ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής
Πολυτεχνική Σχολή

Έρευνα και ανάπτυξη συστήματος για την ανίχνευση μεροληψίας σε συστήματα συστάσεων

Διπλωματική Εργασία

του

ΡΩΜΑΝΟΥ ΚΑΨΑΛΗ

Επιβλέπων:
Χρήστος Μακρής, Αναπληρωτής καθηγητής

Συνεπιβλέπων:
Ιωάννης Κανελλόπουλος, Ιδρυτής και Διευθύνων Σύμβουλος της εταιρείας Code4Thought

Μέλη εξεταστικής επιτροπής:
Σπυρίδων Σιούτας, Καθηγητής
Δημήτριος Τσώλης, Επίκουρος καθηγητής

Πάτρα, Φεβρουάριος 2022

Πανεπιστήμιο Πατρών
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Έρευνα και ανάπτυξη συστήματος για την ανίχνευση μεροληψίας σε συστήματα συστάσεων

Διπλωματική Εργασία

του

ΡΩΜΑΝΟΥ ΚΑΨΑΛΗ

Επιβλέπων: Χρήστος Μακρής,
Αναπληρωτής καθηγητής
Συνεπιβλέπων: Ιωάννης Κανελλόπουλος,
Ιδρυτής και Διευθύνων Σύμβουλος
της εταιρίας Code4Thought

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Ημερομηνία Εξέτασης.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Χρήστος Μακρής,	Σπυρίδων Σιούτας	Δημήτριος Τσώλης
Αναπληρωτής καθηγητής	Καθηγητής	Επίκουρος καθηγητής

Πάτρα, Φεβρουάριος 2022

Πανεπιστήμιο Πατρών
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Υπεύθυνη Δήλωση
Βεβαιώνω ότι είμαι συγγραφέας αυτής της διπλωματικής εργασίας, και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην διπλωματική εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών.

(Υπογραφή)

.....
Ρωμανός Καψάλης

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον αναπληρωτή καθηγητή κ. Χρήστο Μακρή, για την τη δυνατότητα που μου έδωσε να εκπονήσω την παρούσα διπλωματική εργασία και για τις γνώσεις που μου προσέφερε καθόλη την διάρκεια των σπουδών μου. Επίσης, θα ήθελα να ευχαριστήσω ιδιαίτερα τον κ. Ιωάννη Κανελλόπουλο, συνεπιβλέποντα της διπλωματικής εργασίας, ο οποίος μου έδωσε την ευκαιρία να γνωρίσω ένα καινούριο, άμεσα εξελισσόμενο και άκρως ενδιαφέρον ερευνητικό πεδίο, για τον χρόνο που διέθεσε, για τις εποικοδομητικές συμβουλές του και γενικότερα για την άψογη συνεργασία που είχαμε. Ευχαριστώ επίσης και τα υπόλοιπα δύο μέλη της τριμελούς επιτροπής, τον καθηγητή κ. Σπυρίδωνα Σιούτα και τον επίκουρο καθηγητή κ. Δημήτριο Τσώλη. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, και πιο συγκεκριμένα τους γονείς μου Μιλτιάδη και Χρυσούλα και τον αδερφό μου Ευθύμη, για την απέραντη αγάπη με την οποία με περιέβαλαν όλα αυτά τα χρόνια και την συνεχή τους στήριξη σε κάθε καινούρια μου προσπάθεια. Χωρίς εκείνους τίποτα δεν θα ήταν δυνατό.

Περίληψη

Τα συστήματα συστάσεων εντοπίζονται πλέον παντού στον κόσμο του διαδικτύου, καθορίζοντας απλές καθημερινές μας συνήθειες όπως την μουσική που θα ακούσουμε, τα προϊόντα που θα αγοράσουμε, τα βιβλία που θα διαβάσουμε και καταλήγουν να επηρεάζουν έμμεσα το τι σκεφτόμαστε και πως δρούμε ως πολίτες στην κοινωνία. Καθίσταται επομένως επιτακτική η ανάγκη ελέγχου αυτών των συστημάτων και η εύρεση της όποιας μεροληψίας εισάγουν. Το πιο γνωστό και σοβαρό είδος μεροληψίας είναι η μεροληψία δημοφιλίας (popularity bias).
Σε αυτή την διπλωματική εργασία αναφέρονται τα σημαντικότερα ζητήματα μεροληψίας και δικαιοσύνης που εντοπίζονται στην μηχανική μάθηση γενικότερα και στα συστήματα συστάσεων ειδικότερα. Στα πλαίσια αυτής της διπλωματικής εργασίας έχει αναπτυχθεί μια εφαρμογή η οποία επιτρέπει στους χρήστες να δημιουργήσουν ένα σύστημα συστάσεων, χρησιμοποιώντας ένα σύνολο δεδομένων της επιθυμίας τους, και ακολούθως να ελέγξουν εάν έχει εισαχθεί κάποια μεροληψία και να την μετριάσουν με χρήση ενός εκ των τεσσάρων αλγορίθμων που προσφέρονται: FAR, PFAR, FA*IR και Calibrated recommendations.
Με την εφαρμογή αυτή υλοποιήθηκαν πειράματα για τον εντοπισμό της μεροληψίας με χρήση τεσσάρων συνόλων δεδομένων, εκ των οποίων το ένα πραγματικό. Στη συνέχεια, γίνεται η αξιολόγηση των αποτελεσμάτων μέσω τριών διαφορετικών τύπων αναλύσεων: ανάλυση υπερπαραμέτρων των αλγορίθμων, σύγκριση αλγορίθμων και συνόλων δεδομένων και ανάλυση cut-off. Σε όλες τις αναλύσεις που πραγματοποιήθηκαν εξετάστηκε επίσης ο ρόλος των χαρακτηριστικών των δεδομένων, όπως η αραιότητα του μητρώου χρηστών αξιολογήσεων, ο λόγος αξιολογήσεων προς χρήστες, αξιολογήσεων προς αντικείμενα, χρηστών προς αντικείμενα και ο χώρος των αξιολογήσεων, δίνοντας ιδιαίτερη έμφαση στην αραιότητα των δεδομένων. Τέλος, γίνεται ο μετριασμός της μεροληψίας που εντοπίστηκε, με σύγκριση τριών διαφορετικών αλγορίθμων.
Από την ανάλυση που πραγματοποιήθηκε διαπιστώθηκε ότι σε όλα τα σύνολα δεδομένων, τα χαρακτηριστικά των δεδομένων επηρεάζουν έως έναν βαθμό την μεροληψία που εισάγεται. Παράλληλα, οι υπερπαράμετροι των αλγορίθμων παίζουν πολύ μεγάλο ρόλο στην ρύθμιση της μεροληψίας πέρα από την ρύθμιση την ακρίβειας. Μια ακόμη διαπίστωση που προέκυψε από την έρευνά μας είναι ότι οι post-processing αλγόριθμοι μετριασμού της μεροληψίας μπορούν να βελτιώσουν το αντιστάθμισμα μεροληψίας-ακρίβειας, ωστόσο έχουν και σημαντικούς περιορισμούς. Εν κατακλείδι, οι δημιουργοί των συστημάτων είναι αναγκαίο αφενός να έχουν επίγνωση της μεροληψίας που εισάγεται, καθώς και των αιτιών της, και αφετέρου θα πρέπει να φροντίζουν να βρίσκουν ένα αντιστάθμισμα ανάμεσα στην ακρίβεια και την μεροληψία. Αυτό μπορεί να συμβεί είτε με την κατάλληλη ρύθμιση των υπερπαραμέτρων είτε με τον μετριασμό της μεροληψίας. Η εφαρμογή που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας συμβάλλει σημαντικά προς αυτή την κατεύθυνση.

Λέξεις κλειδιά
Μηχανική μάθηση, Συστήματα συστάσεων, Μεροληψία δημοφιλίας, Δικαιοσύνη

Abstract

Recently, researchers have increased their scrutiny of ethical issues on artificial intelligence (AI), especially on the field of Machine Learning. However, most previous studies on the area of Ethical Machine Learning have only focused on classification and regression tasks, while only a few studies have investigated ethical issues on recommender systems. The aim of this Diploma Thesis is to contribute to the understanding of biases that appear in recommender systems.
In this direction, a web-app was developed to help users understand how biases are introduced in recommender systems. Moreover, we could thereby estimate the extend of bias in these systems. The app is comprised by four main pages. The first page visualizes datasets to help users find possible biases. In the second page, the user can build a recommender system by choosing between a vast collection of algorithms and hyperparameter tuning options in a user-friendly way. Additionally, we developed a page for the evaluation of a recommender system as per popularity bias, fairness, diversity, novelty and coverage. The evaluation consists of a) bias monitoring through different types of plots for a single dataset or dataset comparison b) cut-off analysis and c) hyperparameter analysis. Finally, we developed a page for popularity bias mitigation using one of the four algorithms that are available: FAR, PFAR, FA*IR and Calibrated recommendations.
With reference to the broader field of ethical issues, this thesis shares special interest to popularity bias, diversity, novelty and item coverage. An extensive experimental study was conducted to gain a better understanding of the sources of bias and analyze the effect of different bias mitigation algorithms. This was implemented by utilizing the aforementioned web app. Four datasets were used in the present study: one real dataset provided by a major electronics retailer, and three datasets collected from the internet. The first part of the study examines the role of the hyperparameter tuning for every algorithm that was used and the role of dataset characteristics, in bias and accuracy. It also compares the above-mentioned datasets. The second part consists of bias mitigation using three re-ranking algorithms: FAR, PFAR and Calibrated recommendations and an in-processing algorithm.
This study has identified that data characteristics, and especially the sparsity of user-item matrix, can highly affect the bias that is introduced. Moreover, another significant finding is that the post-processing mitigation algorithms that were examined can improve the bias-accuracy tradeoff, but have several limitations too. In conclusion, developers of recommender systems need to be aware of sources of biases and of the accuracy-bias tradeoff. This work contributes to this direction and lays the groundwork for future research into bias in recommender systems.

Keywords
Machine learning, Recommender systems, Popularity bias, Fairness

Στην οικογένειά μου

Περιεχόμενα

Ευρετήριο Εικόνων

Ευρετήριο Πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Η ραγδαία ανάπτυξη των αλγορίθμων τεχνητής νοημοσύνης την τελευταία δεκαετία, έχει οδηγήσει πολλές εταιρείες, κυβερνήσεις και οργανισμούς να υιοθετήσουν τέτοιες λύσεις και να επωφεληθούν από αυτές. Οι χρήσεις της τεχνητής νοημοσύνης εκτείνονται πλέον σε ένα πολύ μεγάλο εύρος από τα ρομποτικά συστήματα, τη βιομηχανία και τον τραπεζικό τομέα, έως την βιολογία, την γεωργία και την αρχαιολογία. Το γεγονός όμως ότι πλέον ορισμένοι αλγόριθμοι επηρεάζουν τις ζωές μας και την καθημερινότητά μας, είτε έμμεσα είτε άμεσα, έχει φέρει στο προσκήνιο μια σειρά συζητήσεων για τα ηθικά ζητήματα που προκύπτουν από τη χρήση τους. Στην παρούσα εργασία θα μας απασχολήσουν κυρίως οι αλγόριθμοι μηχανικής μάθησης, ενός πεδίου της τεχνητής νοημοσύνης. Τα τελευταία χρόνια η επιστημονική κοινότητα δείχνει ιδιαίτερο ενδιαφέρον για ζητήματα ηθικής όπως η δικαιοσύνη (fairness), η μεροληψία (bias) και η ανάγκη για λογοδοσία (accountability), διαφάνεια (transparency), επεξηγησιμότητα (explainability) και ερμηνευσιμότητα (interpretability) στη μηχανική μάθηση. Η περισσότερη έρευνα φαίνεται πως έχει γίνει για την κατηγοριοποίηση (classification), κυρίως για το binary classification, και για την παλινδρόμηση (regression). Σε αυτή την κατεύθυνση έχουν δημιουργηθεί διάφορα εργαλεία ανοικτού κώδικα (open-source), με το πιο γνωστό και ίσως πιο πλήρες να είναι το εργαλείο AIF360, το οποίο περιέχει διάφορες μετρικές για το fairness και το bias, επεξηγήσεις τους και αλγορίθμους μετριασμού της μεροληψίας. Ωστόσο, στα συστήματα συστάσεων δεν φαίνεται να έχει γίνει ανάλογη προσπάθεια, τουλάχιστον όχι σε τέτοιο βαθμό, ενώ δεν υπάρχει διαθέσιμο κάποιο παρόμοιο εργαλείο.
Έναυσμα για την ενασχόληση με το συγκεκριμένο θέμα αποτέλεσε το άρθρο με τίτλο “How Facebook got addicted to spreading misinformation” της Karen Hao [1] που δημοσιεύθηκε στο MIT Technology Review, ενός διμηνιαίου περιοδικού με ποικίλα θέματα επιστήμης και τεχνολογίας που εκδίδεται από το φημισμένο Πανεπιστήμιο των Η.Π.Α. Massachusetts Institute of Technology (M.I.T.). Στο συγκεκριμένο άρθρο, γίνεται εκτενής αναφορά στο πως οι αλγόριθμοι των συστημάτων συστάσεων που χρησιμοποιούνται από τον τεχνολογικό κολοσσό Facebook με στόχο την αύξηση της αφοσίωσης (engagement) έχουν ολέθρια αποτελέσματα στην κοινωνία. Προκειμένου να αντιληφθούμε το μέγεθος του κακού που έχει προκληθεί στην κοινωνία, αρκεί να αναφέρουμε πως η προώθηση όλο και πιο αρνητικού υλικού που κινδύνευε να επιδεινώσει περαιτέρω την ψυχική υγεία ατόμων που ήταν αρκετά ευάλωτοι ψυχολογικά, η παραπληροφόρηση και ο εξτρεμισμός που οδήγησαν σε έναν εμφύλιο πόλεμο στην Μιανμάρ, επηρέασαν εκλογικές αναμετρήσεις και κατεύθυναν πολλούς ανθρώπους στο να μην κάνουν το εμβόλιο στον καιρό της πανδημίας COVID-19 μέσα από την συστηματική

προώθηση ψευδών ειδήσεων, είναι ορισμένα από όσα αναφέρει το άρθρο. Επομένως, όπως γίνεται αντιληπτό καθίσταται επιτακτική η ανάγκη δημιουργίας ενός εργαλείου για τον εντοπισμό και τον έλεγχο της μεροληψίας στα συστήματα συστάσεων.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. δημιουργία διαδικτυακής εφαρμογής στην οποία ένας χρήστης μπορεί να:
    - αναλύσει ένα σύνολο δεδομένων
    - να δημιουργήσει συστάσεις προς τους χρήστες επιλέγοντας μέσα από μια πληθώρα αλγορίθμων και ρυθμίζοντας κατάλληλα τις υπερπαραμέτρους αυτών εύκολα και γρήγορα
    - να αξιολογήσει τα αποτελέσματα ως προς την ακρίβεια, το diversity, το novelty, το popularity bias και την κάλυψη των αντικειμένων επιλέγοντας τις μετρικές αξιολόγησης που επιθυμεί
    - να μετριάσει την μεροληψία χρησιμοποιώντας 4 διαφορετικούς αλγορίθμους και στη συνέχεια να δει σε τι ποσοστό έχει υπάρξει βελτίωση
    - να προβάλλει σε γλώσσα φιλική προς όλους τους χρήστες ανεξαρτήτως των γνώσεών τους, επεξηγήσεις για όλες τις μετρικές αξιολόγησης
2. υλοποίηση πειράματος, μέσω της ανωτέρω εφαρμογής, χρησιμοποιώντας ένα πραγματικό σύνολο δεδομένων και τρία σύνολα δεδομένων που συλλέξαμε από το διαδίκτυο, και δημιουργία συστημάτων συστάσεων με χρήση 11 διαφορετικών αλγορίθμων από 5 διαφορετικές οικογένειες. Στη συνέχεια αξιολόγηση αυτών των συστημάτων ως προς την ακρίβεια και την μεροληψία που (ενδεχομένως) εισάγουν.
3. διερεύνηση του κατά πόσο οι υπερπαράμετροι των αλγόριθμων συστάσεων επηρεάζουν την ακρίβεια και το popularity bias και πως μπορεί να βρεθεί ένα αντιστάθμισμα ανάμεσα σε αυτά τα δύο.
4. σύγκριση των αλγορίθμων και των συνόλων δεδομένων και αναζήτηση του κατά πόσο το popularity bias και η ακρίβεια επηρεάζονται από τα χαρακτηριστικά των δεδομένων.
5. σύγκριση δύο διαφορετικών τεχνικών μετριασμού της μεροληψίας και σύγκριση τριών αλγορίθμων που ανήκουν στην ίδια κατηγορία.

1.2 Διάρθρωση της διπλωματικής εργασίας

Η παρούσα εργασία αποτελείται από πέντε κεφάλαια.

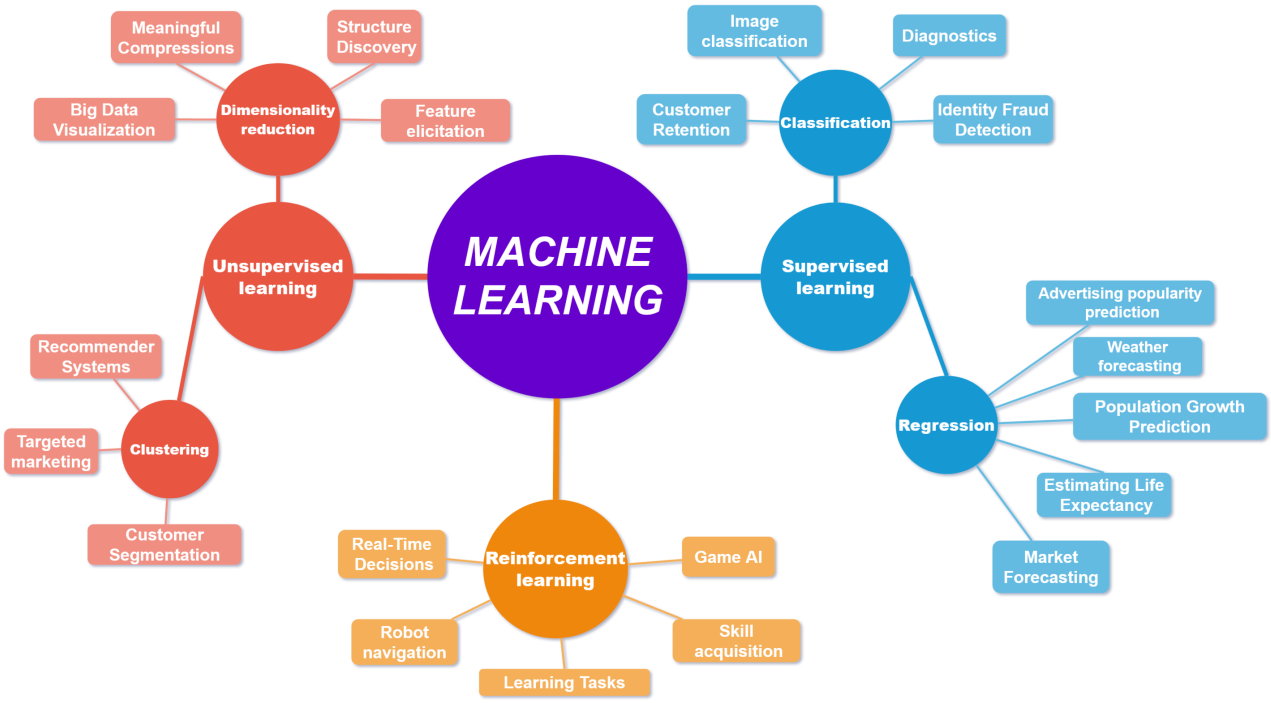
- Στο πρώτο κεφάλαιο γίνεται η παρουσίαση βασικών εννοιών της μηχανικής μάθησης και των συστημάτων συστάσεων.
- Στο δεύτερο κεφάλαιο θίγονται αρχικά ορισμένα ζητήματα που σχετίζονται με την ηθική στην μηχανική μάθηση όπως η αμεροληψία, η δικαιοσύνη, η επεξηγησιμότητα και η διαφάνεια. Στη συνέχεια του κεφαλαίου οι παραπάνω έννοιες επεκτείνονται και αναλύονται πιο ειδικά, για τα συστήματα συστάσεων.

- Στο τρίτο κεφάλαιο γίνεται η παρουσίαση της εφαρμογής που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας.
- Στο τέταρτο κεφάλαιο γίνεται η ανάλυση της πειραματικής αξιολόγησης, παρουσιάζονται αναλυτικά τα σύνολα δεδομένων, οι αλγόριθμοι και οι μετρικές αξιολόγησης που χρησιμοποιήθηκαν γίνεται η παρουσίαση και η ανάλυση των αποτελεσμάτων όπως αυτά προέκυψαν από τα πειράματα που διεξήχθησαν, ενώ παρουσιάζονται και συγκρίνονται τρεις διαφορετικοί αλγόριθμοι μετριασμού της μεροληψίας.
- Στο πέμπτο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα, οι περιορισμοί που συναντήθηκαν και μελλοντικές προεκτάσεις.

