ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΥΣΤΑΣΕΙΣ ΒΑΣΙΣΜΕΝΕΣ ΣΕ ΓΡΑΦΟΥΣ ΓΝΩΣΗΣ

Διπλωματική Εργασία

του

Παπαγεωργίου Δημητρίου

Θεσσαλονίκη, Ιούνιος 2023

ΣΥΣΤΑΣΕΙΣ ΒΑΣΙΣΜΕΝΕΣ ΣΕ ΓΡΑΦΟΥΣ ΓΝΩΣΗΣ
RECOMMENDATIONS BASED ON KNOWLEDGE GRAPHS

Papageorgiou Dimitrios

B.Sc in Chemistry , Aristotle University of Thessaloniki, 1995
- Master in Business Administration (MBA), University of Macedonia 2005

Masters Thesis

Submitted for the partial fulfillment of the requirements for obtaining

M.Sc IN BUSINESS COMPUTING

Supervisor
Assistant professor GEORGIA KOLONIARI

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30/10/2023

Ονοματεπώνυμο 1	Ονοματεπώνυμο 2	Ονοματεπώνυμο 3
Επίκουρη καθηγήτρια	Καθηγητής	Καθηγητής
Κολωνιάρη Γεωργία	Γεωργιάδης Χρήστος	Ξυνόγαλος Στυλιανός

Papageorgiou Dimitrios

.....

Περίληψη

Η εποχή μας χαρακτηρίζεται από τον τεράστιο και συνεχώς αυξανόμενο όγκο δεδομένων , που είναι διαθέσιμα σε σχεδόν όλους τους τομείς, καθώς και από την υπερπληθώρα σε προσφορά προϊόντων και υπηρεσιών. Ο πολύ μεγάλος αυτός αριθμός επιλογών, μπορεί πολλές φορές να οδηγήσει τον μέσο χρήστη σε δυσκολία λήψης αποφάσεων έως και "παράλυση". Για αυτόν ακριβώς το λόγο, σε πάρα πολλές εφαρμογές έχουν ενσωματωθεί Συστήματα Συστάσεων (ΣΣ), ώστε να βοηθήσουν τον χρήστη στην επιλογή του. Στην συγκεκριμένη εργασία μας, στοχεύσαμε στην δημιουργία ενός ΣΣ που εφαρμόσαμε σε μια πλατφόρμα παροχής μικρών εκπαιδευτικών προγραμμάτων πάνω στην χρήση προγραμμάτων πληροφορικής, για στελέχη επιχειρήσεων , *την MOOC.office365-training.com, της Mandarine Academy.* Για τον σκοπό αυτό χρησιμοποιήσαμε σχετική βάση δεδομένων, με την συμπεριφορά και τις επιλογές των χρηστών της πλατφόρμας σε μία διάρκεια ετών και με βάση αυτά τα δεδομένα, υλοποιήσαμε ένα αλγόριθμο Συστάσεων τύπου Συνεργατικού Φίλτρου (Collaborative based filtering) που βασίζεται στη χρήση ενός γράφου γνώσης.

Λέξεις Κλειδιά: Συστήματα Συστάσεων, Γράφοι γνώσης

Abstract

Our era is characterized by the huge and ever-increasing amount of data available in almost all sectors, as well as by the overabundance of products and services on offer. This very large number of options can often lead the user to difficulty in making a choice even up to "paralysis". For this very reason, many applications have integrated Recommender Systems (RS) to help the user in his choice. In our specific work, we worked on creating a RS that we based on an existing platform, Mandarine Academy's **MOOC.office365-training.com**, which provides short e-learning courses, on the use of IT programs and applications, aimed at business employees. For this purpose we used the relevant database with the behavior and choices of the users of the platform, over a period of years and based on this data, we implemented a Collaborative filter-type Recommendation algorithm (Collaborative based filtering) based on the use of a knowledge graph.

Keywords: Neo4j, Recommendation Systems, Recommender systems, e-learning, Knowledge Graphs

Πρόλογος – Ευχαριστίες (προαιρετικό)

Θα ήθελα να πω πολλά ευχαριστώ στην οικογένειά μου, που με στήριξε στην προσπάθειά μου.

Επίσης στην επιβλέπουσα καθηγήτρια κα Κολωνιάρη, για την καθοδήγηση και εν γένει συνδρομή της.

Περιεχόμενα

Συμβολισμοί (αν υπάρχουν)

ΣΣ	**-Συστήματα Συστάσεων**
ΒΔ	**-Βάσεις Δεδομένων**
ΒΔΓ	**-Βάσεις Δεδομένων Γράφου**
ΣΒΔ	**-Σχεσιακή Βάση Δεδομένων**
UB-CFA	**-User Based Collaborative Filtering Algorithm**
ITB-CFA	**-Item Based Collaborative Filtering Algorithm**

1 Εισαγωγή

1.1 Το Πρόβλημα – Η Σημαντικότητα του θέματος

Το αντικείμενο της παρούσας εργασίας, είναι η μελέτη σχετικά με την υλοποίηση ενός Συστήματος Συστάσεων (ΣΣ), βασισμένο πάνω σε έναν Γράφο Γνώσης.

Σύμφωνα με τον Παπαγεωργίου (2021, σ.4), "τα ΣΣ υπάρχουν στην αγορά εδώ και αρκετά χρόνια. Είναι εξαιρετικά δημοφιλή στις εφαρμογές ηλεκτρονικού εμπορίου, καθώς είναι χρήσιμα τόσο στους ενδιαφερόμενους πελάτες , ενώ ταυτόχρονα αυξάνουν τις πωλήσεις των εταιρειών. Δεν είναι έκπληξη λοιπόν που τέτοια συστήματα χρησιμοποιούν όλες οι γνωστές παγκόσμιες πλατφόρμες ηλεκτρονικού εμπορίου, αλλά και social media καθώς και παροχής συνδρομητικών υπηρεσιών, όπως ενδεικτικά, το eBay, η Amazon, το Spotify, το Facebook, το Netflix κ.α.

Η διάδοσή τους στη σύγχρονη ιντερνετική οικονομία, έχει να κάνει βασικά με το γεγονός ότι υπάρχει υπερπληθώρα πληροφοριών καθώς και προσφερόμενων ειδών σήμερα, μέσω του διαδικτύου, οπότε ο σημερινός χρήστης- καταναλωτής δυσκολεύεται πλέον να καταλήξει σε μία επιλογή. Για παράδειγμα, όπως αναφέρεται στο (Παπαγεωργίου, 2021), ένας τουρίστας που επισκέπτεται σήμερα (το 2020) την Μαδρίτη, έχει να επιλέξει μεταξύ 250 διαφορετικών ταξιδιωτικών εφαρμογών από το Google Play για να οργανώσει την ξενάγησή του στην πόλη.

Σήμερα, τα ΣΣ ήδη χρησιμοποιούνται σε ένα μεγάλο εύρος εφαρμογών, σε μια πληθώρα διαφορετικών αντικειμένων, όπως στους τομείς του βιβλίου, της μουσικής, των ταινιών, των ειδήσεων, στην πρόταση για εξεύρεση εστιατορίων, στην πρόταση για καινούργιες επαφές στα social media , αλλά ακόμα και στον τουριστικό τομέα, ο οποίος μάλιστα είναι και αυτός στον οποίο γίνεται η πιο ευρεία αυτήν τη στιγμή χρήση των ΣΣ και σε πολλούς άλλους .

Πιο συγκεκριμένα, τα ΣΣ έρχονται να αντικαταστήσουν τον κλασικό τρόπο πληροφόρησης του καταναλωτικού κοινού σχετικά με τα χαρακτηριστικά των νέων προϊόντων, ιδιαίτερα όταν πρόκειται για νέα κατηγορία, ή όταν αυτά ανανεώνονται συνεχώς με νέα χαρακτηριστικά. Μια που πλέον μεγάλο μέρος των αγορών γίνονται μέσω του διαδικτύου, προέκυψε η ανάγκη και για πληροφόρηση πριν την αγορά , μέσω του διαδικτύου.

Επομένως τα ΣΣ είναι τα online πληροφοριακά "εργαλεία" που επιδιώκουν στην ουσία να συνδέσουν τις προτιμήσεις του καταναλωτή με κάποιο- ή κάποια προϊόντα που

, με διάφορους τρόπους που θα εξετάσουμε παρακάτω, είναι κοντά στις συνήθειες ή τις προτιμήσεις του".

Η υλοποίηση τέτοιων ΣΣ, στην εποχή των Big Data , βασίζεται στην γρήγορη πρόσβαση, επεξεργασία και διαχείριση τεράστιων όγκων δεδομένων που είναι οργανωμένα σε διάφορες Βάσεις Δεδομένων (ΒΔ). Αυτές χωρίζονται βασικά σε δύο ευρείες κατηγορίες: στις Σχεσιακές ΒΔ (RDBMS) και στις Μη Σχεσιακές ή αλλιώς όπως έχει καθιερωθεί, στις NoSQL ΒΔ .

Οι Σχεσιακές ΒΔ βασίζονται στην κλασσική μορφή αρχειοθέτησης των δεδομένων και έχει πλέον αποδειχθεί, ότι δεν είναι οι καταλληλότερες για την διαχείριση των τεράστιων όγκων δεδομένων που συλλέγονται πλέον καθημερινά, μια που υπόκεινται σε πολλούς περιορισμούς, τόσο από άποψη δυνατοτήτων, όσο και από άποψη αποτελεσματικότητας και γρήγορης ανταπόκρισης. Τα τελευταία χρόνια έχει αναδειχθεί η υπεροχή στην διαχείριση των Big Data των NoSQL ΒΔ και εξ΄αυτών ειδικότερα έχουν ξεχωρίσει για την αποτελεσματικότητά τους οι ΒΔ Γράφων (Graph Databases), καθώς εκ κατασκευής συμπεριλαμβάνουν όχι μόνο τα δεδομένα αλλά και τις σχέσεις που αυτά έχουν μεταξύ τους, γεγονός που πολλαπλασιάζει την αξία τους και την χρησιμότητά τους. Παραδείγματα αξιοποίησης τέτοιων ΒΔ είναι κατεξοχήν οι πλατφόρμες Social Media όπως το Facebook, το Instagram, to LinkedIn αλλά ακόμα και πολλά εμπορικά sites όπως το Amazon, κτλ. Πολύ γρήγορα έχουν καθιερωθεί ως το στάνταρ στις πλατφόρμες ηλεκτρονικών πωλήσεων, ακριβώς επειδή επιτρέπουν εύκολα και γρήγορα τις συσχετίσεις μεταξύ τόσο των αντικειμένων που προσφέρονται, όσο και των χρηστών, αξιοποιώντας τις πληροφορίες για τις σχέσεις όλων αυτών των οντοτήτων μεταξύ τους, που είναι κάτι που ενυπάρχει στην δομή του γραφήματος, εξ΄αρχής και εξ' ορισμού. Έχουν αποδειχθεί ιδανικές για εφαρμογές όπου έχει ιδιαίτερη σημασία το είδος και η ένταση της σχέσης μεταξύ των διαφόρων οντοτήτων, όπως για παράδειγμα στα συστήματα κυβερνοασφάλειας, ανάλυσης κοινωνικών δικτύων και στα Συστήματα Συστάσεων, για εμπορικούς ή άλλους σκοπούς

1.2 Σκοπός – Στόχοι

Στόχος της εργασίας είναι η μελέτη της αποτελεσματικότητας των ΒΔ Γράφων και πιο συγκεκριμένα των Γράφων Γνώσεων (Knowledge Graphs) - δλδ των ΒΔ Γράφων

που αποτελούνται από ένα μεγάλο δίκτυο οντοτήτων, το οποίο περιλαμβάνει και την σημασία , τις ιδιότητές τους καθώς και τις μεταξύ τους σχέσεις - σε ένα ΣΣ, μέσα από τη δημιουργία ενός κατάλληλου Γράφου Γνώσης που θα χρησιμοποιείται για την παροχή συστάσεων, σε έναν τομέα όπου μπορεί να βρει χρησιμότητα. Η έμφαση θα δοθεί στη δημιουργία του γράφου και την επιλογή της κατάλληλης πληροφορίας που θα πρέπει να συμπεριλαμβάνει.

Συγκεκριμένα, μετά από ενδελεχή διερεύνηση, των διαφόρων τομέων που θα ήταν πρόσφοροι ως πεδίο για να βασίσουμε την μελέτη μας, επιλέξαμε τον τομέα παροχής συστάσεων στην επιλογή αντικειμένου σπουδών εξ' αποστάσεως, μια που είναι ένας τομέας ο οποίος, ειδικά μετά τον κορονοϊό έχει όλο και μεγαλύτερη ζήτηση, ενώ ταυτόχρονα ο αριθμός των προσφερόμενων προγραμμάτων σπουδών έχει αυξηθεί δραματικά, με αποτέλεσμα να είναι ζητούμενο η βοήθεια προκειμένου ο ενδιαφερόμενος να κάνει την σωστή επιλογή.

Επιπλέον αποφασίσαμε να δημιουργήσουμε τον Γράφο μας στην Neo4j η οποία θεωρείται πλέον η πιο καθιερωμένη Βάση Δεδομένων Γράφου (Guia etal, 2017), όντας σύγχρονη και αποτελεσματική, ενώ διαθέτει και την δική της γλώσσα χειρισμού και δημιουργίας των γράφων καθώς και διατύπωσης των ερωτημάτων, την cypher.

1.3 Τα Ερωτήματα – Οι Υποθέσεις

Η βασική υπόθεση της εργασίας μας, αυτή την οποία διερευνήσαμε πειραματικά, είναι αν γενικά οι ΒΔΓ και ειδικότερα ακόμα η Neo4j, είναι κατάλληλες, για να βασιστούν πάνω τους ΣΣ.

Στα πλαίσια αυτά, θα εξετάσουμε και την εφαρμογή διαφόρων αλγορίθμων, με σκοπό να δούμε την αποτελεσματικότητα ή μη των παραπάνω πλατφορμών, προκειμένου να υποστηρίξουν αποτελεσματικά ένα ΣΣ.

1.4 Η Συνεισφορά- Η Πρωτοτυπία της εργασίας

Οι Βάσεις Δεδομένων Γράφου, είναι καινούργιες, σχετικά, προτάσεις οργάνωσης των Δεδομένων, πολλά υποσχόμενες στην εποχή μας των Big Data. Επομένως η εργασία μας φιλοδοξεί να προσθέσει ένα ακόμη κρίκο στην διερεύνηση των δυνατοτήτων που αυτές προσφέρουν, καθώς και των εφαρμογών που μπορούν να βασιστούν πάνω τους.

Ειδικότερα ακόμη, με την μελέτη μας θεωρούμε ότι προάγουμε επίσης την διάδοση των ΣΣ, που θεωρούμε ότι είναι ένα χρήσιμο εργαλείο για την λήψη αποφάσεων, ειδικότερα δε, στον τομέα της απομακρυσμένης Εκπαίδευσης, η οποία απ' ότι φαίνεται, ειδικά μετα την επιδημία της Covid-19, φαίνεται ότι ήρθε για να μείνει.

1.5 Η Διάρθρωση της μελέτης

Η εργασία διαρθρώνεται συνοπτικά ως παρακάτω.

Στο *Κεφάλαιο 2,* γίνεται παρουσίαση των ευρημάτων της βιβλιογραφικής έρευνας που διενεργήθηκε. Συγκεκριμένα στην πρώτη ενότητα περιγράφουμε το ιστορικό της εξέλιξης των ΒΔΓ, μέχρι σήμερα. Επίσης περιλαμβάνουμε και πληροφορίες για την δομή και τα πεδία στα οποία βρήκαν εφαρμογή. Στην δεύτερη ενότητα αναφέρονται επιγραμματικά δημοφιλείς NoSQL βάσεις δεδομένων, μεταξύ των οποίων και η Neo4j, καθώς και οι ιδιαιτερότητες τους. Η 4η ενότητα, επικεντρώνεται συγκεκριμένα στη δομή και στην λειτουργία της Neo4j, ενώ γίνεται αναφορά και στην δική της γλώσσα που χρησιμοποιεί, για την διατύπωση των ερωτημάτων , την Cypher. Η 5η ενότητα , εξετάζει την διαφορά των Σχεσιακών Βάσεων Δεδομένων (ΣΒΔ), σε σχέση με τις ΒΔΓ και ειδικότερα την Neo4j. Διερευνήθηκαν τα πλεονεκτήματα και τα μειονεκτήματα, καθώς και οι ιδιαιτερότητες, των δύο τύπων βάσεων. Στην ενότητα 6, γίνεται μία παρουσίαση των διαφόρων εφαρμογών ΣΣ, ενώ στην ενότητα 7, μελετώνται διάφοροι αλγόριθμοι που έχουν προταθεί για ΣΣ, ειδικά πάνω σε ΒΔΓ. Στην συνέχεια, η ενότητα 8 , συγκεντρώνεται ειδικότερα στις προτάσεις και στις εφαρμογές που έχουν γίνει σχετικά με αλγόριθμους Συστάσεων, ειδικά όμως για την Neo4j. Πιο συγκεκριμένα, παρουσιάζονται επιγραμματικά οι αλγόριθμοι Cosine Similarity και PageRank.

Στο *Κεφάλαιο 3*, παρουσιάζεται η μεθοδολογία της συγκεκριμένης έρευνας, καθώς και το σκεπτικό που οδήγησε στις επιλογές όσον αφορά στην διάρθρωση αυτής της εργασίας.

Στο *Κεφάλαιο* 4, περιγράφεται η ανάπτυξη της κατασκευής του Γράφου, καθώς και το Dataset που χρησιμοποιήθηκε, τα χαρακτηριστικά του, η επεξεργασία των δεδομένων που έγινε, τα εργαλεία που χρησιμοποιήθηκαν, καθώς και ο τρόπος που αναπτύχθηκε η εφαρμογή παροχής συστάσεων.

2 Βιβλιογραφική Επισκόπηση – Το Θεωρητικό Υπόβαθρο

Προκειμένου να κατανοηθεί καλύτερα το θέμα, έγινε αναζήτηση στην βιβλιογραφία σχετικά με άρθρα που αφορούσαν στην δημιουργία και την χρησιμότητα των ΒΔΓ , καθώς και άρθρων που αφορούν στην δημιουργία ΣΣ γενικότερα, όπως επίσης και πιο εξειδικευμένων, συγκεκριμένα σε ΣΣ πάνω σε ΒΔΓ και που έχουν να κάνουν με την υποστήριξη ενδιαφερόμενων σπουδαστών, στην επιλογή προγραμμάτων εξ' αποστάσεως εκπαίδευσης.

Έγινε προσπάθεια τα άρθρα αυτά να είναι όσο το δυνατόν πιο πρόσφατα, κατά προτίμηση μετά το 2020.

Η βιβλιογραφική αναζήτηση, έγινε στο Google Scholar, ενώ οι όροι αναζήτησης ήταν, μεταξύ άλλων, οι: "Graph Databases", " Knowledge Graphs", "Recommendation systems with Neo4j", Recommendation systems applications", "Recommendation systems and e learning", "Recommender systems in Neo4j", "Machine Learning and Recommendation systems" , "Graph Models", Neo4J.

Επίσης αφιερώθηκε πολύς χρόνος και προσπάθεια με σκοπό να βρεθούν κατάλληλα, δωρεάν και διαθέσιμα σετ δεδομένων (Datasets), προκειμένου να επιλεγεί το πιο κατάλληλο. Για τις σχετικές αναζητήσεις χρησιμοποιήθηκαν όροι όπως "Free datasets", "e learning courses", "on line learning courses", "MOOC", "edX", "Coursera" κ.α., "e learning courses", "on line learning courses", "MOOC", "edX", "Coursera" κ.α. Ως κύρια, ως πιο πλήρης και αξιόπιστη πηγή δωρεάν datasets, χρησιμοποιήθηκε η Kaggle, ενώ σχετική έρευνα έγινε μελετώντας και πηγές που προέκυψαν από το υλικό της βιβλιογραφικής έρευνας.

