία περίπτωση κάποιου χρήστη ο οποίος εντοπίσθηκε 30 φορές και ο μέσος όρος εντοπισμού των άλλων χρηστών είναι το 5 τότε αυτό σημαίνει ότι οι τιμές, με τις οποίες θα αξιολογούμε τους χρήστης για αυτό το κατάστημα, θα είναι πολύ μικρές γιατί ο τύπος κανονικοποίησης διαιρεί με το (μέγιστο – ελάχιστο). Επομένως όσο πιο μεγάλη είναι η τιμή του μέγιστου, τόσο πιο μικρές τιμές θα παίρνουμε. Συνεπώς, ο αλγόριθμος δεν έδινε καλά αποτελέσματα με αυτή τη τεχνική και αποφάσισα να ακολουθήσω την εξής τεχνική:

2η δοκιμή:

Average-min κανονικοποίηση: Βρίσκουμε ποια ήταν η μέση τιμή των φορών που εντοπίστηκε κάποιος χρήστης και ποια η ελάχιστη.

Interest για το χρήστη(x) στο κατάστημα(y) =

$$\frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης } \chi \ - \text{ελάχιστη τιμή})}{(\text{μέση τιμή} - \text{ ελάχιστη τιμή})}$$

,αν οι φορές που εντοπίστηκε ένας χρήστης είναι μικρότερη της μέσης τιμής.

Παράδειγμα κανονικοποίησης:

Αρχείο:

<χρόνος, MAC διεύθυνση του σαρωτή, [MAC διεύθυνση της συσκευής που εντοπίστηκε]>

<34432, 004e24167398, [004e24165d0e, 004e241660b6, 0062df12c3a4]>

<34566, 004e24167398, [0062df12c3a4, 004e24165d0e, 004e24165d0f, 004e241660b6]>

<34700, 004e24167398, [004e24165d0e, 004e24165d0f, 0062df12c3a4, 004e241660b6]>
<34833, 004e24167398, [004e24165d0e, 004e241660be, 0062df12c3a4, 004e24165d0f]>
<34967, 004e24167398, [004e24165d0e, 0062df12c3a4, 004e241660be, 004e24165d0f]>
<35101, 004e24167398, [0062df12c3a4, 004e24165d0e, 004e32748ge7]>

Τρέχουμε Wordcount λογική στα πιο πάνω δεδομένα και αποθηκεύουμε πόσες φορές εντοπίστηκε ο κάθε χρήστης στο πιο πάνω αρχείο. Τα αποτελέσματα είναι τα εξής με σειρά προτεραιότητας τους χρήστες οι οποίοι εμφανίστηκαν τις περισσότερες φορές.

Αποτελέσματα:

1. 004e24165d0e , 6
2. 0062df12c3a4 , 6
3. 004e24165d0f , 4
4. 004e241660b6 , 3
5. 004e241660be , 2
6. 004e32748ge7 , 1

Ελάχιστη τιμή = οι λιγότερες φορές που εντοπίστηκε κάποια συσκευή = 1

$$\text{Μέση τιμή} = \frac{\text{άθροισμα των φορών που εντοπίστηκαν όλοι οι χρήστες}}{\text{αριθμός των χρηστών}}$$

$$\text{Μέση τιμή} = \frac{6+3+4+6+2+1)}{5}$$

$$\text{Μέση τιμή} = \frac{22}{5}$$

$$\text{Μέση τιμή} = 4.4$$

Αποτελέσματα χρηστών για το πιο πάνω κατάστημα (=αρχείο):

$1^{η}$ περίπτωση: Αν ο χρήστης(x) έχει βρεθεί περισσότερες ή ίσες φορές από τη μέση τιμή τότε αυτό σημαίνει ότι η βαθμολογία που θα πάρει θα είναι το απόλυτο, δηλαδή το

1.0 αφού ξεπερνά τη τιμή του μέσου όρου. Αρά για του εξής χρήστες έχουμε τα εξής αποτελέσματα:

- 004e24165d0e: Interest = 1.0
- 0062df12c3a4: Interest = 1.0

2η περίπτωση: Αν το ο χρήστης(x) έχει βρεθεί λιγότερες φορές από τη μέση τιμή τότε θα κανονικοποίησουμε τη τιμή στο διάστημα τιμών από 0 μέχρι 1 και παίρνουμε τα εξής αποτελέσματα:

- 004e24165d0f:

$$\text{Interest} = \frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης } \chi - \text{ελάχιστη τιμή})}{(\text{μέση τιμή} - \text{ελάχιστη τιμή})}$$

$$= \frac{4-1}{4.4-1}$$

$$= 0.8823$$

- 004e241660b6:

$$\text{Interest} = \frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης } \chi - \text{ελάχιστη τιμή})}{(\text{μέση τιμή} - \text{ελάχιστη τιμή})}$$

$$= \frac{3-1}{4.4-1}$$

$$= 0.5882$$

- 004e241660be:

$$\text{Interest} = \frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης } \chi - \text{ελάχιστη τιμή})}{(\text{μέση τιμή} - \text{ελάχιστη τιμή})}$$

$$= \frac{2-1}{4.4-1}$$

$$= 0.2941$$

- 0062de120fd3:

$$\text{Interest} = \frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης } \chi - \text{ελάχιστη τιμή})}{(\text{μέση τιμή} - \text{ελάχιστη τιμή})}$$

$$= \frac{1-1}{4.4-1}$$

$$= 0.0$$

3η περίπτωση: Αν, επιπλέον, είχαμε επιπλέον πληροφορία ότι ο χρήστης(x) έχει αγοράσει κάτι από το συγκεκριμένο κατάστημα τότε εισάγουμε στις μετρικές μας την έννοια της βαρύτητας στις 2 μετρικές. Στη μετρική Interest δίνουμε βαρύτητα 10% και

στη μετρική BuySomething δίνουμε βαρύτητα 90%. Τα αποτελέσματα που παίρνουμε είναι τα εξής:

- 004e24165d0f :

  Interest + BuySomething = 0.8823 * 0.1 + 1.0 * 0.9 = 0.9882
- 004e241660be :

  Interest + BuySomething = 0.2941 * 0.1 + 1.0 * 0.9 = 0.9294

6.3.2 Έχουν γίνει 3 είδη πειραμάτων στο 2^{ο} Σετ δεδομένων

- Το **1^{ο}** πείραμα αφορά τη διαδικασία δίκαιης και ορθής διαμοίρασης των δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και ελέγχου.
- Το **2^{ο}** πείραμα αφορά τη διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank, Lambda και Iterations μέσω των οποίων ο αλγόριθμος χρησιμοποιώντας ALS προσέγγιση μαθαίνει τα δεδομένα με τρόπο του οποίου του επιτρέπει να γενικεύει, έτσι ώστε να δίνει τα κατάλληλα αποτελέσματα που αφορούν τι αντικείμενα θα προτείνουμε και σε ποιους χρήστες.
- Το **3^{ο}** πείραμα αφορά ένα επιπρόσθετο πείραμα που έχει γίνει με σκοπό τη παρατήρηση των αποτελεσμάτων για πολύ μικρές τιμές του Lambda σε αντίθεση με αυτές που ελέγχουμε στο 2^{ο} πείραμα.
- Το **4^{ο}** πείραμα αφορά την ορθή διαμοίραση των ποσοστών για το ενδιαφέρον που δείχνει ένας χρήστης για ένα κατάστημα (=Interest) και για το αν αγόρασε κάτι από αυτό το κατάστημα (=BuySomething).

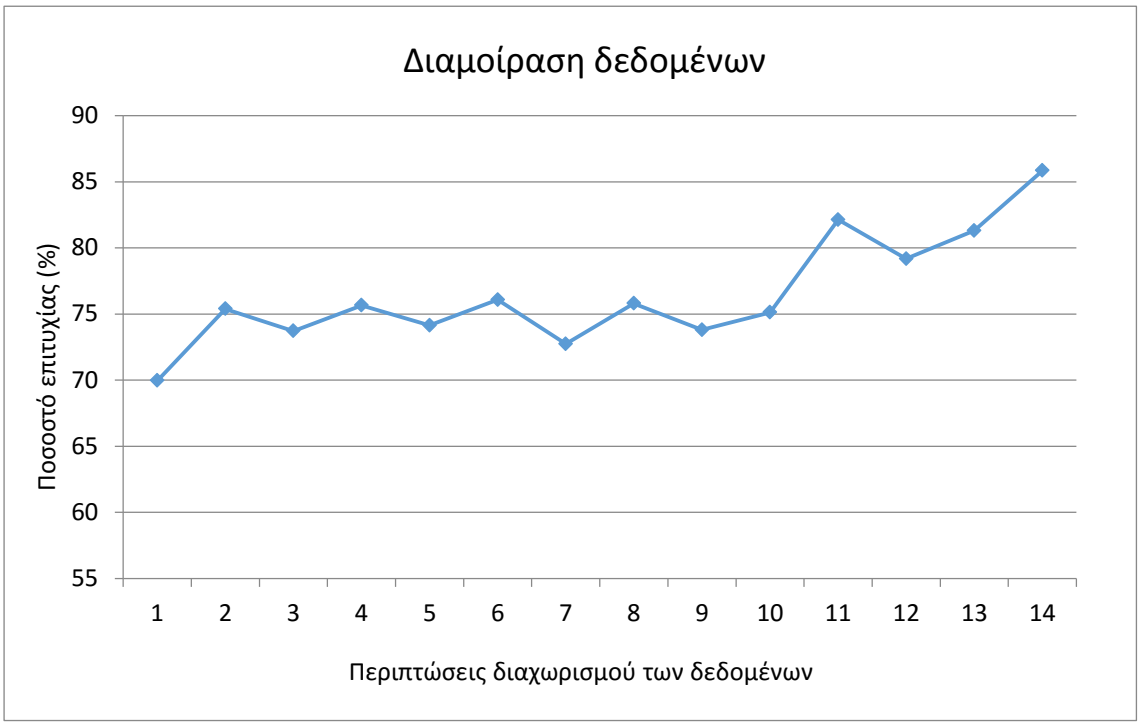
6.3.3 1^{ο} πείραμα: Δίκαιος διαχωρισμός των δεδομένων μας στα 3 σύνολα εκπαίδευσης, επαλήθευσης και ελέγχου.

6.3.3.1 Περιγραφή πειράματος: Ο *πίνακας 6.1.6* όπως φαίνεται πιο κάτω, αναγράφει 14 συνδυασμούς διαχωρισμού των δεδομένων, ενώ στην αριστερότερη στήλη αναγράφεται το ποσοστό επιτυχίας το οποίο αναλογεί σε κάθε περίπτωση.

Ποσοστό επιτυχίας (%)	Δεδ. εκπαίδευσης (%)	Δεδ. επαλήθευσης (%)	Δεδ. ελέγχου(%)
69.54	20	10	70
75.12	20	20	60
73.55	30	10	60
75.45	30	20	50
74.14	40	10	50

76.07	40	20	40
72.73	50	10	40
75.65	50	20	30
73.43	60	10	30
75.23	60	20	20
83.45	70	10	20
79.38	70	15	15
81.41	70	20	10
85.65	80	10	10

Πίνακας 6.1.6: Πίνακας καταγραφής περιπτώσεων οι οποίες έχουν εκτελεστεί στα πλαίσια του 1ου πειράματος



Γραφική παράσταση 6.2.6: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.6

6.3.3.2 Συμπεράσματα: Ο διαχωρισμός 80% - 10% -10% μας δίνει το καλύτερο, με μικρή διαφορά από το δεύτερο 70% - 10% - 20%, ποσοστό επιτυχίας. Παρόλο που η διαφορά είναι μικρή σε σύγκριση, θα επιλέξουμε να διαχωρίσουμε τα δεδομένα σε **80% για την εκπαίδευση, 10% για την επαλήθευση και 10% για τον έλεγχο,** ούτως ώστε να λάβουμε το καλύτερο δυνατό καλά αποτελέσματα στη συνέχεια των πειραμάτων που θα κάνουμε.