Κεφάλαιο 2

Μηχανική μάθηση και συστήματα συστάσεων

2.1 Μηχανική μάθηση

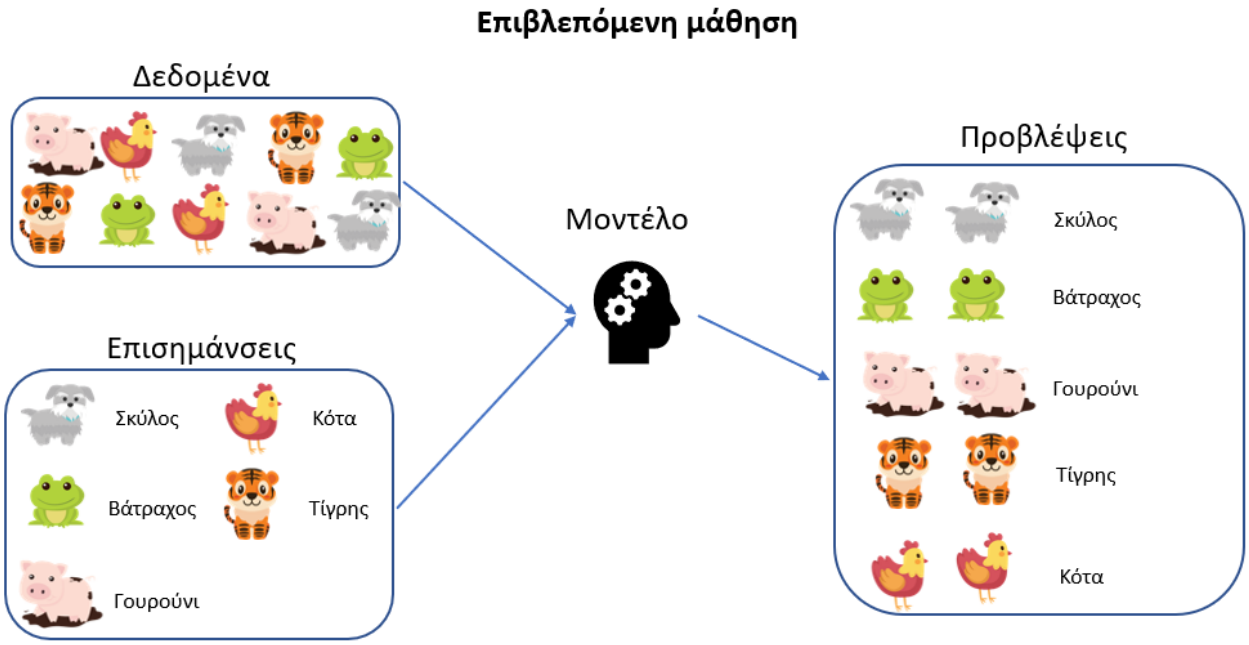


Εικόνα 2.1: Είδη μηχανικής μάθησης

Η μηχανική μάθηση είναι ένα από τα πιο ραγδαίως αναπτυσσόμενα πεδία τα τελευταία χρόνια με εφαρμογές στη βιοπληροφορική, στην οικονομία (τραπεζικός τομέας, ανάλυση μετοχών), στις τηλεπικοινωνίες, στις ιατρικές διαγνώσεις, στην τεχνολογία λογισμικού, στο μάρκετινγκ, στην αναγνώριση ομιλίας, στη ρομποτική και σε αρκετά ακόμη πεδία. Αποτελεί ένα υποπεδίο της τεχνητής νοημοσύνης, ενώ έχουν αναπτυχθεί αρκετοί ορισμοί για αυτό. Ένας από τους πρώτους και πιο απλούς ορισμούς δόθηκε από τον Samuel[2] «Η μηχανική μάθηση αποτελεί πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί», ενώ ένας πιο επίσημος ορισμός δίνεται από τον Mitchell [3]: «Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει

από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P, αν η απόδοσή του σε εργασίες από το T, όπως μετριέται από το P, βελτιώνεται μέσω της εμπειρίας E.». Όπως περιγράφεται στο [4] η συνάρτηση ενός συστήματος μηχανικής μάθησης μπορεί να είναι **περιγραφική (descriptive)**, το οποίο σημαίνει ότι το σύστημα χρησιμοποιεί τα δεδομένα για να περιγράψει τι έχει συμβεί, **προγνωστική (predictive)**, που σημαίνει ότι το σύστημα χρησιμοποιεί τα δεδομένα για να προβλέψει τι θα συμβεί ή **προστακτική (prescriptive)**, όπου το σύστημα χρησιμοποιεί τα δεδομένα για να δημιουργήσει προτάσεις σχετικές με το ποια ενέργεια θα πρέπει να γίνει. Οι αλγόριθμοι μηχανικής μάθησης εντάσσονται σε τρεις κύριες κατηγορίες, ανάλογα με το είδος της μάθησης όπως φαίνεται και στην Εικόνα 2.1. Οι κατηγορίες αυτές είναι η επιβλεπόμενη μηχανική μάθηση (supervised learning), η μη-επιβλεπόμενη (unsupervised learning) και η ενισχυμένη (reinforcement learning) και περιγράφονται αναλυτικά στην επόμενη υποενότητα. Η περιγραφή αυτών των εννοιών έγινε ύστερα από εμβριθή μελέτη των πιο κλασικών βιβλίων για αυτό το είδος [5], [6]

2.1.1 Κατηγορίες αλγορίθμων μάθησης



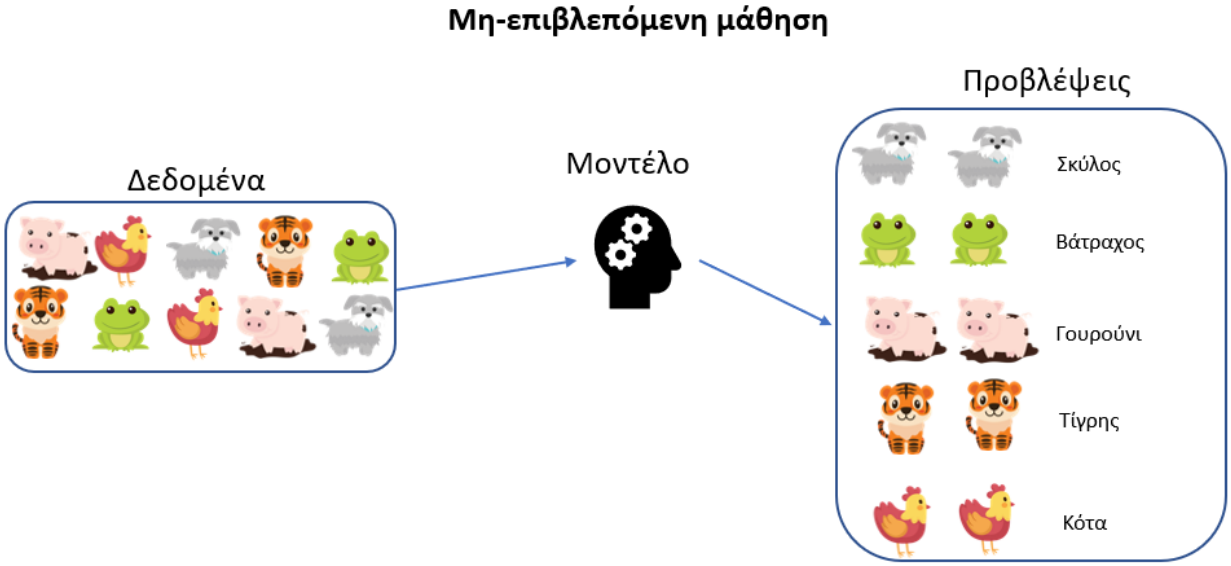
Εικόνα 2.2: Επιβλεπόμενη μάθηση

Επιβλεπόμενη μάθηση

Στην επιβλεπόμενη μηχανική μάθηση (supervised learning) οι αλγόριθμοι δέχονται ως είσοδο δεδομένα τα οποία έχουν επισημανθεί με ετικέτες (labelled data) και τα χρησιμοποιούν για την εκπαίδευσή τους. Ονομάζεται επιβλεπόμενη διότι στην εκπαίδευση λαμβάνει μέρος και κάποιος ειδικός, ο οποίος διακρίνει πότε ο αλγόριθμος παράγει σωστά αποτελέσματα και προβαίνει στις κατάλληλες διορθώσεις του αλγορίθμου έως ότου παραχθούν σωστά αποτελέσματα. Η επιβλεπόμενη μάθηση διαιρείται περαιτέρω σε δύο κύριες κατηγορίες: την παλινδρόμηση (regression) και την κατηγοριοποίηση (classification).

- **Κατηγοριοποίηση**
  Σύμφωνα με την περιγραφή που δίνει η Dunham στο [7], ένα πρόβλημα κατηγοριοποίησης χρησιμοποιεί μια συνάρτηση αντιστοίχισης (mapping function) για την αντιστοίχιση μεταβλητών εισόδου (X) σε διακριτές μεταβλητές εξόδου (Y). Σε ένα πρόβλημα κατηγοριοποίησης ο αλγόριθμος λαμβάνει επισημασμένα δεδομένα ως είσοδο για την εκπαίδευσή του και παράγει ένα μοντέλο το οποίο αν πάρει ως είσοδο μη-προσημασμένα δεδομένα δίνει ως πρόβλεψη επισημάνσεις για αυτά. Μια επισήμανση ανήκει σε ένα πεπερασμένο σύνολο κλάσεων. Εάν το πλήθος των κλάσεων είναι ίσο με δύο, για παράδειγμα έστω δύο κλάσεις για να περιγράψουμε την κατάσταση της υγείας ενός ανθρώπου («άρρωστος», «υγιής»), τότε έχουμε δυαδική κατηγοριοποίηση (binary classification), εάν υπάρχουν περισσότερες από δύο κλάσεις τότε έχουμε κατηγοριοποίηση πολλών κλάσεων (multiclass classification). Οι πιο γνωστοί αλγόριθμοι κατηγοριοποίησης είναι οι Support Vector Machines (SVM), Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbors (KNN) και οι αλγόριθμοι νευρωνικών δικτύων.

- **Παλινδρόμηση**
  Η παλινδρόμηση είναι μια στατιστική διαδικασία πρόβλεψης, η οποία εν αντιθέσει με την κατηγοριοποίηση παράγει ως πρόβλεψη συνεχείς μεταβλητές [6]. Χρησιμοποιείται κυρίως για την κατανόηση της σχέσης που υπάρχει ανάμεσα σε ανεξάρτητες και εξαρτημένες μεταβλητές, δηλαδή σε προβλήματα όπως η πρόβλεψη της θερμοκρασίας, προβλέψεις σχετικές με την οικονομία και η πρόβλεψη χρονοσειρών. Οι πιο δημοφιλείς αλγόριθμοι παλινδρόμησης είναι οι Logistic Regression, Linear Regression, Ridge Regression και Polynomial Regression.



Εικόνα 2.3: Μη-επιβλεπόμενη μάθηση

Μη-επιβλεπόμενη μάθηση
Η μη επιβλεπόμενη μηχανική μάθηση (unsupervised learning) δέχεται ως είσοδο δεδομένα χωρίς επισήμανση (unlabeled data) και κατηγοριοποίηση και χρησιμοποιεί αλγόριθμους για την εξαγωγή σημαντικών χαρακτηριστικών που απαιτούνται για την επισήμανση, την ταξινόμηση και την κατηγοριοποίηση των δεδομένων σε πραγματικό χρόνο, χωρίς ανθρώπινη παρέμβαση. Ο κύριος στόχος

της μη-επιβλεπόμενης μάθησης είναι η ανακάλυψη προτύπων. Όταν ένα μοντέλο μάθει να αναπτύσσει πρότυπα, μπορεί εύκολα να προβλέψει πρότυπα για κάθε νέο σύνολο δεδομένων με τη μορφή συστάδων. Ακολούθως περιγράφονται ορισμένες εργασίες (tasks) μη-επιβλεπόμενης μάθησης.

- **Συσταδοποίηση (Clustering)**
  Είναι ο διαχωρισμός των σημείων ενός συνόλου δεδομένων σε κατηγορίες που ονομάζονται συστάδες (clusters). Κάθε συστάδα περιέχει δεδομένα που έχουν παρόμοια χαρακτηριστικά. Η διαφορά της συσταδοποίησης από την κατηγοριοποίηση είναι ότι δεν υπάρχουν προκαθορισμένες κατηγορίες. Υπάρχουν αρκετοί αλγόριθμοι συσταδοποίησης: K-Means, BIRCH, DBSCAN, Gaussian Mixture Models (GMM) είναι ορισμένοι από αυτούς. Τέλος, αξίζει να αναφερθεί πως η συσταδοποίηση είναι μια πάρα πολύ χρήσιμη τεχνική για εργασίες όπως η ανάλυση δεδομένων, τα συστήματα συστάσεων, η μείωση της διαστατικότητας (dimensionality reduction), οι μηχανές αναζήτησης, η κατάτμηση εικόνας (image segmentation) και αρκετές ακόμη.

- **Μείωση διαστατικότητας (dimensionality reduction)**
  Όπως μαρτυράει και το όνομα, στόχος εδώ είναι η μείωση των διαστάσεων των δεδομένων. Η πιο συνηθισμένη προσέγγιση που ακολουθείται εδώ είναι η Principal Component Analysis (PCA)

Ημι-επιβλεπόμενη μάθηση

Η ημι-επιβλεπόμενη μάθηση (semi-supervised learning) είναι ένας συνδυασμός της επιβλεπόμενης και της μη-επιβλεπόμενης μάθησης. Πιο αναλυτικά, ορισμένοι αλγόριθμοι μπορούν να διαχειριστούν μεγάλη ποσότητα μη-επισημασμένων δεδομένων και αρκετά μικρή ποσότητα επισημασμένων δεδομένων.

Ενισχυμένη μάθηση

Η ενισχυμένη μηχανική μάθηση (Reinforcement machine learning) [8] έχει αρκετές ομοιότητες με την επιβλεπόμενη μηχανική μάθηση. Η κύρια διαφορά τους ωστόσο, είναι ότι η ενισχυμένη μάθηση δεν χρησιμοποιεί επισημασμένα δεδομένα για την εκπαίδευση του μοντέλου, αλλά ο agent (ο μαθητής - learner και υπεύθυνος λήψης αποφάσεων) παρατηρεί το περιβάλλον (environment), επιλέγει και εκτελεί μια ενέργεια (action) και μαθαίνει μέσω ενός συστήματος επιβράβευσης/ποινών. Το περιβάλλον αφού λάβει την επιλεγμένη ενέργεια και την κατάσταση (state) του agent ως είσοδο, παράγει ως έξοδο, για κάθε σωστή ενέργεια μια ανταμοιβή (positive reward) και για κάθε λανθασμένη μια ποινή (negative reward), καθώς και την επόμενη κατάσταση. Ο agent επιλέγει την επόμενη ενέργειά του, μέσω μια στρατηγικής (policy), η οποία στην πραγματικότητα είναι μια συνάρτηση, με βάση την τρέχουσα κατάσταση στην οποία βρίσκεται, που παίρνει ως είσοδο. Ο στόχος ενός agent είναι να επιλέξει τη βέλτιστη στρατηγική ώστε να μεγιστοποιήσει τις ανταμοιβές που λαμβάνει. Σε αυτή την κατεύθυνση ο agent αξιοποιεί (exploit) την γνώση που ήδη έχει ώστε να λάβει μια ανταμοιβή και εξερευνά (explore) το περιβάλλον ώστε να λάβει επιπλέον πληροφορίες για αυτό και να μεγιστοποιήσει αυτή την ανταμοιβή. Οι εφαρμογές της ενισχυμένης μηχανικής μάθησης είναι πολλές και ποικίλες: επεξεργασία φυσικής γλώσσας (Natural Language Processing - NLP), ρομποτική, βαθιά μάθηση, υγειονομική περίθαλψη, χρηματοοικονομικά, ηλεκτρονικά παιχνίδια, είναι μόνο μερικές από αυτές.

2.1.2 Εκπαίδευση και αξιολόγηση της απόδοσης των μοντέλων μηχανικής μάθησης

Αφού καθοριστεί η κατηγορία στην οποία ανήκει το πρόβλημα μηχανικής μάθησης που καλούμαστε να αντιμετωπίσουμε και επιλεγεί ο καταλληλότερος αλγόριθμος, ακολουθεί η εκπαίδευση του αλγορίθμου στα δεδομένα που θα του παρέχουμε ως είσοδο. Πριν από αυτό όμως, κρίνεται αναγκαία μια προεπεξεργασία των δεδομένων. Μια από τις πιο γνωστές και συχνότερα χρησιμοποιούμενες στρατηγικές, είναι η διάσπαση του συνόλου δεδομένων που έχουμε στη διάθεσή μας σε δύο υποσύνολα. Το μεγαλύτερο θα είναι το σύνολο εκπαίδευσης (training set) το οποίο δίνεται ως είσοδος στον αλγόριθμο μηχανικής μάθησης και στο οποίο θα γίνει η εκπαίδευση του αλγορίθμου μας και το άλλο θα είναι το σύνολο δοκιμής (test set) στο οποίο θα γίνει η δοκιμή της απόδοσης του μοντέλου ύστερα από την εκπαίδευσή του. Τα πιο συνηθισμένα ποσοστά διάσπασης είναι 80% ή 70% του αρχικού συνόλου δεδομένων να αποτελεί το σύνολο εκπαίδευσης και το 30% ή 20% να αποτελεί το σύνολο δοκιμής. Σε ορισμένες περιπτώσεις η διαδικασία της δοκιμής της απόδοσης του μοντέλου γίνεται σε δύο βήματα και για αυτόν τον λόγο χρειάζεται και ένα ακόμη σύνολο, το σύνολο επικύρωσης (validation set). Στο πρώτο βήμα το σύνολο αυτό χρησιμοποιείται για την αξιολόγηση και την επιλογή του βέλτιστου μοντέλου και τη ρύθμιση υπερπαραμέτρων (hyperparameter tuning), ενώ στο δεύτερο βήμα χρησιμοποιείται το σύνολο δοκιμής όπως ακριβώς και πριν.
Σε αυτήν την υποενότητα παρουσιάζονται ορισμένες μετρικές αξιολόγησης της απόδοσης των μοντέλων μηχανικής μάθησης, κάτι που κρίνεται απαραίτητο προκειμένου να γίνουν πιο κατανοητές ορισμένες έννοιες που θα παρουσιαστούν στις επόμενες ενότητες.
Το πρώτο πράγμα που μας ενδιαφέρει συνήθως όσον αφορά την αξιολόγηση της απόδοσης ενός μοντέλου μηχανικής μάθησης είναι η *ακρίβεια (accuracy)*:

$$\text{Ακρίβεια} = \frac{\text{αριθμός σωστών προβλέψεων}}{\text{συνολικός αριθμός προβλέψεων}}$$

Όταν έχουμε να επιλύσουμε προβλήματα κατηγοριοποίησης, τότε υπάρχουν τέσσερα πιθανά αποτελέσματα που μπορεί να προκύψουν, πρώτα όμως ας δούμε ένα πολύ συγκεκριμένο παράδειγμα από την καθημερινή ζωή. Έστω ένα διαγνωστικό τεστ το οποίο μας δείχνει αν κάποιος νοσεί από κορονοϊό ή όχι. Ορίζεται ως κλάση των θετικών (positive class) το γεγονός κάποιος άνθρωπος να νοσεί από κορονοϊό και ως κλάση των αρνητικών (negative class) το γεγονός κάποιος άνθρωπος να μην νοσεί από κορονοϊό. Πιο γενικά, στην κλάση των θετικών έχουμε δεδομένα που έχουν κάποιο χαρακτηριστικό, ενώ αντίθετα στην κλάση των αρνητικών έχουμε δεδομένα που δεν έχουν κάποιο χαρακτηριστικό. Σε αυτό το σημείο μπορούμε να εξετάσουμε τα τέσσερα αποτελέσματα που μπορεί να προκύψουν από τις προβλέψεις ενός μοντέλου κατηγοριοποίησης:

Αληθώς θετικά – True positives (TP): το μοντέλο προβλέπει επιτυχώς τη θετική (positive) κλάση. Ο ασθενής νοσεί από κορονοϊό και το τεστ βγαίνει θετικό.

Ψευδώς θετικά – False positives (FP): το μοντέλο προβλέπει εσφαλμένα τη θετική (positive) κλάση. Ο ασθενής δεν νοσεί από κορονοϊό και το τεστ βγαίνει θετικό.

Αληθώς αρνητικά – True negatives (TN): το μοντέλο προβλέπει επιτυχώς την αρνητική (negative) κλάση. Ο ασθενής δεν νοσεί από κορονοϊό και το τεστ βγαίνει αρνητικό.

Ψευδώς αρνητικά – False negatives (FN): το μοντέλο προβλέπει εσφαλμένα την αρνητική (negative) κλάση. Ο ασθενής νοσεί από κορονοϊό και το τεστ βγαίνει αρνητικό.