2.1. Η εξέλιξη των Βάσεων Δεδομένων Γράφου (ΒΔΓ)

Η θεωρία των Γράφων πάνω στην οποία βασίζονται οι ΒΔΓ, διατυπώθηκε για πρώτη φορά από τον Euler ήδη από τον 18ο αιώνα, ενώ έκτοτε έχει μελετηθεί, επεκταθεί και βελτιωθεί, χάρη στην ενασχόληση με αυτήν επιστημόνων από διάφορα γνωστικά πεδία, όπως Μαθηματικοί, Ανθρωπολόγοι, Κοινωνιολόγοι κα.. Είναι όμως μόλις τα τελευταία χρόνια, στα οποία απασχόλησε την επιστήμη της Πληροφορικής. Η ανάγκη για αυτό προέκυψε κυρίως από την τεράστια ανάπτυξη και εξάπλωση που

γνώρισαν εμπορικές εφαρμογές, όπως το Facebook, η Google και το Twitter, βασιζόμενες πάνω σε τέτοιου είδους, Βάσεις Δεδομένων Γράφων (Robinson et al, 2015).

Οι πρώτες ΒΔ που ομοιάζουν με τις ΒΔΓ, δημιουργήθηκαν από την IBM περίπου στα μέσα του 1960, με μορφή δέντρων δεδομένων, σε ιεραρχικό μοντέλο, ενώ οι πρώτοι Γράφοι με ετικέτα (Labelled Graph), εμφανίστηκαν στις αρχές του 1980 ως Logical Data Model. Στην μετέπειτα πορεία υπήρξαν πολλές βελτιώσεις, ώσπου στα μέσα με τέλη της δεκαετίας του 2000, παρουσιάστηκαν εμπορικές ΒΔΓ, όπως η Neo4j και η Oracle Spatial and Graph ΒΔΓ, οι οποίες πληρούσαν τις προδιαγραφές αξιοπιστίας ACID και ήταν πλέον διαθέσιμες στο ευρύ κοινό. Ακολούθησαν και άλλες εκδοχές από διαφορετικές εταιρείες όπως η OrientDB, MarkLogic και η ArangoDB. Η πραγματική καθιέρωσή τους όμως έγινε με την εμφάνιση και την εξάπλωση των Social Media, μια που ήταν ιδανικές γι' αυτές τις εφαρμογές. Τέλος στην διάρκεια της ίδιας δεκαετίας έγιναν διαθέσιμες και ΒΔΓ βασισμένες στο cloud, όπως η Amazon Neptune και η Neo4jAuraDB. (Wikipedia, 2023)

2.2. Περιγραφή των Γραφικών Βάσεων Δεδομένων

Οι Βάσεις Δεδομένων Γράφου, είναι βάσεις που απεικονίζουν τα δεδομένα όπως συνδέονται μεταξύ τους εννοιολογικά. Ξεχωρίζουν δε από τις σχεσιακές βάσεις δεδομένων, ιδιαίτερα από το γεγονός ότι δίνουν την ίδια σημασία στα δεδομένα, όσο και στις σχέσεις μεταξύ τους (Robinson et al, 2015). Τα δομικά τους στοιχεία είναι οι *Κόμβοι* (Nodes), και οι *Ακμές*(Edges). Οι κόμβοι αναπαριστούν στο γράφο τα δεδομένα, ενώ ταυτόχρονα περιέχουν και ιδιότητες (attributes). Οι ακμές αντιπροσωπεύουν τις *Σχέσεις* (relationships) μεταξύ των δεδομένων, οι οποίες ενώνουν και δίνουν δομή στους κόμβους, ενώ επίσης έχουν φορά και επισήμανση . Οι Σχέσεις έχουν και αυτές, τις δικές τους ιδιότητες (ShefaliPatil, G. et al, 2014). Οι ΒΔΓ βρίσκουν ιδιαίτερη χρησιμότητα, ακριβώς επειδή περιλαμβάνουν στην δομή τους πέρα από τα ίδια τα δεδομένα και τις σχέσεις που τα συνδέουν και ιδιαίτερα στις περιπτώσεις εκείνες που οι σχέσεις μεταξύ των δεδομένων είναι εξίσου, ή και πιο σημαντικές ακόμα, από τα ίδια τα δεδομένα που συσχετίζουν.

2.2.1 Παρουσίαση των διαφόρων τύπων NoSQL Βάσεων Δεδομένων

Ο όρος NoSQL, χρησιμοποιήθηκε για πρώτη φορά το 1998 για να περιγράψει μία σχεσιακή βάση δεδομένων που δεν βασίζεται στην χρήση της γλώσσας SQL, ενώ αρκετά αργότερα, γύρω στο 2009, χρησιμοποιήθηκε από τους υποστηρικτές των μη σχεσιακών βάσεων δεδομένων. οι οποίοι αναζητούσαν πιο αποδοτικές και φθηνότερες εφαρμογές διαχείρισης δεδομένων (Strauch et al., 2011).

Οι NoSQL ΒΔ είναι μια προσέγγιση στο σχεδιασμό βάσεων δεδομένων που επιτρέπει την αποθήκευση και την ερώτηση δεδομένων εκτός των παραδοσιακών δομών που βρίσκονται στις σχεσιακές βάσεις δεδομένων[1] . Τα NoSQL έχουν διάφορους τύπους βάσεων δεδομένων, όπως "έγγραφα", "κλειδί-τιμή", "ευρεία-στήλη" και "γράφημα". Παρέχουν ευέλικτα σχήματα και κλιμακώνουν εύκολα με μεγάλες ποσότητες δεδομένων και υψηλά φορτία χρηστών[2], και χρησιμοποιούνται όλο και περισσότερο σε εφαρμογές big data και real-time web.

Οι NoSQL ΒΔ λειτουργούν αποθηκεύοντας και επεξεργάζοντας δεδομένα σε μορφές που διαφέρουν από τους πίνακες σχέσεων που χρησιμοποιούνται από τις σχεσιακές βάσεις δεδομένων. Ανάλογα με τον τύπο της βάσης δεδομένων NoSQL, τα δεδομένα μπορούν να αποθηκευτούν και να ανακτηθούν ως έγγραφα (όπως JSON), ζεύγη κλειδί-τιμή, στήλες ή κόμβοι και ακμές ενός γραφήματος. Οι NoSQL ΒΔ δίνουν στους προγραμματιστές περισσότερη ευελιξία και ταχύτητα στη διαχείριση των δεδομένων, καθώς δε χρειάζεται να ορίσουν ένα σταθερό σχήμα εκ των προτέρων ή να χρησιμοποιήσουν SQL για τη δημιουργία ερωτημάτων (mongo.com). Επίσης, οι NoSQL ΒΔ μπορούν να κλιμακωθούν οριζοντίως, δηλαδή να διαχωρίσουν τα δεδομένα σε πολλούς κόμβους ή servers, για να αυξήσουν τη διαθεσιμότητα και την απόδοση.

Υπάρχουν πολλά παραδείγματα από NoSQL βάσεις δεδομένων στην αγορά, που ανήκουν σε διαφορετικούς τύπους και καλύπτουν διαφορετικές ανάγκες. Συγκεκριμένα, υπάρχουν 4 γενικές κατηγορίες NoSQL ΒΔ : α) Οι τύπου *κλειδί-τιμή*, όπου η αποθήκευση των δεδομένων γίνεται κάτω από έναν τίτλο, που ονομάζεται κλειδί. Τέτοια ΒΔ είναι η Redis[3], που λειτουργεί στη μνήμη και προσφέρει υψηλή ταχύτητα και απλότητα. Χρησιμοποιείται για τη διαχείριση session, caching, messaging και real-time analytics.

[1] ibm.com
[2] https://www.mongodb.com
[3] https://redis.io/

β) Οι τύπου *ευρείας στήλης,* στις οποίες η αποθήκευση των δεδομένων γίνεται ανά στήλες, αντί σε σειρές. Ένα παράδειγμα είναι η Cassandra[4], μια βάση δεδομένων ευρείας-στήλης, που είναι σχεδιασμένη για να χειρίζεται μεγάλους όγκους δεδομένων σε πολλούς κόμβους. Είναι από τις πιο αξιόπιστες και ανθεκτικές βάσεις δεδομένων NoSQL. γ) Αυτές που βασίζονται σε *έγγραφα,* όπου έγγραφα είναι μία σειρά από πεδία με χαρακτηριστικά. Λειτουργούν με τρόπο παραπλήσιο αυτών του τύπου κλειδί-τιμή, μόνο που που σ' αυτήν τη περίπτωση, το κλειδί είναι ένα πεδίο προσδιοριστικό του εγγράφου, το οποίο αποτελεί την τιμή και είναι σε μορφή π.χ. JSON ή XML η οποία επιτρέπει αναζήτηση στα πεδία του εγγράφου. Παράδειγμα τέτοιας ΒΔ είναι η MongoDB[5], μια βάση δεδομένων βασισμένη σε έγγραφα, που χρησιμοποιεί JSON για να αποθηκεύσει δεδομένα. Είναι μια γενικής χρήσης βάση δεδομένων που προσφέρει ευελιξία, κλιμάκωση και υψηλή απόδοση. δ) Τέλος οι *τύπου Γράφου,* στις οποίες η αποθήκευση των δεδομένων γίνεται σε μορφή Γράφου. Ένα σύγχρονο παράδειγμα, ΒΔ Γράφου, αποτελεί η Neo4j[6] ,που χρησιμοποιεί κόμβους και ακμές για να αποθηκεύσει και να αποκαλύψει σχέσεις μεταξύ δεδομένων. Είναι ιδανική για την ανάλυση συσχέτισης, την ανίχνευση απάτης και τη σύσταση προϊόντων και είναι αυτή την οποία και θα χρησιμοποιήσουμε σ' αυτήν την εργασία (Corbellini et al, 2017).

2.2.2 Παρουσίαση της Neo4j

Στην παρούσα εργασία επιλέξαμε να στήσουμε το ΣΣ μας, πάνω στην ΒΔΓ Neo4j, η οποία είναι η πιο δημοφιλής ΒΔΓ. Είναι ευρύτατα χρησιμοποιούμενη σε τομείς όπως η Υγεία, η Δημόσια Διακυβέρνηση, η παραγωγή αυτοκινήτων, στρατιωτικούς τομείς κ.α. (Guia et al, 2017)

Η Neo4j είναι μία open-source ΒΔΓ , δημιουργημένη με την γλώσσα Java. Η πρώτη της εκδοχή παρουσιάστηκε το 2007 και από τότε εξελίσσεται συνεχώς.Τα κυριότερα πλεονεκτήματά της (Guia et al, 2017), είναι η δυνατότητα για οριζόντια κλιμάκωση, που σημαίνει ότι επιτρέπει εύκολα την προσθήκη επιπλέον κόμβων στο γράφημα, το γεγονός ότι έχει την δική της γλώσσα χειρισμού δεδομένων, την Cypher, η οποία έχει φτιαχτεί ειδικά για τον χειρισμό των δεδομένων στην Neo4j, το γεγονός ότι η

[4] https://cassandra.apache.org/_/index.html
[5] https://www.mongodb.com/
[6] https://neo4j.com/

αξιοπιστία της είναι εγγυημένη σύμφωνα με τις αρχές ACID (*ατομικότητα, συνέπεια, απομόνωση, μονιμότητα*) και τέλος το ότι διαθέτει ένα πολύ φιλικο προς τον χρήστη interface χειρισμού.

2.3. Σύγκριση των NoSQL Βάσεων Δεδομένων με τις Σχεσιακές Βάσεις Δεδομένων

Οι Σχεσιακές Βάσεις Δεδομένων, είναι καλές στον χειρισμό δεδομένων συναλλαγών (transactional data) και αποτελούν ακόμα και σήμερα την πιο δημοφιλή κατηγορία ΒΔ. Παρ' όλα αυτά, τα τελευταία χρόνια, με την επικράτηση των εφαρμογών, που βασίζονται έντονα στην σχέση μεταξύ των οντοτήτων, όπως π.χ. Social media, οι ΒΔΓ κερδίζουν όλο και περισσότερο έδαφος, ως το πιο προτιμώμενο είδος ΒΔ, καθώς αποδεικνύεται ως το πιο κατάλληλο για να αντιπροσωπεύσει και να αποθηκεύσει αυτού του είδους τα δεδομένα (Guia et al, 2017). Το κυριότερο πλεονέκτημα των ΒΔΓ έναντι των σχεσιακών, είναι η αστραπιαία ταχύτητα τους στην πρόσβαση στα δεδομένα, με την διατύπωση σχετικών ερωτημάτων (queries), *όταν στην διαδικασία αυτή απαιτείται συνδυασμένη αναζήτηση, με χρήση φίλτρων και πολλαπλών συζεύξεων*, ιδιότητα πάνω στην οποία βασίζονται πλήθος σύγχρονων εφαρμογών, όπως για παράδειγμα οι εφαρμογές Social media, τα Συστήματα Συστάσεων, εφαρμογές Εξόρυξης Δεδομένων κ.α.

Δεν αποτελούν ένα υποκατάστατο των Σχεσιακών Βάσεων Δεδομένων (ΣΒΔ), είναι απλώς η πιο αποδοτική επιλογή στις περιπτώσεις που χρειάζεται να επεξεργαστούμε τεράστιους όγκους αλληλοσυνδεδεμένων δεδομένων.

Τα κύρια χαρακτηριστικά που διαφοροποιούν τις ΒΔΓ και τις δίνουν προβάδισμα έναντι των ΣΒΔ, συνοψίζονται στα εξής στοιχεία (Robinson et al, 2013): η αναζήτηση στις ΒΔΓ είναι πολύ πιο αποτελεσματική και βελτιστοποιημένη σε σχέση με τις ΣΒΔ, *στην περίπτωση που θέλουμε να αξιοποιήσουμε τις συσχετίσεις μεταξύ δεδομένων*, γιατί αξιοποιεί τις σχέσεις που ενυπάρχουν εκ κατασκευής μεταξύ των δεδομένων στις ΒΔΓ, είναι πιο κοντά στην πραγματική απεικόνιση των σχέσεων μεταξύ των δεδομένων, και επίσης υποστηρίζουν την αποθήκευση τεράστιων όγκων δεδομένων, της τάξεως των petabytes (10^15). Επιπλέον είναι πολύ ευέλικτες στην ανάπτυξή τους, μια και μπορούν να προσαρμόζονται εύκολα στις αλλαγές με το πέρασμα του χρόνου, είτε αυτές αφορούν

στην προσθήκη, είτε στην διαγραφή πληροφορίας, επιτρέπουν την αποθήκευση διαφόρων τύπων δεδομένων και είναι κατάλληλες για δεδομένα που σχετίζονται μεταξύ τους, όπως συμβαίνει πολύ συχνά στις πραγματικές περιπτώσεις. Τέλος είναι σχεδιασμένες εξ' αρχής για διεργασίες εξόρυξης δεδομένων και είναι πολύ πιο αποτελεσματικές από τις ΣΒΔ, όταν η απαίτησή μας είναι για αναζήτηση και συσχέτιση δεδομένων σε πολύ μεγάλο βάθος (Corbellini et al., 2017).

2.4. Περιγραφή των Συστημάτων Συστάσεων (ΣΣ)

Όπως είδαμε και στην εισαγωγή, τα ΣΣ έχουν βρει ευρεία εφαρμογή στις μέρες μας. Σύμφωνα με τον Παπαγεωργίου (2017, σ. 4-9), η διάδοσή τους στη σύγχρονη ιντερνετική οικονομία, έχει να κάνει βασικά με το γεγονός ότι υπάρχει υπερπληθώρα πληροφοριών καθώς και προσφερόμενων ειδών σήμερα, μέσω του διαδικτύου, οπότε ο σημερινός χρήστης- καταναλωτής δυσκολεύεται πλέον να καταλήξει σε μία επιλογή. Οπότε, τα ΣΣ έρχονται να αντικαταστήσουν τον κλασικό τρόπο πληροφόρησης του καταναλωτικού κοινού σχετικά με τα χαρακτηριστικά των νέων προϊόντων, πριν την αγορά , μέσω του διαδικτύου.

Προκειμένου να το πετύχουν αυτό, τα ΣΣ, μέχρι σήμερα, χρησιμοποιούσαν και χρησιμοποιούν κάποιες τεχνικές και μοντέλα προκειμένου κατ' αρχήν να κατανοήσουν τις προτιμήσεις του καταναλωτή - χρήστη. Οι τεχνικές αυτές των παραδοσιακών ΣΣ διακρίνονται χονδρικά στις παρακάτω:

- Στα ΣΣ που βασίζονται στο φιλτράρισμα των συστάσεων με βάση το περιεχόμενο (content based recommendation systems)

Οι τεχνικές αυτές βασίζονται στις προτιμήσεις που έχει δείξει ο καταναλωτής για κάποια παραπλήσια προϊόντα στο παρελθόν. Η ομοιότητα των προϊόντων ή των κατηγοριών προϊόντων, πάνω στην οποία βασίζονται αυτά τα ΣΣ, επιδιώκεται να εκτιμηθεί, μέσω της ομοιότητας των χαρακτηριστικών τους. Για να δουλέψουν τα συστήματα αυτά, πρέπει λοιπόν να χτιστεί ένα προφίλ χαρακτηριστικών για κάθε προϊόν, κάτι το οποίο γίνεται συνήθως αναλύοντας με διάφορες μεθόδους - π.χ. AI- τις λέξεις που χρησιμοποιούνται στην περιγραφή τους, ώστε να συσχετιστούν με κάποια χαρακτηριστικά πχ. *είδος* στην οποία ανήκει μία ταινία (π.χ. δράσης, ρομαντική κτλ). Επίσης μπορεί να αξιοποιηθεί η ομοιότητα στις λέξεις κλειδιά με τα οποία γίνονται οι

αναζητήσεις για το κάθε προϊόν, με την χρήση τεχνικών Τεχνητής Νοημοσύνης, με κυρίαρχες συγκεκριμένα τις τεχνικές Βαθιάς Μάθησης (Deep Learning) . Μάλιστα σύμφωνα με τον , τα ΣΣ μαζί με τις τεχνικές προβλέψεων (predictive analytics), είναι οι δύο πιο σημαντικοί κλάδοι στους οποίους εστιάζουν οι τεχνικές της Μηχανικής Μάθησης (Machine Learning) και που ζητούνται από τους οργανισμούς και τις επιχειρήσεις.

- Στα ΣΣ συνεργατικού φιλτραρίσματος (collaborative filtering recommendation systems)

Η λογική στην οποία βασίζονται τα συστήματα αυτά, είναι ότι χρήστες με παρόμοιες προτιμήσεις και επιλογές προϊόντων και υπηρεσιών στο παρελθόν, μπορεί κανείς να υποθέσει ότι θα εξακολουθούν να έχουν τις ίδιες ή παραπλήσιες προτιμήσεις και στο μέλλον.