6.3.4 2ο πείραμα: Η διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank, Lambda και Iterations για την εκπαίδευση και επικύρωση του αλγορίθμου.

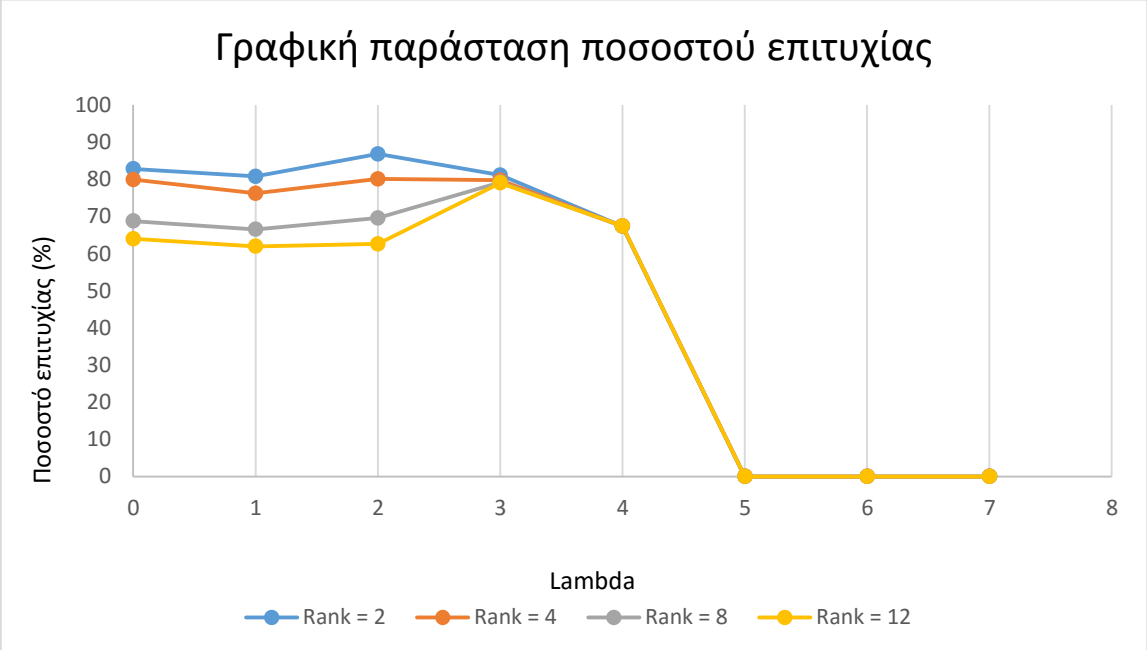
6.3.4.1 Περιγραφή πειράματος: Σε αυτό το πείραμα σκοπός μας είναι να βρούμε ποιο συνδυασμός των τριών παραμέτρων Rank, Lambda και Iterations θα μας δώσει το μεγαλύτερο ποσοστό επιτυχίας.

Σημείωση: Οι σημειώσεις οι οποίες έχουν αναφερθεί στο κομμάτι 6.1.4.1 ισχύουν και σε αυτό το σετ δεδομένων με την ίδια λογική αντίστοιχα. Οι πιο κάτω *πίνακες 6.1.7(α), 6.1.7(β), 6.1.8(α), 6.1.8(β), 6.1.9(α) και 6.1.9(β)* και αντίστοιχα *οι γραφικές παραστάσεις 6.2.7(α), 6.2.7(β), 6.2.8(α), 6.2.8(β), 6.2.9(α) και 6.2.9(β)* παρουσιάζουν τα αποτελέσματα όπως αυτά έχουν ληφθεί από τα πειράματα.

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 10:

	2	4	8	12
0.01	82.82	79.92	68.76	64.01
0.05	80.8	76.21	66.55	61.98
0.1	86.85	80.11	69.58	62.63
0.5	81.12	79.77	79.19	79.05
1.0	67.39	67.39	67.4	67.42
5.0	0	0	0	0
10.0	0	0	0	0
20.0	0	0	0	0

Πίνακας 6.1.7(α): Rank/Lambda πίνακας χρησιμοποιώντας iterations ίσο με 10.

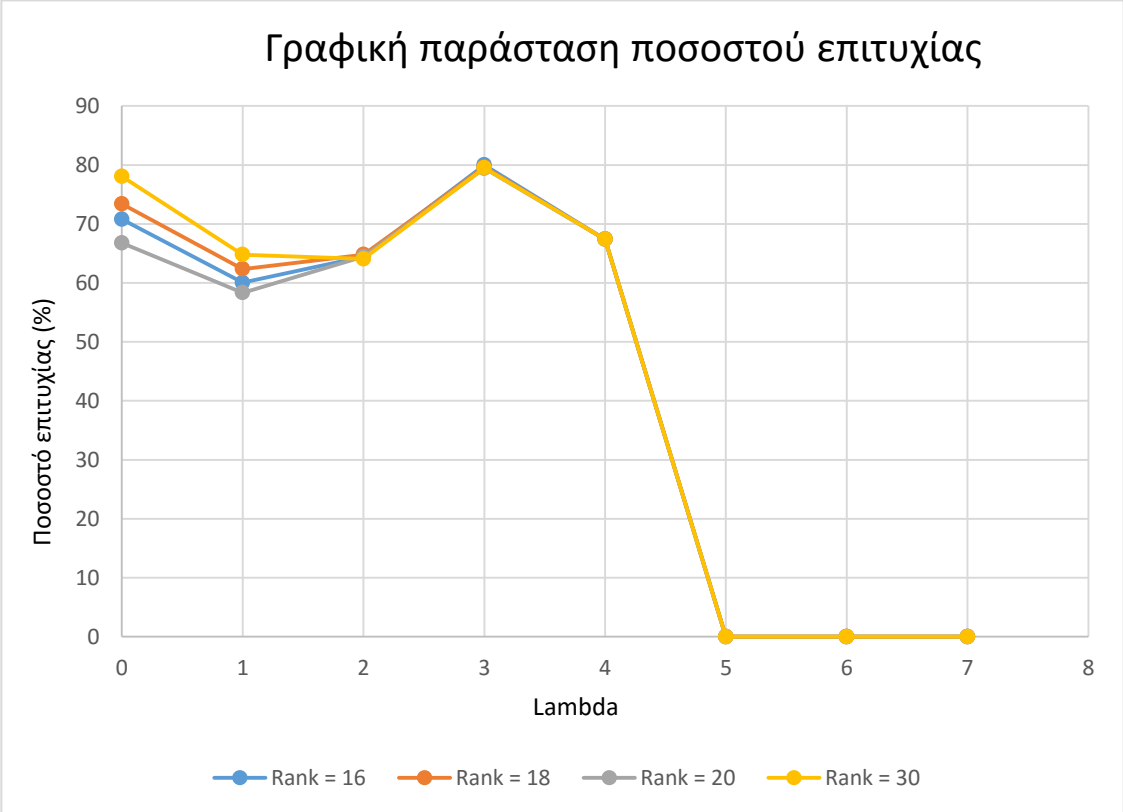


Γραφική παράσταση 6.2.7(α): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.7(α)

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 10:

	16	18	20	30
0.01	70.79	73.39	66.77	78.06
0.05	60.03	62.34	58.34	64.8
0.1	64.43	64.79	64.43	64.06
0.5	79.99	79.64	79.41	79.5
1.0	67.39	67.39	67.4	67.39
5.0	0	0	0	0
10.0	0	0	0	0
20.0	0	0	0	0

Πίνακας 6.1.7(β): Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 10.

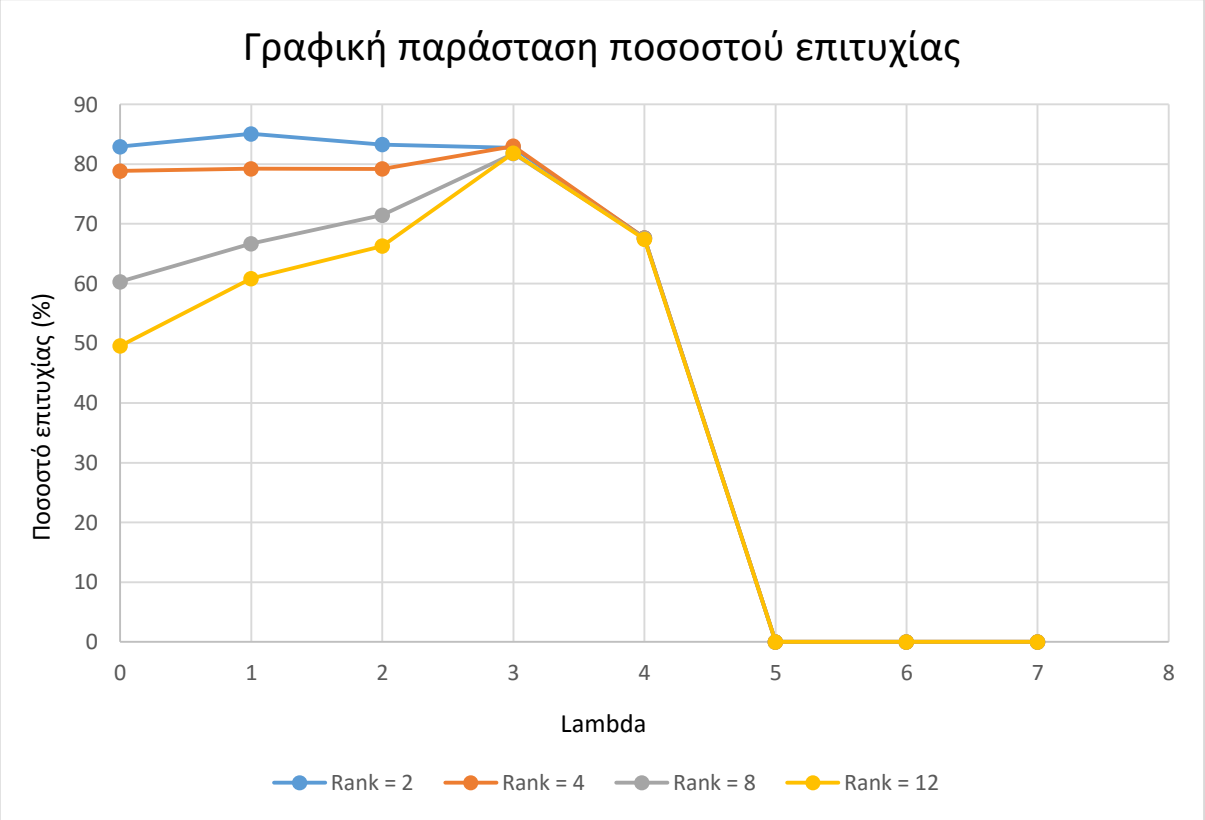


Γραφική παράσταση 6.2.7(β): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.7(β)

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 20:

	2	4	8	12
0.01	82.92	78.85	60.32	49.55
0.05	85.07	79.24	66.7	60.85
0.1	83.28	79.21	71.47	66.29
0.5	82.74	83.01	81.79	81.87
1.0	67.64	67.64	67.61	67.44
5.0	0	0	0	0
10.0	0	0	0	0
20.0	0	0	0	0

Πίνακας 6.1.8(α): Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 20.