Ποσοστό αληθώς θετικών - True Positive Rate (TPR): ονομάζεται και Recall ή sensitivity και είναι το ποσοστό των σωστών προβλέψεων στην κλάση των θετικών, δηλαδή όταν ένας άνθρωπος νοσεί πραγματικά, πόσο συχνά προβλέπει ότι νοσεί;

$$TPR = \frac{\textit{αριθμός των αληθώς θετικών}}{\text{αριθμός των αληθώς θετικών + αριθμός των ψευδώς αρνητικών}} = 1 - FNR$$

Ποσοστό ψευδώς θετικών - False Positive Rate (FPR): το ποσοστό των λανθασμένων προβλέψεων στην κλάση των θετικών, δηλαδή όταν ένας άνθρωπος δεν νοσεί πραγματικά, πόσο συχνά προβλέπει ότι νοσεί;

$$FPR = \frac{\text{αριθμός των ψευδώς θετικών}}{\text{αριθμός των ψευδώς θετικών + αριθμός των αληθώς αρνητικών}} = 1 - TNR$$

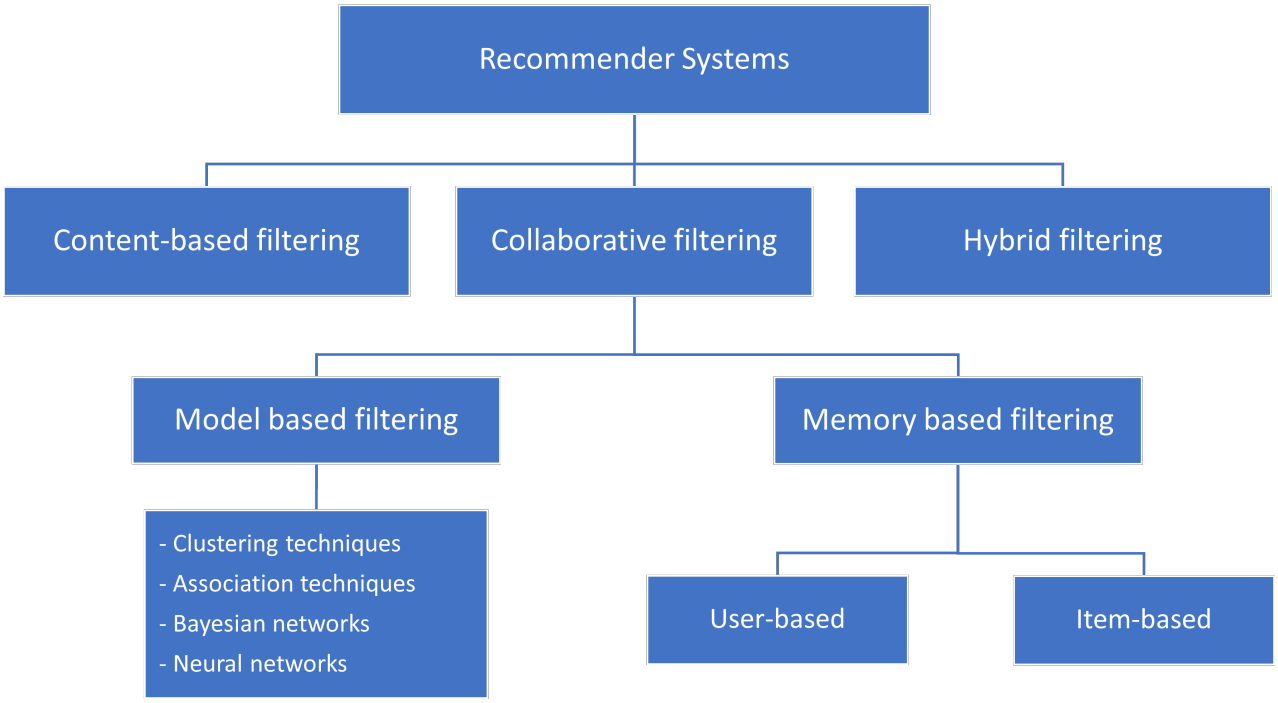
Ποσοστό αληθώς αρνητικών - True Negative Rate (TNR): είναι γνωστό και ως specificity και περιγράφει το ποσοστό των σωστών προβλέψεων στην κλάση των αρνητικών, δηλαδή όταν ένας άνθρωπος δεν νοσεί πραγματικά, πόσο συχνά προβλέπει ότι δεν νοσεί;

$$TNR = \frac{\text{αριθμός των αληθώς αρνητικών}}{\text{αριθμός των αληθώς αρνητικών + αριθμός των ψευδώς θετικών}} = 1 - FPR$$

Ποσοστό ψευδώς αρνητικών - False Negative Rate (FNR): το ποσοστό των λανθασμένων προβλέψεων στην κλάση των αρνητικών, δηλαδή όταν ένας άνθρωπος νοσεί πραγματικά, πόσο συχνά προβλέπει ότι δεν νοσεί; Είναι γνωστό και ως ποσοστό αποτυχιών (miss rate).

$$FNR = \frac{\text{αριθμός των ψευδώς αρνητικών}}{\text{αριθμός των αληθώς θετικών + αριθμός των ψευδώς αρνητικών}} = 1 - TPR$$

2.2 Συστήματα συστάσεων



Εικόνα 2.4: Κατηγορίες αλγορίθμων συστημάτων συστάσεων

Στην υποενότητα αυτή θα γίνει μια αναφορά στην έννοια των συστημάτων συστάσεων, στα είδη αυτών καθώς και στα πιο συνηθισμένα προβλήματα που εντοπίζονται σε αυτά. Η ανάλυση που ακολουθεί βασίζεται στα βιβλία [9], [10], [11] στα οποία μπορείτε να ανατρέξετε, αν επιθυμείτε να εμβαθύνετε περισσότερο.

Ως σύστημα συστάσεων (recommendation ή recommender system ή RecSys εν συντομία) ορίζεται το σύστημα όπου οι άνθρωποι παρέχουν προτάσεις ως εισόδους οι οποίες έπειτα συγκεντρώνονται και κατευθύνονται σε συγκεκριμένους παραλήπτες [12]. Αυτός ήταν ένας από τους πρώτους ορισμούς που δόθηκαν για τα συστήματα συστάσεων. Οι βασικές οντότητες που υπάρχουν σε αυτά τα συστήματα είναι οι χρήστες (users) και τα αντικείμενα (items). Η χρήση των συστημάτων συστάσεων έχει αυξηθεί κατακόρυφα τελευταία στο χώρο του διαδικτύου, καθώς τα συστήματα αυτά χρησιμοποιούνται στον χώρο του ηλεκτρονικού εμπορίου, στα κοινωνικά δίκτυα, σε συστήματα παροχής υπηρεσιών ψυχαγωγίας, στις μηχανές αναζήτησης και σε πολλές ακόμη υπηρεσίες. Ιδιαίτερη σημασία σε ένα σύστημα συστάσεων έχει η συλλογή των δεδομένων από τους χρήστες, δηλαδή η λήψη feedback από αυτούς. Υπάρχουν δύο κύριοι τρόποι για να γίνει αυτό:

1. **Implicit feedback (έμμεση ανάδραση) [13]:** σε αυτή την περίπτωση δεν υπάρχει άμεση εμπλοκή του χρήστη στην αξιολόγηση των αντικειμένων. Το σύστημα συλλέγει στοιχεία σχετικά με τον χρήστη όπως είναι το ιστορικό αναζήτησης, τα κλικ που έχει κάνει, τα αντικείμενα που έχει δει ή/και έχει αγοράσει και το προφίλ του (ηλικία, φύλο κτλ.). Μέσω αυτών των στοιχείων το σύστημα έχει τη δυνατότητα να προβλέψει ποια είναι τα αντικείμενα που ενδιαφέρουν τον χρήστη, ώστε να δημιουργήσει τις κατάλληλες συστάσεις για αυτόν. Στην έμμεση ανάδραση οι τιμές στο μητρώο χρηστών-αντικειμένων αντιστοιχούν στην σιγουριά (confidence) που έχουμε

για μια παρατήρηση που έχει γίνει για την σχέση ενός χρήστη με ένα αντικείμενο.

2. **Explicit feedback (ρητή ανάδραση):** οι χρήστες, εν αντιθέσει με το implicit feedback, παρέχουν αξιολογήσεις σε αντικείμενα που έχουν δει ή έχουν αγοράσει, είτε δίνοντας κάποια βαθμολογία, συνήθως σε κλίμακα 1 έως 5 ή 5 έως 10, είτε δηλώνοντας απλά αν τους άρεσε ή όχι κάποιο αντικείμενο (like και dislike), δηλαδή παρέχοντας μια δυαδική αξιολόγηση (binary rating), είτε -πιο σπάνια- επιλέγοντας μια γραπτή αξιολόγηση (ordinal rating) όπως στα ερωτηματολόγια («καθόλου», «λίγο», «πολύ», «πάρα πολύ»). Στη ρητή ανάδραση οι τιμές στο μητρώο χρηστών-αντικειμένων αντιστοιχούν σε προτιμήσεις (preferences) των χρηστών για τα αντικείμενα.

Φυσικά σε πολλές περιπτώσεις υπάρχει και ο συνδυασμός του implicit και του explicit feedback, δημιουργώντας ένα υβριδικό feedback.
Υπάρχουν τρεις κύριοι τύποι συστημάτων συστάσεων και άρα τρεις τύποι αλγορίθμων, όπως φαίνεται και στην Εικόνα 2.4, οι οποίοι περιγράφονται αναλυτικά στην επόμενη υποενότητα.

2.2.1 Συστήματα βασισμένα στη συνεργατική διήθηση (Collaborative-filtering systems)

Τα συστήματα αυτά, βασίζονται στην υπόθεση ότι αν σε κάποιον χρήστη αρέσει ένα αντικείμενο A, και σε έναν άλλο χρήστη αρέσει το αντικείμενο A μαζί με κάποιο άλλο αντικείμενο B, τότε ο πρώτος χρήστης είναι αρκετά πιθανό να ενδιαφέρεται και για το αντικείμενο B. Οπότε, βασίζεται σε παλιές αλληλεπιδράσεις μεταξύ χρηστών και αντικειμένων, ώστε να προβλέψει νέες. Είναι μια αρκετά δημοφιλής προσέγγιση, η οποία χρησιμοποιείται από τα μεγαλύτερα και τα πιο ευρέως χρησιμοποιούμενα συστήματα, όπως η Amazon, το YouTube, το Netflix, το LinkedIn και το Spotify. Υπάρχουν δύο τύποι μεθόδων που χρησιμοποιούνται στα συστήματα που βασίζονται στη συνεργατική μάθηση:

- **Μέθοδοι με βάση το περιεχόμενο (memory-based methods):** γνωστές και ως μέθοδοι με βάση τη γειτονιά (neighborhood-based) ή ευρετικές (heuristic-based) μέθοδοι. Βασίζονται στην υπόθεση ότι παρόμοιοι χρήστες εμφανίζουν παρόμοια μοτίβα συμπεριφοράς ως προς την αξιολόγηση των αντικειμένων και ότι παρόμοια αντικείμενα λαμβάνουν παρόμοιες αξιολογήσεις. Στις μεθόδους με βάση το περιεχόμενο, οι αξιολογήσεις που έχουν υποβάλλει οι χρήστες σε ορισμένα αντικείμενα και έχουν αποθηκευτεί στο σύστημα, χρησιμοποιούνται απευθείας για την πρόβλεψη αξιολογήσεων για νέα αντικείμενα. Υπάρχουν δύο τρόποι για να γίνει αυτό.

  - Ο πρώτος είναι η **διήθηση χρήστη προς χρήστη (user-user ή user based filtering)**[14], η οποία για ένα συγκεκριμένο χρήστη U, βρίσκει K χρήστες οι οποίοι έχουν παρόμοια ενδιαφέροντα με αυτόν, με βάση την ομοιότητα των αξιολογήσεων, και προτείνει αντικείμενα στον χρήστη U τα οποία άρεσαν σε αυτούς τους χρήστες. Οι K παρόμοιοι αυτοί χρήστες είναι οι γείτονες του χρήστη U. Η εύρεση της ομοιότητας δύο αντικειμένων γίνεται υπολογίζοντας κάποια μετρική ομοιότητας, όπως για παράδειγμα η μετρική συνημιτόνου (cosine similarity), η μετρική Chebyshev και πολλές άλλες. Μέσω αυτών των μετρικών βρίσκονται οι K πλησιέστεροι γείτονες κάθε αντικειμένου. Η μαθηματική περιγραφή των όσων περιγράψαμε δίνεται από τον τύπο:

$$\widehat{r_{u_1 i}} = \mu_{u_1} + \frac{\sum_{j \in N_u^k(i)} \text{sim}\left(u_1, u_2\right) \cdot r_{u_2 j}}{\sum_{j \in P_u^k(i)} \text{sim}\left(u_1, u_2\right)} \tag{2.1}$$

όπου $sim\left(u_1, u_2\right)$ είναι η συνάρτηση που υπολογίζει την ομοιότητα ανάμεσα στο χρήστη u_1 για τον οποίο θέλουμε να δημιουργήσουμε τις συστάσεις και σε έναν χρήστη u_2, μ_u είναι η μέση αξιολόγηση του χρήστη u: $\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{|I_u|}$ με I_u το σύνολο των δεικτών των αντικειμένων που έχει αξιολογήσει ο χρήστης u, $P_{u_1}^k(i)$ είναι το σύνολο των k πλησιέστερων χρηστών στον χρήστη u_1 και $r_{u_2 j}$ η αξιολόγηση που έχει δώσει ο χρήστης u_2 στο αντικείμενο j.

– Αντίθετα, στον δεύτερο τρόπο, τη **διήθηση αντικείμενο προς αντικείμενο (item-item ή item-based filtering)**[15], για τη δημιουργία συστάσεων για έναν χρήστη U δημιουργείται ένα σύνολο παρόμοιων αντικειμένων με όσα έχει αλληλεπιδράσει θετικά ο χρήστης U. Ως θετική αλληλεπίδραση ορίζουμε μια θετική αξιολόγηση, μια αγορά ενός αντικειμένου ή ακόμη και ένα κλικ, δηλαδή μια προβολή του αντικειμένου από τον χρήστη. Σε αυτό το σημείο θα πρέπει να επισημάνουμε πως δύο αντικείμενα θεωρούνται παρόμοια αν οι περισσότεροι χρήστες έχουν αλληλεπιδράσει με παρόμοιο τρόπο και με τα δύο αυτά αντικείμενα, για παράδειγμα έστω ένα σύνολο δεδομένων από μια επιχείρηση ηλεκτρονικού εμπορίου, αν οι περισσότεροι χρήστες που αγόρασαν το προϊόν “Sony PlayStation 5”, αγόρασαν επίσης και το βιντεοπαιχνίδι “Fifa 21”, τότε προκύπτει το συμπέρασμα ότι αυτά τα δύο προϊόντα είναι παρόμοια. Η αξιολόγηση γίνεται υπολογίζοντας αποστάσεις (distances) ανάμεσα σε αυτά τα αντικείμενα. Η μέθοδος αυτή για ένα αντικείμενο I βρίσκει χρήστες στους οποίους άρεσε αυτό το αντικείμενο, καθώς και άλλα αντικείμενα τα οποία άρεσαν στους συγκεκριμένους ή σε «παρόμοιους» χρήστες. Το όνομα αυτής της μεθόδου προκύπτει από το γεγονός ότι παίρνει ως είσοδο αντικείμενα και δίνει ως έξοδο άλλα αντικείμενα ως προτάσεις. Η πρόβλεψη υπολογίζεται με παρόμοιο τρόπο με την μέθοδο διήθησης χρήστη προς χρήστη:

$$\widehat{r_{ui}} = \mu_u + \frac{\sum_{j \in N_u^k(i)} \operatorname{sim}(i, j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} \operatorname{sim}(i, j)} \tag{2.2}$$

όπου $sim(i, j)$ είναι η συνάρτηση που υπολογίζει την ομοιότητα ανάμεσα στο αντικείμενο i και στο αντικείμενο j, μ_u είναι η μέση αξιολόγηση του χρήστη u: $\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{|I_u|}$ με I_u το σύνολο των δεικτών των αντικειμένων που έχει αξιολογήσει ο χρήστης u, $N_u^k(i)$ είναι το σύνολο των αντικειμένων στην γειτονιά που ο χρήστης u έχει αξιολογήσει και r_{uj} η αξιολόγηση που έχει δώσει ο χρήστης u στο αντικείμενο j.

■ **Μέθοδοι με βάση το μοντέλο (model-based):** σε αυτήν την περίπτωση τα μοντέλα συνεργατικής διήθησης παρέχουν συστάσεις στους χρήσεις μέσω αλγορίθμων μηχανικής μάθησης. Ο στόχος είναι να εκπαιδεύσουμε τα μοντέλα, προκειμένου να μας δώσουν ως προβλέψεις λίστες συστάσεων. Συναντάμε τρεις κύριους τύπους: α) Matrix Factorization (MF) β) υλοποιήσεις που βασίζονται σε τεχνικές βαθιάς μάθησης (κυρίως σε νευρωνικά δίκτυα) και γ) αλγόριθμους συσταδοποίησης (clustering based).

Στην παρούσα εργασία θα μας απασχολήσουν κυρίως τα συστήματα που βασίζονται στη συνεργατική διήθηση και πιο συγκεκριμένα τα συστήματα βασισμένα στο μοντέλο, επομένως όπως γίνεται αντιληπτό θα δοθεί λίγη περισσότερη έμφαση στην ανάλυση αυτών. Στη συνέχεια αυτής της υποενότητας θα αναφέρουμε τις κυριότερες κατηγορίες αλγορίθμων.

Μοντέλα λανθανόντων παραγόντων (Latent factor models)

Στη γραμμική άλγεβρα ένα μητρώο (matrix) $m \times n$ ορίζεται ως ένας ορθογώνιος πίνακας αριθμών, συμβόλων ή εκφράσεων, διατεταγμένων σε m γραμμές και n στήλες [16]. Ένα διάνυσμα (vector) είναι μια ειδική περίπτωση ενός μητρώου, που έχει μία μόνο στήλη. Στα συστήματα συστάσεων κάθε αντικείμενο αναπαρίσταται από ένα διάνυσμα, με τόσες γραμμές όσοι και οι χρήστες, όπου κάθε κελί (i, j) περιέχει την αξιολόγηση του χρήστη i για το αντικείμενο j. Ένα μητρώο αξιολογήσεων (rating matrix) $R \in \mathbb{R}^{m \times n}$ αποτελείται από όλα τα διαθέσιμα αντικείμενα, πλήθους n, και όλους τους διαθέσιμους χρήστες, πλήθους m, ενός συνόλου δεδομένων, με άλλα λόγια στο μητρώο αποθηκεύονται όλα τα διανύσματα που προαναφέρθηκαν.

Το μητρώο αξιολογήσεων μπορεί να περιέχει χιλιάδες ή ακόμη και εκατομμύρια γραμμές και στήλες. Η μεγάλη αύξηση των διαστάσεων έχεις ως συνέπεια την εκθετική αύξηση του χώρου διαστάσεων και αυξάνει κατά πολύ την αραιότητα (sparsity) των δεδομένων, προκαλώντας μια πληθώρα προβλημάτων. Αυτό το φαινόμενο έχει γίνει γνωστό από τον Richard E. Bellman [17] ως «κατάρα της διαστατικότητας» (curse of dimensionality). Στα μοντέλα λανθανόντων παραγόντων, χρησιμοποιούνται τεχνικές μείωσης της διαστατικότητας (dimensionality reduction) για τον υπολογισμό του μητρώου δεδομένων (data matrix). Πιο συγκεκριμένα, η βασική υπόθεση σε αυτά είναι η αξιοποίηση των συσχετίσεων που υπάρχουν ανάμεσα σε αρκετές γραμμές και στήλες του μητρώου δεδομένων, δηλαδή ανάμεσα σε χρήστες και αντικείμενα. Μέσω αυτής της υπόθεσης, είναι εφικτή η προσέγγιση του μητρώου δεδομένων με αρκετά καλή ακρίβεια από ένα μητρώο μικρότερης τάξης. Το ερώτημα εδώ είναι πως επιτυγχάνεται αυτό.

Έστω ένα δείγμα από D-διάστατα πραγματικά διανύσματα, που έχουν παραχθεί από μια άγνωστη κατανομή. Η κατανομή στο χώρο δεδομένων γίνεται στην πραγματικότητα εξαιτίας της παρουσίας ενός μικρού αριθμού μεταβλητών (L<D) που δρουν σε συνδυασμό μεταξύ τους, για παράδειγμα ταινίες που έχουν θετική συσχέτιση μεταξύ τους, οι μεταβλητές αυτές ονομάζονται λανθάνουσες (latent variables) ή κρυφές (hidden) καθώς δεν είναι άμεσα ορατές αλλά η εύρεσή τους γίνεται από έναν αλγόριθμο. Με αυτόν τον τρόπο δημιουργείται ένα σημείο στον λανθάνοντα χώρο σύμφωνα με μια προηγούμενη κατανομή και αντιστοιχίζεται στο χώρο δεδομένων με μια ομαλή αντιστοίχιση. Αυτό έχει ως αποτέλεσμα τη δημιουργία ενός L-διάστατου υπόχωρου στον χώρο δεδομένων. Προκειμένου να επεκταθεί αυτό σε ολόκληρο τον D-διάστατο χώρο δεδομένων, ορίζουμε ένα μοντέλο θορύβου (σφάλματος). Το μοντέλο λανθανουσών μεταβλητών ορίζεται από την αντιστοίχιση από τον λανθάνοντα χώρο στον χώρο δεδομένων και το μοντέλο θορύβου σε χώρο δεδομένων. Οι παράμετροι ενός τέτοιου μοντέλου συνήθως βελτιστοποιούνται με τη χρήση ενός κριτηρίου μέγιστης πιθανοφάνειας (maximum likelihood) με χρήση του αλγόριθμου Μεγιστοποίησης Προσδοκίας (Expectation Maximization - EM). Η μείωση των διαστάσεων επιτυγχάνεται με τον ορισμό μιας αντίστροφης αντιστοίχισης από τον χώρο δεδομένων στον λανθάνοντα χώρο, έτσι ώστε σε κάθε σημείο δεδομένων να εκχωρείται ένας αντιπρόσωπος στον λανθάνοντα χώρο. Υπάρχουν αρκετές τεχνικές μείωσης της διαστατικότητας, η πιο ευρέως χρησιμοποιούμενη από αυτές είναι η παραγοντοποίηση μητρώου.