Χρησιμοποιούνται διάφορες τεχνικές και αλγόριθμοι, που προσπαθούν να συλλέξουν και να αναλύσουν τις συμπεριφορές των χρηστών ώστε να διακρίνουν ομοιότητες, προκειμένου να παράγουν τις κατάλληλες συστάσεις. Η συλλογή δεδομένων από τους χρήστες, γίνεται είτε με φανερό τρόπο είτε στο παρασκήνιο, χωρίς ο χρήστης να μπορεί να το αντιληφθεί

Η συλλογή δεδομένων με *ρητό τρόπο* μπορεί να γίνει με τους παρακάτω τρόπους:

- Παρακινώντας τον χρήστη να βαθμολογήσει διάφορα αντικείμενα, δλδ να προβεί σε κάποιου είδους αξιολόγηση (rating).
- Παρακινώντας τον χρήστη να ιεραρχήσει διάφορα αντικείμενα κατά σειρά προτίμησης
- Ζητώντας από τον χρήστη να επιλέξει μεταξύ δύο αντικειμένων
- Ζητώντας από τον χρήστη να δηλώσει εμφανώς κάποια αντικείμενα που του αρέσουν

Η συλλογή στοιχείων *εμμέσως, με σιωπηλό τρόπο*, γίνεται με τους παρακάτω τρόπους :

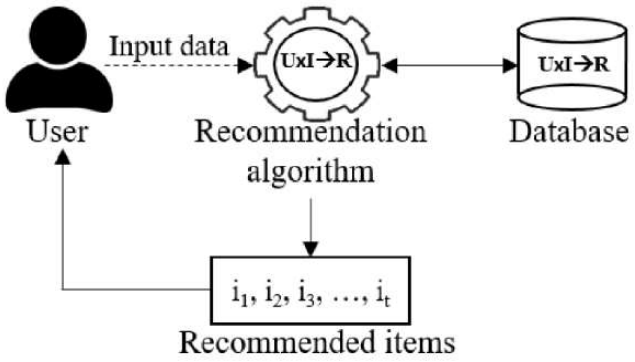
- Καταγράφοντας τις επιλογές αγοράς που κάνει ο χρήστης
- Συγκεντρώνοντας στοιχεία σχετικά με τα αντικείμενα ή/και τις υπηρεσίες που επισκέπτεται ή χρησιμοποιεί ο χρήστης
- Καταγράφοντας το πλήθος των επισκέψεων σει κάποια sites/ αντικείμενα

- Αξιοποιώντας την δραστηριότητα του χρήστη στα κοινωνικά δίκτυα (social media) για να καταγραφούν οι προτιμήσεις του κοινωνικού του περίγυρου.

Παραδείγματα γνωστών εταιρειών που χρησιμοποιούν ευρέως τα Συστήματα Συνεργατικού Φιλτραρίσματος, είναι η Amazon, για να προτείνει προϊόντα στους πελάτες της και το Facebook και το Linkedln για να προτείνουν νέους χρήστες ή ομάδες, αλλά επιπλέον στις μέρες μας και από όλα πρακτικά τα εμπορικά sites..

- Στα ΣΣ υβριδικού φιλτραρίσματος

Όπως υποδηλώνει και το όνομά τους, τα συστήματα αυτά χρησιμοποιούν τεχνικές που συνδυάζουν αλγορίθμους που βασίζονται και στα δύο παραπάνω μοντέλα, ώστε να αξιοποιήσουν τα πλεονεκτήματα και να μειώσουν τις αδυναμίες, *σαν αυτές που παρουσιάζονται στην επόμενη παράγραφο*, της κάθε μίας.



Εικόνα 1. Απλοποιημένη απεικόνιση του τρόπου λειτουργίας των παραδοσιακών Συστημάτων Συστάσεων. Από τους M. del Carmen Rodríguez-Hernández and S. Ilarri

Γενικά τα παραδοσιακά ΣΣ έχουν σημειώσει επιτυχία στους σκοπούς για τους οποίους δημιουργήθηκαν, δηλαδή στην υποβοήθηση στο πρόβλημα της υπερπληθώρας των προσφερόμενων προϊόντων και υπηρεσιών, στην αύξηση των πωλήσεων και ειδικότερα των πιο καινοτόμων και νεοφανών προϊόντων, και σε τελική ανάλυση στην αύξηση της ικανοποίησης των χρηστών. Υπάρχουν όμως ακόμα πολλά προβλήματα καθώς και πολλοί τομείς και πολλά περιθώρια για βελτίωση και προκλήσεις για περαιτέρω έρευνα, που συνοψίζονται στους παρακάτω τομείς :

Το λεγόμενο ως πρόβλημα της Αργής Εκκίνησης (Cold Start Problem), το οποίο συνίσταται στο γεγονός ότι για τους νέους χρήστες, που δεν έχουν ακόμα κάνει πολλές επιλογές/ αγορές, δεν έχουν οι αλγόριθμοι των ΣΣ αρκετά δεδομένα για να προχωρήσουν σε συσχετίσεις και άρα προτάσεις. Αφορά κυρίως τα Συνεργατικα συστήματα ΣΣ, τα οποία εξ' ορισμού βασίζονται στις αξιολογήσεις που έχουν κάνει άλλοι χρήστες με παραπλήσια συμπεριφορά στις προτιμήσεις τους. Το πρόβλημα αυτό είναι ακόμα πιο έντονο όταν πρόκειται για καινούργια προϊόντα/ υπηρεσίες, για τα οποία εκ των πραγμάτων δεν έχουν γίνει ακόμα αρκετές κριτικές.

Το πρόβλημα της Αραιότητας (Data Sparsity problem), το οποίο συνίσταται στο γεγονός ότι πολλοί χρήστες αν δεν έχουν κάποιο απτό κίνητρο, σπανίως αφήνουν αξιολογησεις και γενικά δίνουν ανατροφοδότηση στο σύστημα σχετικά με τις επιλογές τους, επομένως πάλι τα ΣΣ δυσκολεύονται να παράγουν αξιόπιστες συστάσεις, ή στη καλύτερη περίπτωση περιορίζουν το εύρος των συστάσεων. Το πρόβλημα της Αραιότητας μπορεί να οφείλεται στο φαινόμενο της Αργής Εκκίνησης ή στην ιδιοσυγκρασία του χρήστη, όπου για παράδειγμα είναι επιφυλακτικός στο να αφήνει τα "ίχνη" του στο Ίντερνετ.

Το πρόβλημα της Κλιμάκωσης (Scalability problem). Είναι γνωστό ότι η εποχή μας είναι η εποχή των λεγόμενων Big Data. Χαρακτηρίζεται δε έτσι, μια και ο όγκος των δεδομένων αυξάνεται εκθετικά, και έτσι πλέον στην περίπτωση των ΣΣ, όπου τα δεδομένα των προτιμήσεων συσσωρεύονται,, ο όγκος των σχετικών βάσεων δεδομένων από τις οποίες αντλούν δεδομένα οι αλγόριθμοι των ΣΣ γίνεται τεράστιος και ως εξ΄αυτού, η διαχείριση τους γίνεται πολύ δύσκολη. Ταυτόχρονα η απαίτηση για όλο και μεγαλύτερη επεξεργαστική ισχύ έχει γίνει επιτακτική. Η χρήση της τεχνολογίας cloud καθώς και η προσπαθεια για απλοποίηση της αντιπροσώπευσης των προϊόντων και των χαρακτηριστικών τους, με συντμημένο και απλουστευμένο τρόπο, π.χ. με εκπροσώπησή τους με δυαδικούς κώδικες αντί για περιγραφές κ.α., είναι κατευθύνσεις στις οποίες αναζητούνται λύσεις, για το πρόβλημα αυτό.

Η πρόκληση στη διαχείριση επιμέρους αξιολογήσεων, που αποτελούνται από αξιολογήσεις σε πολλαπλούς διαφορετικούς τομείς που συνθέτουν την κριτική ενός χρήστη για ένα προϊόν ή μία υπηρεσία. Για παράδειγμα, η κριτική ενός

χρήστη για ένα εστιατόριο, δεν είναι συνήθως μία μοναδική συνολική βαθμολόγηση της υπηρεσίας, αλλά πολλές βαθμολογήσεις επιμέρους πτυχών της υπηρεσίας όπως π.χ. η ποιότητα και η γεύση του φαγητού, η διακόσμηση του εστιατορίου, η εξυπηρέτηση, η τιμή κτλ. όπου για κάθε ξεχωριστό παράγοντα, ο χρήστης μπορεί να εισάγει μία ξεχωριστή, διαφοροποιημένη αξιολόγηση. Σε αυτό τον τομέα υπάρχει περιθώριο και ανάγκη από τους ερευνητές του χώρου, να επινοήσουν ή να συνδυάσουν τεχνικές με τις οποίες τα ΣΣ θα μπορούν να ενσωματώσουν, με τη κατάλληλη στάθμιση, όλες αυτές τις αξιολογήσεις σε ένα συγκεντρωτικό αποτέλεσμα παράγοντας έτσι ένα αξιόπιστο αποτέλεσμα.

- ***Η πρόκληση της συμπερίληψης στις συστάσεις, πολλών και διαφορετικών προϊόντων και υπηρεσιών (diversity challenge),*** προσφέροντας στον χρήστη μία ποικιλία κατάλληλων γι' αυτόν επιλογών και όχι μία περιορισμένη ομάδα των ίδιων και των ίδιων προτάσεων κάθε φορά. Εδώ μπορούν να βοηθήσουν οι Γράφοι Γνώσης τα ΣΣ, καθώς μέσω αυτών μπορεί η μηχανή συστάσεων να αναζητήσει συσχετίσεις πέρα από τις προφανείς και άμεσες, διατρέχοντας τον Γράφο σε ικανή απόσταση, βάσει των σχέσεων που ενώνουν τα δεδομένα.
- ***Η πρόκληση του απρόσμενου ή της έκπληξης στον χρήστη (the serendipity challenge)***, με την πρόταση καινοτόμων προϊόντων για τα οποία μπορεί να μην υπάρχουν, εκ των πραγμάτων, ήδη πολλές κριτικές, αλλά που με κάποια κριτήρια, να συναχθεί ότι μπορεί να ενδιαφέρουν τον χρήστη, κάτι που βλέπουμε ήδη π.χ. στην πλατφόρμα Netflix, όπου τελευταία δημιουργήθηκε και η επισήμανση για νέες εισόδους, ή επίσης των πιο δημοφιλών ταινιών, για την συγκεκριμένη χρονικη περίοδο, άσχετα με τις άμεσες προτιμήσεις του χρήστη.
- ***Η πρόκληση της αξιοποίησης κατανεμημένων αρχιτεκτονικών (distributed architectures), που να βασίζονται σε δίκτυα μεταξύ χρηστών (P2P networks)***, προκειμένου και να παραχθούν καλύτερες προτάσεις, αλλά και να αντιμετωπιστεί το πρόβλημα της έλλειψης ικανής υπολογιστικής ισχύς από τον κάθε μεμονωμένο χρήστη.
- ***Η παραγωγή συστάσεων για ολόκληρα γκρουπ χρηστών*** π.χ. γκρουπ τουριστών, ή μαθητές σε σχολεία, περίπτωση εμφανώς διαφορετική από αυτή των ΣΣ για μεμονωμένους χρήστες, μια που πρέπει οι συστάσεις που παράγονται να είναι κατά το δυνατόν αποδεκτές από τα περισσότερων από τα μέλη του. Το

χαρακτηριστικό αυτό είναι πολύ σημαντική πρόκληση, με μεγάλη πρακτική σημασία, μια που τα ΣΣ χρησιμοποιούνται κατ' εξοχήν στον τουριστικό τομέα, για να βοηθήσουν τουρίστες που θέλουν να επισκεφθούν τα αξιοθέατα ενός τόπου και αυτοί συνήθως ταξιδεύουν ως γκρουπ .

2.4.2 Περιπτώσεις εφαρμογών των Συστημάτων Συστάσεων (ΣΣ)

Μια που τα ΣΣ υπάρχουν πλέον εδώ και πολλά χρόνια στην αγορά, περίπου από το 1990, υπάρχουν στη βιβλιογραφία και πληθώρα εφαρμοσμένων περιπτώσεων, τόσο εμπορικών εφαρμογών όσο και μελετών και δοκιμών διαφόρων μεθόδων, τεχνικών και αλγορίθμων, σε διάφορους τομείς δραστηριοτήτων.

Στην μελέτη των Tran et al. (2020), παρουσιάζεται μία συστηματική βιβλιογραφική έρευνα ειδικά στον τομέα της Υγείας, όπου έχουν προταθεί διάφορες παραλλαγές ΣΣ για την βελτίωση τόσο της πρόληψης, όσο και της αντιμετώπισης προβλημάτων υγείας. Συγκεκριμένα περιγράφονται συνοπτικά, εργασίες που χρησιμοποιούν τα ΣΣ στην παροχή συστάσεων για την σωστή διατροφή, ανάλογα με την κατάσταση του χρήστη, στην σύσταση των κατάλληλων φαρμάκων, στην εκτίμηση της υγειονομικής κατάστασης του ασθενούς, στην παροχή συμβουλών σχετικά με προτάσεις για την ιδανικότερη φυσική άσκηση, ανάλογα με τα χαρακτηριστικά του χρήστη και τέλος για την σύσταση των καλύτερων επαγγελματιών από τον χώρο της υγείας, ανάλογα με την πάθηση και το προφίλ του κάθε χρήστη. Στο τεχνικό κομμάτι , γίνεται αναφορά στις πολλές και διάφορες μεθόδους ΣΣ που έχουν χρησιμοποιηθεί στις μελέτες αυτές, καθώς και στους διαφορετικούς σε κάθε μελέτη σχετικούς αλγορίθμους.

Ένας άλλος τομέας στον οποίο τα ΣΣ έχουν χρησιμοποιηθεί με ιδιαίτερη επιτυχία (και εμπορική π.χ. Amazon), είναι αυτός των παροχών συστάσεων στον χρήστη για την επιλογή βιβλίων. Σε αυτόν τον τομέα αφορά και η μελέτη των Sarma et al. (2021), στην οποία οι ερευνητές δημιούργησαν ένα ΣΣ το οποίο προτείνει στον χρήστη κάποιο βιβλίο, με βάση την ομοιότητα αυτού με κάποιο άλλο βιβλίο το οποίο ο χρήστης έχει ήδη αξιολογήσει θετικά. Για τον μηχανισμό του ΣΣ χρησιμοποιούν τον αλγόριθμο Cosine Similarity για να προσδιορίσουν την ομοιότητα του κάθε βιβλίου με τις συστάδες (clusters) των υπόλοιπων βιβλίων, ώστε να προτείνουν ένα κοντινό στην προτίμηση του χρήστη.

Σε άλλο τομέα, αυτόν του Τουρισμού, οι Abuzir και Dwieb (2021), δημιούργησαν ένα ΣΣ βασισμένο σε πραγματικά datasets, σε ΒΔΓ, προκειμένου να προτείνουν στους χρήστες κατάλληλα ξενοδοχεία, βάσει των προτιμήσεων και αξιολογήσεων άλλων χρηστών, χρησιμοποιώντας έναν αλγόριθμο Μηχανικής Εκμάθησης, τύπου ομοιότητας.

2.4.2.2 Συστημάτων Συστάσεων (ΣΣ) για εξ' αποστάσεως μαθήματα

Ένα άλλο πεδίο στο οποίο έχουν βρει εφαρμογή τα ΣΣ και το οποίο είναι και το αντικείμενο αυτής της εργασίας, είναι και το πεδίο Σύστασης για εξ' αποστάσεως μαθήματα για το οποίο το ενδιαφέρον κορυφώθηκε στην εποχή του Covid-19.

Η πρόσβαση στη Ανώτατη Εκπαίδευση είναι συνήθως ακριβή και δύσκολη, ενώ η ζήτηση γι' αυτή όλο και μεγαλώνει. Λύση στο πρόβλημα αυτό, τα τελευταία δέκα περίπου χρόνια παρέχει η απομακρυσμένη, on-line, πρόσβαση σε μαζικά, ανοικτής πρόσβασης, διαδικτυακά μαθήματα, που έχουν καθιερωθεί διεθνώς με τον όρο MOOCs, τα οποία επιτρέπουν την ελεύθερη, συμμετοχική και αποκεντρωμένη ικανοποίηση των αναγκών για πρόσβαση στην συνεχή μάθηση (Khalid et al, 2020). Ο αριθμός των MOOCs, καθώς και ο αριθμός των φοιτητών σ' αυτά ανέρχεται συνεχώς ενώ εκτιμάται ότι το 2018, προσφέρονταν διεθνώς πάνω από 11400 μαθήματα διαδικτυακά, από περισσότερα από 900 Πανεπιστήμια, ενώ ο αριθμός των φοιτητών που τα παρακολουθούσε ήταν περίπου 101 εκατομμύρια (Khalid et al, 2020).

Μέσα σ' αυτήν την πληθώρα των προσφερόμενων επιλογών, τα ΣΣ, είναι ουσιαστικά αλγόριθμοι και τεχνικές που επιλέγουν και προτείνουν στους ενδιαφερόμενους τμήματα και μαθήματα που ταιριάζουν στα ενδιαφέροντά τους, όπως αυτά συμπεραίνονται από το προφίλ τους και το ιστορικό τους. Από την άλλη πλευρά τα ΣΣ βοηθούν και τους παρόχους των μαθημάτων, δλδ τα εκπαιδευτικά ιδρύματα, να παρέχουν μαθήματα που να ενδιαφέρουν την μεγάλη πλειοψηφία των φοιτητών.

Ενώ η αναζήτηση μέσω της μηχανής αναζήτησης της Google είναι ο πιο δημοφιλής τρόπος για να βρει κάποιος μία πληροφορία στο Ίντερνετ, *στην περίπτωσή μας το κατάλληλο διαδικτυακό μάθημα*, επειδή *δεν* λαμβάνει υπόψη της τα ενδιαφέροντα του χρήστη, επιστρέφει χιλιάδες αποτελέσματα στον ενδιαφερόμενο, οπότε τελικά δεν τον βοηθά πάρα πολύ. Αντιθέτως τα ΣΣ επιστρέφουν προσωποποιημένες συστάσεις και έτσι διευκολύνουν *πραγματικά* τον χρήστη στην επιλογή. Η αναζήτηση μέσω μηχανών

αναζήτησης είναι γενικά πιο κατάλληλη όταν ο χρήστης ξέρει τι θέλει, ενώ τα ΣΣ είναι πιο αποδοτικά όταν ο ενδιαφερόμενος αναζητεί πληροφόρηση για αντικείμενα με τα οποία δεν είναι πολύ εξοικειωμένος (Chicaiza et al 2020).

Στον τομέα αυτό, οι Estrela et al (2017), παρουσίασαν ένα ΣΣ που προτείνει στους χρήστες μαθήματα ανάλογα με το προφίλ τους και την ομοιότητα με άλλους χρήστες. Για το σκοπό αυτό χρησιμοποίησαν τόσο το φιλτράρισμα βασισμένο στο περιεχόμενο, όσο και το συνεργατικό φιλτράρισμα, καθώς και συνδυασμό αυτων. Στην αρχή από τον χρήστη ζητείται να διαμορφώσει ένα προφίλ όπου του ζητείται να δηλώσει τις κατηγορίες εκπαιδευτικών αντικειμένων που τον ενδιαφέρουν. Έπειτα, βάσει αυτών, το ΣΣ προτείνει κάποια μαθήματα, βασισμένα στο φιλτράρισμα περιεχομένου. Στη συνέχεια ο χρήστης καλείται να δηλώσει ποιες από τις προτάσεις που του έγιναν, του αρέσουν και ποιες όχι, οπότε σε δεύτερο επίπεδο, το ΣΣ θα προχωρήσει σε συνεργατικό φιλτράρισμα, συγκρίνοντας τις προτιμήσεις του χρήστη με αυτές άλλων χρηστών και έτσι θα καταλήξει στις τελικές συστάσεις.