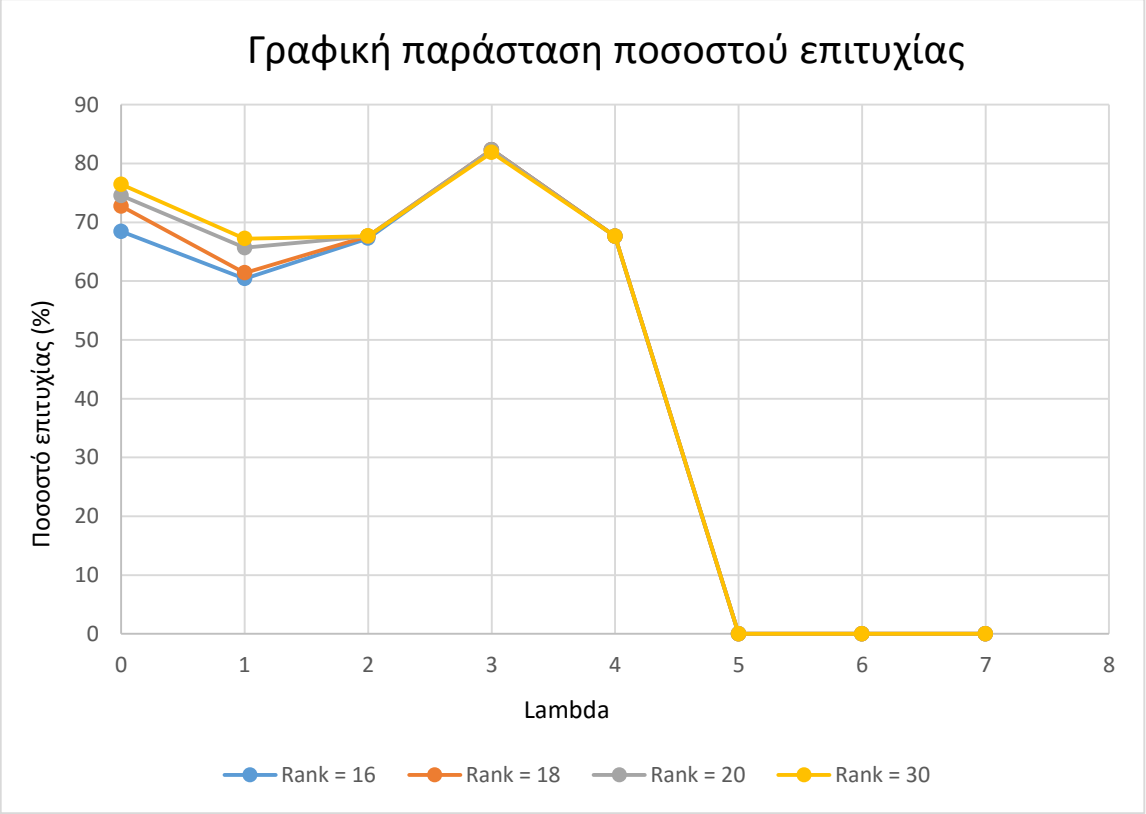


Γραφική παράσταση 6.2.8(α): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.8(α)

Το πιο κάτω πείραμα έχει γίνει με τη χρήση Iterations = 20:

	16	18	20	30
0.01	68.45	72.75	74.55	76.45
0.05	60.42	61.39	65.66	67.22
0.1	67.28	67.64	67.64	67.65
0.5	82.32	82.32	82.33	81.87
1.0	67.63	67.62	67.61	67.64
5.0	0	0	0	0
10.0	0	0	0	0
20.0	0	0	0	0

Πίνακας 6.1.8(β): Rank/Lambda πίνακας χρησιμοποιώντας αριθμό από iterations ίσο με 20.



Γραφική παράσταση 6.2.8(β): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.8(β)

6.3.4.2 Αποτελέσματα & παρατηρήσεις 2^{ου} πειράματος

Τρόπος διαχωρισμού των δεδομένων στις πιο πάνω γραφικές παραστάσεις:

Πήρα τα αποτελέσματα που μου έδωσε το πρόγραμμα και ακολούθως μελέτησα τη συμπεριφορά των γραμμών στις γραφικές παραστάσεις. Με αυτό τον τρόπο παρατήρησα ότι υπάρχουν 2 ομάδες οι οποίες η κάθε μία από αυτές ακολουθούν μία συγκεκριμένη συμπεριφορά. Η πρώτη ομάδα αποτελείται από τα rank που έχουν εύρος από 2 μέχρι 12 και η δεύτερη ομάδα αποτελείται από τα rank τα οποία ανήκουν σε ένα εύρος τιμών από 16 μέχρι 30.

Συμπεριφορά πρώτης ομάδας:

1. Παρατηρούμε ότι για τιμές του lambda οι οποίες είναι μικρότερες του 0.5 ότι οι γραμμές ακολουθούν μία κοινή συμπεριφορά ενώ τελικά παρατηρούμε ότι όλες οι γραμμές, οι οποίες αντιπροσωπεύουν συνδυασμούς rank-lambda, καταλήγουν σε μία κοινή τιμή η οποία αντιπροσωπεύει το κατώφλι των αποτελεσμάτων μας. Η κοινή αυτή συμπεριφορά μπορεί να αποτελεί μία ανοδική πορεία, μία πορεία προς τα κάτω ή ένα συνδυασμό από αυτές σε κάποιες περιπτώσεις. Στη συνέχεια παρατηρούμε ότι όσο αυξάνουμε το lambda > 0.5 τότε η τιμή του ποσοστού επιτυχίας που λαμβάνουμε μειώνεται αισθητά.

2. Κρατώντας σταθερή τη τιμή του lambda, τότε όσο περισσότερο αυξάνουμε τη τιμή του rank, τόσο περισσότερο μειώνεται η τιμή του ποσοστού επιτυχίας που λαμβάνουμε. Αυτή η μείωση στο ποσοστό επιτυχίας όμως, παρατηρείται σε μικρές τιμές του lambda. Δίνοντας δηλαδή στο lambda τιμή κυρίως μεγαλύτερη του 0.5 (>0.5) τότε αυτή η μείωση γίνεται πολύ μικρή μέχρι τελικά να μην επηρεάζει καθόλου τα αποτελέσματα που λαμβάνουμε σε σχέση με άλλες τιμές του rank. Αυτή την παρατήρηση μπορούμε να τη δούμε επίσης από τις πιο πάνω γραφικές παραστάσεις καθώς οι 4 γραμμές, οι οποίες αντιπροσωπεύουν κάποια τιμή του rank, φαίνεται να γίνονται σαν μία κοινή γραμμή, δηλαδή ακολουθούν την ίδια πορεία.

Συμπεριφορά δεύτερης ομάδας:

1. Παρατηρούμε ότι όταν αυξάνουμε το lambda από 0.01 σε 0.05 τότε έχουμε μείωση του ποσοστού επιτυχίας στις γραμμές μας όπως φαίνονται και στη γραφική παράσταση. Ακολούθως όταν το lambda πάρει τιμή ίση με 0.1 και 0.5 αντίστοιχα παρατηρούμε μία

ανοδική πορεία, ενώ επίσης οι γραμμές ακολουθούν μια κοινή συμπεριφορά. Όπως και στη συμπεριφορά της πρώτης ομάδας, έτσι και τώρα όταν το lambda πάρει τιμή > 0.5 τότε η τιμή του ποσοστού επιτυχίας που λαμβάνουμε μειώνεται αισθητά.

2. Όπως και στο δεύτερο σημείο στη περίπτωση της πρώτης ομάδας, τότε το ίδιο ισχύει και εδώ, αλλά ακριβώς αντίθετη λογική. Δηλαδή αντί να παρατηρείται μείωση, παρατηρείται αύξηση. Κρατώντας σταθερή τη τιμή του lambda, τότε όσο περισσότερο αυξάνουμε τη τιμή του rank, τόσο περισσότερο αυξάνεται η τιμή του ποσοστού επιτυχίας που λαμβάνουμε. Αυτή η αύξηση στο ποσοστό επιτυχίας όμως, παρατηρείται σε μικρές τιμές του lambda. Δίνοντας δηλαδή στο lambda τιμή κυρίως μεγαλύτερη του 0.5 (>0.5) τότε αυτή η αύξηση γίνεται πολύ μικρή μέχρι τελικά να μην επηρεάζει καθόλου τα αποτελέσματα που λαμβάνουμε σε σχέση με άλλες τιμές του rank.

Παρατηρήσεις και στις 2 περιπτώσεις των ομάδων:

Ο αριθμός των επαναλήψεων, δηλαδή τα **iterations** φαίνεται ότι βοηθά ως ένα βαθμό να εκλάβουμε καλύτερο ποσοστό επιτυχίας κάτι το οποίο μπορούμε να παρατηρήσουμε από τις πιο πάνω γραφικές παραστάσεις. Βέβαια ο αριθμός των επαναλήψεων επιβαρύνει το χρόνο εκπαίδευσης του αλγορίθμου. Ελέγχοντας μεμονωμένο παράδειγμα παρατήρησα ότι το ποσοστό επιτυχίας αυξάνεται όσο αυξάνουμε τα iterations αλλά σε ένα σημείο και μετέπειτα τότε μπορεί να παρατηρηθεί το φαινόμενο της υπέρ-εκπαίδευσης καθώς το ποσοστό επιτυχίας, πλέον, δεν επηρεάζεται, δηλαδή η αλλαγή είναι πολύ μικρή. Η δική μου γνώμη λέει ότι ο αριθμός των *10 iterations* είναι μια δίκαια επιλογή σύμφωνα με τα πιο πάνω.

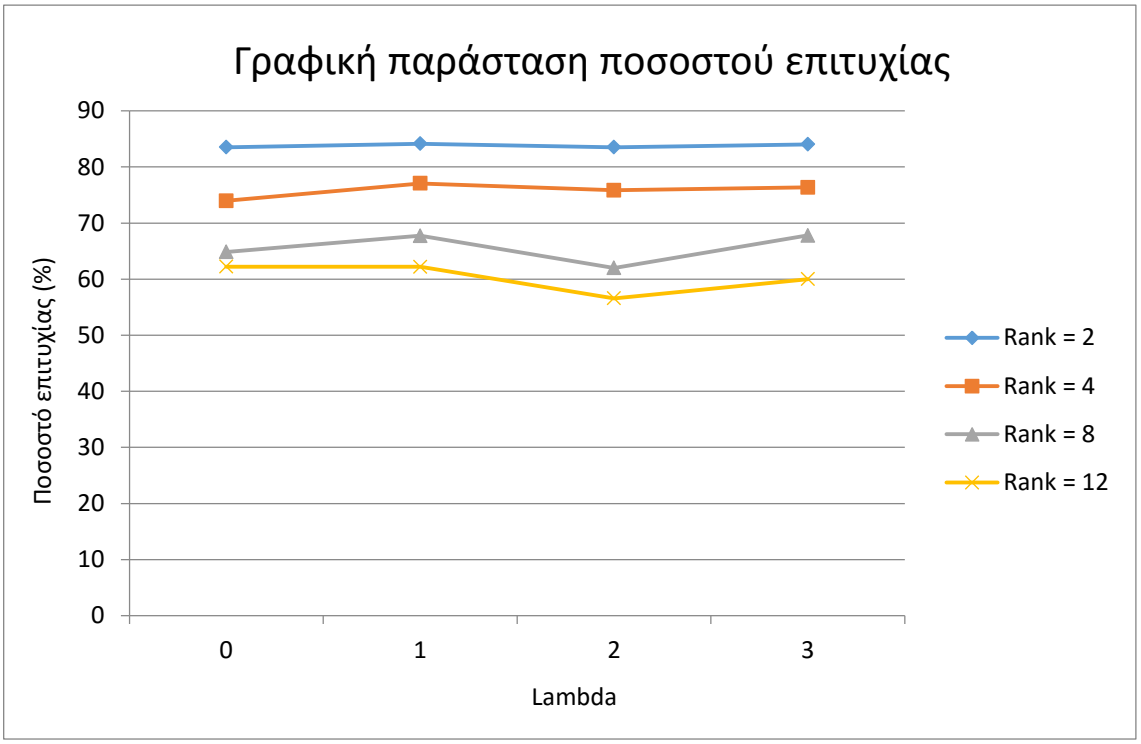
6.3.5 $3^{ο}$ πείραμα: Δοκιμές σε μικρότερες του 0.1 (<= 0.1) τιμές του Lambda.