- **Παραγοντοποίηση μητρώου (Matrix Factorization)**

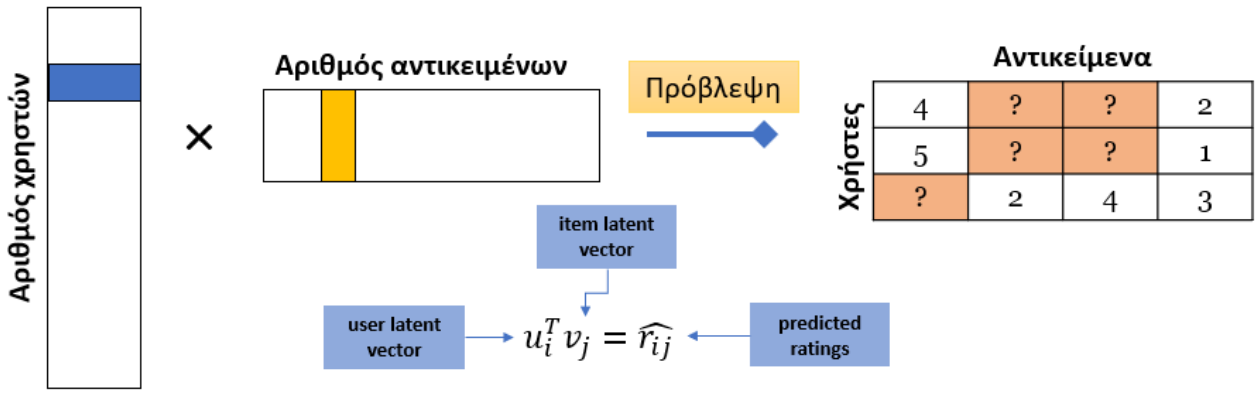
  Όπως προαναφέρθηκε, ένα ζήτημα που προκύπτει με το μητρώο αξιολογήσεων R είναι ότι στις περισσότερες περιπτώσεις δεν υπάρχει αλληλεπίδραση μεταξύ όλων των χρηστών και όλων των αντικειμένων και ως εκ τούτου ορισμένα κελιά του μητρώου αξιολογήσεων μένουν κενά, δημιουργώντας ένα αραιό μητρώο. Το ερώτημα που τίθεται επομένως είναι πώς

γεμίζουμε αυτά τα κενά κελιά.

Μια λύση σε αυτό αποτελεί η μείωση της διαστατικότητας μέσω της παραγοντοποίησης μητρώου [18]. Η παραγοντοποίηση είναι η μέθοδος με την οποία διασπάμε μια μεγάλη ποσότητα, για παράδειγμα έναν μεγάλο αριθμό, σε μικρότερες ποσότητες οι οποίες ονομάζονται παράγοντες (factors). Εάν αυτή η ποσότητα δεν είναι ένας αριθμός, αλλά ένα μητρώο, στην περίπτωσή μας το μητρώο R, τότε αυτό διασπάται σε δύο μητρώα μικρότερων διαστάσεων από το R, το $U \in \mathbb{R}^{m \times k}$ και το $V \in \mathbb{R}^{n \times k}$. Το μητρώο R παραγοντοποιείται ως εξής:

$$R \approx UV^{\top}$$

Το ένα από τα δύο μητρώα είναι το μητρώο χρήστη (user matrix) και το άλλο το μητρώο αντικειμένου (item matrix). Οι γραμμές των δύο μητρώων ονομάζονται λανθάνοντες παράγοντες (latent factors) και οι στήλες λανθάνοντα διανύσματα (latent vectors). Κάθε γραμμή του μητρώου χρήστη είναι ένα διάνυσμα το οποίο περιέχει k λανθάνοντες παράγοντες οι οποίοι περιγράφουν τον χρήστη u και αντίστοιχα κάθε γραμμή του μητρώου αντικειμένου είναι ένα διάνυσμα το οποίο περιέχει k λανθάνοντες παράγοντες οι οποίοι περιγράφουν το αντικείμενο i. Κάθε αξιολόγηση r_{ui} του χρήστη u στο αντικείμενο i στο R υπολογίζεται μέσω του εσωτερικού γινομένου των δύο διανυσμάτων: $r_{ui} = p_u q_i$. Στόχος επομένως, είναι η εύρεση



Εικόνα 2.5: Παραγοντοποίηση μητρώου

των μητρώων U και V, ώστε να προσεγγίζουν με την καλύτερη δυνατή ακρίβεια το αρχικό μητρώο R. Το κριτήριο για το πόσο καλή είναι μια προσέγγιση δίνεται από μια συνάρτηση κόστους η οποία θα πρέπει να ελαχιστοποιηθεί χρησιμοποιώντας κάποια τεχνική μηχανικής μάθησης. Αυτό είναι ένα κλασικό πρόβλημα ελαχιστοποίησης. Μια επιλογή για τη συνάρτηση κόστους αποτελεί το μέσο τετραγωνικό σφάλμα (Mean Square Error – MSE) λαμβάνοντας υπόψη μόνο τις τιμές που έχουν παρατηρηθεί (observed values):

$$\min_{U \in \mathbb{R}^{m \times d}, V \in \mathbb{R}^{n \times d}} \sum_{i,j \in \mathbb{Z}} \left(R_{ij} - U_i V_j^T \right)^2$$

Αυτό όμως οδηγεί σε συστήματα συστάσεων με αρκετά χαμηλή απόδοση. Μια άλλη σκέψη επομένως είναι να υποθέσουμε ότι οι τιμές που δεν έχουν παρατηρηθεί (unobserved values) είναι ίσες με μηδέν. Σε αυτή την περίπτωση αντί για το μέσο τετραγωνικό σφάλμα στόχος

είναι η ελαχιστοποίηση της νόρμας Frobenius ($\|\cdot\|^2$) που είναι ίση με το άθροισμα των τετραγώνων των καταχωρήσεων του μητρώου:

$$\min_{U \in R^{m \times d}, V \in R^{n \times d}} ||R - UV^{\mathrm{T}}||^2$$

Σε ένα πρόβλημα συστάσεων, όπως και σε όλα τα προβλήματα μηχανικής μάθησης, είναι πάντα ορατός ο κίνδυνος της υπερεκπαίδευσης (overfitting) του μοντέλου. Για να το αποφύγουμε αυτό προσθέτουμε στην εξίσωση τον δεύτερο όρο κανονικοποίησης λ που είναι το L1 regularization και λ είναι η παράμετρος κανονικοποίησης υψωμένης στο τετράγωνο ($\|\cdot\|^2$) [18]

$$J = \min_{U \in R^{m \times d}, V \in R^{n \times d}} \sum_{i,j \in \mathbb{Z}} \left(R_{ij} - U_i V_j^T \right)^2 + \lambda(||U||_2 + ||V||_2) \tag{2.3}$$

Εκτός όμως από την ελαχιστοποίηση της συνάρτησης κόστους, είναι αναγκαίος και ο προσδιορισμός της βέλτιστης τιμής της παραμέτρου κανονικοποίησης λ, καθώς και των βέλτιστων τιμών των i, j. Αυτό μπορεί να γίνει με κάποιον αλγόριθμο βελτιστοποίησης όπως η μέθοδος καθόδου κλίσης (Gradient Descent) η οποία είναι αρκετά δημοφιλής σαν επιλογή, ωστόσο δεν είναι κατάλληλη αν το μητρώο αξιολογήσεων R είναι πολύ μεγάλο, καθώς αυξάνει σημαντικά το υπολογιστικό κόστος. Στη μέθοδο καθόδου κλίσης τα μητρώα U και V αρχικοποιούνται με τυχαίες τιμές και στη συνέχεια σε κάθε επανάληψη υπολογίζεται το γινόμενο του U με το V και γίνεται η σύγκριση με το αρχικό μητρώο R. Αν το γινόμενο αυτό αποτελεί μια καλή προσέγγιση του R τότε η τρέχουσα επανάληψη τερματίζει, αλλιώς οι τιμές των U και V θα πρέπει να αλλάξουν ώστε να έχουμε μια καλύτερη προσέγγιση του R. Η διαδικασία συνεχίζεται έως ότου φτάσουμε σε κάποιο τοπικό ακρότατο στο εκτιμώμενο σφάλμα ανάμεσα στο μητρώο R και στο εκτιμώμενο μητρώο.

Μοντέλα βασισμένα στα τεχνητά νευρωνικά δίκτυα

Τα τελευταία χρόνια ο τεράστιος όγκος των πληροφοριών, σε συνδυασμό με τους περιορισμούς και τα προβλήματα που υπάρχουν στις κλασικές τεχνικές, οδήγησε τους ερευνητές στην υιοθέτηση τεχνικών βαθιάς μάθησης για τη δημιουργία νέων αλγορίθμων συστημάτων συστάσεων. Στο [19] που αποτελεί την πληρέστερη έρευνα που έχει γίνει έως τώρα, συγκεντρώνονται όλες οι τεχνικές βαθιάς μάθησης που χρησιμοποιούνται στα συστήματα συστάσεων.

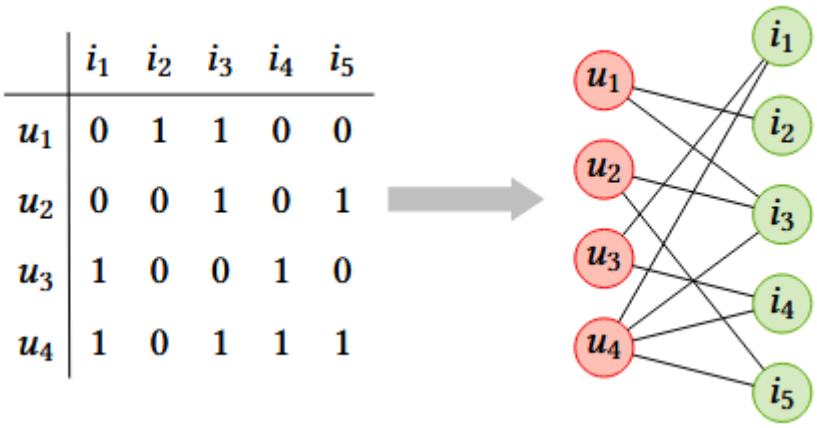
Μοντέλα βασισμένα στα γραφήματα

Τα τελευταία χρόνια, τα νευρωνικά δίκτυα γράφων (Graph Neural Networks) χρησιμοποιούνται για την επίλυση πολλών προβλημάτων της μηχανικής και της βαθιάς μάθησης. Προτού περιγράψουμε όμως τι είναι ένα Graph Neural Network και στη συνέχεια εξετάσουμε το πως αυτό χρησιμοποιείται στα συστήματα συστάσεων, χρειάζεται πρώτα να ορίσουμε ορισμένες βασικές έννοιες.

- **Γράφημα (graph):** στον χώρο των διακριτών μαθηματικών ένα γράφημα (ή γράφος) G = (V, E) είναι μια δομή η οποία αποτελείται από ένα σύνολο ακμών E και ένα σύνολο κορυφών (ή κόμβων) V [20].
- **Graph embeddings:** σύμφωνα με τους Goyal και Ferrara [21] ένα graph embedding είναι η μετατροπή των στοιχείων των γραφημάτων (ακμές, κορυφές, χαρακτηριστικά) σε διανύσματα, διατηρώντας παράλληλα τις ιδιότητες των γραφημάτων. Κατά αυτόν τον τρόπο μειώνεται η διάσταση κατά 1 και άρα μειώνεται το υπολογιστικό κόστος.

- **Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNNs):** στα Αναδρομικά Νευρωνικά Δίκτυα [22] η έξοδος δεν εξαρτάται μόνο από την είσοδο αλλά και από την προηγούμενη έξοδο και εσωτερική κατάσταση του δικτύου. Τα RNNs είναι κατάλληλα για δεδομένα που είναι διατεταγμένα σε μια σειρά, όπως οι χρονοσειρές και όπως είναι δομημένα μπορούν να λειτουργήσουν σε κατευθυνόμενα «γραμμικά» γραφήματα. Κατ' αυτόν τον τρόπο τα RNNs μπορούν να θεωρηθούν ένας ειδικός τύπος GNN, βοηθώντας μας έτσι να τα κατανοήσουμε καλύτερα.

Στη βαθιά μάθηση, ένα Graph Neural Network (GNN) [23] είναι ένας τύπος νευρωνικού δικτύου για την εξαγωγή συμπερασμάτων και γνώσης από δεδομένα που περιγράφονται από γραφήματα. Ένα γράφημα δίνεται ως είσοδος σε ένα GNN αφού πρώτα γίνει η μετατροπή που απαιτείται. Οι κορυφές και οι ακμές των γράφων στα GNNs αναπαριστώνται από διανύσματα και η σύνδεση μεταξύ κορυφών και ακμών από ένα μητρώο γειτνίασης (adjacency matrix). Σε κάθε επανάληψη, κάθε κόμβος C λαμβάνει πληροφορίες (embeddings) από τους γειτονικούς του κόμβους, υπολογίζει το άθροισμά τους και το μεταφέρει στο επόμενο επίπεδο μαζί με το embedding του. Έτσι δημιουργείται ένα νέο embedding, το οποίο περιέχει τις πληροφορίες του κόμβου C μαζί με τις πληροφορίες για τους γειτονικούς του κόμβους. Στην επόμενη επανάληψη ο κόμβοι μαθαίνουν επιπλέον και τις πληροφορίες της γειτονιάς των γειτονικών τους κόμβων (γείτονες 2ης τάξης). Η διαδικασία συνεχίζεται έως ότου οι κόμβοι μάθουν τις πληροφορίες για όλους τους υπόλοιπους κόμβους του γραφήματος. Τέλος, γίνεται η συλλογή όλων των embeddings και η άθροιση τους, δίνοντας μας ένα ενιαίο embedding για όλο το γράφημα. Ο στόχος των GNNs είναι η εκμάθηση αναπαραστάσεων/embeddings των οντοτήτων ή και των κόμβων χρησιμοποιώντας πληροφορίες γειτνίασης. Τα GNNs μπορούν επίσης να χρησιμοποιηθούν για την εκμάθηση αναπαραστάσεων ακμών και γραφημάτων.
Στα συστήματα συστάσεων τα Graph Neural Networks οι αλληλεπιδράσεις χρηστών-αντικειμένων (αγορές, κλικς, προσθήκη προϊόντων στο καλάθι, αξιολογήσεις) περιγράφονται από γραφήματα. Αναλυτικότερα, οι χρήστες και τα αντικείμενα αναπαριστώνται από κορυφές και οι αλληλεπιδράσεις από τις ακμές.



Εικόνα 2.6: Γράφημα χρηστών-αντικειμένων [Πηγή: `https://arxiv.org/pdf/2011.02260.pdf`]

2.2.2 Συστήματα βασισμένα στο περιεχόμενο (Content-based filtering systems)

Τα συστήματα βασισμένα στη συνεργατική διήθηση έχουν ως προαπαιτούμενο τη γνώση αλληλεπιδράσεων μεταξύ χρηστών και αντικειμένων. Ένα ερώτημα που γεννάται εδώ, είναι τι συμβαίνει στην περίπτωση που δεν είναι διαθέσιμη αυτή η γνώση. Μια λύση σε αυτό το ζήτημα αποτελούν τα συστήματα βασισμένα στο περιεχόμενο. Τα συστήματα αυτά βασίζονται στα περιγραφικά χαρακτηριστικά των αντικειμένων, όπως το όνομα, η περιγραφή, το είδος κτλ., προκειμένου να δημιουργήσουν νέες συστάσεις. Πιο συγκεκριμένα, σε αυτήν τη μέθοδο οι αλγόριθμοι προτείνουν στον χρήστη παρόμοια αντικείμενα με αυτά που έχει αλληλεπιδράσει στο παρελθόν. Σύμφωνα με τον Falk [9] ένα σύστημα βασισμένο στο περιεχόμενο αποτελείται από τα εξής μέρη:

1. Αναλυτής περιεχομένου (Content analyzer) — δημιουργεί ένα μοντέλο βασισμένο στο περιεχόμενο. Κατά μία έννοια, δημιουργεί ένα προφίλ για κάθε αντικείμενο. Σε αυτό το στάδιο γίνεται η προεπεξεργασία των δεδομένων και η μετατροπή τους σε μορφή κατάλληλη για το επόμενο στάδιο και η εκπαίδευση του μοντέλου.

2. Δημιουργία προφίλ χρήστη (User profiler) — δημιουργεί το προφίλ ενός χρήστη, με τεχνικές μηχανικής μάθησης. Ορισμένες φορές το προφίλ αυτό αποτελείται από μια απλή λίστα των αντικειμένων που έχει καταναλώσει ο χρήστης.

3. Ανάκτηση αντικειμένων (Item retriever) — συγκρίνει τα προφίλ των χρηστών με τα προφίλ των αντικειμένων, με σκοπό την εύρεση και την ανάκτηση σχετικών αντικειμένων για κάθε χρήστη. Εάν το προφίλ του χρήστη είναι μια λίστα αντικειμένων, αυτή η λίστα προσπελαύνεται και βρίσκονται παρόμοια αντικείμενα για κάθε αντικείμενο που υπάρχει στην λίστα του χρήστη.

2.2.3 Υβριδικά συστήματα (Hybrid systems)

Τα συστήματα αυτά αποτελούν συνδυασμό των συστημάτων που βασίζονται στο περιεχόμενο και των συστημάτων που βασίζονται στη συνεργατική διήθηση. Το πιο γνωστό παράδειγμα υβριδικού συστήματος είναι το Netflix, το οποίο χρησιμοποιεί μοντέλα συνεργατικής διήθησης για να συγκρίνει τις αναζητήσεις και τις προβολές ταινιών παρόμοιων χρηστών και μοντέλα βασισμένα στο περιεχόμενο για να προτείνει ταινίες που έχουν κοινά χαρακτηριστικά με ταινίες που ο χρήστης έχει δώσει υψηλές αξιολογήσεις.

2.2.4 Προβλήματα στα συστήματα συστάσεων

Σε αυτή την υποενότητα αναφέρονται τα πιο σημαντικά ζητήματα στα συστήματα συστάσεων, τα οποία επηρεάζουν σημαντικά την απόδοση των αλγορίθμων και αποτελούν αιτία εμφάνισης επιπλέον προβλημάτων. Τα ζητήματα αυτά απασχολούν εδώ και πολλά χρόνια την επιστημονική κοινότητα.

Χαρακτηριστικά των δεδομένων

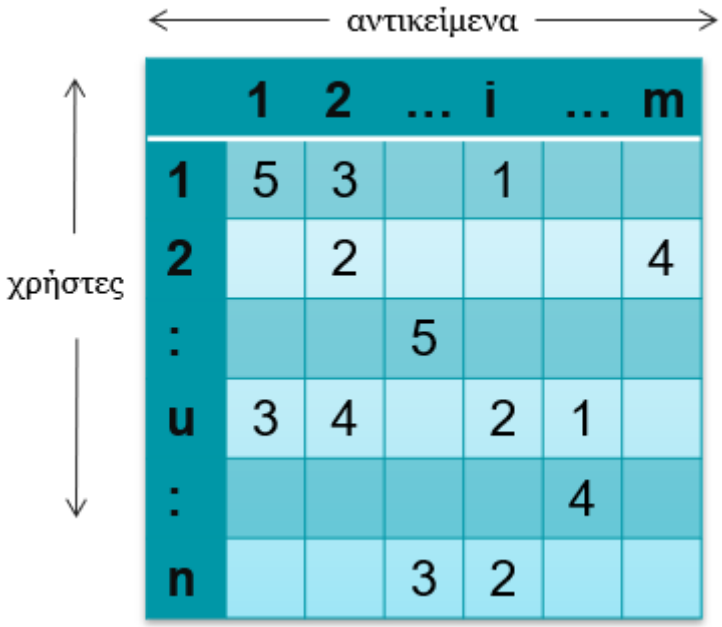
Στο [24] και στο [25] εξετάζεται το πως τα χαρακτηριστικά των δεδομένων μπορούν να επηρεάσουν την απόδοση των συστημάτων συστάσεων. Ως χαρακτηριστικά των δεδομένων ορίζονται:

1. *RatingSpace (RS)* $= |U| \times |I|$, το μέγεθος του χώρου των αξιολογήσεων

2. *User Item Ratio (UIR)* $= \frac{|User|}{|Item|}$, ο λόγος χρηστών προς αντικείμενα αποτυπώνει το σχήμα του χώρου αξιολογήσεων.
3. *Rating Per User (RPU)* $= \frac{|Ratings|}{|Users|}$, ο λόγος των αξιολογήσεων προς χρήστες
4. *Rating Per Item (RPI)* $= \frac{|Ratings|}{|Items|}$, ο λόγος των αξιολογήσεων προς αντικείμενα
5. *Αραιότητα του μητρώου αξιολογήσεων χρηστών-αντικειμένων*, το οποίο αποτελεί ένα πολύ σημαντικό πρόβλημα στα συστήματα συστάσεων και για τον λόγο αυτό αναλύεται περαιτέρω στη συνέχεια

Αραιότητα (Sparsity)

Ένα από τα πιο σημαντικά, αν όχι το πιο σημαντικό, προβλήματα στα συστήματα συστάσεων το οποίο απασχολεί τις τελευταίες δύο δεκαετίες την επιστημονική κοινότητα είναι η αραιότητα των δεδομένων. Έστω το μητρώο χρηστών-αντικειμένων, το οποίο στις γραμμές περιέχει τους χρήστες και στις στήλες τα αντικείμενα. Κάθε κελί περιέχει την αξιολόγηση ενός χρήστη για ένα αντικείμενο. Για παράδειγμα στην Εικόνα 2.7 στο κελί (1,1) βρίσκεται η αξιολόγηση του χρήστη 1 για το αντικείμενο 1 και είναι ίση με 5. Ένα αρκετά σημαντικό ζήτημα που προκύπτει εδώ, είναι πως στις



Εικόνα 2.7: Παράδειγμα μητρώου χρηστών-αντικειμένων

περισσότερες περιπτώσεις το μητρώο αυτό είναι πολύ αραιό, δηλαδή υπάρχουν πολλά κενά κελιά, καθώς καταλαβαίνουμε πως δεν γίνεται όλοι οι χρήστες να έχουν αξιολογήσει όλα τα αντικείμενα σε ένα σύνολο δεδομένων. Η αραιότητα του μητρώου μετριέται σύμφωνα με τον τύπο (με # συμβολίζεται ο αριθμός):

$$\text{Αραιότητα} = 1 - \frac{\#\text{αξιολογήσεων}}{\#\text{χρηστών} \times \#\text{αντικειμένων}}$$

Συνεπώς, μειώνεται η πιθανότητα να βρεθούν συσχετίσεις ανάμεσα σε χρήστες και αντικείμενα, κάτι που με τη σειρά του οδηγεί σε χαμηλής ποιότητας προβλέψεις. Για τον λόγο αυτό αρκετές είναι οι προσπάθειες τα τελευταία χρόνια να μετριαστεί αυτό το πρόβλημα σε διαφορετικά συστήματα συστάσεων [26].