Οι Chicaiza,et al. (2020), σχεδίασαν ένα πλαίσιο που παρέχει συστάσεις για διαδικτυακά μαθήματα, βασισμένο στις πληροφορίες που είναι διαθέσιμες για τον κάθε χρήστη. Συγκεκριμένα, βάση αυτών των πληροφοριών επιδιώκεται να δημιουργηθεί ένα κατά το δυνατόν αντιπροσωπευτικό προφίλ των ενδιαφερόντων του κάθε χρήστη , βασισμένο στα "ίχνη" που αφήνουν οι αναζητήσεις του στο διαδίκτυο, ενώ ταυτόχρονα γίνεται προσπάθεια να κατηγοριοποιηθούν οι ανοικτές και διαθέσιμες πηγές Εκπαιδευτικών Πληροφοριών , με την χρήση τεχνικών όπως το Semantic Web. Τέλος, το μοντέλο τους, περιλαμβάνει μία μηχανή Συστάσεων , η οποία θα συσχετίσει το προφίλ του χρήστη με τις πιο σχετικές, στα ενδιαφέροντα του χρήστη, διαθέσιμες πηγές, ώστε να παρέχει τις πιο χρήσιμες συστάσεις.

2.4.2.2 Αλγόριθμοι για ΣΣ στην Neo4j

Οι Αλγόριθμοι που εφαρμόζονται στην Neo4j, ειδικά πάνω στα ΣΣ, στοχεύουν στην εύρεση μοτίβων και δομών που αποκαλύπτουν την συσσώρευση των δεδομένων γύρω από μία κοινότητα (cluster), ή την αναζήτηση των πιο κατάλληλων διαδρόμων μεταξύ οντοτήτων (path finding). Πολλοί από αυτούς βασίζονται σε επαναλαμβανόμενες

διαδικασίες και διασχίζουν τον γράφο, είτε τυχαία , είτε με κάποιο σχέδιο, ή προτεραιοποίηση.

Υπάρχουν τέσσερις μεγάλες κατηγορίες αλγορίθμων γράφων στην Neo4j, που ανήκουν στις παρακάτω κατηγορίες (Neo4j Graph Data Science/Graph algorithms):

1. Ε**ύρεση Κεντρικότητας(Centrality),** που αναλύουν την σημασία κάθε κόμβου σε ένα δίκτυο
2. Α**νίχνευση Κοινοτήτων** (Community Detection), που αναζητούν και επιδιώκουν να ομαδοποιήσουν τα δεδομένα σε clusters τα οποία να έχουν κάποια κοινά χαρακτηριστικά, αναλύοντας και την συνοχή τους
3. **Path Finding**, οι οποίοι αναζητούν τον συντομότερο δρόμο μεταξύ των δεδομένων, ή αναζητούν την ύπαρξη και την καταλληλότητα των σχετικών διαδρομών.
4. Ο**μοιότητας,** που εστιάζουν στην αναζήτηση ομοιοτήτων μεταξύ των δεδομένων

Στα ΣΣ , κυρίως βρίσκουν εφαρμογή, οι αλγόριθμοι Κεντρικότητας τύπου Page Rank και οι αλγόριθμοι Ομοιότητας.

Οι αλγόριθμοι τύπου PageRank, μετράνε την συνδεσιμότητα και άρα την σπουδαιότητα ή επιρροή των κόμβων του γράφου. Ο τρόπος λειτουργίας τους είναι είτε επαναλαμβανόμενα κατανέμοντας την αξιολογηση κάθε κόμβου σε σχέση με τους γειτονικούς του, είτε διασχίζοντας με τυχαίο τρόπο το γράφημα και καταγράφοντας στην πορεία την συχνότητα με την οποία συναντάται κάθε κόμβος.

Ο αλγόριθμος αυτός πήρε το όνομά του από τον συνιδρυτή της Google Larry Page και ουσιαστικά μετράει τον αριθμό και την ποιότητα των συνδέσεων μιας ιστοσελίδας, θεωρώντας ότι αυτό καταδεικνύει και την ποιότητα της σελίδας, στο διαδίκτυο. Ο τύπος στον οποίο βασίζεται είναι ο παρακάτω:

$$PR(A) = (1-d) + d(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)})$$

όπου θεωρείται ότι η σελίδα Α έχει τις σελίδες Τ1 μέχρι και Τn που “δείχνουν” σ’ αυτήν, ενώ C(A) ορίζεται ως ο αριθμός των σελίδων που φεύγουν από την σελίδα Α

Στην παρούσα εργασία θα εστιάσουμε στους αλγορίθμους Ομοιότητας, ως πιο δημοφιλών και καθιερωμένων.

Οι αλγόριθμοι Ομοιότητας και συγκεκριμένα ο αλγόριθμος Ομοιότητας Συνημιτόνου (Cosine Similarity), βασίζονται στην εύρεση του συνημιτόνου της γωνίας μεταξύ δύο διανυσμάτων στον n-διαστάσεων χώρο. Μπορούν επίσης να οριστούν ώς το σημείο που προκύπτει αν διαιρέσουμε δύο διανύσματα με το γινόμενο δύο διανυσματικών μηκών ή μεγεθών. Οι τιμές κυμαίνονται μεταξύ -1 και 1, όπου το -1 δηλώνει την απουσία οποιαδήποτε ομοιότητας ενω το 1 την πλήρη ομοιότητα μεταξύ δύο διανυσμάτων (κόμβων). Η σχετική βιβλιοθήκη της Neo4j, "Neo4j Graph Data Science", παρέχει συναρτήσεις και διαδικασίες για την ανεύρεση ομοιοτήτων μεταξύ δύο σετ δεδομένων, με την προϋπόθεση αυτά να έχουν κάποια κοινά στοιχεία.

Ο μαθηματικός τύπος είναι ο παρακάτω :

$$similarity(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

.

3 Μεθοδολογία

Για την εργασία αυτή χρειάστηκε να γίνει σύνθεση γνώσεων και πληροφοριών από διάφορους τομείς, προκειμένου να γίνει πλήρως αντιληπτό το αντικείμενο των ΣΣ

και να ξεκαθαρίσουν όλες οι πλευρές και οι ανάγκες του εγχειρήματος . Συνεπεία αυτών, χρειάστηκε κατ΄αρχήν να αφιερωθεί πολύς χρόνος για να μελετηθεί εξαντλητικά, μεγάλο μέρος της διαθέσιμης βιβλιογραφίας σχετικά με ΣΣ, προκειμένου να γίνουν αντιληπτές και να αφομοιωθούν όλες οι πλευρές της σημαντικότητας, της χρησιμότητας και χρήσης, καθώς και διάφοροι τρόποι υλοποίησης τέτοιων Συστημάτων.

Επιπλέον χρειάστηκε να αφιερωθεί αρκετός χρόνος στην παρακολούθηση tutorials στο διαδίκτυο, καθώς και σε ένα σχετικό on line course από το ΕΚΠΑ, σχετικά με την γλώσσα προγραμματισμού Python και ειδικά στην χρήση της στην επιστήμη των δεδομένων, προκειμένου να εμβαθύνουμε στο τεχνικό κομμάτι κατ' αρχήν, επεξεργασίας των δεδομένων.

Ακολούθως, ήταν δύσκολη η επιλογή στόχευσης της εργασίας, καθώς τα ΣΣ βρίσκουν τα τελευταία χρόνια εκτεταμένη εφαρμογή, σε πολλούς τομείς και με διάφορες μεθόδους, οπότε από την μελέτη περιπτώσεων και εφαρμογών προσπαθήσαμε να καταλήξουμε σε ένα πεδίο που να έχει ενδιαφέρον και πραγματική χρησιμότητα για τους χρήστες, δλδ τον τομέα συστάσεων στα εξ΄αποστάσεως μαθήματα. Το σκεπτικό που επικράτησε είναι ότι αφού ζούμε σε μία εποχή συνεχών και ταχύτατων αλλαγών, ή ανάγκη για συνεχή εκπαίδευση και ενημέρωση είναι κρίσιμη.

Επιπλέον κριτήριο για την επιλογή του συγκεκριμένου πεδίου, ήταν και η ύπαρξη σχετικών και ελεύθερα διαθέσιμων βάσεων δεδομένων, που να είναι ταυτόχρονα κατάλληλες και με επαρκη στοιχεία, μια που ένα ΣΣ βασίζεται, κατ' εξοχήν, σε μεγάλο αριθμό, κατάλληλων δεδομένων, που μπορούν να οδηγήσουν σε συσχετισμούς μεταξύ των χρηστών και των αντικειμένων (μαθημάτων). Δεν υπάρχουν πολλές τέτοιες πηγές, οπότε η εξεύρεση μιας τέτοιας, που εκτός των άλλων να είναι και από έγκυρη πηγή, έχοντας ταυτόχρονα πρόσφατα δεδομένα, της MARS dataset από την Mandarine Academy, όπως αυτή είναι διαθέσιμη από το αποθετήριο του Harvard University (Hafsa et al, 2022), ήταν καταλυτικός παράγοντας για την επιλογή στόχευσης της εργασίας.

Έπειτα προχωρήσαμε στην ενδελεχή μελέτη της λειτουργίας και των δυνατοτήτων της βάσης δεδομένων πάνω στην οποία βασίστηκε το ΣΣ μας, της Neo4J. Πολύ χρόνο αφιερώσαμε στην παρακολούθηση tutorials σχετικά με την εισαγωγή των δεδομένων, καθώς και με τις αρχές μοντελοποίησης των δεδομένων, προκειμένου για την δημιουργία του κατάλληλου γράφου.

Διενεργήθηκαν αρκετά τεστ, δημιουργώντας αρκετά παράλληλα μοντέλα , για να βρεθεί το πιο κατάλληλο μοντέλο οργάνωσης των δεδομένων μας στον Γράφο και φυσικά, πριν από αυτό, για την προεπιλογή των χρήσιμων και απαραίτητων για την περίπτωσή μας δεδομένων από το συνολικό dataset και την απόρριψη των περιττών. Για το σκοπό αυτό εμβαθύναμε σε δύο κρίσιμους παράγοντες για την δημιουργία του ΣΣ μας, τον τρόπο δηλαδή που συνδέονται οι χρήστες μεταξύ τους, ποια είναι τα χαρακτηριστικά που μπορούν να χρησιμοποιηθούν για να τεκμηριωθεί η ομοιότητα μεταξύ τους, καθώς και τι κοινό μπορεί να έχουν τα μαθήματα μεταξύ τους, ώστε να ενδιαφέρουν τους κοντινούς χρήστες.

Συνεχίσαμε με τη μελέτη της γλώσσας Cypher που χρησιμοποιεί η Neo4j για να διερευνηθούν οι εναλλακτικές οδοί αλλά και οι αλγόριθμοι που θα μπορούσαν να οδηγήσουν στο επιθυμητό αποτέλεσμα, την όσο το δυνατόν πιο κατάλληλη για τον χρήστη, σύσταση, του πιο ταιριαστού γι' αυτόν μαθήματος.

4. Υλοποίηση της Εφαρμογής

Στην παράγραφο αυτή θα παρουσιάσουμε όλα τα βήματα για την δημιουργία της εφαρμογής μας. Συνολικά ή όλη ερευνητική μας προσπάθεια κατέτεινε στην εξεύρεση του καταλληλότερου σχεδιασμού, για την πιο αποτελεσματική υλοποίηση ενός ΣΣ που θα βοηθάει τον ενδιαφερόμενο χρήστη, στην επιλογή του καταλληλότερου γι' αυτόν εξ' αποστάσεως μαθήματος, προκειμένου αυτός να συμπληρώσει την γνώση του στο συγκεκριμένο πεδίο και να μπορεί να διεκπεραιώσει καλύτερα τις υποχρεώσεις του στα πλαίσια της δουλειάς του.

4.1. Σταδια υλοποίησης της εφαρμογής

Τα στάδια που ακολουθήσαμε ήταν επιγραμματικά τα παρακάτω:

4.1.1 Κατανόηση του στόχου της εφαρμογής

Το στάδιο αυτό είναι πολύ σημαντικό, καθώς αφορά στην επικέντρωση στον σκοπό της εφαρμογης μας, απ' όπου προέκυψαν και οι στόχοι για τα δεδομένα που έπρεπε να συλλεχθούν, καθώς και το πλάνο του σχεδιασμού του ΣΣ.

Στην προκειμένη περίπτωση, ο στόχος είναι η παροχή συστάσεων σε χρήστες / επαγγελματίες της πλατφόρμας **-MOOC.office365-training.com** της Mandarine Academy- υποστήριξης εργαζομένων σε θέματα υιοθέτησης εφαρμογών της σύγχρονης ψηφιακής εποχής, προκειμένου αυτοί να επιλέξουν το πρόγραμμα που καλύπτει τις εκπαιδευτικές τους ανάγκες.

4.1.2 Εξεύρεση και προετοιμασία των δεδομένων

Αφιερώθηκε πολύς χρόνος στα πλαίσια αυτής της εργασίας, στην εύρεση ενός κατάλληλου Dataset, το οποίο να είναι κατάλληλο, ελεύθερα διαθέσιμο, επαρκές, έγκυρο , πρόσφατο και χρήσιμο για τον συγκεκριμένο σκοπό της εργασίας μας. Επιπλέον, προκειμένου να κατανοήσουμε τι δεδομένα είχαμε στην διάθεσή μας και τι αξία και χρησιμότητα μπορούν αυτά να έχουν στο ΣΣ. Επίσης πως συσχετίζονται μεταξύ τους, ποια είναι χρήσιμα και ποια όχι, στο ΣΣ που θα φτιάχναμε.

Με βάση όλα τα παραπάνω κριτήρια, καταλήξαμε στην χρησιμοποίηση του Mandarine Academy dataset /MARS dataset (στο Harvard Dataverse -E-learning Recommender System Dataset/ https://doi.org/10.7910/DVN/BMY3UD), που βασίστηκε στην εργασία των Hafsa et al.,(2023) Απαρτίζεται από πραγματικά δεδομένα χρηστών της πλατφόρμας της Mandarine Academy, **MOOC.office365-training.com,** η οποία ειδικεύεται στην παροχή καινοτόμων και εξειδικευμένων εξ' αποστάσεως εκπαιδευτικών προγραμμάτων που απευθύνονται σε εργαζόμενους σε μεγάλες επιχειρήσεις, με σκοπό την εξοικείωση και την εκπαίδευσή τους στην χρήση εφαρμογών ψηφιακής τεχνολογίας στα πλαίσια της εργασίας τους.

Τα δεδομένα είναι αρκετά πρόσφατα, καθώς έχουν συλλεχθεί μεταξύ του 2016 και του 2021. Οι χρήστες έχουν αξιολογήσει τουλάχιστον ένα αντικείμενο/ course , ενώ τα στοιχεία τους είναι ανωνυμοποιημένα. Συλλέχθηκαν στοιχεία για την αλληλεπίδραση των χρηστών με τα μαθήματα, με έμμεσο τρόπο, συγκεκριμένα με τον αριθμό και το ποσοστό παρακολούθησης του κάθε βιντεο -μαθήματος, ενώ μετά οι Hafsa et al, (2023), με την παραδοχή ότι αφού κάποιος χρήστης παρακολούθησε το μάθημα τον ενδιαφέρει και επίσης ότι το χρονικό ποσοστό παρακολούθησης του κάθε μαθήματος σε σχέση με την συνολική του διάρκεια, δηλώνει το πόσο σημαντικό θεωρεί ο χρήστης το μάθημα, μετέτρεψαν τα έμμεσα στοιχεία (implicit data) για τον κάθε χρήστη σε άμεσα (explicit

data) και συγκεκριμένα σε ratings, με κλίμακα από το 1 μέχρι το 10, όπου 10 είναι τα μαθήματα τα οποία παρακολούθησαν πλήρως.

4.1.2.1 Αρχικά δεδομένα

Η βάση δεδομένων μας, αποτελείται από 4 csv αρχεία, τα users_en.csv, items.csv, implicit_ratings_en.csv και explicit_ratings_en.csv, εκ των οποίων χρησιμοποιήσαμε τα τρία τα οποία παρουσιάζουμε στους παρακάτω πίνακες, με τα δεδομένα/ στήλες τους. Το αρχείο implicit_ratings_en.csv αποφασίσαμε να μην το χρησιμοποιήσουμε, καθώς η όποια πληροφορία "κουβαλάει", έχει μετασχηματιστεί και τελικά αξιοποιηθεί και ενσωματωθεί, στο αρχείο explicit_ratings.csv, μια που από εκεί προέκυψε με το σκεπτικό που αναλύσαμε παραπάνω.

ΠΙΝΑΚΑΣ 4.1.3- περιγραφή users_en.csv

Χαρακτηριστικό	**Περιγραφή**	**Κατηγορία**	**Πλήθος**
user id	ο μοναδικός κωδικός του χρήστη	integer	9902
job	το επάγγελμα που δήλωσε ο χρήστης	category	1409

ΠΙΝΑΚΑΣ 4.1.4- περιγραφή items.csv

Όνομα πεδίου	**Περιγραφή**	**Κατηγορία**	**Πλήθος**
item _id	μοναδικός κωδικός	Integer	1167
language	γλώσσα του course	category	1167
title	τίτλος του μαθήματος	text	1167
views	αριθμός θεάσεων	integer	1119

description	περιγραφή του περιεχομένου	text	1009
creation date	ημερομηνία εισαγωγής στην πλατφόρμα	date	1167
duration	διάρκεια σε δευτερόλεπτα	integer	1167
type	tutorial, use case, webcast	category	1167
level	beginner, intermediate, advanced, undefined	category	475
Job	related professions	category	1167
software	related software	category	1167
theme	related theme	category	1167

ΠΙΝΑΚΑΣ 4.1.5- περιγραφή explicit_ratings_en.csv

Όνομα πεδίου	**Περιγραφή**	**Κατηγορία**	**Πλήθος**
item_id	μοναδικός κωδικός	integer	776 (μοναδικά) 3659 (συνολικά)
user_id	μοναδικός κωδικός	integer	822 (μοναδικοί)
Watch %	ποσοστό παρακολούθησης του video-μαθήματος	float	
rating	ποσοστό παρακολούθησης του video-μαθήματος,	category	

	αναγόμενο σε κλίμακα του 10		
creation date	ημερομηνία παρακολούθησης	date	

4.1.2.2 Προεπεξεργασία των δεδομένων με χρήση της Python και επιλογή

Απ' όλα τα παραπάνω διαθέσιμα δεδομένα, κρίναμε ότι πολλά δεν είναι απαραίτητα για την εφαρμογή μας, οπότε για να μην επιβαρύνουμε τον γράφο μας, τα απαλείψαμε.

Πιο συγκεκριμένα, από το αρχείο users_en.csv, κρατήσαμε τα πεδία user_id και job. Από το αρχείο items.csv κρατήσαμε το item_id καθώς και Job, nb_views και title. Τέλος από το αρχείο explicit_en.csv κρατήσαμε επιπλέον το πεδίο rating.

Επίσης, προχωρήσαμε στην αφαίρεση των χρηστών καθώς και των μαθημάτων, όταν έλειπαν κρίσιμα δεδομένα, όπως το *job*, ώστε να διασφαλίσουμε ότι θα κρατήσουμε μόνο τους χρήστες για τους οποίους έχουμε επαρκή στοιχεία για να σχηματίσουμε το προφίλ τους, ώστε να τους παρέχουμε συστάσεις που θα τους ενδιαφέρουν

Επιπλέον , κάναμε μετασχηματισμό των δεδομένων, ώστε η παράμετρος job να έχει ονόματα ταυτόσημα μεταξύ των users και των courses, ώστε να είναι εφικτή η σύγκριση τους στον αλγόριθμο που δημιουργήσαμε. Τέλος, όλα τα πεδία αυτά τα ενοποιήσαμε σε ένα αρχείο το οποίο ονομάσαμε allData.csv και είναι αυτό που χρησιμοποίησαμε στην δημιουργία του γράφου μας.