6.3.5.1 Περιγραφή πειράματος: Ο λόγος επιλογής αυτού του πειράματος είναι να παρατηρήσουμε πως το ποσοστό επιτυχίας επηρεάζεται και κυρίως σε ποιο βαθμό σε κάθε περίπτωση όπως φαίνεται στους πιο κάτω πίνακες 6.1.9(α) και 6.1.9(β).

Μέρος (α) $3^{ου}$ πειράματος:

	2	4	8	12
0.01	83.53	73.96	64.87	62.23
0.025	84.15	77.08	67.74	62.2
0.05	83.53	75.84	62.01	56.57
0.1	84.03	76.35	67.81	60.02

Πίνακας 6.1.9(α): Rank/Lambda πίνακας



Γραφική παράσταση 6.2.9(α): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.9(α)

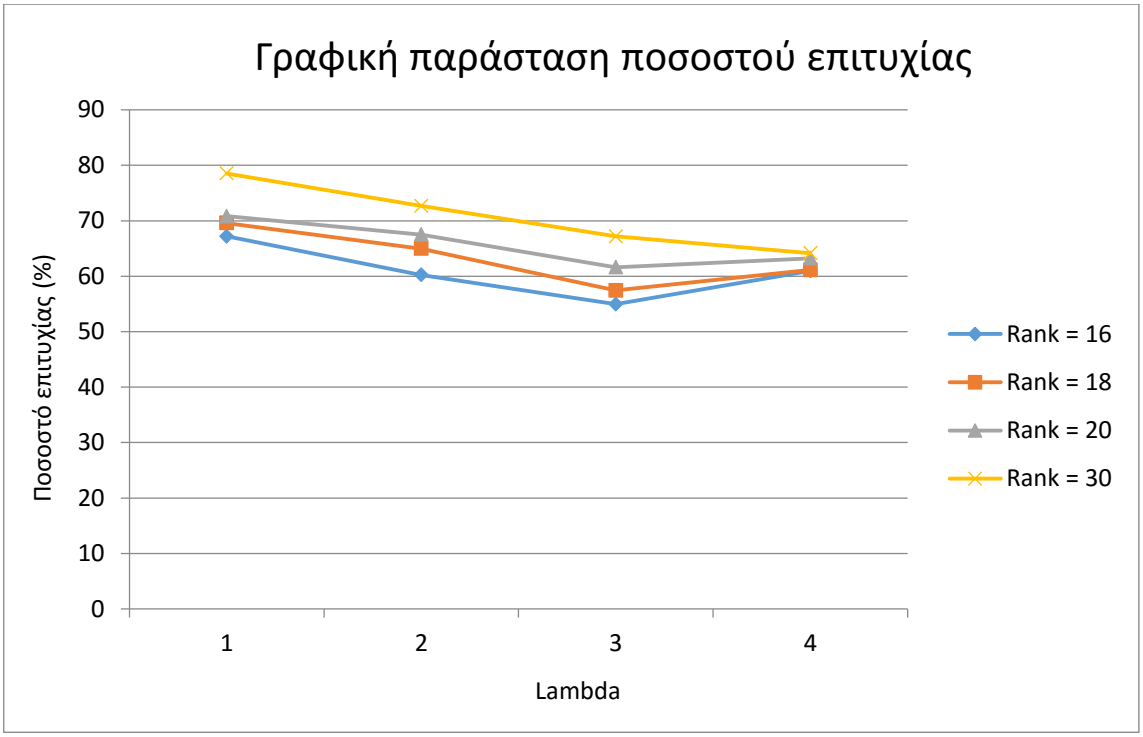
6.3.5.2 (α) Αποτελέσματα & παρατηρήσεις μέρους (α), 3ου πειράματος

Όπως μπορούμε να παρατηρήσουμε και στη πιο πάνω γραφική παράσταση και οι 4 γραμμές οι οποίες αντιπροσωπεύουν τα rank με εύρος τιμών από 2 μέχρι 12 ακολουθούν σε γενικές γραμμές σταθερές πορείες και όπως βλέπουμε το ποσοστό επιτυχίας με lambda = 0.01 είναι περίπου ίσο με το ποσοστό επιτυχίας που λαμβάνουμε στη 4η και τελευταία περίπτωση με lambda = 0.1. Άρα, δεν παρατηρείται κάποια αλλαγή ως προς τη τιμή του ποσοστού επιτυχίας.

Μέρος (β) 3ου πειράματος:

	16	18	20	30
0.01	67.21	69.59	70.83	78.53
0.025	60.23	64.98	67.52	72.69
0.05	54.95	57.45	61.62	67.19
0.1	60.95	61.12	63.25	64.16

Πίνακας 6.1.9(β): Rank/Lambda πίνακας



Γραφική παράσταση 6.2.9(β): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.9(β)

6.3.5.2 (β) Αποτελέσματα & παρατηρήσεις μέρους (β), 3ου πειράματος

Όπως μπορούμε να παρατηρήσουμε και στη πιο πάνω γραφική παράσταση οι 4 γραμμές οι οποίες αντιπροσωπεύουν τα rank εύρους από 16 μέχρι 30 καταλήγουν σε ένα κοινό σημείο στη τελευταία περίπτωση του lambda = 0.1. Η κάθε γραμμή ακολουθά μια συγκεκριμένη πορεία αρά εκείνο που μας ενδιαφέρει τελικά είναι ότι με rank = 0.1 φτάνουμε να έχουμε μικρότερες τιμές ποσοστού επιτυχίας σε σχέση με την αρχική τιμή του lambda = 0.01.

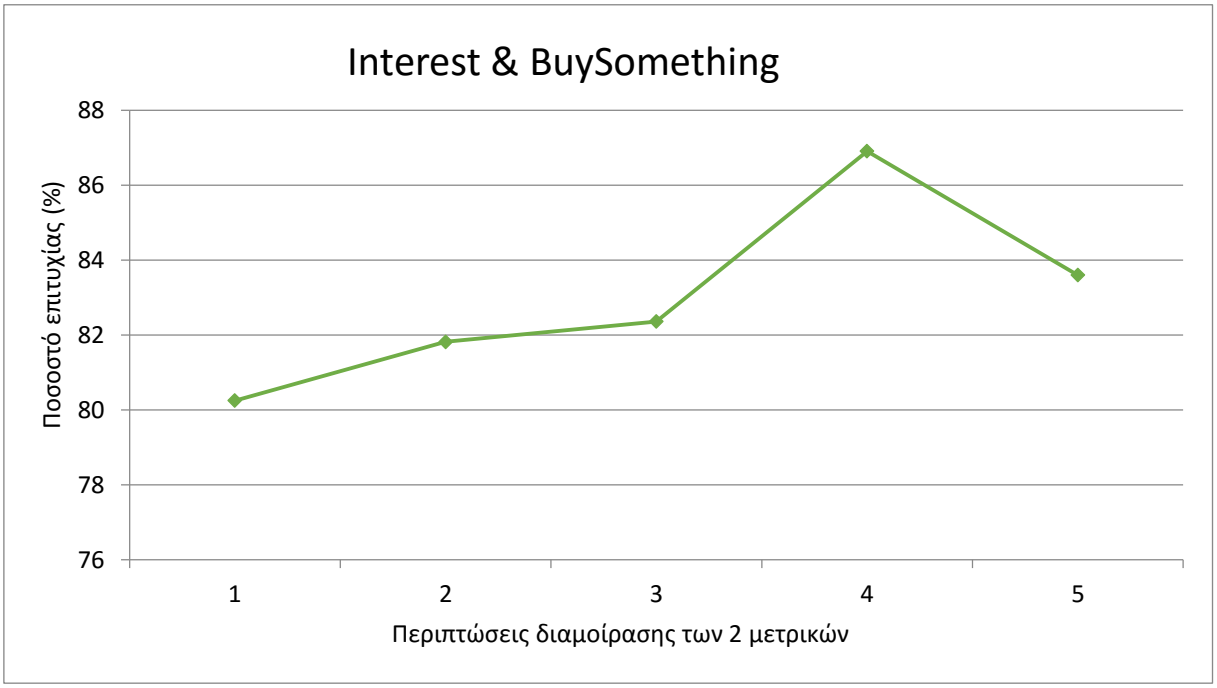
6.3.6 4ο πείραμα: αφορά την ορθή διαμοίραση των ποσοστών για το ενδιαφέρον που δείχνει ένας χρήστης για ένα κατάστημα (=Interest) και για το αν αγόρασε κάτι από αυτό το κατάστημα (=BuySomething).

6.3.6.1 Περιγραφή πειράματος: ορθή διαμοίραση των ποσοστών για το ενδιαφέρον που δείχνει ένας χρήστης για ένα κατάστημα και για το αν αγόρασε κάτι από αυτό το κατάστημα. Δηλαδή έχουμε 2 μετρικές, την Interest και την BuySomething. Θέλουμε να δούμε ποια διαμοίραση μας δίνει το καλύτερο αποτέλεσμα, δηλαδή το καλύτερο

ποσοστό επιτυχίας. Για το πείραμα αυτό χρησιμοποιήσαμε rank = 2 και lambda = 0.1 γιατί μας δίνουν το καλύτερο ποσοστό επιτυχίας για αυτά τα δεδομένα.

Interest – BuySomething (%)	Ποσοστό επιτυχίας (%)
40 – 60	80.25
30 – 70	81.82
20 – 80	82.36
10 – 90	86.91
5 – 95	83.6

Γραφική παράσταση 6.1.10



Γραφική παράσταση 6.2.10

6.3.6.2 Αποτελέσματα 4ου πειράματος

Όπως παρατηρούμε από το πίνακα ή από τη γραφική παράσταση η διαμοίραση η οποία δίνει ποσοστό βαρύτητας 90% στη μετρική BuySomething και 10% στη μετρική Interest μας δίνουν το καλύτερο δυνατό ποσοστό επιτυχίας.

6.4 3^{ο} Σετ δεδομένων

6.4.1 Περιγραφή 3^{ου} Σετ δεδομένων

Έχουμε στη διάθεση μας δεδομένα τα οποία έχουν συλλεχθεί μέσα από τη καταγραφή των GPS συντεταγμένων κάποιων ανθρώπων στο KAIST της Αμερικής. Συγκεκριμένα το πείραμα αφορά 92 χρήστες οι οποίοι έχουν στη κατοχή τους μία κινητή συσκευή η οποία καταγράφει τις συντεταγμένες τους για περισσότερες από 12 ώρες και η καταγραφή αυτή γίνεται κάθε 30 δευτερόλεπτα.