Cold start

Στα συστήματα συστάσεων απαραίτητη προϋπόθεση είναι η ύπαρξη αρκετών δεδομένων και στοιχείων που να σχετίζονται με τα αντικείμενα και τους χρήστες, προκειμένου να μπορέσουν να βρουν τις απαραίτητες συσχετίσεις ανάμεσα στους χρήστες, στα αντικείμενα και στον συνδυασμό χρηστών και στα αντικειμένων, ώστε να δημιουργήσει συστάσεις προς τους χρήστες με όσο καλύτερη ακρίβεια γίνεται. Ένα ζήτημα που προκύπτει εδώ είναι το λεγόμενο cold start [27], το οποίο συμβαίνει κατά κύριο λόγο όταν εισάγεται ένας νέος χρήστης ή ένα νέο αντικείμενο στο σύστημα. Υπάρχουν 3 περιπτώσεις κατά τις οποίες μπορεί να προκύψει αυτό το ζήτημα [28]:

- **Νέος χρήστης:** όταν ένας χρήστης κάνει εγγραφή στο σύστημα, στην αρχή το σύστημα δεν γνωρίζει τίποτα για εκείνον. Πιο συγκεκριμένα δεν υπάρχει ιστορικό, δηλαδή παλαιότερες αλληλεπιδράσεις όπως αξιολογήσεις που να έχει πραγματοποιήσει σε αντικείμενα, προβολές, κλικ ή αγορές αν πρόκειται για σύστημα ηλεκτρονικού εμπορίου.
- **Νέο αντικείμενο:** αντίστοιχα όταν ένα νέο αντικείμενο εισάγεται στο σύστημα τότε θα έχει ελάχιστες ή και καθόλου αλληλεπιδράσεις, συνεπώς σύμφωνα με τον τρόπο που λειτουργούν οι αλγόριθμοι συστημάτων συστάσεων, αυτό το αντικείμενο δεν θα προταθεί ποτέ στους χρήστες αν δεν υπάρχουν καθόλου αλληλεπιδράσεις. Εάν υπάρχουν ελάχιστες αλληλεπιδράσεις, τότε η ακρίβεια θα είναι πολύ κακή σε σχέση με τα αντικείμενα με πολλές αλληλεπιδράσεις. Αυτό οδηγεί στο φαινόμενο του popularity bias που περιγράφεται λεπτομερώς σε επόμενα κεφάλαια.
- **Νέα κοινότητα:** αναφέρεται στην αρχική κατάσταση του συστήματος όπου όλοι οι χρήστες και τα αντικείμενα είναι καινούρια. Στην ουσία είναι ο συνδυασμός των δύο ανωτέρω περιπτώσεων, του νέου χρήστη και του νέου αντικειμένου.

Κεφάλαιο 3

Ηθικά ζητήματα

3.1 Ηθικά ζητήματα στη μηχανική μάθηση

3.1.1 Δικαιοσύνη (Fairness)

Αρχικά, θα πρέπει να ορίσουμε τι είναι δικαιοσύνη στον πραγματικό κόσμο και στη λήψη αποφάσεων από τους ανθρώπους και στη συνέχεια θα προχωρήσουμε σε μια έρευνα σχετικά με το πως μεταφράζονται όλα αυτά στον τομέα της τεχνητής νοημοσύνης και πιο συγκεκριμένα στη μηχανική μάθηση.
Το πλήθος των ορισμών για την έννοια δικαιοσύνη και η απουσία ενός γενικού ορισμού, αποτελούν μια αρκετά καλή ένδειξη για το πόσο σύνθετο είναι αυτό το πρόβλημα και επομένως δύσκολο να αντιμετωπιστεί. Υπάρχουν πάρα πολλοί ορισμοί για το fairness τόσο μαθηματικοί όσο και θεωρητικοί. Ενδεικτικά αναφέρουμε:

- Στη φιλοσοφία υπάρχουν πολλές και διαφορετικές προσεγγίσεις αυτής της έννοιας. Ο Πλάτωνας στην Πολιτεία αναφέρει πως δικαιοσύνη είναι «το να έχει κανείς και να κάνει τη δική του τη δουλειά και ό,τι του ταιριάζει». Στο ίδιο έργο η δικαιοσύνη προβάλλεται από διαφορετικές οπτικές. Σε μεταφυσικό πλαίσιο η δικαιοσύνη προϋποθέτει τη διαίρεση της ανθρώπινης ψυχής σε 3 μέρη στο επιθυμητικό, το θυμοειδές και το λογιστικό. Στο πλαίσιο της ηθικής είναι η επίτευξη της ισορροπίας αυτών των τριών μερών της ψυχής υπό τον έλεγχο του λογιστικού. Ο Αριστοτέλης στα Ηθικά Νικομάχεια διακρίνει τρία είδη δικαιοσύνης: τη διανεμητική που έχει σχέση με τις διανομές τιμητικών διακρίσεων, χρημάτων ή γενικά αγαθών που μοιράζονται σε όσους ζουν σε ένα συγκεκριμένο πολιτειακό καθεστώς, τη διορθωτική (ή επανορθωτική) που σχετίζεται με τις σχέσεις μεταξύ των ανθρώπων και διακρίνεται σε ακούσιες (χωρίς την θέληση των ανθρώπων) και εκούσιες (με την θέληση των ανθρώπων) σχέσεις και έχει ως στόχο την εύρεση του μέσου μεταξύ «ζημιάς» και «κέρδους» και της αμοιβαιότητας ως μια αναλογική ανταπόδοση αμοιβαίων υπηρεσιών.

- Στη νομική ορολογία, η δικαιοσύνη είναι η τήρηση και η εφαρμογή των νόμων με αμερόληπτο τρόπο.

- Στην επιστήμη των υπολογιστών και στα μαθηματικά δικαιοσύνη είναι η απουσία οποιασδήποτε προκατάληψης ή ευνοιοκρατίας προς ένα άτομο ή μια ομάδα που βασίζεται στα εγγενή ή επίκτητα χαρακτηριστικά [29].

Στη μηχανική μάθηση υπάρχουν αρκετοί ορισμοί, ταξινομήσεις και μετρικές για τη δικαιοσύνη. Στη συνέχεια θα περιγράψουμε τους πιο σημαντικούς από αυτούς. Μια προσέγγιση είναι ο ορισμός ανάλογα με το ποιος επιθυμούμε να έχει δίκαια αποτελέσματα μια ομάδα, ένα άτομο ή μια υποομάδα; Σύμφωνα με αυτό το σκεπτικό ορίζονται τρεις τύποι δικαιοσύνης: η **ομαδική δικαιοσύνη (group fairness)**, η **ατομική δικαιοσύνη (individual fairness)** και η **δικαιοσύνη υποομάδας (subgroup fairness)**.

Στην ομαδική δικαιοσύνη ή αλλιώς statistical parity ή equal acceptance rate ή benchmarking, ο στόχος είναι η ίση μεταχείριση διαφορετικών ομάδων ατόμων με κοινά προστατευόμενα χαρακτηριστικά, μια ομάδα ατόμων με κοινά προστατευόμενα χαρακτηριστικά είναι για παράδειγμα οι γυναίκες. Η έννοια αυτή έγινε αρκετά δημοφιλής ύστερα από μια έρευνα που διεξήχθη το 2016 για τη μεροληψία που εισάγει εναντίον των Αφροαμερικάνων ένα σύστημα πρόβλεψης μελλοντικών κρατούμενων [30]. Πιο συγκεκριμένα, το 1998 αναπτύχθηκε στις Η.Π.Α. από την εταιρεία Northpointe (νυν equivant) το Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), ένα εργαλείο πρόβλεψης υποτροπής κατηγορουμένων εντός δύο ετών από την αξιολόγησή τους, με βασικά κριτήρια το ποινικό τους μητρώο και 137 χαρακτηριστικά τους. Τα συστήματα αυτά χρησιμοποιούνται για την αξιολόγηση των ποσοστών υποτροπής και την απόδοση βαθμολογιών που υποδεικνύουν εάν ένας συγκεκριμένος κατηγορούμενος έχει χαμηλό, μεσαίο ή υψηλό κίνδυνο να διαπράξει εγκλήματα στο μέλλον. Το COMPAS ύστερα από έρευνες αποδείχθηκε πως δίνει λανθασμένα περισσότερες πιθανότητες (σχεδόν διπλάσιες) στους Αφροαμερικανούς να τελέσουν ξανά κάποιο αδίκημα από ότι δίνει στους λευκούς και πως είναι δύο φορές πιο πιθανό να προσδιορίσει εσφαλμένα τους λευκούς κατηγορούμενους ως χαμηλού κινδύνου όσον αφορά τη διάπραξη μελλοντικών εγκλημάτων. Έκτοτε, έχουν βρεθεί αρκετά συστήματα στα οποία υπάρχει τέτοιου είδους μεροληψία. Μαθηματικά, το statistical parity ή Demographic Parity ορίζει ότι ένας προγνώστης (predictor) είναι αμερόληπτος εάν η πρόβλεψή του $\hat{y}$ είναι στατιστικά ανεξάρτητη από το προστατευόμενο χαρακτηριστικό (protected attribute) p, δηλαδή ότι κάθε ομάδα έχει την ίδια πιθανότητα να ταξινομηθεί με το θετικό αποτέλεσμα:

$$\Pr(\hat{y}|p) = \Pr(\hat{y})$$

Ορισμός

Προστατευόμενα χαρακτηριστικά (Protected attributes): Κατηγορία ατόμων που τυπικά υπόκεινται σε διακρίσεις σε έναν πληθυσμό. Σύμφωνα με τη Διεθνή Αμνηστία [31] αυτά είναι: η ηλικία, η αναπηρία, το φύλο, ο σεξουαλικός προσανατολισμός, η θρησκεία ή τα πιστεύω, η φυλή, η οικογενειακή κατάσταση, η εγκυμοσύνη και η μητρότητα και ο επαναπροσδιορισμός φύλου.

Στην ατομική δικαιοσύνη, όπως αυτή παρουσιάστηκε στο [32] διασφαλίζεται ότι δίνονται παρόμοιες προβλέψεις σε παρόμοιους χρήστες και πως αυτοί οι χρήστες λαμβάνουν την ίδια αντιμετώπιση. Τα άτομα (individuals) ορίζονται με βάση μια μετρική απόστασης η οποία αναπαριστά πόσο όμοια είναι τα άτομα μεταξύ τους όσον αφορά τα χαρακτηριστικά που σχετίζονται με το πλαίσιο λήψης αποφάσεων και γίνεται η υπόθεση πως υπάρχουν τυχαίες απεικονίσεις (randomized mappings) από τα άτομα σε πιθανοτικές κατανομές επί των αποτελεσμάτων. Οι κατανομές που ανατίθενται σε παρόμοιους χρήστες πρέπει να είναι παρόμοιες. Η έννοια αυτή βρίσκει εφαρμογή σε περιπτώσεις όπως μαθητές που έχουν κάνει αίτηση εισαγωγής σε κάποιο εκπαιδευτικό ίδρυμα, άτομα που κάνουν αίτηση για δουλειά και άτομα που κάνουν αίτηση για χορήγηση δανείου σε κάποια τράπεζα.

Τέλος, η δικαιοσύνη υποομάδας [33] χρησιμοποιεί τις έννοιες της ομαδικής και της ατομικής δικαιοσύνης. Σύμφωνα με αυτή, εφαρμόζονται κλασικές στατιστικές έννοιες της δικαιοσύνης σε μεγάλες συλλογές υποομάδων που ορίζονται από ένα σύνολο συναρτήσεων των προστατευόμενων χαρακτηριστικών.
Δύο ακόμη θεωρήσεις δίνονται στο [34]. Σύμφωνα με την πρώτη θεώρηση, η οποία ονομάζεται *equalized odds*, ένας predictor $\hat{Y}$ ικανοποιεί το equalized odds σε σχέση με το προστατευόμενο χαρακτηριστικό A και το αποτέλεσμα Y, αν ο $\hat{Y}$ και το A είναι ανεξάρτητα από το Y. Ένας κατηγοριοποιητής (classifier) h (X) πρέπει να έχει ίσα ποσοστά αληθώς θετικών και ψευδώς θετικών (true positive και false positive rates) για όλες τις ομάδες:

$$\mathrm{P}\left[\hat{Y} = 1 \mid A = 0,\ Y = y\right] = P\left[\hat{Y} = 1 | A = 1, Y = y\right],\ \forall a, y \tag{3.1}$$

$$\begin{cases} y = 0, \text{ ποσοστά ψευδώς θετικών} \\ y = 1, \text{ ποσοστά αληθώς θετικών} \end{cases}$$

Πολλές φορές όμως μας ενδιαφέρει περισσότερο το ποσοστό των αληθώς θετικών από το ποσοστό των αληθώς αρνητικών, όπως είναι η εργασία της πρόβλεψης του αν η αίτηση ενός μαθητή για την εισαγωγή του σε ένα εκπαιδευτικό ίδρυμα γίνει δεκτή. Σε αυτό το πλαίσιο ορίζεται μια παραλλαγή, μια πιο «χαλαρή» εκδοχή του equalized odds, η *ίση ευκαιρία (equal opportunity)*, όπου θέτουμε $y = 1$, καθώς θέλουμε μόνο το ποσοστό των αληθώς θετικών. Έτσι, θα λέμε ότι ο δυαδικός predictor $\hat{Y}$ ικανοποιεί την equal opportunity ως προς το A και το Y αν:

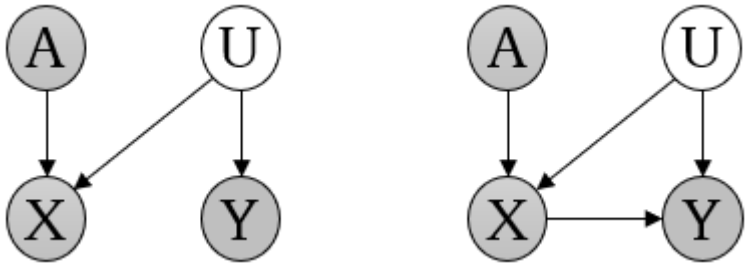
$$\Pr\left[\hat{Y} = 1 \mid A = 0,\ Y = 1\right] = \Pr\left[\hat{Y} = 1 \mid A = 1,\ Y = 1\right], \forall a, y \tag{3.2}$$

Η Counterfactual fairness [35] (δικαιοσύνη μέσω αντιπαραδειγμάτων) αποτελεί μια αρκετά διαφορετική προσέγγιση στον ορισμό της δικαιοσύνης. Πιο συγκεκριμένα, εξηγεί τον αντίκτυπο της μεροληψίας μέσω ενός αιτιώδους γράφου (causal graph), όπως αυτοί στην Εικόνα 3.1 που περιγράφουν δύο παραδείγματα που δίνονται στο [35]. Στον πρώτο γράφο υποθέτουμε ότι υπάρχει ένας παράγοντας που δεν έχει παρατηρηθεί που αντιστοιχεί στην επιθετική οδήγηση (U), ο οποίος αυξάνει την πιθανότητα των οδηγών να τους συμβεί κάποιο ατύχημα και έχεις ως αποτέλεσμα τα άτομα να προτιμούν τα κόκκινα αυτοκίνητα (η μεταβλητή X που έχει παρατηρηθεί). Επιπλέον, άτομα που ανήκουν σε μια συγκεκριμένη φυλή (race) A είναι πιο πιθανό να οδηγούν κόκκινα αυτοκίνητα. Ωστόσο, αυτά τα άτομα δεν είναι πιο πιθανό να οδηγούν επιθετικά ή να εμπλακούν σε ατυχήματα σε σχέση με άλλη άτομα. Έτσι, η χρήση του χαρακτηριστικού του κόκκινου αυτοκινήτου X για την πρόβλεψη του ποσοστού ατυχήματος Y φαίνεται να είναι μια άδικη πρόβλεψη. Η αντιπαραστατική δικαιοσύνη συμφωνεί με αυτήν την έννοια: η αλλαγή του A ενώ διατηρείται σταθερό το U θα αλλάξει επίσης το X και, κατά συνέπεια, το $\hat{Y}$. Ο δεύτερος γράφος περιγράφει το εξής παράδειγμα. Η κυβέρνηση μιας πόλης θέλει να εκτιμήσει τα ποσοστά εγκληματικότητας ανά γειτονιά για να διαθέσει πόρους αστυνόμευσης. Τα δεδομένα που έχει στη διάθεσή της προέκυψαν συγχωνεύοντας ένα μητρώο κατοίκων που περιέχει τη γειτονιά τους X και τη φυλή A, με αστυνομικά αρχεία συλλήψεων, δίνοντας σε κάθε κάτοικο μια δυαδική ετικέτα με $Y = 1$ που υποδεικνύει το ποινικό μητρώο σύλληψης. Η τοποθεσία X εξαρτάται από την A. Οι τοποθεσίες X με περισσότερους αστυνομικούς πόρους έχουν μεγαλύτερο αριθμό συλλήψεων Y. Και τέλος, το U αντιπροσωπεύει το σύνολο των κοινωνικοοικονομικών παραγόντων και των πρακτικών αστυνόμευσης που επηρεάζουν τόσο το πού μπορεί να ζήσει ένα άτομο, όσο και πόσο πιθανό είναι να συλληφθεί και να κατηγορηθεί. Σύμφωνα με αυτή την θεώρηση, ένα μοντέλο είναι δίκαιο εάν η πρόβλεψή του για ένα συγκεκριμένο άτομο ή μια συγκεκριμένη

ομάδα στον πραγματικό κόσμο παραμένει η ίδια με αυτή που θα έδινε σε έναν «counterfactual» κόσμο όπου το άτομο ανήκει σε μία διαφορετική δημογραφική ομάδα. Έστω A τα προστατευόμενα χαρακτηριστικά, X οι μεταβλητές που έχουν παρατηρηθεί και δεν αποτελούν προστατευόμενα χαρακτηριστικά και Y η πρόβλεψη που παράγει ως αποτέλεσμα. Τότε ένας προγνώστης (predictor) $\hat{Y}$ θεωρείται counterfactually δίκαιος, εάν για οποιαδήποτε X=x και A=a ισχύει ότι:

$$\mathbb{P}\left(\hat{Y}_{A\leftarrow a} = y \middle| X = x,\ \ A = a\right) = \mathbb{P}(Y_{A\leftarrow a'} = y | X = x, A = a), \forall y\ \kappa\alpha\iota\ a' \neq a \tag{3.3}$$

Δηλαδή, μια αλλαγή στο A, διατηρώντας ταυτόχρονα τα πράγματα που δεν είναι αιτιολογικά εξαρτημένα (causally dependent) με το A σταθερά, δεν θα επιφέρει αλλαγή στην κατανομή του $\hat{Y}$.



Εικόνα 3.1: Παραδείγματα αιτιωδών γράφων

Αναφέραμε τις κυριότερες και πιο γνωστές μετρικές, ωστόσο επειδή τα τελευταία χρόνια έχουν δημιουργηθεί πάρα πολλές μετρικές, δεν είναι δυνατό να αναλυθούν όλες σε αυτή την εργασία. Περισσότερες μετρικές μπορούν να βρεθούν στο [36].

3.1.2 Διαφάνεια (Transparency) και Λογοδοσία (Accountability)

Η λογοδοσία αναφέρεται στο εάν μια απόφαση έχει ληφθεί ακολουθώντας διαδικασίες ενός προτύπου και τον ορισμό ενός υπεύθυνου εάν αυτά τα πρότυπα δεν πληρούνται [37]. Εισάγει δηλαδή και την σημασία του ανθρώπινου παράγοντα ως απαραίτητη προϋπόθεση για την αμερόληπτη και σωστή λειτουργία των συστημάτων. Σε αυτή την κατεύθυνση συμβάλλει αρκετά, η αύξηση της διαφάνειας των αλγορίθμων [38]. Για τον όρο διαφάνεια τον αλγορίθμων υπάρχουν πολλοί ορισμοί και θεωρήσεις. Ωστόσο αυτό από μόνο του δεν αρκεί. Ο Frank Pasquale στο βιβλίο του “The black box society” [39] αναφέρεται στο πως οι επιχειρήσεις αλλά και οι κυβερνήσεις συλλέγουν και επεξεργάζονται τεράστιους όγκους δεδομένων, μέσω των οποίων δημιουργούνται εξατομικευμένα προφίλ και λαμβάνονται αποφάσεις για βασικές πτυχές της καθημερινότητας και της ζωής μας, όπως για παράδειγμα η λήψη ενός δανείου ή η επιλογή μαθητών που θα φοιτήσουν σε ένα κολλέγιο [40], χωρίς να μας δίνουν την δυνατότητα για λεπτομερή έλεγχο της λειτουργίας των αλγορίθμων που επεξεργάζονται τα δεδομένα και λαμβάνουν αποφάσεις. Παράλληλα ο συγγραφέας επισημαίνει, ότι «εχθρός» της διαφάνειας είναι η μεγάλη πολυπλοκότητα. Η λογοδοσία στην τεχνητή νοημοσύνη όμως πέρα από τη διαδικασία λήψης αποφάσεων από έναν αλγόριθμο και τον ορισμό ενός υπεύθυνου για αυτή, σχετίζεται και με την παροχή επεξηγήσεων (explanations).