Η παραπάνω επεξεργασία και προετοιμασία των δεδομένων μας, έγινε με την χρήση της Python και πιο συγκεκριμένα αξιοποιώντας τις δυνατότητες της βιβλιοθήκης Pandas. Ο σχετικός κώδικας που γράψαμε γι' αυτόν το σκοπό, παρατίθεται στο παράρτημα Ι

ΠΙΝΑΚΑΣ 4.1.6- περιγραφή allData.csv

Όνομα πεδίου	**Περιγραφή**	**Κατηγορία**
item_id	μοναδικός κωδικός	integer
user_id	μοναδικός κωδικός	integer
rating	ποσοστό παρακολούθησης του video-μαθήματος, αναγόμενο σε κλίμακα του 10	category

title	τίτλος του μαθήματος	text
views	αριθμός θεάσεων	integer
job	το επάγγελμα που δήλωσε ο χρήστης	category
Job	related professions	category

4.1.3 Δημιουργία του Μοντελου

Σημαντικό κομμάτι της εφαρμογής υπήρξε η δημιουργία του μοντέλου με το οποίο περιγράφουμε το πρόβλημά μας και απεικονίζουμε την συσχέτιση των δεδομένων. Το σκεπτικό με βάση το οποίο κινηθήκαμε, ήταν να κρατήσουμε αφ' ενός το μοντέλο μας όσο πιο απλό γίνεται, ώστε να είναι όσο πιο κατανοητό, αποτελεσματικό από άποψη χρονικής απόκρισης στα ερωτήματά μας και διαχειρίσιμο, ενώ όμως ταυτόχρονα είχαμε την έγνοια, να μην χάσουμε πολύτιμες συσχετίσεις μεταξύ των δεδομένων, που θα μας περιόριζαν στις αναζητήσεις μας και θα μας στερούσαν πολύτιμων διασυνδέσεων μεταξύ των οντοτήτων που θα μπορούσαν να εμπλουτίσουν τα αποτελέσματά μας. Με αυτό το πλαίσιο σκεφτήκαμε αρχικά να μοντελοποιήσουμε και τις αξιολογήσεις ως ξεχωριστούς κόμβους, μια που παίζουν καθοριστικό ρόλο στα ερωτήματά μας, σε ένα ΣΣ, όπως ήταν ο στόχος μας. Άλλη οδός που επεξεργαστήκαμε ήταν να δημιουργήσουμε ξεχωριστούς κόμβους για το πεδίο Job δλδ, το επάγγελμα του χρήστη, ή ακόμα και για το επαγγελματικό αντικείμενο που σχετίζεται με το κάθε course. Εκτός από τη σκέψη για επιπλέον κόμβους, διερευνήσαμε και τις διαφορετικές συνδέσεις / σχέσεις που θα ήταν χρήσιμο να συνδέσουν τους κόμβους μας.

Όσον αφορά στα πεδία attributes που είχαμε στην διάθεσή μας και έπρεπε να επιλέξουμε ποια θα κρατήσουμε, πάλι το σκεπτικό μας ήταν να μην "βαρύνουμε" υπερβολικά το μοντέλο μας, αλλά και να μην "πετάξουμε" οποιαδήποτε πληροφορία που θα μπορούσε να βελτιώσει την ποιότητα των αποτελεσμάτων / συστάσεών μας. Έτσι εξετάσαμε αν θα κρατήσουμε επιπλέον τα πεδία :

creation date
duration

type
level
software

που είχαμε στην διάθεσή μας, όμως κρίναμε ότι δεν ήταν απαραίτητο. Συγκεκριμένα καταλήξαμε ότι δεν ήταν σκόπιμο να περιορίσουμε τις προτάσεις μας με βάση την ημερομηνία δημιουργίας του course, καθώς γενικά τα μαθήματα είναι εισαγωγικού επιπέδου και άρα οι γνώσεις που παρέχουν δεν είναι κάτι καινοτόμο, αλλά κυρίως βασικές οδηγίες χρήσης των εφαρμογών ή βασικού χειρισμού των βασικών προγραμμάτων και εφαρμογών, επομένως η αξία τους δεν υποβαθμίζεται γρήγορα και αισθητά με την πάροδο του χρόνου. Επιπλέον σε ότι αφορά την διάρκεια των courses, επειδή πάλι γενικά δεν είναι πολύ μεγάλη, μερικές ώρες στην πλειοψηφία τους, και καθώς αφορούν γνώσεις χειρισμού βασικών εφαρμογών και προγραμμάτων, ο χρήστης θα τα παρακολουθήσει έτσι κι αλλιώς αν είναι απαραίτητα για την εργασία του. Για τον ίδιο ακριβώς λόγο δεν θεωρήσαμε σημαντικό τον τύπο των μαθημάτων, αφού κάποιος αν του είναι απαραίτητη η γνώση για να κάνει την δουλειά του, θα το παρακολουθήσει σε όποια μορφή και να είναι το μάθημα (πάντα απομακρυσμένα και μέσω υπολογιστή). Για το ίδιο σκεπτικό αποφασίσαμε να μην χρησιμοποιήσουμε το πεδίο level, με επιπλέον βασικό κριτήριο ότι δεν έχουμε γνωστό το αντίστοιχο επίπεδο του χρήστη, αλλά δεν μπορεί να γίνει ουσιαστικά, αξιοποίηση αυτής της πληροφορίας. Τέλος κρίναμε ότι ο παράγοντας software (που είναι απαραίτητο για την παρακολούθηση του course) δεν είναι πραγματικά ουσιώδης, μια που με τον ένα ή με τον άλλο τρόπο, κάποιος που χρειάζεται τις γνώσεις για την εργασία του, θα το προμηθευτεί.

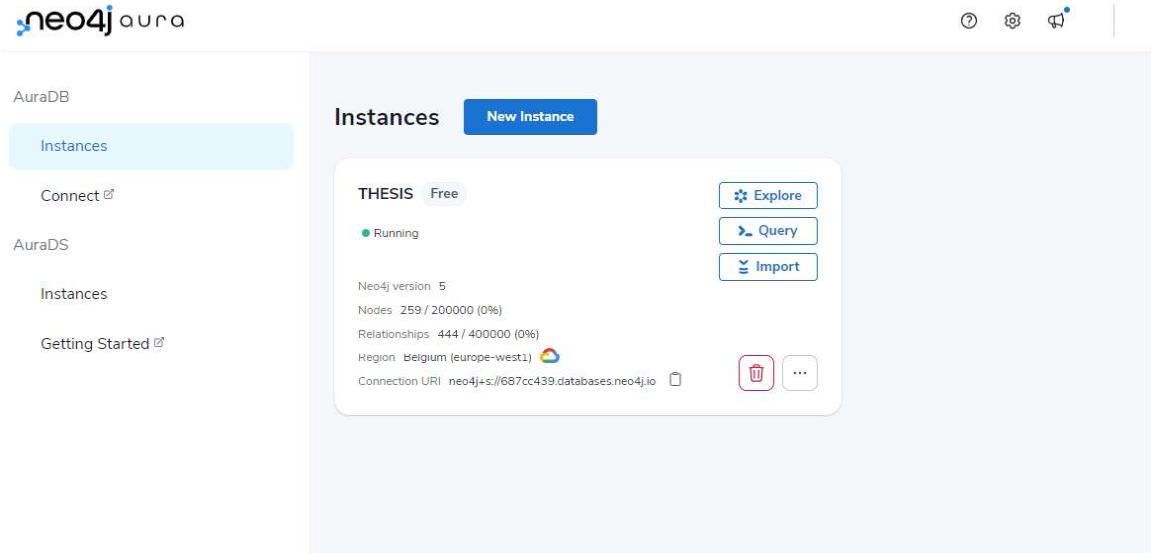
4.1.4 Υλοποίηση του αλγόριθμου

Δοκιμάσαμε διάφορες παραμέτρους που θα μπορούσαν να χρησιμοποιηθούν, ώστε να προταθούν στον χρήστη, κατά το δυνατόν, τα πιο χρήσιμα γι' αυτόν μαθήματα. Συγκεκριμένα, πειραματιστήκαμε με τον αριθμό των θεάσεων (nb-views attribute), minimum κριτήριο ομοιότητας (similarity factor), ομοιότητα των προτεινόμενων μαθημάτων με την εργασία του χρήστη κτλ. Επίσης κάναμε αρκετές δοκιμές με διαφορετικούς χρήστες για τους οποίους ήταν διαθέσιμος διαφορετικός αριθμός

δεδομένων και οι οποίοι είχαν κάνει λίγες ή περισσότερες αξιολογήσεις, για να δούμε την ευαισθησία του αλγόριθμού μας, σε χρήστες με διαφορετικό προφίλ.

4.2 Περιγραφή της διαδικασίας εισαγωγής δεδομένων στην Neo4j

Για την δημιουργια του γράφου μας χρησιμοποιούμε την γραφική βάση δεδομένων Neo4j και πιο συγκεκριμένα την εκδοχή της στο cloud την Neo4j Aura (Εικόνα 2).

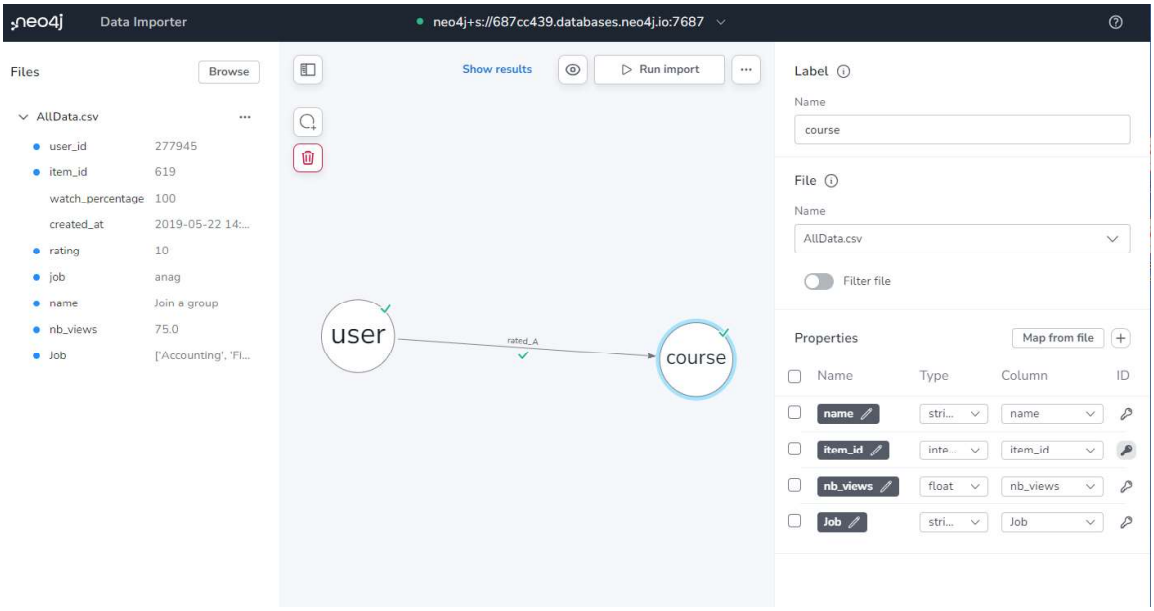


Εικόνα 2: Neo4j Aura Instance

Για να εισάγουμε τα δεδομένα μας στο γράφο, αξιοποιήσαμε το σχετικό και ειδικό γι' αυτό το σκοπό εργαλείο Data Importer (Εικόνα 3), που δημιουργήθηκε από τους ίδιους τους προγραμματιστές της Neo4j προκειμένου να διευκολύνει την εισαγωγή των δεδομένων που βρίσκονται σε μορφή π.χ. ενός αρχείου .csv όπως τα δικά μας, στον Neo4j γράφο, χωρίς να χρειάζεται κανείς να γράψει κώδικα στην γλώσσα της Neo4j, την Cypher. Το εργαλείο αυτό διευκολύνει πράγματι ουσιαστικά και εύκολα, τόσο στην επιλογή και εισαγωγή των δεδομένων μας, καθώς και στην δημιουργία του μοντέλου μας, όπου καθορίζουμε τις συσχετίσεις μεταξύ τους. Επιπλέον με πολύ εύκολο τρόπο, μπορούμε να διαλέξουμε ποια από τα πεδία των δεδομένων μας θα κρατήσουμε -ως

attributes- για τον κάθε κόμβο του γράφου. Επίσης μπορούμε να κάνουμε και ένα preview του γράφου, ώστε να δούμε αν πράγματι είναι αυτό που έχουμε σχεδιάσει, πριν από την καθ' εαυτό δημιουργία του.

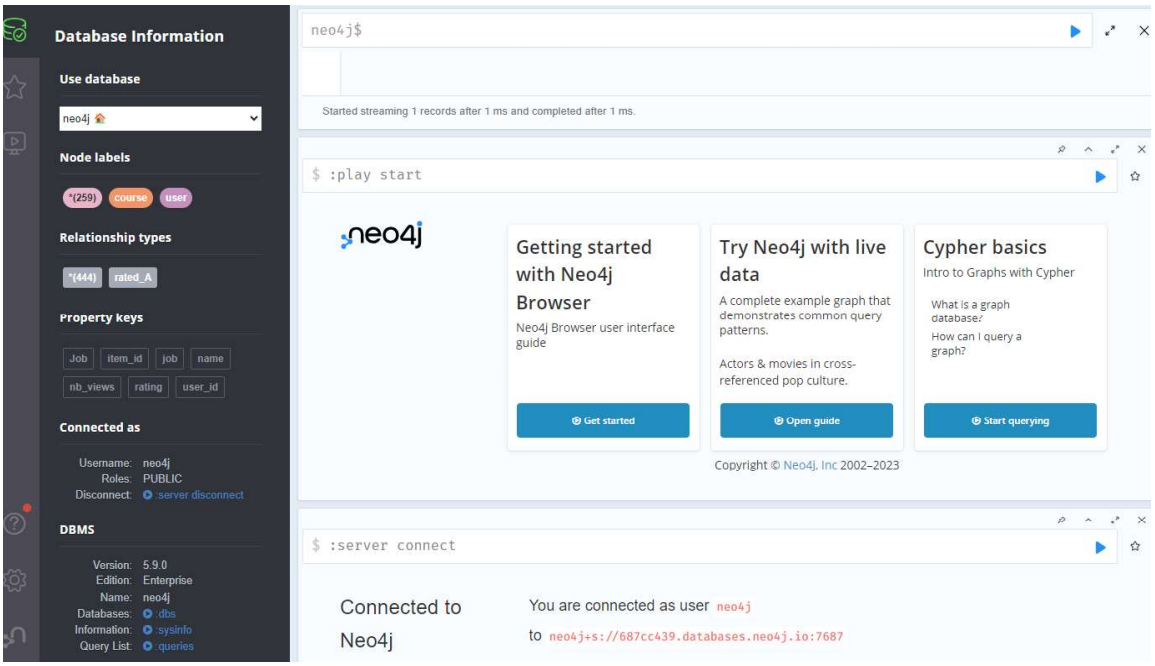
Τέλος, αν μείνουμε ικανοποιημένοι από την προεπισκόπηση, κάνοντας μία απλή επιλογή, το εργαλείο αυτό μας δημιουργεί τον γράφο, φορτώνοντας όλα τα δεδομένα σύμφωνα με το μοντέλο που έχουμε ορίσει.



Εικόνα 3: Απεικόνιση του dashboard του Data Importer tool

Με την εφαρμογή αυτή, δημιουργήσαμε επίσης και το μοντέλο του γράφου μας, πριν εισάγουμε τα δεδομένα μας στον Neo4j Browser (Εικόνα 4), που είναι η διεπαφή στην οποία φορτώνεται η βάση δεδομένων καθώς και το μοντέλο του γράφου και στην συνέχεια διατυπώνονται τα ερωτήματα (queries), στην γλώσσα της Neo4j, την Cypher.

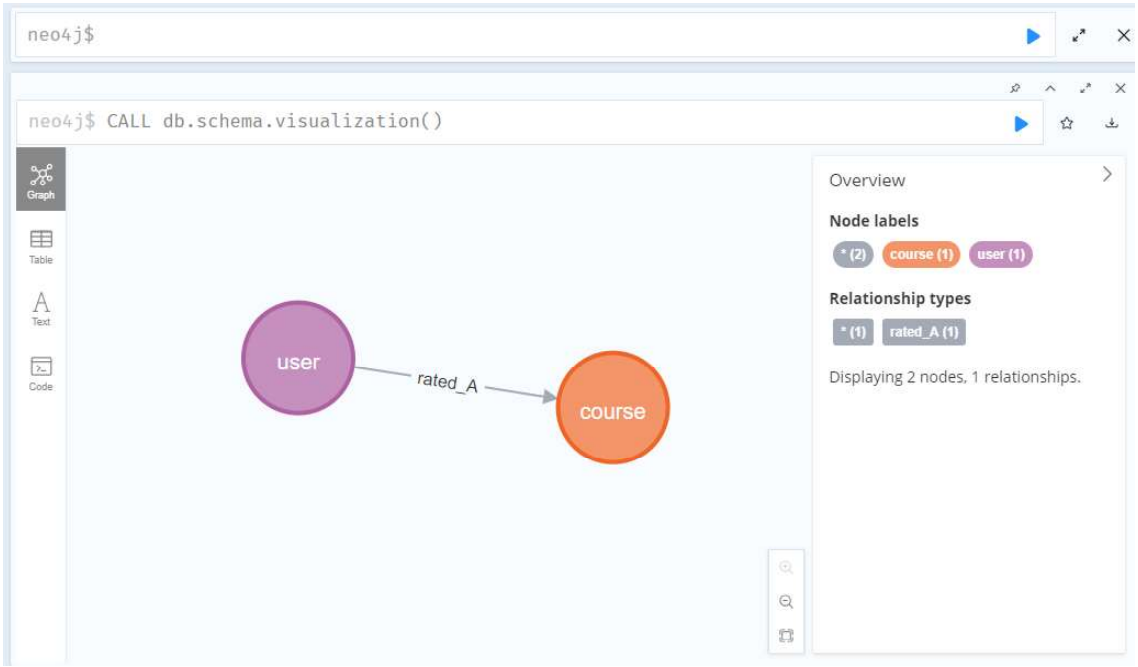
Στην διεπαφή αυτή γράφουμε τα ερωτήματα (queries) και παίρνουμε τα αποτελέσματα. Επίσης μπορούμε να ελέγξουμε τα δεδομένα που είναι φορτωμένα στον γράφο, τους κόμβους και τις σχέσεις που έχουν δημιουργηθεί, να απεικονίσουμε σχηματικά τον γράφο, καθώς επίσης και να δούμε στατιστικά στοιχεία του, να αλλάξουμε την απεικόνισή του, - χρώματα κτλ. Τέλος υπάρχουν και ενσωματωμένα tutorials για την εξοικείωση του νέου χρήστη.



Εικόνα 4: Εισαγωγική εικόνα του Neo4j Aura Browser

4.3 Παρουσίαση του μοντέλου Γράφου

Το *αρχικό* μοντέλο, στο οποίο καταλήξαμε προκειμένου πάνω του να στήσουμε το ΣΣ , είναι απλό και απεικονίζεται στην παρακάτω Εικόνα 5:



Εικόνα 5: Εικόνα του μοντέλου του γράφου μας, όπου απεικονίζονται οι κόμβοι (2) καθώς και οι σχέση που τους ενώνει.