Για κάθε χρήστη έχουμε ένα αρχείο καταγραφής των συντεταγμένων του της εξής μορφής:
<Χρονική στιγμή, <χ - συντεταγμένη, ψ - συντεταγμένη>>

Σε αυτό το σετ δεδομένων δεν υπάρχει η έννοια του αντικειμένου άμεσα. Το αντικείμενο σε αυτή τη περίπτωση θα αντιπροσωπεύει μια τοποθεσία όπου η τοποθεσία αυτή θα αναπαριστάτε από τις χ και ψ συντεταγμένες της. Επίσης ένα αντικείμενο μπορούμε να το αναπαραστήσουμε ως ένα εύρος τιμών γύρω από μία <χ , ψ> συντεταγμένη, δηλαδή < χ ± α, ψ ± α> όπου α το εύρος τιμών.

Επομένως, ο αλγόριθμος θα προτείνει, σε ένα χρήστη, μία τοποθεσία η οποία αναπαριστά ένα αντικείμενο.

Ο στόχος μας είναι να κανονικοποίησουμε τα δεδομένα αυτά με ένα ορθό τρόπο ο οποίος θα μας εγγυάται σωστά αποτελέσματα. Για να το πετύχουμε αυτό θα διατρέξουμε τη λογική του Wordcount στα δεδομένα μας για κάθε χρήστη. Δηλαδή θα βρούμε για κάθε χρήστη πόσες φορές εντοπίστηκε σε μία τοποθεσία όπως πράξαμε και στο 2^{ο} σετ δεδομένων.
Π.χ. Χρήστης = 1, Τοποθεσία = <χ, ψ> συντεταγμένες, Εντοπίστηκε 20 φορές

Στη συνέχεια, θα ακολουθήσουμε την ίδια λογική όπως πιο πάνω αλλά με την εισαγωγή εύρος τιμών στις συντεταγμένες. Ο λόγος που εισάγαμε το εύρος τιμών είναι γιατί είναι πολύ δύσκολο ένας χρήστης να εντοπισθεί πολλές φορές στο ίδιο ακριβώς σημείο. Για παράδειγμα, αν κάποιος βρίσκεται σε ένα κατάστημα μπορεί να εντοπισθεί σε

διαφορετικές συντεταγμένες με ένα εύρος τιμών της τάξης των 100 μέτρων το οποίο εξαρτάται από το πόσο μεγάλο σε έκταση είναι αυτό το κατάστημα.

Μετρικές που χρησιμοποιήθηκαν στο 3^{ο} Σετ δεδομένων:

- Ποσοστό επιτυχίας
- Interest σε τοποθεσίες

Μέγεθος του σετ δεδομένων:

- Έχουμε στη διάθεση μας περίπου 130 χιλιάδες γραμμές από δεδομένα του τύπου <χρήστης, <x-συντεταγμένη, y-συντεταγμένη>. Μετά τη κανονικοποίηση και το συνδυασμό που έγινε στα δεδομένα έχουμε περίπου 3 χιλιάδες του τύπου <χρήστης, τοποθεσία, Interest(τοποθεσία)>

Τρόπος κανονικοποίησης δεδομένων:

Interest για το χρήστη (x) στη τοποθεσία (y) =

$$\frac{(\text{φορές που εντοπίστηκε αυτός ο χρήστης } \chi \ - \text{ελάχιστη τιμή})}{(\text{μέση τιμή} - \text{ελάχιστη τιμή})}$$

,αν οι φορές που εντοπίστηκε ένας χρήστης είναι μικρότερη της μέσης τιμής.

6.4.2 Έχουν γίνει 3 είδη πειραμάτων:

- Το **1^{ο}** πείραμα αφορά τη διαδικασία δίκαιης και ορθής διαμοίρασης των δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και ελέγχου.

- Το **2^{ο}** πείραμα αφορά τη διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank και Lambda μέσω των οποίων ο αλγόριθμος χρησιμοποιώντας ALS (Alternative least square) μαθαίνει τα δεδομένα με τρόπο του οποίου του επιτρέπει να γενικεύει, έτσι ώστε να δίνει τα κατάλληλα αποτελέσματα που αφορούν τι προϊόντα θα προτείνουμε και σε ποιους χρήστες. Σε αυτό το πείραμα θέτουμε το όριο του εύρος τιμών στις συντεταγμένες να είναι το 200.

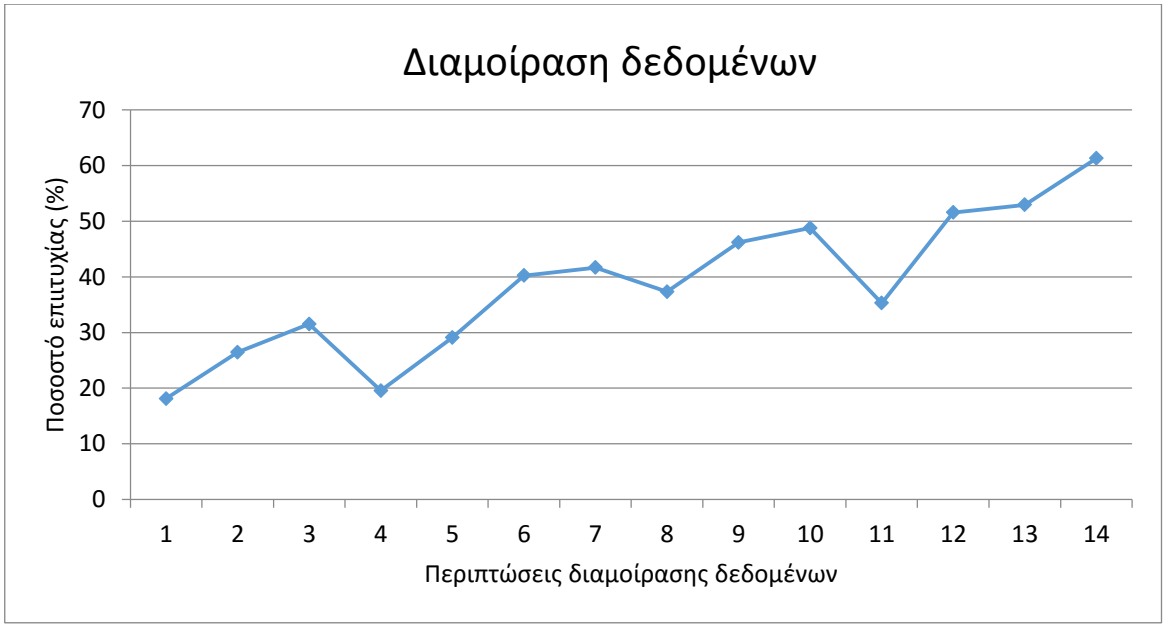
- Το **3^{ο}** πείραμα αφορά ένα επιπρόσθετο πείραμα που έχει γίνει με σκοπό τη παρατήρηση των αποτελεσμάτων δίνοντας μεγαλύτερα εύρη τιμών στις συντεταγμένες τα οποία ξεπερνούν το όριο του 200 που θέσαμε στο 2^{ο} πείραμα.

6.4.3 1ο πείραμα: Δίκαιος διαχωρισμός των δεδομένων μας στα 3 σύνολα εκπαίδευσης, επαλήθευσης και ελέγχου.

6.4.3.1 Περιγραφή πειράματος: Ο *πίνακας 6.1.11* όπως φαίνεται πιο κάτω, αναγράφει 13 συνδυασμούς διαχωρισμού των δεδομένων, ενώ στην αριστερότερη στήλη αναγράφεται το ποσοστό επιτυχίας το οποίο αναλογεί σε κάθε περίπτωση.

Ποσοστό επιτυχίας (%)	Δεδ. εκπαίδευσης (%)	Δεδ. επαλήθευσης (%)	Δεδ. ελέγχου(%)
18.12	20	20	60
26.45	20	10	70
31.51	30	20	50
19.55	30	10	60
29.1	40	10	50
40.22	40	20	40
41.66	50	20	30
37.32	50	10	40
46.19	60	20	20
48.78	60	10	30
35.29	70	20	10
51.55	70	10	20
52.94	70	15	15
61.29	80	10	10

Πίνακας 6.1.11: Πίνακας καταγραφής περιπτώσεων οι οποίες έχουν εκτελεστεί στα πλαίσια του 1ου πειράματος.



Γραφική παράσταση 6.2.11: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.11.

6.4.3.2 Συμπεράσματα: Ο διαχωρισμός 80% - 10% -10% μας δίνει το καλύτερο, με μικρή διαφορά από το δεύτερο 70% - 10% - 20%, ποσοστό επιτυχίας. Παρόλο που η διαφορά είναι μικρή σε σύγκριση, θα επιλέξουμε να διαχωρίσουμε τα δεδομένα σε **80% για την εκπαίδευση, 10% για την επαλήθευση και 10% για τον έλεγχο,** ούτως ώστε να λάβουμε το καλύτερο δυνατό καλά αποτελέσματα στη συνέχεια των πειραμάτων που θα κάνουμε.

6.4.4 2ο πείραμα: Η διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank, Lambda και Iterations για την εκπαίδευση και επικύρωση του αλγορίθμου.

6.4.4.1 Περιγραφή πειράματος: Σε αυτό το πείραμα σκοπός μας είναι να βρούμε ποιο συνδυασμός των τριών παραμέτρων Rank, Lambda και Iterations θα μας δώσει το μεγαλύτερο ποσοστό επιτυχίας. Αυτό το πείραμα το έχουμε χωρίσει σε διάφορες φάσεις για καλύτερη οργάνωση των αποτελεσμάτων που λαμβάνουμε. Σε κάθε φάση του πειράματος παίρνουμε αποτελέσματα δίνοντας στον αλγόριθμο ένα σετ δεδομένων το οποίο έχει προκύψει μέσα από την ανάλυση των δεδομένων που έχουμε στη διάθεση μας και χρησιμοποιώντας εύρος τιμών, έστω α, στις συντεταγμένες.

Σημείωση: Οι σημειώσεις οι οποίες έχουν αναφερθεί στο κομμάτι 6.1.4.1 ισχύουν και σε αυτό το σετ δεδομένων με την ίδια λογική αντίστοιχα.

Οι πιο κάτω *πίνακες 6.1.12, 6.1.13, 6.1.14, 6.1.15, 6.1.16, 6.1.17, 6.1.18, 6.1.19, 6.1.20, 6.1.21, 6.1.22* και *οι γραφικές παραστάσεις 6.2.12, 6.2.13(α) και 6.2.13(β)* παρουσιάζουν τα αποτελέσματα όπως αυτά έχουν ληφθεί από τα πειράματα με αριθμό Iterations = 10.

Πρώτη φάση: Εύρος τιμών από 0 μέχρι 50:

Συμπεράσματα: Το ποσοστό επιτυχίας δεν ξεπερνούσε το 5% στη καλύτερη περίπτωση. Εν κατακλείδι δεν υπάρχει λόγος να παρουσιάσουμε κάποια αποτελέσματα στη πρώτη φάση.

Δεύτερη φάση: Εύρος τιμών από 60 μέχρι 100:

Αυξάνοντας το εύρος τιμών πάνω από 50, παρατηρούμε βελτίωση στα αποτελέσματα τα οποία λαμβάνουμε χωρίς, όμως, να είναι ικανοποιητικά. Σε αυτή τη φάση παραθέτουμε τα εξής αποτελέσματα σε πίνακες.

Εύρος τιμών = α = 70:

	1	2	4	8	12	16
0.05	16.67	19.72	20.37	13.95	13.63	18.89
0.1	18.43	20.56	16.67	13.25	15.89	16.67
0.5	17.39	18.18	16.67	17.39	16.67	16.67
1.0	0	0	0	0	0	0

Πίνακας 6.1.12: Rank/Lambda πίνακας

Εύρος τιμών = α = 100:

	1	2	4	8	12	16
0.05	16.11	22.84	17.67	12.74	7.07	12.36
0.1	16.11	16.67	15.85	12.74	12.36	13.98
0.5	17.85	18.89	16.67	16.67	15.62	16.67
1.0	0	0	0	0	0	0

Πίνακας 6.1.13: Rank/Lambda πίνακας

Συμπεράσματα: Το καλύτερο δυνατό ποσοστό επιτυχίας που πήραμε στις πιο πάνω περιπτώσεις για εύρη τιμών 70 και 100 αντίστοιχα είναι 22.84 % όπου Rank = 2, Lambda = 0.05, Iterations = 10 και α = 100. Επομένως, στη $3^{η}$ φάση θα μελετήσουμε τα αποτελέσματα για εύρος τιμών πάνω από 100 και μέχρι το 200.