3.1.3 Ερμηνευσιμότητα (Interpretability) και επεξηγησιμότητα (Explainability)

Σε ένα σύστημα μηχανικής μάθησης ένα ζήτημα που υπάρχει είναι ότι τα μοντέλα λειτουργούν σαν ένα μαύρο κουτί (black-box), ειδικά σε ιδιαίτερα περίπλοκα συστήματα όπως είναι τα νευρωνικά δίκτυα. Σύμφωνα με τους Velez και Kim [41] η ερμηνευσιμότητα (interpretability) ορίζεται ως «η ικανότητα να εξηγήσουμε ένα σύστημα ή να το παρουσιάσουμε με έναν τρόπο που είναι κατανοητός στους ανθρώπους», ενώ πιο ειδικά στον τομέα της τεχνητής νοημοσύνης ερμηνεύσιμο είναι *ένα σύστημα στο οποίο μπορούμε μελετήσουμε και να κατανοήσουμε πως οι είσοδοι αντιστοιχίζονται μαθηματικά στις εξόδους του [42]*. Πιο συγκεκριμένα, θέλουμε να μάθουμε τι προκάλεσε μια συγκεκριμένη απόφαση που ελήφθη από ένα σύστημα. Σύμφωνα με τον Molnar [43] οι δύο κύριες κατηγορίες μεθόδων ερμηνευσιμότητας είναι οι αγνωστικές μέθοδοι ως προς το μοντέλο (model-agnostic), στις οποίες διαχωρίζουμε τις εξηγήσεις από το μοντέλο μηχανικής μάθησης, μπορούν να εφαρμοστούν σε οποιοδήποτε μοντέλο μηχανικής μάθησης και εφαρμόζονται μετά την εκπαίδευση του μοντέλου και οι γνωστικές μέθοδοι ως προς το μοντέλο (model-specific) οι οποίες μπορούν να εξηγήσουν μόνο κάποια συγκεκριμένη κλάση μοντέλων. Πολλές φορές η έννοια της ερμηνευσιμότητας συγχέεται με την έννοια της επεξηγησιμότητας (explainability). Παράλληλα, αρκετοί είναι οι ορισμοί που έχουν προταθεί για την επεξηγησιμότητα και αρκετές οι διαφορετικές προσεγγίσεις που ακολουθούνται για τη δημιουργία επεξηγήσιμων συστημάτων μηχανικής μάθησης. Ένας, από αυτούς τους ορισμούς στο πεδίο της τεχνητής νοημοσύνης και κατ'επέκταση στο πεδίο της μηχανικής μάθησης δίνεται στο [44]: «Ένα σύστημα τεχνητής νοημοσύνης είναι επεξηγήσιμο εάν είναι εγγενώς ερμηνεύσιμο ή εάν το μη ερμηνεύσιμο μοντέλο εργασίας συμπληρώνεται με μια ερμηνεύσιμη και πιστή εξήγηση (εδώ το σύστημα τεχνητής νοημοσύνης περιέχει επίσης μια εκ των υστέρων (post-hoc) εξήγηση)». Τα μοντέλα που είναι επεξηγήσιμα είναι και ερμηνεύσιμα εξ'ορισμού, όμως το αντίστροφο δεν ισχύει πάντα.

3.1.4 Μεροληψία (Bias)

Η μεροληψία σχετίζεται άμεσα με την δικαιοσύνη και πολλές φορές τα όρια ανάμεσα σε αυτές τις δύο έννοιες είναι αρκετά δυσδιάκριτα για αρκετούς ανθρώπους.
Στο [29] τα είδη μεροληψίας κατηγοριοποιούνται σύμφωνα με τον βρόχο ανατροφοδότησης που σχηματίζεται στα συστήματα μηχανικής μάθησης ανάμεσα στα δεδομένα, τον αλγόριθμο και τις αλληλεπιδράσεις του χρήστη. Έτσι, διακρίνεται η μεροληψία των δεδομένων τα οποία δίνονται στον αλγόριθμο, η μεροληψία που εισάγεται από τους αλγορίθμους και μπορεί να επηρεάσει τη συμπεριφορά των χρηστών και η μεροληψία που εισάγεται στα δεδομένα από τους χρήστες.

Τύποι αλγορίθμων μετριασμού μεροληψίας

Οι τεχνικές μετριασμού της μεροληψίας μπορούν να κατηγοριοποιηθούν σε pre-processing, in-processing και post-processing, όπως αναφέρουν οι Caton και Haas [36]. Οι pre-processing τεχνικές λαμβάνουν χώρα πριν την εκπαίδευση του μοντέλου και προσπαθούν να μετριάσουν την μεροληψία που υπάρχει στα δεδομένα που δίνονται στο μοντέλο για την εκπαίδευσή του. Θεωρείται αρκετά ευέλικτη τεχνική καθώς οι αλγόριθμοι που την χρησιμοποιούν δεν επηρεάζουν καθόλου το μοντέλο που θα χρησιμοποιηθεί. Οι in-processing τεχνικές εφαρμόζονται κατά τη διάρκεια της εκπαίδευσης του μοντέλου και τροποποιούν ήδη υπάρχοντες αλγορίθμους προκειμένου αυτοί να είναι πιο δίκαιοι στις αποφάσεις που παίρνουν. Η τελευταία τεχνική, η post-processing έχει ως στόχο να κάνει πιο

δίκαιες τις προβλέψεις που έχει παράξει ένας αλγόριθμος, άρα εφαρμόζεται μετά την εκπαίδευση του μοντέλου, είναι και αυτή αρκετά ευέλικτη τεχνική καθώς είναι ανεξάρτητη από το μοντέλο και δεν χρειάζεται να αποκτήσει πρόσβαση σε αυτό παρά μόνο στα αποτελέσματα. Φυσικά υπάρχει και ο συνδυασμός δύο ή και τριών ειδών.

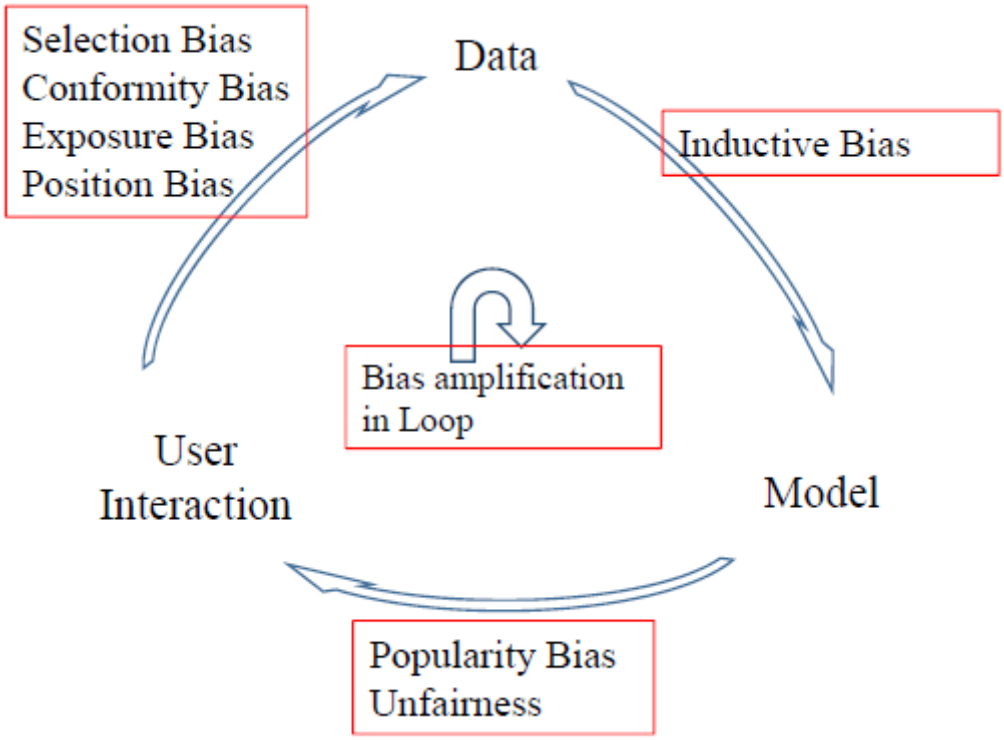
Ένα από τα πιο γνωστά και πιο πλήρη ανοικτού κώδικα (open-source) εργαλεία για εύρεση και μετριασμό της μεροληψίας μέσω διάφορων τεχνικών και αλγορίθμων της μεροληψίας, αποτελεί το AI Fairness 360 (AIF360) της IBM [45] σε γλώσσα python και R. Το εργαλείο αυτό, περιέχει 71 μετρικές μεροληψίας, 9 αλγορίθμους μετριασμού της μεροληψίας, καθώς και περιγραφή κάθε μετρικής σε γλώσσα απλή και κατανοητή από όλους τους χρήστες. Θα πρέπει να σημειωθεί πως έχουν αναπτυχθεί αρκετά παρόμοια εργαλεία, με τα πιο γνωστά από αυτά να είναι το FairLearn [46] της Microsoft και το Aequitas [47] που αναπτύχθηκε από το πανεπιστήμιο του Σικάγο των Η.Π.Α..
Είναι γεγονός ότι έως σήμερα έχει διεξαχθεί αρκετά εκτενής έρευνα σχετική με ζητήματα δικαιοσύνης, λογοδοσίας, διαφάνειας και ερμηνευσιμότητας, ενώ αρκετές είναι και οι προσπάθειες για τον μετριασμό της μεροληψίας όπως προαναφέρθηκε, σε διάφορους τομείς της μηχανικής μάθησης (κατηγοριοποίηση, παλινδρόμηση, Graph Embedding/Clustering, εκμάθηση αναπαραστάσεων (representation learning), επεξεργασία φυσικής γλώσσα (NLP) είναι τα κυριότερα). Σε αυτούς τους τομείς ωστόσο δεν συμπεριλαμβάνονται τα Συστήματα συστάσεων (recommender/recommendation systems). Παρόλα αυτά, τα τελευταία χρόνια αρχίζουν και γίνονται αξιόλογες προσπάθειες και σε αυτόν τον τομέα.

3.2 Ηθικά ζητήματα στα συστήματα συστάσεων

Στο [48] παρουσιάζονται διάφορα είδη μεροληψίας που συναντάμε στα συστήματα συστάσεων καθώς και διάφοροι αλγόριθμοι και τεχνικές για τον μετριασμό της. Επιπροσθέτως, παρουσιάζεται ο κύκλος ζωής των συστημάτων συστάσεων ως ένας βρόχος ανατροφοδότησης (feedback loop) ανάμεσα σε τρεις κυρίαρχους παράγοντες: τους χρήστες, τα δεδομένα και το μοντέλο, όπως φαίνεται και στην Εικόνα 3.2. Η μεροληψία στα δεδομένα μπορεί να κατηγοριοποιηθεί περαιτέρω σε μεροληψία στο explicit feedback και στο implicit feedback, δύο έννοιες που είδαμε αναλυτικά στην Ενότητα 2.2. Τα είδη μεροληψίας που παρουσιάζονται συνοπτικά είναι:

Μεροληψία επιλογής (Selection Bias)
Η μεροληψία επιλογής, κάνει την εμφάνισή της όταν οι χρήστες είναι ελεύθεροι να επιλέξουν ποια αντικείμενα θα αξιολογήσουν και έτσι οι παρατηρηθείσες αξιολογήσεις (observed ratings) δεν είναι αντιπροσωπευτικό δείγμα όλων των αξιολογήσεων, δηλαδή τα δεδομένα αξιολογήσεων (rating data) συχνά είναι missing not at random (MNAR) το οποίο σημαίνει πως τα δεδομένα λείπουν για λόγους που δεν είναι γνωστοί σε εμάς.

Μεροληψία κομφορμισμού (Conformity Bias)
Ο κομφορμισμός είναι μια αρκετά γνωστή έννοια στην επιστήμη της ψυχολογίας και αναφέρεται στην αλλαγή της συμπεριφοράς ενός ατόμου προκειμένου να ταιριάξει με τις συμπεριφορές των ατόμων γύρω του [49]. Στα συστήματα συστάσεων οι χρήστες έχουν την τάση να αξιολογούν αντικείμενα ακολουθώντας τον τρόπο αξιολόγησης των άλλων χρηστών, ακόμη και αν αυτό έρχεται



Εικόνα 3.2: Βρόχος ανατροφοδότησης στα συστήματα συστάσεων [Πηγή: `https://arxiv.org/pdf/2010.03240v1.pdf`]

σε αντίθεση με την προσωπική τους κρίση. Για παράδειγμα, αν μια ταινία έχει χιλιάδες αξιολογήσεις και μέσο όρο αξιολόγησης που αγγίζει ή και ξεπερνάει τα 4 στα 5 αστέρια, είναι αρκετά πιθανό να μην το αξιολογήσει αρνητικά ακόμη και αν αυτή ήταν η αρχική του πρόθεση. Αυτό έχει ως συνέπεια οι αξιολογήσεις πολλές φορές να μην αντικατοπτρίζουν την πραγματική κρίση/προτίμηση του χρήστη. Το conformity bias φαίνεται πως συνδέεται και με ένα άλλο είδος μεροληψίας το popularity bias (υποενότητα 3.2.2), καθώς οι χρήστες συνηθίζουν να αλληλεπιδρούν με δημοφιλή αντικείμενα, για παράδειγμα αγοράζουν ένα προϊόν έχοντας ως βασικό κριτήριο τη δημοφιλία του, δηλαδή τον αριθμό των πωλήσεων ή των αξιολογήσεων του [50].

Μεροληψία έκθεσης (Exposure Bias)

Αυτό το είδος μεροληψίας συμβαίνει καθώς οι χρήστες εκτίθενται μόνο σε ένα μέρος των αντικειμένων μιας λίστας και εξαιτίας αυτού οι μη παρατηρηθείσες αλληλεπιδράσεις (unobserved interactions) δεν αντιπροσωπεύουν πάντα αρνητική προτίμηση. Με άλλα λόγια, αν ένας χρήστης δεν έχει δει ένα αντικείμενο δεν μπορεί να προκύψει το αυθαίρετο συμπέρασμα ότι δεν του αρέσει αυτό το αντικείμενο. Οι ερευνητές έχουν εξετάσει από διάφορες οπτικές αυτό το είδος μεροληψίας. Μια εξ αυτών θεωρεί πως το popularity bias είναι μια μορφή μεροληψίας έκθεσης, διότι το τμήμα των αντικειμένων στο οποίο εκτίθεται περισσότερο ένας χρήστης είναι τα δημοφιλή αντικείμενα. Ενώ μια άλλη, σχετίζεται με την αναζήτηση που πραγματοποιούν οι χρήστες προκειμένου να βρουν τα αντικείμενα που τους ενδιαφέρουν και την επιλογή αυτών. Η επιλογή αυτή θεωρείται ως ένας τύπος έκθεσης, ο οποίος δίνει αρκετά περισσότερες πιθανότητες στα πολύ σχετικά αντικείμενα να προβληθούν/εκτεθούν και επομένως το exposure bias μπορεί να ονομαστεί αλλιώς και selection bias.

Μεροληψία θέσης (Position Bias)
Σχετίζεται με την τάση των ανθρώπων να αλληλεπιδρούν με αντικείμενα τα οποία βρίσκονται στις πιο υψηλές θέσεις των λιστών συστάσεων (recommendation lists) ανεξάρτητα από την πραγματική σχετικότητα των αντικειμένων αυτών. Το position bias θεωρείται πως είναι ένας τύπος selection bias. Ένα χαρακτηριστικό παράδειγμα αυτού αποτελούν τα βίντεο που μας προτείνει το YouTube. Ένας χρήστης συνήθως θα επιλέξει να παρακολουθήσει κάποιο από τα πρώτα βίντεο που του προτείνει αγνοώντας όσα είναι πιο χαμηλά στη λίστα, τα οποία όμως μπορεί να είναι πιο ενδιαφέροντα και πιο σχετικά με τις προτιμήσεις του.

Μεροληψία συναισθήματος (Sentiment bias)
Στο [51] ορίζεται ένας ακόμη τύπος μεροληψίας, η μεροληψία συναισθήματος ως «*η απόκλιση μεταξύ της απόδοσης των συστημάτων συστάσεων σε χρήστες/στοιχεία με θετικό feedback και σε χρήστες/στοιχεία με αρνητικό feedback*». Πιο αναλυτικά, τα συστήματα συστάσεων κάνουν αρκετά πιο ακριβείς προτάσεις σε χρήστες/αντικείμενα που έχουν θετικό feedback (positive feedback) σε σχέση με χρήστες/αντικείμενα που έχουν αρνητικό feedback (negative feedback). Συνέπεια αυτού είναι τόσο η χαμηλή ποιότητα στις συστάσεις που προσφέρονται στους χρήστες, όσο και μη δίκαιη εκπροσώπηση των αντικειμένων που λαμβάνουν θετικά σχόλια από ένα μικρό μέρος του πληθυσμού (τα λεγόμενα και niche items). Αξίζει να σημειωθεί πως αυτό το είδος μεροληψίας αν και εκ πρώτης όψεως μοιάζει να έχει αρκετά κοινά με άλλη είδη, όπως για παράδειγμα με το popularity bias, εντούτοις είναι διαφορετικό.

Στις επόμενες δύο υποενότητες αναλύονται εκτενέστερα, η μεροληψία που εισάγει το μοντέλο στα αποτελέσματα που παράγει και δίνει στους χρήστες λόγω της σημαντικότητάς τους. Πιο συγκεκριμένα, αναλύονται: η έννοια της δικαιοσύνης στα αποτελέσματα που παράγει ένα σύστημα συστημάτων συστάσεων, ενώ παράλληλα περιγράφονται τα πιο σημαντικά είδη αυτής, και ένα πολύ σοβαρό είδος μεροληψίας, το popularity bias.

3.2.1 Δικαιοσύνη

Ο Yao στο [52] παρουσιάζει μορφές αδικίας (unfairness) που υπάρχουν στα μοντέλα συνεργατικής διήθησης. Πιο συγκεκριμένα, περιγράφει μια διαδικασία μέσω της παραγοντοποίησης μητρώου, η οποία οδηγεί σε άδικες συστάσεις (unfair recommendations) - δηλαδή συστάσεις που εισάγουν κάποιο είδος μεροληψίας -, ακόμη και όταν τα δεδομένα αξιολογήσεων αντικατοπτρίζουν με ακρίβεια τις προτιμήσεις των χρηστών. Αυτού του είδους το unfairness, προκύπτει από την ελλιπή εκπροσώπηση (underrepresentation). Ανιχνεύτηκαν, 2 μορφές αυτής: η άνιση κατανομή πληθυσμιακών ομάδων (Population imbalance) και η μεροληψία παρατήρησης (observation bias) η οποία είναι αρκετά παρόμοια με την μεροληψία έκθεσης που περιγράψαμε στην προηγούμενη υποενότητα.
Στο [53] η διάκριση των τύπων δικαιοσύνης στα συστήματα συστάσεων είναι η εξής: ατομική/ομαδική, στατική/δυναμική, μονόπλευρη/πολύπλευρη, αντικειμένου/χρήστη και συνειρμική/αιτιώδης. Όπως και σε άλλα πεδία της μηχανικής μάθησης και στα συστήματα συστάσεων υπάρχει η διάκριση μεταξύ ομαδικής (Group) και ατομικής (Individual) δικαιοσύνης.
Στο [54] ορίζονται τα εμπλεκόμενα μέρη στα συστήματα συστάσεων:

- **οι καταναλωτές (consumers - C)**, δηλαδή εκείνοι που δέχονται τις προτάσεις
- **οι πάροχοι (providers - P)**, δηλαδή εκείνοι που έχουν κάποιο κέρδος από τις επιλογές του

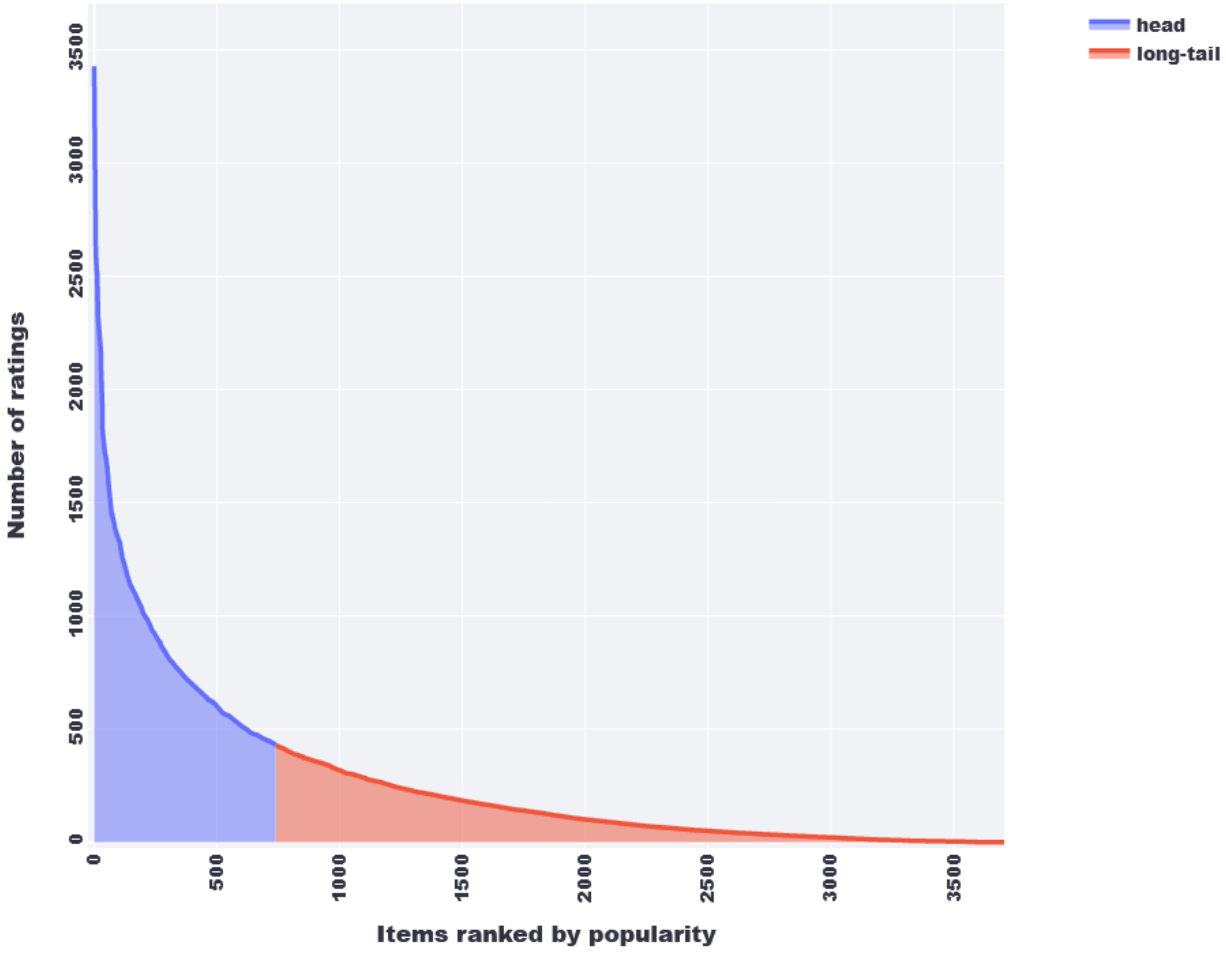
καταναλωτή και υποστηρίζουν (χορηγούν) τα προτεινόμενα αντικείμενα

- **η πλατφόρμα ή αλλιώς το σύστημα (System – S)** που δημιουργεί τις συστάσεις.