Δημιουργήσαμε τους παρακάτω κόμβους και σχέσεις

(user)- [:rated_A]-> (course)

Ουσιαστικά δημιουργήσαμε την βασική συσχέτιση που ενώνει τις δύο οντότητες, στη περίπτωσή μας, των χρηστών και των μαθημάτων. Βέβαια, "πίσω" από κάθε κόμβο, αλλά και από την σχέση, υπάρχουν και ιδιότητες (attributes), που μπορούν να αξιοποιηθούν στα queries που μπορούμε να διαμορφώσουμε.

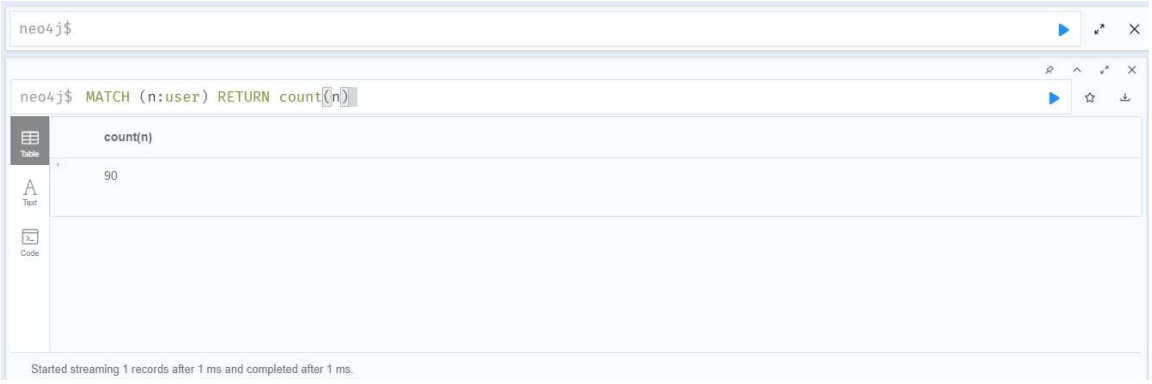
4.4. Παρουσίαση των metaData των δεδομένων του γράφου

Στην παράγραφο αυτή, θα παρουσιαστούν στοιχεία από το γράφο που δημιουργήσαμε, ώστε να έχουμε μία συνολική, απολογιστική εικόνα του .

Ζητήσαμε κατ' αρχήν πόσους κόμβους course, συνολικά, έχει ο γράφος που δημιουργήσαμε. Αποτέλεσμα : 169 (Εικόνα 6).

Εικόνα 6: Συνολικός αριθμός κόμβων του γράφου μας

Έπειτα ρωτήσαμε τον αριθμό των users, με αποτέλεσμα 90 (Εικόνα 7).



Εικόνα 7: Συνολικός αριθμός κόμβων users

Τέλος, αναζητήσαμε τον αριθμό των σχέσεων rated_A και πήραμε το αποτέλεσμα 888 (Εικόνα 8).

Εικόνα 8: Συνολικός αριθμός σχέσεων rated_A

4.5 Σκεπτικό δημιουργίας του ΣΣ και υλοποίηση

Το ΣΣ μας βασίστηκε στην τεχνική του Συνεργατικού φιλτραρίσματος (ΣΦ, Collaborative Filtering), καθώς από την βιβλιογραφική έρευνα που διενεργήσαμε, καταλήξαμε ότι είναι το πιο χρησιμοποιούμενο, μια που έχει το πλεονέκτημα των πιο

πρωτότυπων συστάσεων, καθώς επίσης έχει αποδειχθεί και ως το πιο αποτελεσματικό.

Επίσης στην απόφαση αυτή καταλήξαμε και από το γεγονός ότι τα αρχεία μας, στην πηγή δεδομένων που χρησιμοποιήσαμε, μας έδιναν αυτήν την δυνατότητα, μια που μπορούμε να συσχετίσουμε τους χρήστες με τα μαθήματα, μέσω της σχέσης "αξιολόγηση"/ratings. Τέλος, η μέθοδος αυτή κρίναμε ότι ήταν εφικτή, μια και είχαμε μεγάλο αριθμό χρηστών, μαθημάτων και αξιολογήσεων, γεγονός που μας επιτρέπει να δημιουργήσουμε ένα αποτελεσματικό ΣΣ.

Το ΣΣ μας βασίστηκε τόσο σε ένα user-based, όσο και σε item-based συνεργατικό φιλτράρισμα, προκειμένου τα αποτελέσματα να έχουν μεγαλύτερη ποικιλία, καθώς επίσης και για να ξεπεράσουμε το cold-start, καθώς και το πρόβλημα αραιότητας των δεδομένων, για κάποιους χρήστες με λίγες αξιολογήσεις. Συγκεκριμένα προτείνουμε δύο αλγόριθμους, έναν user-based και έναν item-based , για τον ίδιο χρήστη και έπειτα ενώνουμε τα δύο αποτελέσματα συστάσεων.

Προκειμένου να εξευρεθεί η ομοιότητα μεταξύ των χρηστών που έχουν κοινές συμπεριφορές- παρακολούθησαν στο παρελθόν κοινά μαθήματα και έδωσαν παραπλήσιες αξιολογήσεις- εφαρμόσαμε έναν αλγόριθμο Ομοιότητας Συνημιτόνου (Cosine Similarity), και δημιουργήθηκαν έτσι στον γράφο μας, νέες επιπλέον σχέσεις μεταξύ των χρηστών, που βασίζονται στην σχέση [:SIMILARITY].

Τέλος, προκειμένου να βελτιώσουμε ακόμα περισσότερα την ποιότητα των αποτελεσμάτων καθώς και και την χρησιμότητά τους για τον ενδιαφερόμενο χρήστη, προσθέσαμε στον αλγόριθμό μας και επιπλέον κριτήρια επιλογής που αφορούν στην δημοτικότητα (popularity) των μαθημάτων, καθώς και στην ταύτιση του επαγγέλματος του χρήστη, με τα επαγγελματικά αντικείμενα στα οποία αφορά το προτεινόμενο μάθημα.

4.6 Παρουσίαση αποτελεσμάτων χρησιμοποιώντας ενδεικτικό χρήστη

Στην παράγραφο αυτή παρουσιάζουμε τους δύο αλγόριθμους συστάσεων στους οποίους καταλήξαμε, ο ένας βασίζεται στο user-based συνεργατικό φιλτράρισμα, ενώ ο δεύτερος στο item-based συνεργατικό φιλτράρισμα.

4.6.1 Ερώτημα/Query 1: Αλγόριθμος user- based συνεργατικού φιλτραρίσματος (UB-CFA)

Το παρακάτω είναι ένα παράδειγμα εφαρμογής. Το ερώτημα μας έχει δύο διακριτά υποερωτήματα, συγκεκριμένα, αρχικά και προκειμένου να είναι πιο γρήγορη η απόκριση των ερωτημάτων μας, τρέχουμε τον παρακάτω αλγόριθμο/ query, που δημιουργεί συσχετίσεις μεταξύ των κόμβων user, δημιουργώντας μεταξύ τους, μια καινούργια σχέση την SIMILARITYUSER, με attribute, το χαρακτηριστικό similarity_value, που παίρνει τιμή από το αποτέλεσμα του Cosine Similarity Algorithm, όπως αυτός εφαρμόζεται για κάθε ζεύγος user που έχουν κάνει αξιολόγηση για το ίδιο course. Ο σχετικός αλγόριθμος/query είναι ο παρακάτω:

```
MATCH (u1:user)-[r1:rated_A]->(c1:course)<-[r2:rated_A]-(u2:user)
where u1<>u2
with SUM(tofloat(r1.rating)*tofloat(r2.rating)) as rating, count (r1) as ratings,
SQRT(Reduce(int1=0.0, x in collect(toFloat(r1.rating))| int1+x^2)) as length1,
SQRT(Reduce(int2=0.0, y in collect(toFloat(r2.rating))| int2+y^2)) as length2,
u1,u2
MERGE
(u1)-[S:SIMILARITYUSER{similarity_value:rating/(length1*length2)}]-(u2)
RETURN *
```

Από την ανάλυση των μεταδεδομένων, διαπιστώνουμε όπως φαίνεται και από την Εικόνα 10, ότι με την εφαρμογή του παραπάνω αλγορίθμου, δημιουργήθηκαν και νέοι δεσμοί/σχέσεις μεταξύ των οντοτήτων/κόμβων του γράφου μας. Πιο συγκεκριμένα , τώρα μεταξύ των κόμβων “user” και “course” δημιουργήθηκε και μία νέα σχέση, η [:SIMILARITYUSER].

Στη συνέχεια, χρησιμοποιούμε τον User Based Collaborative Filtering αλγόριθμο, πάνω στον τυχαίο χρήστη , με την γενική ονομασία “candidate”, για τον οποίο κατά την εφαρμογή του ερωτήματος, θα μας ζητηθεί να προσδιορίσουμε τον κωδικό του- και άρα να τον συγκεκριμενοποιήσουμε, για κάθε ξεχωριστή περίπτωση π.χ. 307569:

MATCH(u1:user{user_id:$candidate})-[r:SIMILARITYUSER]-(u2:user)-[q:SIMILARITYUSER]-(u3:user)-[z:rated_A]-(c) WHERE NOT EXISTS ((u1)-[:rated_A]->(c)) AND (c.nb_views)>200 AND c.Job CONTAINS u1.job AND u1<>u2<>u3<>u1

*WITH *, u1,u3, r.similarity_value as Similarity1*

*WITH *, q.similarity_value as Similarity2*

*WITH *, round((tofloat(Similarity1+Similarity2)),2) as finalSimilarity*

*WITH *, finalSimilarity, c.name AS RecommendedCourseName2, u3, z.rating AS rate*

ORDER BY finalSimilarity DESC

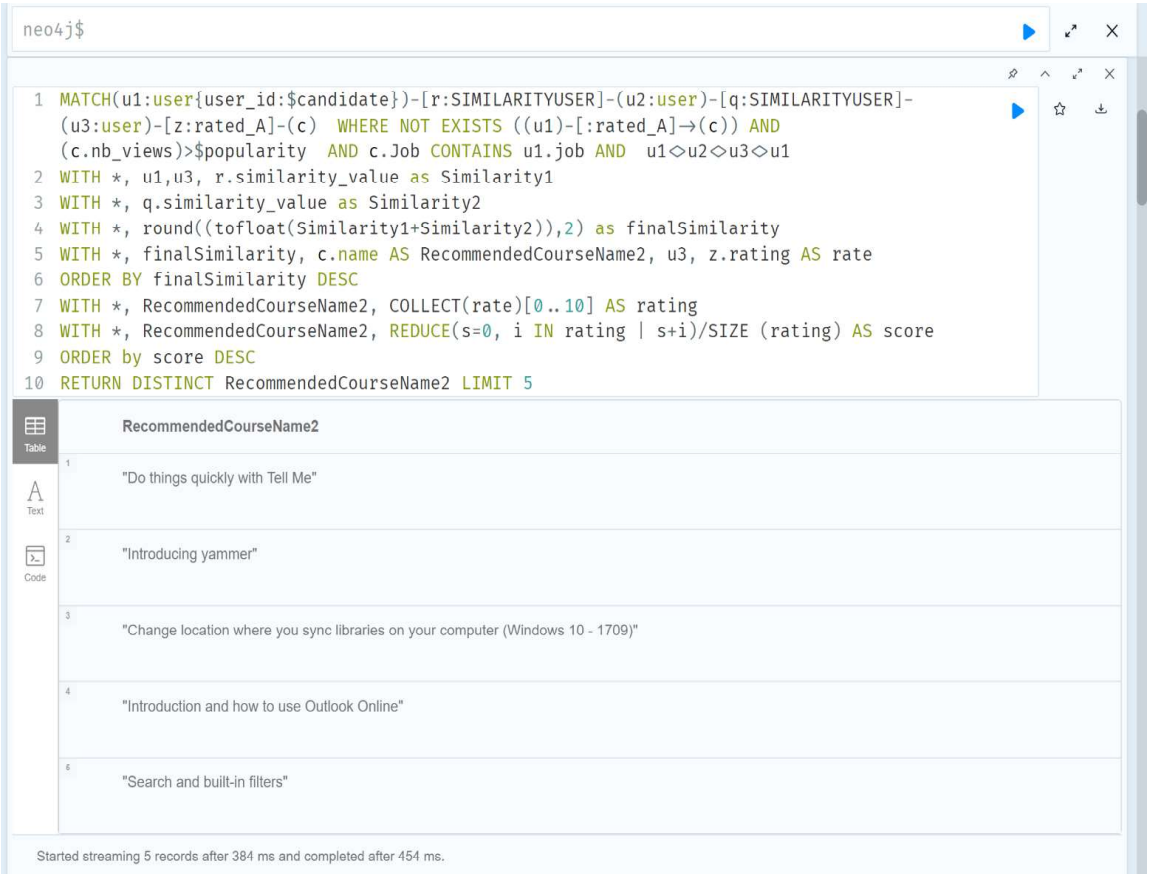
*WITH *, RecommendedCourseName2, COLLECT(rate)[0..10] AS rating*

*WITH *, RecommendedCourseName2, REDUCE(s=0, i IN rating | s+i)/SIZE (rating) AS score*

ORDER by score DESC

RETURN DISTINCT RecommendedCourseName2 LIMIT 5

Το printscreen από τον Neo4j Browser στην περίπτωση αυτή, απεικονίζεται στην παρακάτω Εικόνα 9, όπου φαίνεται το ερώτημα (query) που κάναμε στον γράφο , καθώς και το αποτέλεσμα του.



Εικόνα 9: Printscreen από το πρώτο Query , που βασίζεται στον UB-CFA

Συγκεκριμένα στο ερώτημα μας, ζητήσαμε να βρεθούν οι κοντινοί στις προτιμήσεις τους χρήστες, με κριτήριο την τιμή που έχει η παράμετρος similarity_value της σχέσης SIMILARITYUSER που δημιουργήσαμε νωρίτερα και έπειτα αναζητήσαμε τα courses που αυτοί έχουν παρακολουθήσει με σκοπό να τα προτείνουμε ως σύσταση στον ενδιαφερόμενο, αρχικό, χρήστη μας. Προκειμένου οι συστάσεις μας να έχουν και τον παράγοντα πρωτοτυπία , δλδ να μην είναι πολύ προφανείς και πολύ κοντά σε αυτά τα μαθήματα που πιθανόν έχει ήδη παρακολουθήσει, καινοτομήσαμε συνδυάζοντας και τη λογική των αλγορίθμων path finding και “ρυθμίζοντας” τον αλγόριθμό μας, ώστε να παίρνει ιδέες για συστάσεις, όχι από τους απολύτως κοντινούς στον ενδιαφερόμενο μας, χρήστες, αλλά στους χρήστες που έχουν παραπλήσιες προτιμήσεις με τους τελευταίους. Έτσι θέλαμε να δώσουμε “φρέσκες” προτάσεις/ συστάσεις, που παραμένουν φυσικά κοντινές με τις προτιμήσεις του χρήστη μας, αλλά έχουν και ένα παράγοντα έκπληξης, ώστε να προκαλέσουμε το ενδιαφέρον του. Για να το πετύχουμε αυτό είναι προφανές ότι

χρησιμοποιήσαμε τις δυνατότητες εύκολης διάβασης / traversal που μας δίνει μία βάση γράφου όπως η Neo4j.

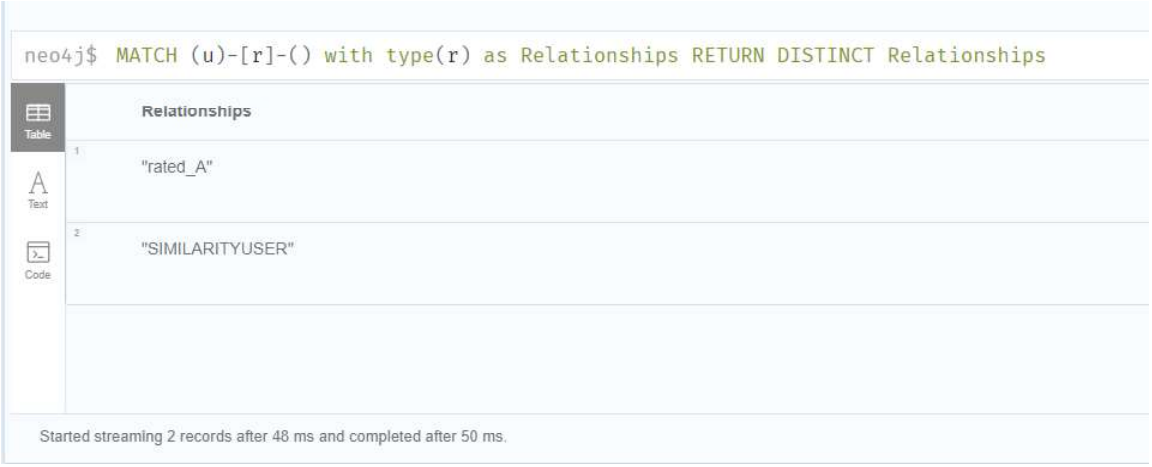
Ακόμα, πριν κάνουμε την σύστασή μας, εφαρμόσαμε επίσης και κάποια επιπλέον κριτήρια, αξιοποιώντας τις δυνατότητες που μας δίνει η βάση δεδομένων σε μορφή γράφου, όπως ο αριθμός των θεάσεων του κάθε course καθώς και το πεδίο το οποίο αφορά, ώστε αυτό να ταυτίζεται με το επάγγελμα του ενδιαφερόμενου χρήστη, προκειμένου οι συστάσεις μας να είναι όσο το δυνατόν πιο ενδιαφέρουσες.

Ο παραπάνω αλγόριθμος που δημιουργήσαμε, φροντίσαμε να έχει και δυνατότητα για ρυθμίσεις ώστε να προσαρμόζεται καλύτερα ανάλογα με την περίπτωση. Στο παραπάνω query για παράδειγμα, *μπορούμε να δώσουμε σαν παραμέτρο (την ονομάσαμε popularity στον αλγόριθμο)*, minimum αριθμό θεάσεων των course που θα μας προτείνει -στο παραπάνω παράδειγμα δώσαμε τις 200-, ώστε να μας επιλέξει τα course που θα προτείνει, συμπεριλαμβάνοντας και ένα ακόμα κριτήριο, αυτό της δημοφιλίας των μαθημάτων.

Τέλος , τα αποτελέσματα που παράγει ο αλγόριθμος, λαμβάνουν υπόψη το επάγγελμα του χρήστη και το συσχετίζουν με το αντικείμενο του course, πριν το προτείνουν, ώστε να ενδιαφέρει και να είναι όσο πιο χρήσιμο γίνεται για τον χρήστη.

Το “προϊόν” / αποτέλεσμα του ερωτήματος αυτού, είναι η σύσταση 5 μαθημάτων, ταξινομημένα από τα πιο κοντινά στον χρήστη προς τα πιο μακρινά από τα ενδιαφέροντά του, όπως φαίνονται στην Εικόνα 9 (“Do things quickly with Tell, me”, “Introducing Yammer” κτλ), που, με όλο το παραπάνω σκεπτικό, είναι πολύ πιθανό να βρει ο χρήστης μας αξιόλογα και χρήσιμα.