Τρίτη φάση: Εύρος τιμών από 110 μέχρι 200:

Σε αυτή τη φάση αυξήσαμε το εύρος τιμών στις συντεταγμένες από 110 μέχρι 200 συμπεριλαμβανομένων. Πιο κάτω παρουσιάζουμε τα αποτελέσματα τα οποία λάβαμε και επιλέξαμε τα εύρη τιμών 120, 140, 160, 180 και 200 ή με άλλα λόγια κάθε φορά αυξάνουμε κατά 20 το εύρος τιμών των συντεταγμένων.

Εύρος τιμών = α = 120:

	1	2	4	8	12	16
0.01	23.80	26.02	24.45	22.22	13.95	18.02
0.05	21.95	23.25	15.47	20.21	22.28	17.26
0.1	18.29	20.5	18.60	18.45	15.85	20.12
0.5	17.5	17.5	17.5	17.5	17.5	17.5

Πίνακας 6.1.14: Rank/Lambda πίνακας

Εύρος τιμών = α = 140:

	1	2	4	8	12	16
0.01	37.38	35.83	28.14	27.77	29.06	19.37
0.05	36.40	32.90	27.17	26.51	26.98	24.60
0.1	35.64	34.25	23.10	28.04	19.51	23.98
0.5	50.0	44.73	38.88	30.0	30.0	30.0

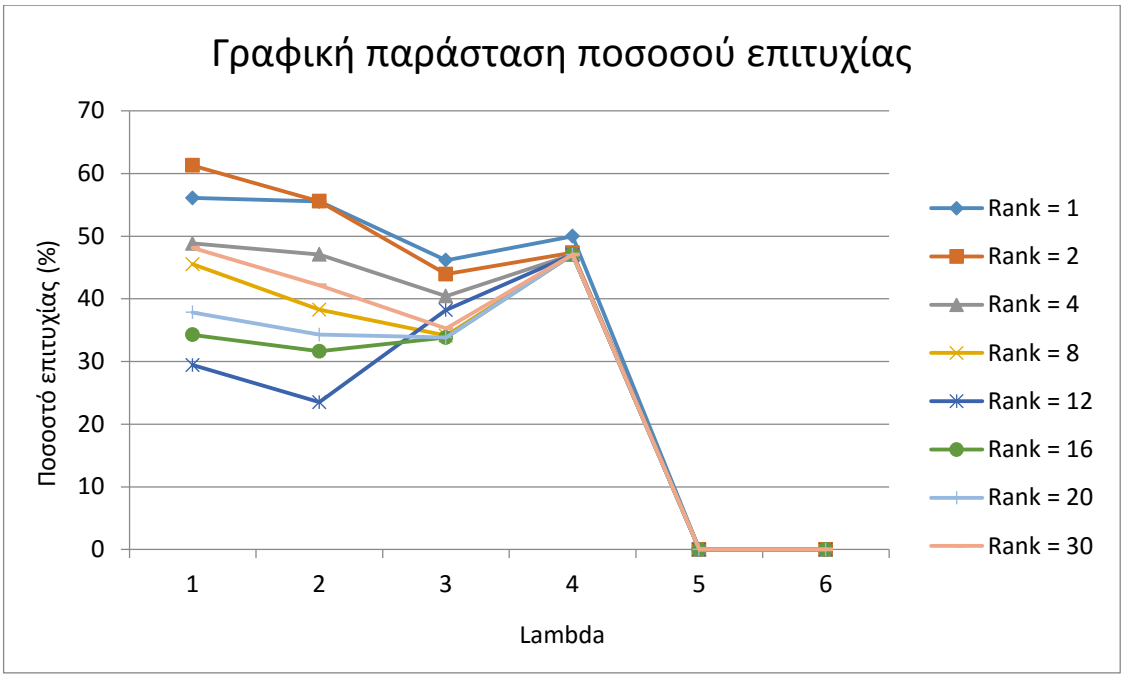
Πίνακας 6.1.15: Rank/Lambda πίνακας

Εύρος τιμών = α = 160:

	1	2	4	8	12	16	20	30
0.01	56.11	61.29	48.84	45.53	29.44	34.25	37.83	48.14
0.05	55.55	55.57	47.08	38.27	23.52	31.62	34.28	42.15
0.1	46.16	43.95	40.44	34.14	38.23	33.82	33.82	35.29
0.5	50	47.36	47.05	47.05	47.05	47.05	47.05	47.05
1.0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	0

Πίνακας 6.1.16: Rank/Lambda πίνακας

Για εύρος τιμών = α = 160, τρέχουμε τον αλγόριθμο με περισσότερες παραμέτρους ως προς το Lambda και το Rank για να παρατηρήσουμε με μεγαλύτερη λεπτομέρεια και ακρίβεια πως συμπεριφέρεται ο αλγόριθμος σε αυτή τη περίπτωση και επίσης παρουσιάζουμε τη πιο κάτω γραφική παράσταση αναπαριστά το πιο πάνω πίνακα για α = 160.



Γραφική παράσταση 6.2.12: Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.16

Εύρος τιμών = α = 180

	1	2	4	8	12	16
0.01	33.34	26.47	30.88	15.71	18.42	16.17
0.05	27.27	24.24	20.58	17.18	8.82	11.76
0.1	30.0	26.47	20.58	22.72	17.18	15.15
0.5	28.57	34.61	21.42	21.42	21.42	21.42

Πίνακας 6.1.17: Rank/Lambda πίνακας

Εύρος τιμών = α = 200

	1	2	4	8	12	16
0.01	28.0	18.96	23.43	15.63	15.15	19.35
0.05	25.92	27.58	25.75	16.12	9.67	13.33
0.1	29.16	21.42	8.06	11.67	13.33	10.0
0.5	16.67	28.57	23.52	17.64	18.75	23.52

Πίνακας 6.1.18: Rank/Lambda πίνακας

6.4.4.2 Συμπεράσματα: Αυξάνουμε το εύρος τιμών και παρατηρούμε ότι λαμβάνουμε καλύτερα αποτελέσματα ως προς το ποσοστό επιτυχίας. Όπως παρατηρούμε και στους

πιο πάνω πίνακες όταν το εύρος τιμών που δίνουμε είναι ίσο με α = 160 τότε παίρνουμε τα καλύτερα ποσοστά επιτυχίας σε σχέση με τις άλλες περιπτώσεις που του α. Επίσης, όταν το εύρος τιμών ξεπεράσει το 160 τότε έχουμε λιγότερο καλά αποτελέσματα.

6.4.5 3ο πείραμα: Σε αυτό το πείραμα εκτελούμε την ίδια διαδικασία κατά την οποία γίνεται η ορθή επιλογή των παραμέτρων Rank και Lambda για την εκπαίδευση και επικύρωση του αλγορίθμου όπως και στη περίπτωση του 2ου πειράματος, αλλά για τιμές μεγαλύτερες του 200 σε εύρος τιμών.

6.4.5.1 Περιγραφή πειράματος: Σε αυτό το πείραμα σκοπός μας είναι να βρούμε ποιο συνδυασμός των τριών παραμέτρων Rank και Lambda θα μας δώσει το μεγαλύτερο ποσοστό επιτυχίας για εύρος τιμών μεγαλύτερου του 200 και συγκεκριμένα έχουμε λάβει αποτελέσματα για α = 250, α = 300 και α = 350.

Εύρος τιμών = α = 250

	1	2	4	8	12	16
0.01	25.0	18.96	13.33	10.41	15.38	8.33
0.05	25.0	25.0	14.58	4.54	4.23	11.36
0.1	21.42	21.05	16.67	9.53	4.76	4.76
0.5	31.81	40.0	20.83	15.38	18.18	18.18

Πίνακας 6.1.19: Rank/Lambda πίνακας

Εύρος τιμών = α = 300

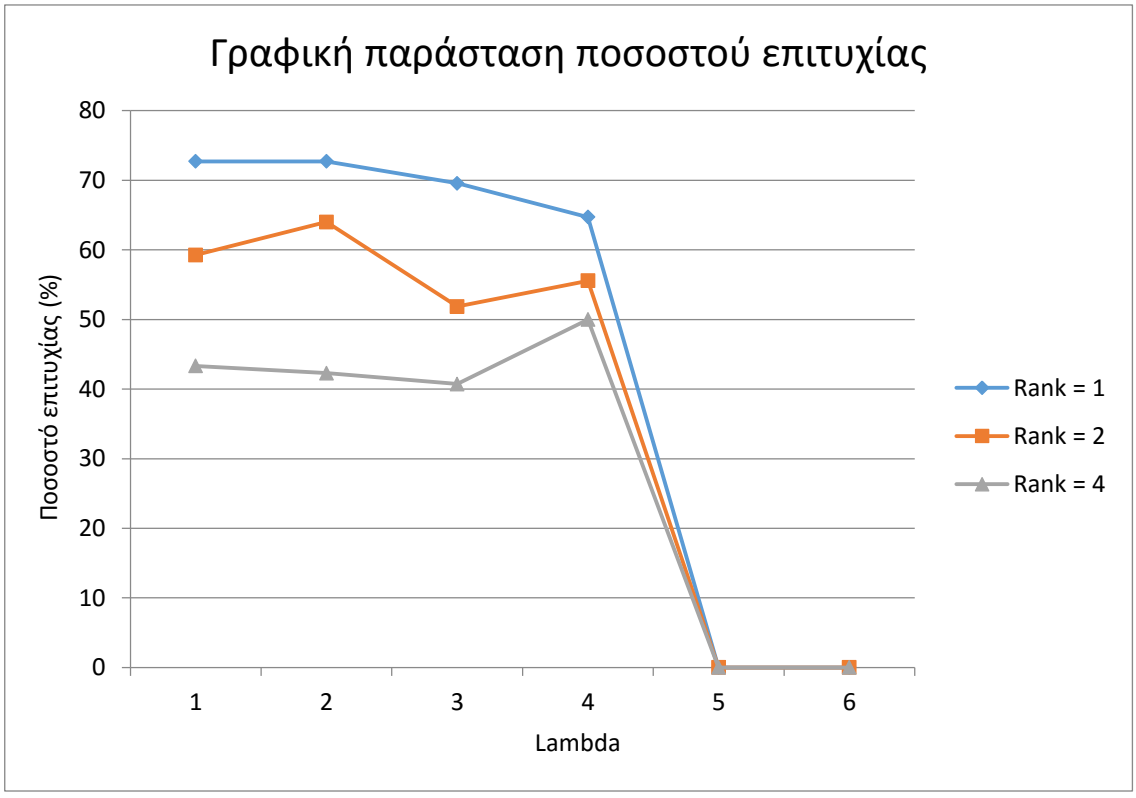
	1	2	4	8	12	16	20	30
0.01	72.72	59.25	43.33	40.21	35.71	39.92	46.42	51.29
0.05	72.72	64	42.3	33.07	31.32	35.46	36.4	37.03
0.1	69.56	51.85	40.74	21.73	17.39	17.39	17.39	21.73
0.5	64.7	55.55	50	47.05	47.05	50	50	50
1.0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	0