Σύμφωνα με αυτή την κατηγοριοποίηση ορίζονται αντίστοιχα οι έννοιες και στο fairness: C-fairness, P-fairness και CP-fairness. Οι δύο πρώτες έννοιες, εντάσσονται στην κατηγορία της δικαιοσύνης που ονομάζεται μονόπλευρη (single-sided fairness) καθώς εξετάζει την ύπαρξη δίκαιων αποτελεσμάτων μόνο για την μία πλευρά. Αντίθετα η περίπτωση της CP-fairness, στην οποία λαμβάνεται υπόψη τόσο η δικαιοσύνη από την πλευρά του καταναλωτή, όσο και από την πλευρά του παρόχου, ανήκει στην ευρύτερη κατηγορία της πολύπλευρης δικαιοσύνης (multi-sided fairness). Έκτοτε, στην βιβλιογραφία συναντάμε αρκετές μετρικές που ανήκουν σε μια από τις προαναφερθείσες κατηγορίες.
Η συνειρμική δικαιοσύνη (Associative Fairness) αναφέρεται στην απόκλιση των στατιστικών μετρικών μεταξύ ατόμων ή υποπληθυσμών. Οι περισσότερες έννοιες δικαιοσύνης στα συστήματα συστάσεων που συναντάμε στη βιβλιογραφία βασίζονται σε αυτή τη θεώρηση. Μία διαφορετική θεώρηση από αυτή αποτελεί η αιτιώδης δικαιοσύνη (Causal Fairness). Μια από τις πρώτες προσπάθειες, αν όχι η πρώτη, είναι η τεχνική που παρουσιάζεται στο [55] για την οποία δημιουργήθηκαν αιτιώδεις γράφοι (causal graphs) για να περιγραφεί το πως επηρεάζουν τα προστατευόμενα χαρακτηριστικά τη δημιουργία των συστάσεων.
Μια άλλη θεώρηση είναι η δικαιοσύνη για τον χρήστη (User Fairness) και η δικαιοσύνη για το αντικείμενο (Item Fairness). Η δικαιοσύνη για τον χρήστη αναφέρεται στην μεροληψία που εισάγεται κατά την παραγωγή συστάσεων για ορισμένους χρήστες ή (συχνότερα) ομάδες χρηστών και η δικαιοσύνη για το αντικείμενο σχετίζεται με την δικαιοσύνη που πρέπει να υπάρχει στα αντικείμενα που προτείνονται. Χαρακτηριστικότερο παράδειγμα στο οποίο εξετάζεται η δικαιοσύνη για τα αντικείμενα είναι το popularity bias, που αναλύεται λεπτομερώς στην αμέσως επόμενη υποενότητα.
Μια ακόμη έννοια που συναντάμε είναι αυτή της στατικής δικαιοσύνης. Στατική δικαιοσύνη (Static fairness) σημαίνει ότι τα προστατευόμενα χαρακτηριστικά ή οι ετικέτες των ομάδων (group labels), όπως για παράδειγμα το φύλο και ηλικία, είναι σταθερά σε όλη την κατάταξη ή τη διαδικασία δημιουργίας συστάσεων. Αντίθετα, η δυναμική δικαιοσύνη (Dynamic fairness) [56] λαμβάνει υπόψη της τους δυναμικούς παράγοντες στο περιβάλλον των συστημάτων συστάσεων. Αυτοί εμπεριέχουν τις αλλαγές στη χρησιμότητα ή τα χαρακτηριστικά των αντικειμένων σε αυτό το περιβάλλον. Αφού γίνει αυτό σειρά έχει η εκμάθηση μιας στρατηγικής για να ανταποκρίνεται το σύστημα σε τέτοιες δυναμικές.
Στο [57] παρουσιάζεται μια αρκετά διαφορετική κατηγοριοποίηση της δικαιοσύνης, η οποία δεν εντάσσεται στο πλαίσιο της δικαιοσύνης που σχετίζεται με τους αλγορίθμους, όπως όσες προαναφέρθηκαν και όσες αναφέρονται στη βιβλιογραφία. Πιο συγκεκριμένα, διακρίνονται τρεις μορφές δικαιοσύνης.
Πρώτα παρουσιάζεται η δικαιοσύνη στην αφοσίωση του χρήστη (Engagement), η οποία σχετίζεται με το πώς διαφορετικοί χρήστες δεσμεύονται και παρακινούνται να χρησιμοποιούν το σύστημα και να αλληλεπιδρούν με τις συστάσεις. Ακολούθως, παρουσιάζεται η δικαιοσύνη αναπαράστασης (Representation) η οποία αφορά την παροχή πολλαπλών μέσων εκπροσώπησης για τους χρήστες με στόχο την ενίσχυση διαφορετικών χρηστών, οι οποίοι έχουν τις δικές τους συγκεκριμένες φυσικές και γνωστικές ικανότητες, για την καλύτερη κατανόηση των συστάσεων που τους παρουσιάζονται. Ένα παράδειγμα που μπορεί να δοθεί για αυτό ώστε να γίνει πιο κατανοητό, είναι ένας χρήστης με προβλήματα όρασης. Τέλος, ορίζεται η δικαιοσύνη δράσης και έκφρασης (Action & Expression) όπου το σύστημα θα πρέπει να δίνει τον έλεγχο στον χρήστη ως προς το πώς θέλει να μεταβάλλονται οι συστάσεις και θα πρέπει επίσης να παρέχει πολλαπλά μέσα αλληλεπίδρασης για τους χρήστες.

3.2.2 Μεροληψία δημοφιλίας (Popularity bias)

Στην παρούσα εργασία από όλα τα είδη μεροληψίας που συναντάμε στα συστήματα συστάσεων και περιγράφονται σε αυτή την ενότητα, αυτό που θα μας απασχολήσει είναι το popularity bias καθώς φαίνεται πως είναι το πιο σοβαρό και έχει δημιουργήσει πολυάριθμα και σημαντικά προβλήματα. Το φαινόμενο αυτό *περιγράφει την τάση των συστημάτων συστάσεων να προτείνουν τα πιο δημοφιλή αντικείμενα (εκείνα με τις περισσότερες αξιολογήσεις) πολύ πιο συχνά από τα λιγότερο δημοφιλή αντικείμενα, τα λεγόμενα long tail αντικείμενα, που στην μεγάλη πλειονότητα των περιπτώσεων είναι πολύ περισσότερα [58].*



Εικόνα 3.3: Διάγραμμα long tail

Ο Chris Anderson στο βιβλίο του "The Long Tail: Why the Future of Business Is Selling Less of More" [59] εισάγει για πρώτη φορά τον όρο "long tail", ο οποίος βασίζεται στην «αρχή του Pareto» (Pareto Principle), για να περιγράψει το φαινόμενο στο οποίο οι επιχειρήσεις αποκομίζουν σημαντικά κέρδη από την πώληση προϊόντων με χαμηλή ζήτηση ή χαμηλές πωλήσεις σε πολλούς πελάτες, σε σχέση με αυτά που αποκομίζουν από την πώληση μόνο μεγάλων ποσοτήτων ενός περιορισμένου αριθμού δημοφιλών προϊόντων. Η Αρχή του Pareto (Pareto Principle), γνωστή και ως κανόνας του 80/20 ορίζει ότι το 80% των επιπτώσεων προέρχεται από το 20% των αιτιών. Στον χώρο των επιχειρήσεων αυτό

μεταφράζεται ως: το 20% των προϊόντων αντιπροσωπεύει το 80% των πωλήσεων. Αυτό το 20% των πιο δημοφιλών προϊόντων ονομάζεται “head” και το υπόλοιπο 80% “long tail”, καθώς φαίνεται σαν μια μακριά ουρά, κάτι που είναι ευδιάκριτο στην Εικόνα 3.3. Προκειμένου να γίνει πιο κατανοητό το long tail παρατίθεται το παρακάτω γράφημα (Εικόνα 3.3), το οποίο είναι στο σύνολο δεδομένων Movielens1M που περιέχει αξιολογήσεις ταινιών. Στον άξονα x βρίσκονται τα IDs των ταινιών ταξινομημένα κατά φθίνουσα σειρά, ως προς τον αριθμό των αξιολογήσεων που έχουν λάβει και στον άξονα y ο συνολικός αριθμός των αξιολογήσεων.

Ωστόσο, το γεγονός ότι ένα αντικείμενο είναι δημοφιλές δεν σημαίνει αυτομάτως ούτε πως είναι ποιοτικό ούτε ηθικό. Επιπρόσθετα, το popularity bias έχει αποδειχθεί ότι συνδέεται και με άλλους τύπους μεροληψίας και επίσης συμβάλλει στη δημιουργία των φαινομένων “echo chambers” και “filter bubbles”.

Στο φαινόμενο που έχει γίνει γνωστό ως «θάλαμος αντήχησης» (echo chamber) [60], ένα άτομο ή μια ομάδα ατόμων δημιουργούν ένα περιβάλλον μέσα στο οποίο συναντούν μόνο πληροφορίες ή απόψεις που αντανακλούν ή ενισχύουν τις δικές τους, χωρίς να συναντούν αντίθετες απόψεις. Το φαινόμενο «φυσαλίδες πληροφορίας» (filter bubbles) πολλές φορές συγχέεται με το echo chambers αν και πρόκειται για κάτι διαφορετικό. Πιο αναλυτικά, στο φαινόμενο filter bubbles οι αλγόριθμοι προτείνουν στους ανθρώπους περιεχόμενο παρόμοιο με τα ενδιαφέροντά τους ή σύμφωνα με τα χαρακτηριστικά τους και το προφίλ τους (παλιά κλικ, ιστορικό αναζήτησης κτλ.), αποκόπτοντας τους έτσι από μεγάλο μέρος του διαθέσιμου περιεχομένου και δημιουργώντας μια «φούσκα» η οποία μειώνει κατά πολύ την ποικιλία (diversity).

Εξαιτίας αυτής της μεροληψίας, έχει παρατηρηθεί επίσης το φαινόμενο μεγάλες εταιρίες να πληρώνουν χρήστες προκειμένου να παρέχουν υψηλές αξιολογήσεις και θετικές κριτικές στα προϊόντα τους [61]. Επιπρόσθετα, μπορεί να οδηγήσει σε χειραγώγηση των χρηστών από ψεύτικες κριτικές (fake reviews), των bots που συναντάμε στα μέσα κοινωνικής δικτύωσης (social bots ή social media bots) [1] και στο φαινόμενο που έχει γίνει γνωστό ως “astroturfing”, καθώς και στην έλλειψη ανεξαρτησίας και κοινωνικής επιρροής ανάμεσα στα μέλη ενός πλήθους ανθρώπων - όπως αυτό υπονοείται έμμεσα από τη διαθεσιμότητα των ταξινομήσεων (rankings) - η οποία υπονομεύει σοβαρά την αξιοπιστία των σημάτων ένδειξης της δημοφιλίας (popularity signals), όπως αναφέρει ο Ciampaglia στο [63]. Χρησιμοποιώντας τον όρο “astroturfing”, αναφερόμαστε στην προσπάθεια να δημιουργηθεί μια εντύπωση ευρείας υποστήριξης από τη βάση για μια πολιτική, άτομο ή προϊόν, όπου στην πραγματικότητα υπάρχει μικρή τέτοια υποστήριξη. Με αυτόν τον τρόπο στα μέσα κοινωνικής δικτύωσης προωθούνται πολλές φορές ακραίες, ψευδείς ή ακόμα και επικίνδυνες αναρτήσεις οι οποίες μπορούν να διαβρώσουν και να αλλοιώσουν μια κοινωνία, να προωθήσουν τον εξτρεμισμό και την βία, ή ακόμα και να οδηγήσουν σε αλλοίωση των πολιτικών αποτελεσμάτων ή ακόμα χειρότερα σε εμπόλεμες διαμάχες. Πρόσφατες έρευνες δείχνουν πως τα μέσα κοινωνικής δικτύωσης ευθύνονται για την διάδοση ψευδών ειδήσεων σχετικών με την πανδημία της COVID-19 και τα εμβόλια για διάφορες ασθένειες, αποτρέποντας τους ανθρώπους να εμβολιαστούν, αυξάνοντας έτσι τον αριθμό των θανάτων και την πίεση στο εθνικό σύστημα υγείας. Καθοριστικό ρόλο σε αυτή τη διάδοση φαίνεται πως διαδραματίζουν οι αλγόριθμοι των συστημάτων συστάσεων. Εκτός όμως από τα μέσα κοινωνικής δικτύωσης, popularity bias συναντάμε σχεδόν παντού πλέον στο διαδίκτυο, από τις ιστοσελίδες που χρησιμοποιούμε για την ψυχαγωγία μας και τις αγορές μας, μέχρι και εκείνες που χρησιμοποιούμε για την εύρεση εργασίας και την αναζήτηση

[1] **Social bot:** αλγόριθμος που παράγει αυτόματα περιεχόμενο και αλληλεπιδρά με τους ανθρώπους στα μέσα κοινωνικής δικτύωσης, προσπαθώντας να μιμηθεί και ενδεχομένως να αλλάξει τη συμπεριφορά τους.[62]

πηγών για επιστημονική έρευνα. Στο [64] γίνεται έρευνα σχετικά με την επίδραση του popularity bias στα Massive Open Online Courses (MOOC's). Καθίσταται επομένως επιτακτική η ανάγκη να εξαλείψουμε τη μεροληψία αυτού του είδους των αλγορίθμων.

Αλγόριθμοι μετριασμού μεροληψίας

Σε αυτήν την υποενότητα παρουσιάζονται ορισμένοι από τους πιο σημαντικούς αλγορίθμους και τεχνικές για τον μετριασμό της μεροληψίας, που ανήκουν στις τρεις κατηγορίες που αναφέραμε στην υποενότητα 3.1.4: pre-processing, in-processing και post-processing. Ιδιαίτερη έμφαση δίνεται στην post-processing κατηγορία, διότι αυτή χρησιμοποιήθηκε κατά κύριο λόγο σε αυτή την εργασία, ενώ θα πρέπει να σημειωθεί πως περιγράφονται αναλυτικά μόνο οι αλγόριθμοι που χρησιμοποιήθηκαν στα πειράματά μας.

Pre-processing

Στο [65] η τεχνική που προτείνεται αφήνει το σύνολο εκπαίδευσης ανέγγιχτο και απλά προσθέτει σε αυτό δεδομένα που τα ονομάζει «αντίδοτο», καθώς αποτελούν ένα «αντίδοτο» στην μεροληψία που περιέχουν τα δεδομένα. Αυτή η τεχνική που εφαρμόζεται σε έναν αλγόριθμο παραγοντοποίησης μητρώου στον οποίο έχει γίνει η εκπαίδευση δίνοντας του ως είσοδο τα δεδομένα που επιθυμούμε, προσθέτουμε «ψεύτικους» χρήστες και μαζί ψεύτικες αξιολογήσεις που έχουν πραγματοποιήσει. Αυτές οι νέες αξιολογήσεις είναι στην ουσία τα δεδομένα αντιδότου.

In-processing

Μια προσέγγιση σε αυτή την κατηγορία είναι η προσθήκη όρων κανονικοποίησης στη συνάρτηση απώλειας (loss function) του μοντέλου. Μέσω αυτών των όρων γίνεται η ρύθμιση της μεροληψίας που εισάγεται. Ο αλγόριθμος που πρότεινε ο Kamishima στο [66] έχει ως στόχο την στατιστική ανεξαρτησία, δηλαδή να μην συμπεριλάβει οτιδήποτε αφορά το προστατευόμενο χαρακτηριστικό που να επηρεάζει το αποτέλεσμα. Η τεχνική που παρουσιάζεται στο [67] βασίζεται στην τεχνική των Variational Autoencoders (VAE) [68] στη συνεργατική διήθηση. Σε αυτή αποδεικνύεται πως ο θόρυβος στην κανονική κατανομή και στην κατανομή Gauss κατά τη φάση δοκιμής των VAE αλλάζει τα scores του αποτελέσματος, όταν έχουν τα ίδια δεδομένα ως είσοδο και μπορεί να μειώσει το unfairness, αν και οδηγεί σε μια μικρή μείωση της ακρίβειας.

Post-processing

Η πιο γνωστή τεχνική σε αυτή την κατηγορία είναι η τεχνική του re-ranking (ανακατάταξη). Σε αυτή, οι αλγόριθμοι παίρνουν ως είσοδο την αρχική λίστα συστάσεων που έχει δημιουργήσει ο αρχικός (base) αλγόριθμος, την επεξεργάζονται και δίνουν ως έξοδο μια λίστα μικρότερου μεγέθους από την αρχική. Στο [69] οι Liu και Burke πρότειναν δύο re-ranking αλγορίθμους τον *Fairness-Aware Re-ranking (FAR)* και τον *Personalized Fairness-Aware Re-ranking (PFAR)*, οι οποίοι βασίζονται στον αλγόριθμο eXplicit Query Aspect Diversification (xQuAD) [70]. Με τη χρήση αυτών των αλγορίθμων διασφαλίζεται ότι αντικείμενα από διαφορετικές κατηγορίες, όπως είναι η long tail και η head λαμβάνουν μια δίκαιη έκθεση στη λίστα συστάσεων. Έστω R μια λίστα συστάσεων η οποία έχει παραχθεί από έναν base αλγόριθμο συστάσεων, u είναι ένας χρήστης, S η re-ranked λίστα συστάσεων, K ο αριθμός των αντικειμένων που θα περιέχει η S, $\lambda \in (0,\ 1)$ μια παράμετρος η οποία ελέγχει το ποσοστό δικαιοσύνης του παρόχου, τ_u είναι ένα εξατομικευμένο βάρος (personalized weight) του οποίου η εκμάθηση γίνεται από το ιστορικό της συμπεριφοράς κάθε χρήστη. Αν $\tau_u = 1$

Algorithm 1 (Personalized) Fairness-Aware Re-ranking (FAR/PFAR)

Input: u, R, K, λ, τ_u
Output: S

$S \leftarrow \emptyset$
while $|S| < K$ **do**
  $v^* \leftarrow \arg\max_{v \in R \setminus S} P(v|u) + \lambda \tau_u P(v, \bar{S}|u)$
  $R \leftarrow R \setminus \{v^*\}$
  $S \leftarrow S \cup \{v^*\}$
end while
return S

Εικόνα 3.4: Οι αλγόριθμοι FAR και PFAR [Πηγή: `https://arxiv.org/pdf/1809.02921.pdf`]

τότε έχουμε τον αλγόριθμο FAR, ενώ αν τ_u είναι μια τιμή εξατομικευμένου βάρους του οποίου έχει γίνει η εκμάθηση (personalized learned value) τότε έχουμε τον αλγόριθμο PFAR.

$$\max_{v \in R(u)} \underbrace{(1-\lambda)P(v|u)}_{\text{personalization}} + \underbrace{\lambda \tau_u \sum_c PV_c \mathbb{1}_{\{v \in V_c\}} \prod_{i \in S(u)} \mathbb{1}_{i \notin V_c}}_{\text{fairness}}$$