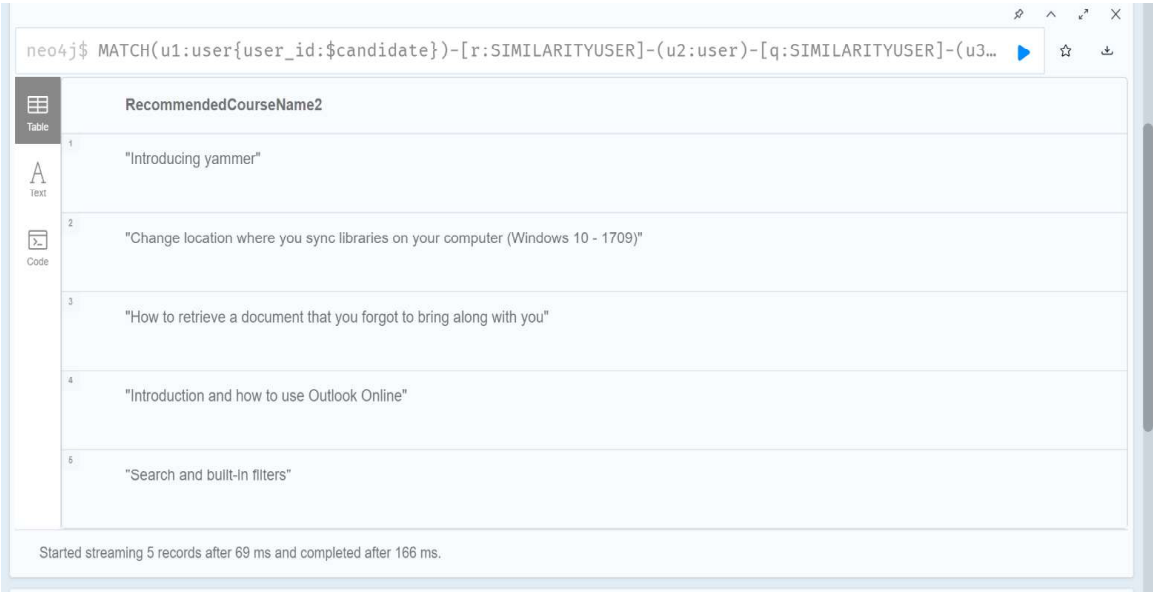
Από την ανάλυση των μεταδεδομένων, διαπιστώνουμε επίσης από την Εικόνα 10, ότι με την εφαρμογή του παραπάνω αλγορίθμου, δημιουργήθηκαν και νέοι δεσμοί/σχέσεις μεταξύ των οντοτήτων/κόμβων του γράφου μας. Πιο συγκεκριμένα , τώρα μεταξύ των κόμβων “user” και “course” δημιουργήθηκε και μία νέα σχέση, η [:SIMILARITYUSER].



Εικόνα 10: Απεικόνιση των σχέσεων του γράφου μας, μετά και από τις καινούργιες σχέσεις ομοιότητας που δημιουργήσαμε

Από την παραπάνω διαδικασία και αφού πρώτα ορίσαμε, όπως μας ζητήθηκε μόλις τρέξαμε το query μας, βλ. Εικόνα 11- ως τιμή της παραμέτρου candidate και της παραμετρου popularity σε 307569 και 200 αντίστοιχα, πήραμε 5 προτάσεις για τον χρήστη 307569, τις παρακάτω, όπως φαίνονται στην Εικόνα 12:

Εικόνα 11: ***Ορισμός των παραμέτρων candidate και popularity σε 307569 kai 200 αντίστοιχα, στην οθόνη που εμφανίζεται αμέσως μόλις τρέξουμε το query μας , για πρώτη φορά***



Εικόνα 12: ***Συστάσεις για τον χρήστη 307569, ως παράδειγμα εφαρμογής του (UB-CFA)***

4.6.3 Αλγόριθμος item- based συνεργατικού φιλτραρίσματος (ITB-CFA)

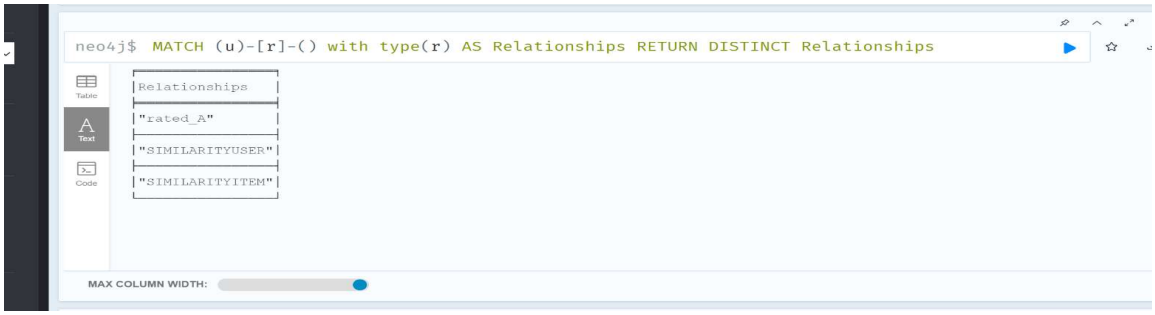
Όπως είπαμε και παραπάνω, προκειμένου να δώσουμε στον ενδιαφερόμενο χρήστη και άλλες επιλογές, με μία ανεξάρτητη, διαφορετική προσέγγιση, ώστε να καλύψουμε κατά το δυνατόν πληρέστερα τα ενδιαφέροντα και τις ανάγκες επιμορφώσεώς του, η διαδικασία μας ακολουθεί μετά την χρήση του UB-CFA και την εφαρμογή πάνω στα ίδια δεδομένα, για τον ίδιο χρήστη και ενός δεύτερου αλγορίθμου ΣΣ, που βασίζεται αυτήν την φορά στο item based συνεργατικό φιλτράρισμα, του ITB-CFA.

Τον αλγόριθμο αυτόν, απεικονίζουμε παρακάτω, με τα αποτελέσματα εφαρμογής του. Ο συγκεκριμένος αλγόριθμος έχει δύο κομμάτια: στο πρώτο κομμάτι που απεικονίζεται παρακάτω, συγκρίνονται τα courses που έχει ήδη παρακολουθήσει ο χρήστης μας, με τα υπόλοιπα -*διαφορετικά*- courses που έχουν παρακολουθήσει οι άλλοι χρήστες που έχουν παρακολουθήσει επιπλέον και τα courses που έχει παρακολουθήσει ο

χρήστης μας. Η σύγκριση αυτή γίνεται στην βάση των αξιολογήσεων που έχουν δώσει ο χρήστης μας, καθώς και οι άλλοι χρήστες, στα courses που έχουν παρακολουθήσει. Με τον τρόπο αυτό δημιουργείται στον γράφο μας, μία καινούργια σχέση *μεταξύ των courses,* την οποία την ονομάσαμε [:SIMILARITYITEM].

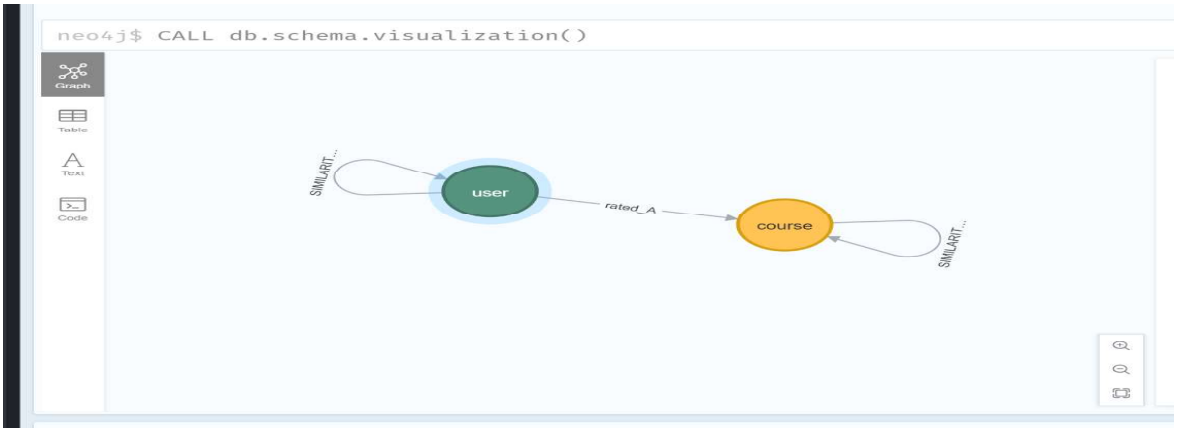
MATCH(c1:course), (c2:course) where c1<>c2
MATCH (c1)<-[r:rated_A]-(u:user)
WITH AVG(toFloat(r.rating)) as c1rate,c1,c2
MATCH (c2)<-[r:rated_A]-(u:user)
WITH AVG (toFloat(r.rating)) as c2rate,c1,c2,c1rate
MATCH (c1)<-[r1:rated_A]-(u:user)-[r2:rated_A]- (c2)
WITH SUM((toFloat(r1.rating)-c1rate)(toFloat(r2.rating)-c2rate)) as sumc1c2,*
SQRT(SUM((toFloat(r1.rating)-c1rate)^2) SUM((toFloat(r2.rating)-c2rate)^2)) AS sqrtc1c2,*
c1,c2, COUNT (r1) as countrelation
WHERE sqrtc1c2<>0 and countrelation>1
MERGE (c1)-[q:SIMILARITYITEM{similarity_item:(sumc1c2/sqrtc1c2)}]-(c2)
*return**

Μπορούμε να ελέγξουμε την δημιουργία αυτής της σχέσης, κάνοντας το κατάλληλο query στον γράφο μας και το αποτέλεσμα είναι το παρακάτω (Εικόνα 13):



Εικόνα 13: Απεικόνιση των σχέσεων του γράφου συμπεριαλμβανόμενης της SIMILARITYITEM

Μετά από την παραπάνω τροποποίηση των σχέσεων μεταξύ των κόμβων του γράφου μας, το μοντέλο συσχέτισης των δεδομένων μας, απεικονίζεται πλέον όπως παρακάτω φαίνεται στην Εικόνα 14.



Εικόνα 14: Απεικόνιση του πλήρους μοντέλου

Τέλος, πάνω σε αυτόν πλέον τον γράφο και πάνω στο παραπάνω μοντέλο, εφαρμόζουμε το δεύτερο μέρος του ITB-CFA αλγόριθμού μας, συγκεκριμένα για έναν μεμονωμένο χρήστη, τον οποίο στον αλγόριθμο τον περάσαμε με την μορφή παραμέτρου με το όνομα candidate. Καθώς τρέχει κανείς τον αλγόριθμο, θα του ζητηθεί να ορίσει τον συγκεκριμένο χρήστη για τον οποίο θέλει να γίνουν οι συστάσεις, καθώς επίσης και τον ελάχιστο βαθμό δημοφιλίας/θεάσεων των προτεινόμενων μαθημάτων θέλει να χρησιμοποιήσει ο αλγόριθμος συστάσεων .

MATCH(u:user
{user_id:$candidate})-[r:rated_A]-(c1:course)-[s:SIMILARITYITEM]->(c2:course)-[z:
SIMILARITYITEM]->(c3:course) WHERE c3.nb_views>$popularity AND c3.Job
CONTAINS u.job

WITH *, s.similarity_item as Similarity1

WITH *, z.similarity_item as Similarity2

WITH *, (tofloat(Similarity1+Similarity2)) as finalItemSimilarity

WITH c3, round(finalItemSimilarity,2) as Similarity

ORDER by Similarity DESC LIMIT 5

WHERE Similarity>0.8

return DISTINCT c3.name

Στο παράδειγμα μας, ζητάμε συστάσεις για τον χρήστη 307569 και minimum αριθμό δημοφιλίας των courses που θα του προτείνουμε τις 200 θεάσεις, οπότε παίρνουμε το παρακάτω αποτέλεσμα (Εικόνα 15), από τον Neo4j Browser:

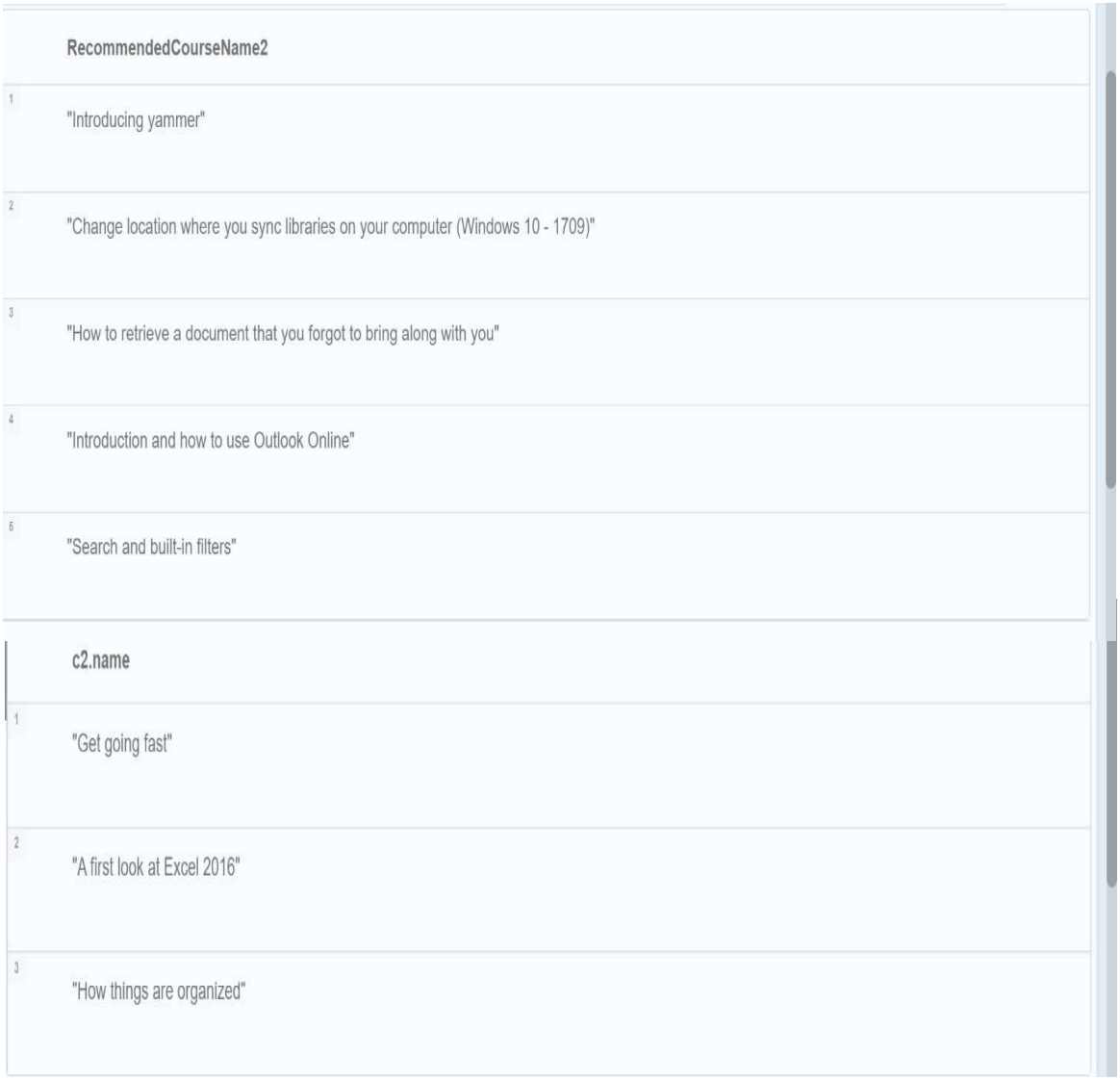
Εικόνα 15: Αποτελέσματα από την εφαρμογή του ITB-CF για τον χρήστη 307569 και minimum αρ. θεάσεων 200

Στον παραπάνω αλγόριθμο, αυτό που κάνουμε είναι να πάμε στους κόμβους course , **δύο βήματα παραπέρα ,** μέσω της σχέσης ομοιότητας SIMILARITYITEM που έχουμε ήδη δημιουργήσει, από τα courses που έχει ήδη παρακολουθήσει και αξιολογήσει ο χρήστης μας, προς παραπλήσια courses. Προτιμήσαμε να πάμε δύο και όχι ένα βήμα παραπέρα, προκειμένου να αυξήσουμε την ποικιλία και την πρωτοτυπία στις συστάσεις μας. Αυτό γίνεται πολύ εύκολα σε μία βάση δεδομένων γράφου, όπως η Neo4j που χρησιμοποιήσαμε, σε αντίθεση με μία σχεσιακή βάση όπου θα ήταν πιο πολύπλοκο.

Επίσης προσθέσαμε και δύο παραπάνω κριτήρια, στην επιλογή των μαθημάτων που θα προτείνουμε, ώστε οι συστάσεις μας να είναι όσο γίνεται πιο χρήσιμες για τον χρήστη. Συγκεκριμένα χρησιμοποιήσαμε την δημοτικότητα των courses, όπως αυτή φαίνεται από τον αριθμό των θεάσεων, αξιοποιώντας το attribute των κόμβων user, **nb_views**, τον οποίο αναπαριστούμε με την παράμετρο popularity, την οποία την ορίζουμε κατά βούληση, την πρώτη φορά που θα τρέξει ο αλγόριθμος. Επιπλέον ορίσαμε σαν φίλτρο των προτεινόμενων courses το να σχετίζονται με το επάγγελμα του χρήστη, αξιοποιώντας πάλι attributes των κόμβων user **- u.job-** και course **- c.job-**,.

4.6.4 Τελικό αποτέλεσμα - σύσταση- για τον χρήστη του case study μας

Κατόπιν όλων των παραπάνω, η σύστασή μας προς τον χρήστη 307569, θα είναι ο συγκερασμός των συστάσεων, που προέκυψαν από τον κάθε ένα από τους δύο αλγόριθμους μας , επομένως:



	RecommendedCourseName2
1	"Introducing yammer"
2	"Change location where you sync libraries on your computer (Windows 10 - 1709)"
3	"How to retrieve a document that you forgot to bring along with you"
4	"Introduction and how to use Outlook Online"
5	"Search and built-in filters"

	c2.name
1	"Get going fast"
2	"A first look at Excel 2016"
3	"How things are organized"

Εικόνα 16: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CF και του ITB-CF για τον χρήστη 307569

Στην προκειμένη περίπτωση , βλέπουμε στην Εικόνα 16, ότι από την χρήση του USB-CF προέκυψαν 5 συστάσεις**,** ενώ από την χρήση του ITB-CF 3 συστάσεις, συνολικά δλδ 8 προτάσεις προς τον χρήστη. Σημειωτέον ότι μπορούμε να προτείνουμε και παραπάνω, αλλά όπως εξηγήσαμε στις προηγούμενες παραγράφους, θέσαμε κριτήρια ώστε τα courses που προτείνουμε να είναι όσο γίνεται πιο χρήσιμα για τον χρήστη.

4.7 Δοκιμές αποτελεσματικότητας και "ευαισθησίας" των αλγορίθμων σε διάφορους χρήστες

Προκειμένου να τεκμηριώσουμε τη λειτουργικότητα των αλγορίθμων μας, δεν μείναμε στην εφαρμογή τους σε ένα μόνο χρήστη, αλλά τους δοκιμάσαμε και σε άλλους, τυχαία επιλεγμένους, από το datasets μας, για να δούμε αν παράγουν αποτελέσματα. Επίσης, κάναμε δοκιμές για να δούμε και την ευαισθησία τους σε διαφορετικές τιμές της παραμέτρου popularity, συγκεκριμένα 100 και 50.

Διενεργήσαμε λοιπόν δοκιμές και για τους χρήστες 277945, 319053, 322262 και τα αποτελέσματα, σύμφωνα με την μεθοδολογία που αναπτύξαμε εκτενώς παραπάνω και για τον χρήστη του case study μας (του 307569), απεικονίζονται στις Εικόνες 17-22.