Πίνακας 6.1.20: Rank/Lambda πίνακας

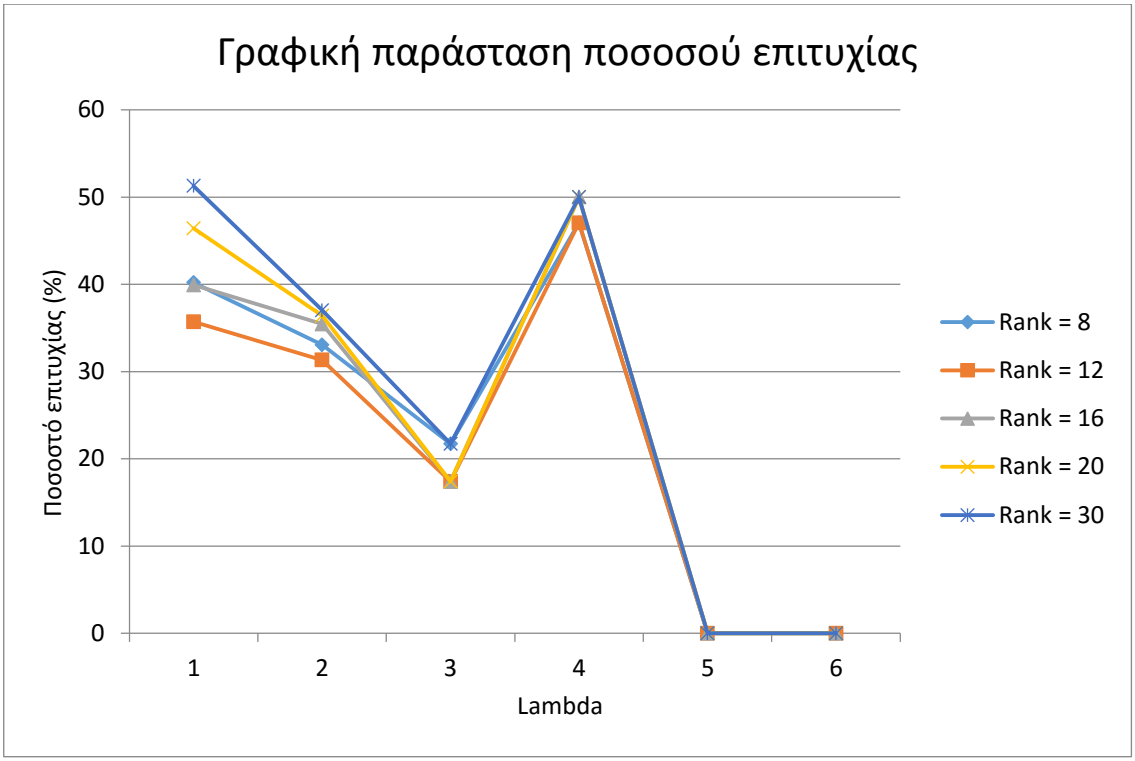
Για εύρος τιμών = α = 300, τρέχουμε τον αλγόριθμο με περισσότερες παραμέτρους ως προς το lambda και το rank για να παρατηρήσουμε με μεγαλύτερη λεπτομέρεια και ακρίβεια πως συμπεριφέρεται ο αλγόριθμος σε αυτή τη περίπτωση και επίσης τα παρουσιάζουμε σε μορφή 2 γραφικών παραστάσεων όπως φαίνονται πιο κάτω.

Τρόπος διαχωρισμού των αποτελεσμάτων για α = 300 στις πιο κάτω γραφικές παραστάσεις:

Πήρα τα αποτελέσματα που μου έδωσε το πρόγραμμα και ακολούθως μελέτησα τη συμπεριφορά των γραμμών στις γραφικές παραστάσεις. Με αυτό τον τρόπο παρατήρησα ότι υπάρχουν 2 ομάδες οι οποίες η κάθε μία από αυτές ακολουθούν μία συγκεκριμένη συμπεριφορά. Η πρώτη ομάδα αποτελείται από τα rank που έχουν εύρος από 1 μέχρι 4 και η δεύτερη ομάδα αποτελείται από τα rank τα οποία ανήκουν σε ένα εύρος τιμών από 8 μέχρι 30.



Γραφική παράσταση 6.2.13(α): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 3.1.10



Γραφική παράσταση 6.2.13(β): Γρ. παράσταση η οποία παρουσιάζει τα αποτελέσματα του πίνακα 6.1.20

Στη συνέχεια αυξήσαμε το εύρος τιμών σε α = 350 και παρατηρήσαμε ότι το ποσοστό επιτυχίας αρχίζει να κατεβαίνει και επομένως σταματήσαμε σε αυτό το σημείο.

Εύρος τιμών = α = 350

	1	2	4	8	12	16
0.01	47.72	43.47	9.09	4.54	25.0	28.0
0.05	43.34	47.82	13.63	8.69	5.56	5.56
0.1	47.72	43.47	21.05	26.67	5.88	5.88
0.5	31.34	37.5	26.67	28.57	28.57	28.57

Πίνακας 6.1.21: Rank/Lambda πίνακας

6.4.5.2 Γενικά συμπεράσματα 2ου και 3ου πειράματος

Στο 2ο πείραμα λάβαμε το καλύτερο ποσοστό επιτυχίας χρησιμοποιώντας α = 160. Στη συνέχεια αυξάνοντας το α παρατηρήσαμε ότι δεν λαμβάνουμε καλά αποτελέσματα για εύρη τιμών μεγαλύτερα του 160 και μικρότερα του 300.

Στη περίπτωση του 3ου πειράματος χρησιμοποιώντας α = 300 πήραμε το καλύτερο δυνατό αποτέλεσμα σε όλες τις περιπτώσεις και των 2 πειραμάτων με ποσοστό επιτυχίας το οποίο άγγιξε το 73%.

6.5 Πείραμα Επεκτασιμότητας και απόδοσης σε χρόνο αλγορίθμου συστάσεων

Περιγραφή πειράματος:
Σε αυτό το πείραμα σκοπός είναι να μελετήσουμε την απόδοση του αλγορίθμου συστάσεων, ως προς το χρόνο που χρειάζεται για να εκπαιδευτεί και να προτείνει αντικείμενα σε χρήστες, δίνοντας του ένα σετ δεδομένων σε 3 διαφορετικά μεγέθη.

Σετ δεδομένων: Τα δεδομένα που χρησιμοποιούμαι σε αυτή τη περίπτωση είναι τα ίδια με αυτά που έχουμε χρησιμοποιήσει το υποκεφάλαιο 6.2 το οποίο αφορά τη συλλογή δεδομένων από τη σελίδα Movielens και έχουν την εξής μορφή:
<χρήστης, αντικείμενο (ταινία στη συγκεκριμένη περίπτωση), βαθμολογία>

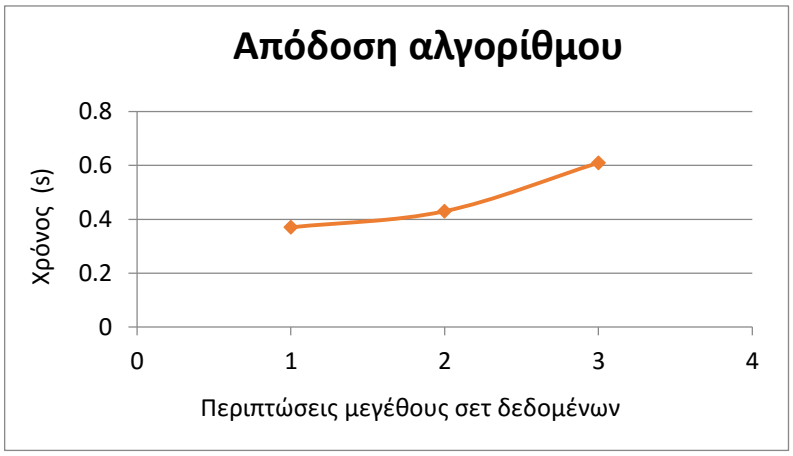
Μέγεθος του σετ δεδομένων: Όπως είπαμε πιο πάνω έχουμε δώσει το σετ δεδομένων στο αλγόριθμο συστάσεων ως είσοδο σε 3 διαφορετικά μεγέθη:
- 100 χιλιάδες βαθμολογίες
- 1 εκατομμύριο βαθμολογίες
- 10 εκατομμύρια βαθμολογίες

Εκτέλεση πειράματος:
Λάβαμε την απόδοση σε χρόνου που χρειάζεται ο αλγόριθμος συστάσεων για να προτείνει σε 800 τυχαίες χρήστες με είσοδο τα 3, διαφορετικά σε μέγεθος, σετ δεδομένων κάθε φορά. Στη συνέχεια, υπολογίσαμε το χρόνο που χρειάζεται για να προτείνουμε μεμονωμένα σε ένα χρήστη.
Σημείωση: Τα πιο κάτω αποτελέσματα λήφθηκαν τοπικά σε ηλεκτρονικό υπολογιστή.

1. **Για 800 χρήστες:**

Μέγεθος του σετ δεδομένων	Απόδοση σε χρόνο (s)
100 χιλιάδες	295
1 εκατομμύριο	346
10 εκατομμύρια	489

Πίνακας 6.1.22

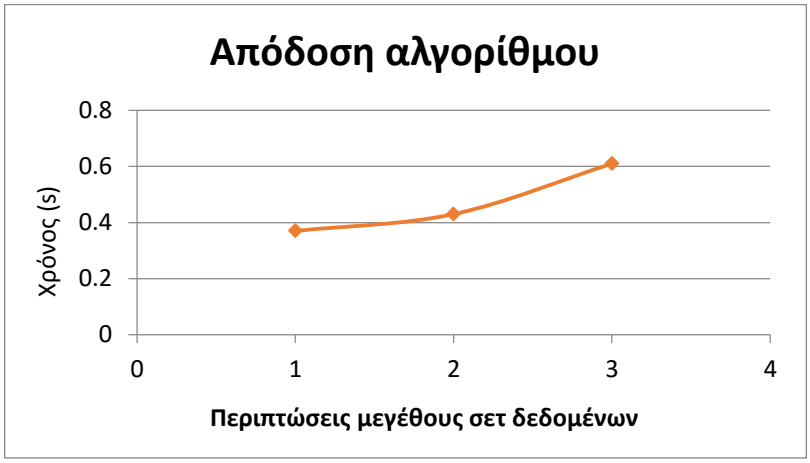


Γραφική Παράσταση 6.2.14

2. **Για 1 χρήστη:**

Μέγεθος του σετ δεδομένων	Απόδοση σε χρόνο (s)
100 χιλιάδες	0.37
1 εκατομμύριο	0.43
10 εκατομμύρια	0.61

Πίνακας 6.1.23



Γραφική Παράσταση 6.2.15

Συμπέρασμα:

Επεκτείνοντας το σετ δεδομένων μας από 100 χιλιάδες σε 1 εκατομμύριο, δηλαδή 10 φορές μεγαλύτερο παρατηρήσαμε ότι ο αλγόριθμος συστάσεων συνεχίζει να έχει καλή απόδοση σε χρόνο αφού ο χρόνος που χρειάζεται για να προτείνει αντικείμενα σε ένα χρήστη αυξήθηκε κατά 0.06 δευτερόλεπτα αφού στις 100 χιλιάδες ο χρόνος που χρειάζεται είναι 0.37 δευτ. και στο 1 εκατ. είναι 0.43 δευτ. Στη συνέχεια δώσαμε στον αλγόριθμο ένα ακόμη πιο μεγάλο σετ δεδομένων το οποίο είναι 100 φορές πιο μεγάλο από το αρχικό και 10 από το δεύτερο. Έχει μέγεθος 10 εκατομμύρια και ο αλγόριθμος φαίνεται να ανταποκρίνεται στις απαιτήσεις της επεκτασιμότητας τους σε μεγαλύτερα σετ αφού ο χρόνος που χρειάστηκε να προτείνει σε ένα χρήστη αυξήθηκε κατά 0.18 δευτερόλεπτα από το σετ του $1^{ος}$ εκατ. και κατά 0.24 από το σετ των 100 χιλιάδων. Επομένως, ο στόχος της επεκτασιμότητας του αλγόριθμου έχει επιτευχθεί σε μεγάλο βαθμό.

Κεφάλαιο 7

Συμπεράσματα και Μελλοντική εργασία

7.1 Γενικά Συμπεράσματα 103
7.2 Μελλοντική εργασία 104

7.1 Γενικά Συμπεράσματα

Στις μέρες φαίνεται ότι υπάρχει μεγάλο ενδιαφέρον για δημιουργία συστημάτων συστάσεων. Η ανάλυση των δεδομένων σε πληροφορίες και μετέπειτα σε γνώση είναι πολύ σημαντική τεχνική, καθώς με αυτό τον τρόπο μπορούμε να παρατηρήσουμε διάφορες συμπεριφορές και να καταλήξουμε σε συμπεράσματα. Για παράδειγμα, οι εταιρίες και οι οργανισμοί έχουν ως βάση να στοχεύουν σε άτομα τα οποία θα ενδιαφερθούν για αυτούς. Η διαφήμιση είναι ένας πολύ γενικός όρος, κουράζει τους ανθρώπους και είναι εντελώς τυχαία, δηλαδή δεν ανταποκρίνεται στο προφίλ των ατόμων που παρουσιάζεται. Επομένως, αφού τα συστήματα συστάσεων βασίζονται στη πραγματική δραστηριότητα, συλλογή και ανάλυση δεδομένων, τότε αυτό σημαίνει ότι θα στοχέψουν στη πρόβλεψη δραστηριοτήτων και γενικά αντικειμένων τα οποία θα ενδιαφέρουν τα άτομα τα οποία θα τους παρουσιαστούν.

Το σύστημα συστάσεων το οποίο έχει υλοποιηθεί ανταποκρίνεται στις αρχικές απαιτήσεις που έχουμε θέσει οι οποίες αφορούσαν την ορθότητα, την απόδοση σε χρόνο και κατά πόσο μπορεί να φορεθεί σε διάφορα σενάρια τα οποία δεν αναφέρονται συγκεκριμένα σε ένας είδος αντικειμένου, αλλά σε ένα γενικό επίπεδο. Με αυτό τον τρόπο και βάση της αρχιτεκτονικής που έχουμε θέσει μπορούμε να προτείνουμε αντικείμενα σε χρήστες τα οποία θα ανταποκρίνονται σε υψηλό βαθμό ποσοστού επιτυχίας και θα παραδίδονται έγκαιρα σε χρόνο.

Η μελέτη των αλγορίθμων και των διαφόρων τεχνικών που έχουν χρησιμοποιηθεί δίνουν τη δυνατότητα σε εμάς να κατανοήσουμε σε καλύτερο βαθμό τι κρύβεται πίσω από ένα αλγόριθμο ο οποίος έχει στόχο να προβλέψει τι θα μας άρεσε στο μέλλον με βάση το παρόν και τη σύγκριση μας με άλλους χρήστες. Επίσης, έχουν κατανοηθεί οι δυνατότητες που μας παρέχουν τα διάφορα εργαλεία και πως μπορούν να αξιοποιηθούν ανάλογα με το σκοπό και τη προσέγγιση του προβλήματος που είχαμε θέσει εξαρχής.

Το γραφικό περιβάλλον με το οποίο θα αλληλεπιδρά ο χρήστης, δηλαδή η Android εφαρμογή έχει κατασκευαστεί και θέσει ως στόχους να είναι απλή σε χρήση χωρίς επιπλέον λεπτομέρειες οι οποίες θα επιβαρύνουν τη λειτουργία της. Αυτός ο στόχος έχει επιτευχθεί σε μεγάλο βαθμό παρέχοντας δυνατότητες αλληλεπίδρασης του χρήστη με το σύστημα συστάσεων μέσα από ένα, εύκολο σε λειτουργία, περιβάλλον λειτουργίας.

7.2 Μελλοντική Εργασία

1. Το σύστημα συστάσεων το οποίο έχει αναπτυχθεί πιο πάνω μπορεί να εγκατασταθεί σε συστοιχία από διακομιστές οι οποίοι θα έχουν εγκαταστημένο το Spark και όλες οι διασυνδέσεις του συστήματος να γίνονται μέσω αυτού. Επίσης, σε αυτό το κομμάτι θα πρέπει να μελετηθεί η διασύνδεση μεταξύ της Android εφαρμογής και του Kafka producer και αντίθετα η διασύνδεση του Kafka receiver με την Android εφαρμογή. Για να επιτευχθεί αυτό το κομμάτι θα γίνει επικοινωνία από τη βάση δεδομένων προς το Kafka και αντίστροφα διαμέσου προγράμματος το οποίο θα είναι γραμμένο σε γλώσσα PHP.

2. Ένα άλλο κομμάτι αφορά τη δημιουργία διαδικτυακής εφαρμογής με την οποία οι χρήστες οι οποίοι βρίσκονται πίσω από την αντιπροσωπεία ενός αντικειμένου να μπορούν να έχουν πρόσβαση σε διάφορες λειτουργίες. Για παράδειγμα, αν τα αντικείμενα μας αφορούν καταστήματα σε ένα εμπορικό κέντρο τότε αυτό σημαίνει ότι ο ιδιοκτήτης του καταστήματος ή κάποιο υπεύθυνο άτομο του καταστήματος να μπορεί να έχει πρόσβαση σε ένα γραφικό περιβάλλον το οποίο θα του παρέχει τις εξής λειτουργίες:
   - Εγγραφή/Είσοδος/Έξοδος στη διαδικτυακή εφαρμογή.

- Να μπορεί να εισάγει και να αλλάζει τις απαραίτητες πληροφορίες τις οποίες περιέχει μια ειδοποίηση όταν σταλεί στη κινητή συσκευή ενός χρήστη του οποίο προτείναμε το συγκεκριμένο αντικείμενο ή κατάστημα, σύμφωνα με το παράδειγμα που δώσαμε πιο πάνω. Δηλαδή, να μπορεί να βάλει τον τίτλο, την περιγραφή και την εικόνα που αποτελούν μιαν ειδοποίηση. Με αυτό τον τρόπο θα μπορεί να αναβαθμίζει τι θέλει να βλέπουν οι χρήστες που λαμβάνουν ειδοποιήσεις από το σύστημα συστάσεων. Παράδειγμα μπορεί να αφορά μια προσφορά η οποία ισχύει για τη συγκεκριμένη μέρα μόνο.

3. Σύστημα διαχείρισης βάσης δεδομένων:

Μπορεί να αναπτυχθεί ένα σύστημα διαχείρισης της βάσης δεδομένων η οποία αφορά την Android εφαρμογή. Με αυτό τον τρόπο, αν στο μέλλον χρειαστεί να γίνει οποιαδήποτε αλλαγή στην εφαρμογή τότε αυτό θα είναι πολύ εύκολο ως προς το κόπο και το χρόνο που χρειάζεται για να αναβαθμιστούν κάποια δεδομένα που περιέχει και επιθυμούμε να τα αλλάξουμε.

Βιβλιογραφία

[1] What Is Apache Spark? Webopedia Definition. 2016. What Is Apache Spark? Webopedia Definition, http://www.webopedia.com/TERM/A/apache-spark.html.

[2] Apache Spark™ - Lightning-Fast Cluster Computing . 2016. Apache Spark™ - Lightning-Fast Cluster Computing , http://spark.apache.org/.

[3] Prem Melville and Vikas Sindhwani. 2010. Recommender Systems, http://vikas.sindhwani.org/recommender.pdf.

[4] Analytics and Visualization of Big Data: Recommendation system - benefits and drawbacks, http://auburnbigdata.blogspot.com.cy/2013/03/recommendation-system-benefits-and.html.

[5] Bugra Akyildiz. 2016. Alternating Least Squares Method for Collaborative Filtering | Bugra Akyildiz, http://bugra.github.io/work/notes/2014-04-19/alternating-least-squares-method-for-collaborative-filtering/.

[6] Wikipedia. 2016. Collaborative filtering - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Collaborative_filtering.

[7] Chapter 9, Mining Massive Datasets, by Jure Leskovec, Anand Rajaraman and Jeff Ullman, Cambridge University Press, 2014

[8] Wikipedia. 2016. Dimensionality reduction - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Dimensionality_reduction.

[9] Machine Learning Library (MLlib) Programming Guide - Spark 1.2.1 Documentation, https://spark.apache.org/docs/1.2.1/mllib-guide.html.

[10] Facebook Code. 2016. Recommending items to more than a billion people | Engineering Blog | Facebook Code | Facebook, https://code.facebook.com/posts/861999383875667/recommending-items-to-more-than-a-billion-people/.

[11] Collaborative Filtering - spark.mllib - Spark 1.6.1 Documentation. 2016. Collaborative Filtering - spark.mllib - Spark 1.6.1 Documentation, http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html.

[12] Wikipedia. 2016. Cold start - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Cold_start.

[13] SearchSOA. 2016. What is Java? - Definition from WhatIs.com, http://searchsoa.techtarget.com/definition/Java.

[14] SearchBusinessAnalytics. 2016. What is Apache Spark? - Definition from WhatIs.com, http://searchbusinessanalytics.techtarget.com/definition/Apache-Spark.

[15] Spark Streaming | Apache Spark . 2016. Spark Streaming | Apache Spark , http://spark.apache.org/streaming/.

[16] Kafka. 2016. Apache Kafka, http://kafka.apache.org/documentation.html.

[17] SearchEnterpriseLinux. 2016. What is MySQL? - Definition from WhatIs.com, http://searchenterpriselinux.techtarget.com/definition/MySQL.

[18] MySQL :: Top Reasons to Use MySQL. 2016. MySQL :: Top Reasons to Use MySQL, https://www.mysql.com/why-mysql/topreasons.html.

[19] Wikipedia. 2016. PHP - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/PHP.

[20] Developers. 2016. What is JSON? — Developers, http://developers.squarespace.com/what-is-json/.

[21] SearchSOA. 2016. What is Eclipse? - Definition from WhatIs.com, http://searchsoa.techtarget.com/definition/Eclipse.

[22] Meet Android Studio | Android Studio . 2016. Meet Android Studio | Android Studio , https://developer.android.com/studio/intro/index.html.

[23] Wikipedia. 2016. Application programming interface - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Application_programming_interface.

[24] Facebook Developers. 2016. Overview - Graph API - Documentation - Facebook for Developers, https://developers.facebook.com/docs/graph-api/overview/.

[25] Wikipedia. 2016. Application programming interface - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Application_programming_interface.

[26] Background monitoring of Estimote Beacons. 2015. Estimote, http://developer.estimote.com/android/tutorial/part-2-background-monitoring/.

[27] Introduction to Hadoop, Lab 1 by Pavlos Antoniou. 2016. Data mining on the web, University of Cyprus, http://www.cs.ucy.ac.cy/courses/EPL451/labs/LAB01/LAB01.pdf.

[28] Wikipedia. 2016. Systems architecture - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Systems_architecture.

[29] Estimote Beacons — real world context for your apps. 2016. Estimote Beacons — real world context for your apps, http://estimote.com/.

[30] Wikipedia. 2016. Bluetooth low energy - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Bluetooth_low_energy.