	RecommendedCourseName2
1	"Introducing yammer"
2	"Create a meeting within the calendar"
3	"Get going fast"
4	"Sending/sharing a large file to/with a contact"
5	"Sharing contacts"

	c2.name
1	"Get going fast"
2	"Getting an overall view of activity on project files and documents"
3	"Starting with planner"
4	"Discovering the interface"

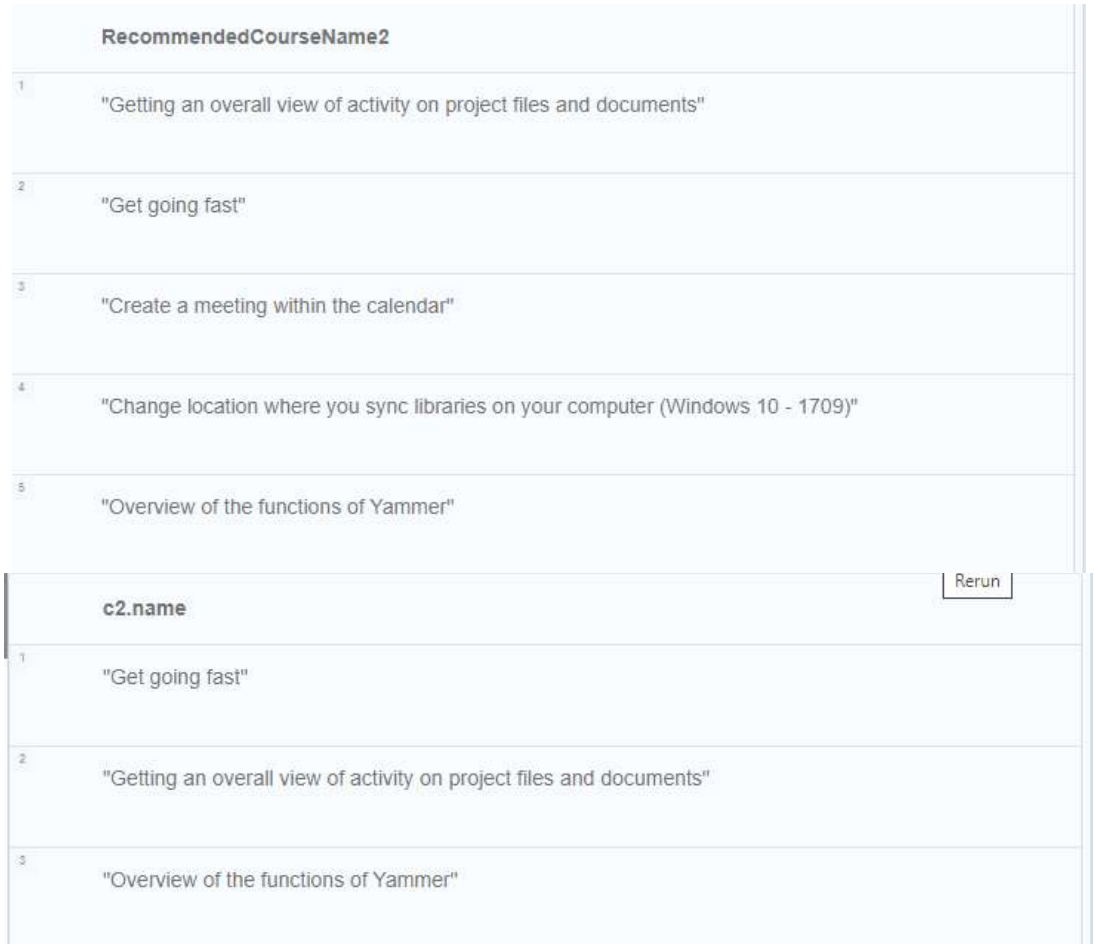
Εικόνα 17: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CFA και του ITB-CFA για τον χρήστη 277945 και popularity 100

	RecommendedCourseName2
1	"Introducing yammer"
2	"Get going fast"
3	"A first look at Excel 2016"
4	"How things are organized"
5	"Start using Excel"

	c2.name
1	"A first look at Excel 2016"
2	"Get going fast"
3	"How things are organized"
4	"Starting with planner"
5	"Getting an overall view of activity on project files and documents"

Εικόνα 18: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CFA και του ITB-CFA για τον χρήστη 277945 και popularity 50

Στη συγκεκριμένη περίπτωση, βλέπουμε ότι για τον χρήστη 277945, όταν θέσουμε το κριτήριο popularity, δλδ τον minimum αριθμό θεάσεων των προτεινόμενων courses, με τιμή 100, παίρνουμε **μία** λιγότερη σύσταση από τον ITB-CFA, και **συνολικά 9 προτεινόμενα courses από τους 2 αλγορίθμους , αντί για 10** όταν το κριτήριο δημοφιλίας το κάνουμε λιγότερο αυστηρο, δίνοντάς του την τιμή 50. Φαίνεται δλδ ο ITB-CFA, να είναι πιο "ευαίσθητος" σ' αυτόν τον παράγοντα.



	RecommendedCourseName2
1	"Getting an overall view of activity on project files and documents"
2	"Get going fast"
3	"Create a meeting within the calendar"
4	"Change location where you sync libraries on your computer (Windows 10 - 1709)"
5	"Overview of the functions of Yammer"

Rerun

	c2.name
1	"Get going fast"
2	"Getting an overall view of activity on project files and documents"
3	"Overview of the functions of Yammer"

Εικόνα 19: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CFA και του ITB-CFA για τον χρήστη 319053 και popularity 100

	RecommendedCourseName2
1	"Join a group"
2	"Create a meeting within the calendar"
3	"How things are organized"
4	"Top tips for working in Excel Online"
5	"Get going fast"

	c2.name
1	"A first look at Excel 2016"
2	"Get going fast"
3	"Getting an overall view of activity on project files and documents"
4	"How things are organized"
5	"Overview of the functions of Yammer"

Εικόνα 20: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CFA και του ITB-CFA για τον χρήστη 319053 και popularity 50

Και σ΄αυτήν την περίπτωση, βλέπουμε ότι η παράμετρος popularity επηρέασε τα αποτελέσματά μας, συγκεκριμένα εδώ μείωσε τον αριθμό και άλλαξε το αποτέλεσμα του ITB-CFA, ο οποίος έδωσε μόνο 3 courses ως προτεινόμενα στην εφαρμογή του αλγορίθμου για popularity = 100, σε σχέση με την εφαρμογή του για popularity=50, και μάλιστα διαφορετικά, γεγονός που δείχνει για μία ακόμη φορά ότι ο συγκεκριμένος αλγοριθμος είναι αρκετά “ευαίσθητος” στο κριτήριο/ φίλτρο των αποτελεσμάτων, “δημοφιλία” των courses.

	RecommendedCourseName2
1	"Getting an overall view of activity on project files and documents"
2	"Get going fast"
3	"Create a PivotTable and analyze your data"
4	"Change location where you sync libraries on your computer (Windows 10 - 1709)"
5	"Create a meeting within the calendar"

	c2.name
1	"Getting an overall view of activity on project files and documents"
2	"Get going fast"

Εικόνα 21: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CFA και του ITB-CFA για τον χρήστη 322262 και popularity 100

	RecommendedCourseName2
1	"Join a group"
2	"Share documents"
3	"Use slicers, timelines and PivotCharts to analyze your pivotetable data"
4	"Top tips for working in Excel Online"
5	"Sort, filter, summarize and calculate your PivoteTable data"

	c2.name
1	"Share documents"
2	"A first look at Excel 2016"
3	"Getting an overall view of activity on project files and documents"
4	"How things are organized"
5	"Get going fast"

ted streaming 5 records after 1 ms and completed after 2 ms.

Εικόνα 22: Τελικό αποτελέσματα/ συστάσεις από την εφαρμογή του USB-CFA και του ITB-CFA για τον χρήστη 322262 και popularity 50

Και στην προκειμένη περίπτωση του χρήστη 322262, βλέπουμε ότι η τιμή της παραμέτρου popularity, αλλάζει τα αποτελέσματα, τόσο ποιοτικά, όσο και ποσοτικά -παίρνουμε 7 συστάσεις με popularity = 100 και 10 με popularity=50 , μιας και καθώς αυστηροποιούμε το κριτήριο δημοφιλίας, δλδ μεγαλώνουμε την τιμή στην παράμετρο popularity, καταλήγουμε να "χάσουμε" κάποια αποτελέσματα, να πάρουμε επομένως λιγότερες συστάσεις, κάτι που φυσικά είναι και λογικό και αναμενόμενο.

Από την άλλη, πιθανότατα αυξάνουμε την χρησιμότητα των προτάσεων μας, αν φυσικά κάνουμε την παραδοχή ότι τα πιο δημοφιλή courses, υπάρχει λόγος που είναι πιο δημοφιλή.

Γενικά, πέρα από το παραπάνω συμπέρασμα, για την ευαισθησία του ITB-CFA στον παράγοντα δημοφιλία, βλέπουμε ότι και οι δύο αλγόριθμοι, κάνουν την δουλειά τους, δηλαδή παράγουν αποτέλεσματα/ συστάσεις στον κάθε ενδιαφερόμενο χρήστη.

5 Επίλογος

Στην παρούσα εργασία αξιοποιήσαμε την ΓΒΔ Neo4j, η οποία είναι αυτήν τη στιγμή, η επικρατούσα ΓΒΔ παγκοσμίως, προκειμένου να δημιουργήσουμε ένα ΣΣ που να στηρίζεται πάνω σε Γράφο Γνώσης (Knowledge Graph), δηλαδή πάνω στις ιδιότητες και κυρίως πάνω στις σχέσεις μεταξύ των οντοτήτων. Για την εφαρμογή του ΣΣ μας, αναζητήσαμε και βρήκαμε μια κατάλληλη πηγή δεδομένων όπου υπάρχουν επαρκή πραγματικά στοιχεία που αφορούν σε συμπεριφορές χρηστών που έκαναν χρήση της πλατφόρμας **MOOC.office365-training.com,** της Mandarine Academy, η οποία παρέχει σύντομα διαδικτυακά μαθήματα σε χρήστες-υπαλλήλους μεγάλων εταιρειών, προκειμένου να λάβουν υποστήριξη σε θέματα που αφορούν στην χρήση εφαρμογών και προγραμμάτων IT στην καθημερινή τους εργασία.

Αρχικά, δημιουργήσαμε το κατάλληλο μοντέλο, όπου απεικονίσαμε τις σχέσεις μεταξύ των οντοτήτων των χρηστών και των μαθημάτων και βάσει αυτού δημιουργήσαμε τον κατάλληλο Γράφο Γνώσης, όπου φορτώθηκαν όλα τα δεδομένα μας.

Στην συνέχεια κατασκευάσαμε δύο διαφορετικά ΣΣ που βασίζονται το πρώτο σε ένα Συνεργατικό Φιλτράρισμα τύπου User-Based και το δεύτερο σε ένα Συνεργατικό

Φιλτράρισμα τύπου Item Based, από όπου προέκυψαν δύο λίστες με μαθήματα προς σύσταση προς τον ενδιαφερόμενο χρήστη, από την μεγάλη γκάμα διαθεσίμων στην πλατφόρμα

5.1 Σύνοψη και συμπεράσματα

Κατά την διάρκεια σύνταξης της παρούσας διπλωματικής εργασίας κληθήκαμε να μελετήσουμε γενικά την όλη φιλοσοφια των ΣΣ, τα οποία τα συναντούμε όλο και περισσότερο πλέον σε πάρα πολλές εφαρμογές που χρησιμοποιούμε στη καθημερινη μας ζωή. Είναι συστήματα που προσφέρουν ευκολία και χρησιμότητα στον χρήστη καθώς και αυξημένα έσοδα στις εταιρείες που τα χρησιμοποιούν.

Εντρυφήσαμε στην μεθοδολογία και στο σκεπτικό που υπάρχει πίσω από τους εδραιωμένους τρόπους που χρησιμοποιούνται σήμερα προκειμένου να προσεγγίσουμε κατά το δυνατόν περισσότερο το προφίλ και τις ανάγκες του κάθε χρήστη.

Παράλληλα, είχαμε τη ευκαιρία να δούμε και να πειστούμε για την δύναμη, τις δυνατότητες και την ευελιξία που κρύβουν οι Βάσεις Δεδομένων Γράφου, γεγονός που τις καθιστούν πάρα πολύ χρήσιμες, ιδιαίτερα για την οργάνωση του τεράστιου όγκου αληλλοδιαπλεκόμενων δεδομένων που παράγονται καθημερινά στη σύγχρονη εποχή.

Πιστεύουμε ότι με την εργασία μας, βάλαμε κι εμείς ένα λιθαράκι στην ανάδειξη της ποιότητας, της ακρίβειας και της χρησιμότητας των ΣΣ, με σκοπό πάντοτε την καλύτερη εξυπηρέτηση του χρήστη των υπηρεσιών.

5.2 Όρια και περιορισμοί της έρευνας

Η παρούσα μελέτη αν και θεωρούμε ότι είναι χρήσιμη και αποτελεσματική, δεν αποτελεί παρά ένα μικρό μέρος της έρευνας που γίνεται για την δημιουργία όλο και καλύτερων και αποτελεσματικότερων ΣΣ**.**

Βασίστηκε, όπως περιγράψαμε σε μία αξιόλογη συλλογή πραγματικών δεδομένων, όμως δεν παύει η ανάγκη για όσο το δυνατόν περισσότερα και πιο πλήρη στοιχεία, προκειμένου να προκύψουν καλύτερες συστάσεις. Επομένως θα πρέπει ίσως να βελτιωθούν και να εξευρεθούν νέοι και πιο αποτελεσματικοί τρόποι έμμεσης συλλογής στοιχείων της συμπεριφοράς των χρηστών, που θα οδηγήσουν σε καλύτερες συστάσεις. Επίσης, θα πρέπει να μελετηθεί ίσως περισσότερο η επίδραση των συνθηκών στις οποίες κάθε φορά καλείται ένα ΣΣ να προτείνει τις πιο ενδιαφέρουσες επιλογές. Η παράμετρος αυτή απουσιάζει από την εργασία μας και πιθανώς να έκανε σημαντική διαφορά αν συμπεριλαμβάνονταν ως ένα επιπλέον φίλτρο στις συστάσεις μας.

5.3 Μελλοντικές Επεκτάσεις

Η μέθοδος συνεργατικού φιλτραρίσματος που χρησιμοποιούμε στην εργασία μας βασίζεται σε προτιμήσεις του χρήστη και των άλλων χρηστών στο παρελθόν. Επειδή όμως οι άνθρωποι με την πολυπλοκότητα της σκέψης που μας διακρίνει αλλάζουμε προτιμήσεις, ενώ επίσης και οι ανάγκες μας αλλάζουν, όπως και οι εκάστοτε περιστάσεις, σκεφτόμαστε ότι θα ήταν ενδιαφέρον να αναλυθούν περισσότερο οι ψυχολογικές και νοητικές παράμετροι με βάση τις οποίες κάποιος χρήστης παίρνει αποφάσεις, ώστε να επινοηθούν αλγόριθμοι που θα παράγουν αποτελέσματα πιο κοντά στις τρέχουσες και μελλοντικές ανάγκες του, με τρόπο πρωτότυπο και ευφάνταστο και όχι ανατρέχοντας μόνο στις προτιμήσεις του χρήστη στο παρελθόν.

Βιβλιογραφία

Παπαγεωργίου, Δ. (2021). *Κινητές Επιχειρηματικές Εφαρμογές Με Επίγνωση Πλαισίου (Context-Aware Mobile Commerce),* Μεταπτυχιακή Εργασία στο μάθημα Κινητό Επιχειρείν και Τεχνολογίες Ηλεκτρονικού Εμπορίου, Σχολή Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας

Robinson, I., Webber, J., & Eifrem, E. (2015). Graph databases: new opportunities for connected data. " O'Reilly Media, Inc.".

Wikipedia, Graph Databases. https://en.wikipedia.org/wiki/Graph_database. (July 6 2023)

ShefaliPatil, G., & Bhatia, A. (2014). Graph databases-an overview. 1Student, ME Computers, Terna College of Engg, Navi Mumbai, 2, 657-660.

Strauch, C., Sites, U. L. S., & Kriha, W. (2011). NoSQL databases. Lecture Notes, Stuttgart Media University, 20(24), 79.

What are NoSQL Databases? | IBM. https://www.ibm.com/topics/nosql-databases.

What Is NoSQL? NoSQL Databases Explained. https://www.mongodb.com/nosql-explained.

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., & Schiaffino, S. (2017). Persisting big-data: The NoSQL landscape. *Information Systems*, *63*, 1-23.

Guia, J., Soares, V. G., & Bernardino, J. (2017, April). Graph Databases: Neo4j Analysis. In ICEIS (1) (pp. 351-356).

Robinson, I., Webber, J., & Eifrem, E. (2013). Graph databases:" O'Reilly Media, Inc. Graph databases:" O'Reilly Media, Inc.

Tran, T. N. T., Felfernig, A., Trattner, C., & Holzinger, A. (2021). Recommender systems in the healthcare domain: state-of-the-art and research issues. Journal of Intelligent Information Systems, 57, 171-201.

Sarma, D., Mittra, T., & Hossain, M. S. (2021). Personalized book recommendation system using machine learning algorithm. International Journal of Advanced Computer Science and Applications, 12(1).

ABUZIR, Y., & DWIEB, M. (2021). Hotel Recommender System based on Knowledge Graph and Collaborative Approach. International Journal of Computing, 20(1), 63-71.

Estrela, D., Batista, S., Martinho, D., & Marreiros, G. (2017). A recommendation system for online courses. In Recent Advances in Information Systems and Technologies: Volume 1 5 (pp. 195-204). Springer International Publishing.

Khalid, A., Lundqvist, K., & Yates, A. (2020). Recommender systems for moocs: A systematic literature survey (January 1, 2012–July 12, 2019). International Review of Research in Open and Distributed Learning, 21(4), 255-291.

Chicaiza, J., Piedra, N., Lopez-Vargas, J., & Tovar-Caro, E. (2017, April). Recommendation of open educational resources. An approach based on linked open data. In 2017 IEEE Global Engineering Education Conference (EDUCON) (pp. 1316-1321). IEEE.

Hafsa, M., Wattebled, P., Jacques, J., & Jourdan, L. (2023). E-learning recommender system dataset. Data in Brief, 47, 108942.

Neo4j Graph Data Science / Graph algorithms, https://neo4j.com/docs/graph-data-science/current/algorithms/ (7 Ιουλίου 2023)

Hafsa, Mounir, (2022), "E-learning Recommender System Dataset", https://doi.org/10.7910/DVN/BMY3UD, Harvard Dataverse, V2, UNF:6:PhD+xVW2pdkKj4z7qz8dtQ== [fileUNF]

(i) Παράρτημα Α

ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΠΗΓΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ PYTHON- ΚΩΔΙΚΑΣ

PYTHON Data Preprocessing

```
#import pandas
import pandas as pd

# read Employee file
data1= pd.read_csv('explicit_ratings_en.csv')

# read Updated fi

data2= pd.read_csv('users_en.csv')
data3=data2.dropna()



mergeUsers = pd.merge(data1, data3,
                on='user_id',
                how='inner')

# displaying result

ratings=mergeUsers.replace({"job":{"accounting": "Accounting", "marketing": "Marketing","administrative": "anag", "consulting": "anag, Marketing, Human Resources, Financial"}})
print (ratings)
```