Όπου $P(v|u)$ είναι η πιθανότητα ενός χρήστη $u \in U$ να ενδιαφέρεται για το αντικείμενο $v \in V$, η οποία έχει προβλεφθεί από τον base αλγόριθμο συστάσεων. Στην Εικόνα 3.4 παρατίθεται αναλυτικά ο αλγόριθμος FAR/PFAR όπως αυτός παρουσιάστηκε στο [69].
Μια ακόμη γνωστή τεχνική είναι αυτή που περιγράφεται μέσω του αλγορίθμου *Calibrated recommendations* (Cali) [71] όπου κατατάσσουμε εκ νέου τις προτάσεις που παρέχουμε στους χρήστες, για να διασφαλίσουμε την πληρέστερη αντιστοίχιση της κατανομής των ενδιαφερόντων του χρήστη στα χαρακτηριστικά των στοιχείων. Στο σημείο αυτό ας δούμε με περισσότερη λεπτομέρεια πως λειτουργεί αυτός ο αλγόριθμος. Έστω δύο κατανομές, η κατανομή πάνω στα είδη g των αντικειμένων, του συνόλου αντικειμένων H με τα οποία έχει αλληλεπιδράσει ο χρήστης u στο παρελθόν:

$$p(g|u) = \sum_{i \in H} p(g|i) \tag{3.4}$$

και η κατανομή πάνω στα είδη των αντικειμένων g, του συνόλου των αντικειμένων I που προτείνονται στον χρήστη u.

$$q(g|u) = \sum_{i \in I} p(g|i) \tag{3.5}$$

Ωστόσο δεν είναι λίγες οι φορές όπου τα αντικείμενα που προτείνονται σε έναν χρήστη δεν προσαρμόζονται κατάλληλα σύμφωνα με τις αλληλεπιδράσεις του στο παρελθόν. Προκειμένου να λυθεί αυτό το πρόβλημα χρειαζόμαστε μια μετρική ρύθμισης/προσαρμογής (calibration metric) C. Για την σύγκριση των δύο κατανομών και την μέτρηση της ομοιότητάς τους υπάρχουν αρκετές μέθοδοι, με την πιο δημοφιλή από αυτές να είναι η απόκλιση (divergence) Kullback-Leibler(KL):

$$C(p,q) = D_{KL}(p||q) = \sum_g p(g|u) \cdot \log \frac{p(g|u)}{\tilde{q}(g|u)} \tag{3.6}$$

Στην περίπτωση που $q(g|u) = 0$ και $p(g|u) > 0$ για ένα είδος g, τότε:

$$\tilde{q}(g|u) = (1 - \alpha) \cdot q(g|u) + \alpha \cdot p(g|u) \tag{3.7}$$

για την εφαρμογή του calibration ο αλγόριθμος λαμβάνει ως είσοδο μια λίστα συστάσεων την οποία ανακατατάσσει, λειτουργώντας ακριβώς όπως ένας post-processing αλγόριθμος. Ο υπολογισμός του βέλτιστου συνόλου I^* των αντικειμένων που θα προταθούν δίνεται από την σχέση μέγιστης οριακής συνάφειας (maximum marginal relevance):

$$I^* = \underset{I,|I|=N}{\operatorname{argmax}} (1 - \lambda) \cdot s(I) - \lambda \cdot C_{KL}(p, q(I)) \tag{3.8}$$

όπου $s(i)$ είναι η βαθμολογία (score) των αντικειμένων $i \in I$ που έχει προβλεφθεί από το σύστημα συστάσεων, $s(I) = \sum_{i \in I} s(i)$, το άθροισμα των βαθμολογιών όλων των αντικειμένων στη λίστα συστάσεων και $\lambda \in [0, 1]$ είναι μια παράμετρος ρύθμισης που καθορίζει την αντιστάθμισμα (trade-off) ανάμεσα στη βαθμολογία που έχει δημιουργηθεί από το σύστημα συστάσεων και από την βαθμολογία του calibration score. Μια παρατήρηση εδώ είναι πως επειδή η μέτρηση του calibration score γίνεται από την KL-divergence, στον τύπο της χρησιμοποιούμε το αρνητικό μιας και είναι μια μετρική όπου όσο χαμηλότερη είναι η τιμή της τόσο το καλύτερο.

Ο αλγόριθμος *FA*IR* [72] δημιουργεί ουρές αντικειμένων που ανήκουν σε προστατευόμενες ομάδες και αντικειμένων που δεν ανήκουν και επιλέγει από κάθε ουρά για να δημιουργήσει την τελική re-ranked λίστα. Ας δούμε λίγο πιο αναλυτικά πώς λειτουργεί ο συγκεκριμένος αλγόριθμος.

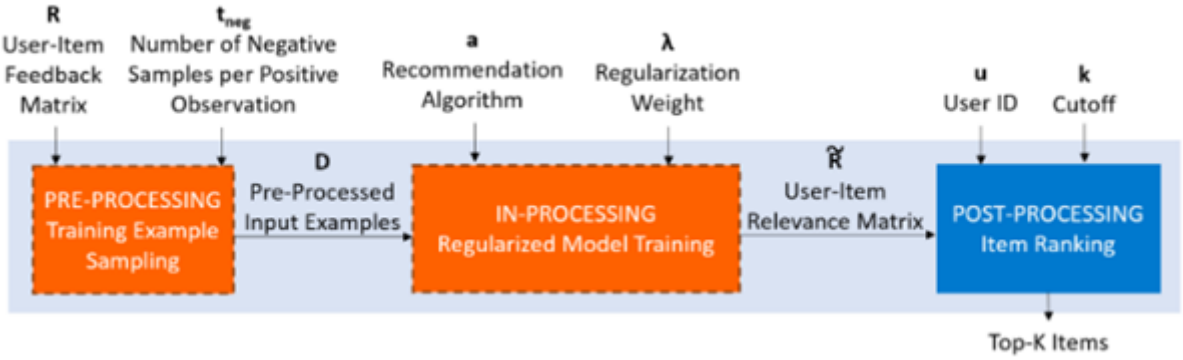
Έστω $[n] = \{1, 2, \dots, n\}$ ένα σύνολο υποψηφίων (candidates) και έστω q_i για $i \in [n]$ η «ποιότητα» του υποψηφίου i. Υπάρχουν δύο είδη υποψηφίων, εκείνοι που ανήκουν σε μια προστατευόμενη ομάδα και εκείνοι που δεν ανήκουν σε κάποια προστατευμένη ομάδα. Εάν ανήκουν σε μια προστατευόμενη ομάδα τότε $q_i = 1$, αλλιώς $q_i = 0$. Σε μια κατάταξη αντικειμένων μεγέθους k που έχει δοθεί ως είσοδος, συγκρίνεται *σε κάθε θέση της κατάταξης* ο αριθμός των αντικειμένων που ανήκουν σε μια προστατευόμενη ομάδα σε σχέση με τον αναμενόμενο αριθμό των αντικειμένων που ανήκουν σε μια προστατευόμενη ομάδα αν αυτά επιλέγονταν τυχαία με χρήση δοκιμών Bernoulli. Ο αριθμός αυτός, δεν θα πρέπει να είναι μικρότερος από την ελάχιστη ποσότητα p. Μαθηματικά αυτό μεταφράζεται σε μία αθροιστική συνάρτηση διωνυμικής κατανομής F με παραμέτρους p, k και α: $F(\tau_p; k, p) > \alpha$, όπου τ_p είναι ο πραγματικός αριθμός των προστατευόμενων αντικειμένων στην κατάταξη και α μια παράμετρος σημαντικότητας η οποίας μας δίνει την πιθανότητα να απορρίψουμε μια δίκαιη κατάταξη. Η προσαρμοσμένη σημαντικότητα $\alpha_c = c(a, k, p)$, χρησιμοποιείται διότι ελέγχει πολλαπλές υποθέσεις, καθώς αν χρησιμοποιήσουμε $\alpha_c = \alpha$ υπάρχει πιθανότητα να δημιουργηθούν πολλά ψευδώς αρνητικά.

Ο αλγόριθμος παίρνει ως είσοδο τον αριθμό των αντικειμένων που θα περιέχει η λίστα κατάταξης για κάθε χρήστη, έναν πίνακα που αντιστοιχίζει κάθε αντικείμενο που υπάρχει στο σύνολο δεδομένων με το αν αυτό ανήκει σε προστατευόμενη ομάδα ή όχι (long tail ή head στην περίπτωσή μας), το προσαρμοσμένο επίπεδο σημαντικότητας α_c και την ελάχιστη ποσότητα p των υποψηφίων που ανήκουν σε κάποια προστατευόμενη ομάδα.

Αρχικά, ο FA*IR δημιουργεί δύο ουρές προτεραιότητας με έως και k υποψηφίους η κάθε μια, τις P_0 και P_1 για τα αντικείμενα για τα αντικείμενα που δεν ανήκουν σε κάποια προστατευόμενη ομάδα και για τα αντικείμενα που ανήκουν, αντίστοιχα. Στην συνέχεια, δημιουργεί έναν πίνακα m ο οποίος σε κάθε γραμμή περιέχει τον ελάχιστο αριθμό των υποψηφίων που ανήκουν στην προστατευόμενη ομάδα σε κάθε μία από τις κορυφαίες k (top k) θέσεις ώστε να ικανοποιείται το κριτήριο, κάθε στήλη του πίνακα περιέχει μια συγκεκριμένη τιμή του k. Αν ο πίνακας m απαιτεί έναν υποψήφιο

που ανήκει σε μια προστατευόμενη ομάδα στη συγκεκριμένη θέση τότε ο αλγόριθμος τοποθετεί τον καλύτερο υποψήφιο από την ουρά P_1 στην κατάταξη, αλλιώς τοποθετεί τον καλύτερο υποψήφιο από την ένωση των 2 ουρών $P_0 \cup P_1$.

Συνδυασμός τεχνικών



Εικόνα 3.5: Η διαδικασία μετριασμού της μεροληψίας που παρουσιάζεται στο "Connecting user and item perspectives in popularity debiasing for collaborative recommendation", L. Boratto, G. Fenu, and M. Marras [Πηγή: `https://arxiv.org/pdf/2006.04275.pdf`]

Στο [73] παρουσιάζεται μια διαδικασία μετριασμού της μεροληψίας η οποία αποτελείται από τρία βήματα. Πρόκειται στην ουσία για τον συνδυασμό των τριών τεχνικών που προαναφέρθηκαν: pre-processing, in-processing και post-processing.
Η pre-processing τεχνική στο σενάριο βελτιστοποίησης ανά-ζεύγη (pair-wise optimization setting), για κάθε χρήστη u, δημιουργούνται t τριπλέτες (u, i, j) ανά παρατηρηθείσα αλληλεπίδραση χρήστη-αντικειμένου (u, i). Το αντικείμενο j που δεν έχει παρατηρηθεί επιλέγεται ανάμεσα σε αντικείμενα λιγότερο δημοφιλή από το i, για t/2 παραδείγματα εκπαίδευσης (training examples), και μεταξύ των πιο δημοφιλών από το i για τα υπόλοιπα t/2 παραδείγματα. Με αυτόν τον τρόπο τα training examples αντιπροσωπεύουν ισότιμα τόσο τα δημοφιλή όσο και τα λιγότερο δημοφιλή αντικείμενα που σχετίζονται με το αντικείμενο που έχει παρατηρηθεί (observed item). Τα training examples συμβολίζονται με D.
Βασίζεται σε έναν pairwise αλγόριθμο, τον BPR και σε έναν pointwise, τον NeuMF. Στην in-processing τεχνική (*Regularized Optimization (reg)*) ο αλγόριθμος προσπαθεί να πετύχει ένα αντιστάθμισμα ανάμεσα στην ακρίβεια και στο popularity bias. Ο αρχικός αλγόριθμος λαμβάνει ως είσοδο το σύνολο εκπαίδευσης D χωρισμένο σε κομμάτια (batches), καθένα από αυτά τα κομμάτια συμβολίζεται με D_{batch}, για να εκτελέσει μια επαναληπτική διαδικασία στοχαστικής καθόδου κλίσης (stochastic gradient descent). Η προσέγγιση που ακολουθείται όσον αφορά τη συνάρτηση βελτιστοποίησης βασίζεται στις αρχικές συναρτήσεις βελτιστοποίησης των pointwise και pairwise αλγορίθμων. Με αυτό το σκεπτικό ορίζεται η ακόλουθη συνάρτηση βελτιστοποίησης:

$$\min_{\theta} (1-\lambda)\,\mathscr{L}\left(D_{batch}|\theta\right) + \lambda C\left(D_{batch}|\theta\right) \tag{3.9}$$

Όπου $\lambda \in [0,1]$ είναι το βάρος (weight) που εκφράζει το αντιστάθμισμα ανάμεσα στην απώλεια ακρίβειας (accuracy loss) και στην απώλεια κανονικοποίησης (regularization loss), δηλαδή στην μεροληψία που εισάγεται. Με $\lambda = 0$ είναι σαφές πως δίνεται προτεραιότητα στην απώλεια ακρίβειας

χωρίς να λαμβάνουμε υπόψη την απώλεια κανονικοποίησης. Αντίθετα, αν $\lambda = 1$ δίνεται προτεραιότητα στην απώλεια κανονικοποίησης. Με $\mathscr{L}$ συμβολίζεται η συνάρτηση απώλειας ακρίβειας, η οποία εξαρτάται από την οικογένεια αλγορίθμων συστημάτων συστάσεων που χρησιμοποιείται. Ενώ με C συμβολίζεται η κανονικοποίηση της συσχέτισης ανάμεσα στα προβλεφθέντα υπόλοιπα (predicted residuals) και στην παρατηρηθείσα δημοφιλία των αντικειμένων (observed item popularity). Για τον υπολογισμό της συσχέτισης ανάμεσα στις κατανομές A1 και A2, χρησιμοποιείται η συνάρτηση Correlation:

$$C\left(D_{batch}|\theta\right) = |\text{Corrrelation}(A_1, A_2)| \tag{3.10}$$

όπου

$$A_1(b) = \mathscr{L}\left(D_{batch}(b)|\theta\right), \text{με } b \in \left\{0, \cdots, |D_{batch}|\right\} \tag{3.11}$$

και:

$$A_2(b) = \frac{1}{\|U\|} \sum_{u \in U} \min\left(R(u, s), 1\right) \text{με } b \in \left\{0, \cdots, |D_{batch}|\right\} \tag{3.12}$$

Όπου $R(u, s)$ το feedback που έδωσε ο χρήσης u για το αντικείμενο s στο σύνολο εκπαίδευσης. Όπως γίνεται αντιληπτό από τον ορισμό των δύο αυτών κατανομών, εάν η ικανότητα του μοντέλου να προβλέψει το εάν ένα αντικείμενο είναι αρκετά σχετικό εξαρτάται από τη δημοφιλία αυτού του αντικειμένου, τότε λαμβάνει κάποια «ποινή» μέσω της παραμέτρου λ.
Η post-processing τεχνική δεν παρουσιάζεται και αφήνεται στην ευχέρεια του χρήστη η επιλογή του κατάλληλου post-processing αλγορίθμου.

Κεφάλαιο 4

Σχεδιασμός εφαρμογής

Εικόνα 4.1: Δομή πειράματος

Το πείραμα που υλοποιήθηκε στα πλαίσια αυτής της εργασίας αποτελείται από 6 βασικά στάδια, όπως φαίνεται και στην Εικόνα 4.1. Στο πρώτο στάδιο γίνεται η εύρεση και η συλλογή των πιο κατάλληλων συνόλων δεδομένων για τη συγκεκριμένη εργασία. Ακολουθεί η κατάλληλη επεξεργασία και οπτικοποίηση των δεδομένων που επιλέχθηκαν, κάτι που είναι απαραίτητο για την καλύτερη κατανόηση των δεδομένων μας και την προετοιμασία τους για να δοθούν ως είσοδος στους αλγορίθμους συστάσεων. Έπειτα, επιλέγονται οι αλγόριθμοι συστάσεων που θα παράξουν τις αρχικές λίστες συστάσεων για τους χρήστες και φυσικά οι πιο κατάλληλες τιμές των υπερπαραμέτρων τους. Αφού παραχθούν οι λίστες, ακολουθεί η αξιολόγηση των αλγορίθμων ως προς την ακρίβεια, το popularity bias, το diversity, το coverage και το novelty. Με τον όρο coverage, αναφερόμαστε στα συνολικά αντικείμενα που καλύπτει ο αλγόριθμος κατά την παραγωγή συστάσεων από τα διαθέσιμα αντικείμενα. Επόμενο στάδιο αποτελεί η χρήση τεχνικών για την μετριασμό της μεροληψίας που (ενδεχομένως) εντοπίστηκε στο προηγούμενο βήμα και η παραγωγή νέων λιστών συστάσεων. Τέλος, γίνεται η αξιολόγηση των νέων αποτελεσμάτων και σύγκριση με τα αρχικά αποτελέσματα που είχαν προκύψει ύστερα από τη δημιουργία των αρχικών λιστών συστάσεων.

Για την εκτέλεση όλων των βημάτων του πειράματος, κρίθηκε απαραίτητη η δημιουργία μιας διαδικτυακής εφαρμογής (web-app), η οποία έλαβε το όνομα *“Bias & Fairness in RecSys”*. Μια διαδικτυακή εφαρμογή είναι μια εφαρμογή η οποία είναι προσβάσιμη μέσω του διαδικτύου, χρησιμοποιώντας έναν οποιοδήποτε φυλλομετρητή (browser). Η επιλογή μιας διαδικτυακής εφαρμογής προκρίθηκε έναντι μιας εφαρμογής για υπολογιστή ή κινητό, καθώς είναι ανεξάρτητη

από το λειτουργικό σύστημα που έχει το σύστημα του χρήστη. Έτσι αφενός μεν δεν χρειάζεται να δημιουργηθούν διαφορετικές εκδόσεις ανάλογα με το λειτουργικό σύστημα και αφετέρου δε είναι αρκετά πιο εύκολη η πρόσβαση για έναν χρήστη, διότι δεν χρειάζεται να εγκαταστήσει κάποιο πρόγραμμα στο σύστημά του.

Η υλοποίηση της διαδικτυακής εφαρμογής (web-app) έγινε στη γλώσσα προγραμματισμού Python (έκδοση 3.8.7) με χρήση του ανοιχτού κώδικα framework Streamlit [74] (έκδοση 1.0.0), το οποίο χρησιμοποιείται ευρέως για τη δημιουργία web-apps για εργασίες μηχανικής μάθησης, επιστήμης των δεδομένων και βαθιάς μάθησης. Το κύριο πλεονέκτημα του Streamlit είναι πως δεν απαιτείται η γνώση front-end τεχνικών, όπως HTML, CSS, JavaScript για τη δημιουργία μιας εφαρμογής και είναι εφικτή η μετατροπή των κωδικών που έχουμε σε γλώσσα Python σε μια καλαίσθητη και φιλική προς τους χρήστες εφαρμογή μέσα λίγα και απλά βήματα. Η γλώσσα προγραμματισμού Python επιλέχθηκε καθώς είναι αρκετά δημοφιλής τόσο για εφαρμογές μηχανικής μάθησης, όσο και για συστήματα συστάσεων και διαθέτει μια πληθώρα βιβλιοθηκών και frameworks. Για τη συγγραφή, την εκτέλεση και την αποσφαλμάτωση του κώδικα, επιλέχθηκε το PyCharm IDE (Integrated Development Environment - ολοκληρωμένο περιβάλλον ανάπτυξης). Οι πιο σημαντικές βιβλιοθήκες που χρησιμοποιήθηκαν είναι:

- **Scikit-learn (Sklearn):** μια από τις πιο πλούσιες, χρήσιμες και απαραίτητες βιβλιοθήκες για μηχανική μάθηση. Προσφέρει μεταξύ άλλων αρκετούς αλγορίθμους μηχανικής μάθησης, αρκετά χρήσιμα εργαλεία και ορισμένα σύνολα δεδομένων.
- **Pandas:** βιβλιοθήκη για τη διαχείριση και την ανάλυση δεδομένων. Προσφέρει δομές δεδομένων και εργαλεία για προβολή και διαχείριση χρονοσειρών.
- **Altair:** βιβλιοθήκη για τη δημιουργία διαδραστικών γραφημάτων.
- **NumPy (Numerical Python):** πρόκειται για την βασική βιβλιοθήκη επιστημονικού υπολογισμού στην Python, προσφέρει διάφορα εργαλεία για την αποδοτική και αποτελεσματική διαχείριση μητρώων και εκτέλεση διάφορων πράξεων επ'αυτών.
- **Plotly:** βιβλιοθήκη για τη δημιουργία διαδραστικών γραφημάτων.

Πίνακας 4.1: Σύγκριση frameworks

	Αλγόριθμοι	Αλγόριθμοι για μετριασμό μεροληψίας	Μετρικές (σύνολο)	Μετρικές C.N.D.B.F. [1]
DaisyRec	19	0	10	0
Elliot	50	0	36	22
Lenskit	6	0	6	0
Librec-auto	55	7	10	14
Surprise	7	0	4	0

Με αυτόν τον τρόπο το "Bias & Fairness in Recsys" αποτελεί όχι μόνο ένα εργαλείο ανάλυσης και μετριασμού της μεροληψίας, αλλά και μια εφαρμογή μέσω της οποίας μπορούν να αναπτυχθούν

[1] C.N.D.B.F.: CoverageNoveltyDiversityBiasFairness

συστήματα συστάσεων με εύκολο και αρκετά φιλικό προς τον χρήστη τρόπο. Για την επιλογή του framework (εργαλείου) που θα μας δώσει τους αλγορίθμους για να δημιουργήσουμε τα συστήματα συστάσεων έγινε σύγκριση των frameworks: Surprise [75], librec-auto [76], Elliot [77], DaisyRec [78], Lenskit [79]. Τα βασικά κριτήρια για την τελική επιλογή ήταν το πλήθος των αλγορίθμων και των μετρικών που προσέφεραν, η επεκτασιμότητα και η ύπαρξη όσο το δυνατόν πιο καλογραμμένων και πληρέστερων εγγράφων τεκμηρίωσης (documentation). Με βάση αυτά έγινε η ανάλυση που βλέπουμε στον Πίνακα 4.1 και επιλέχθηκε το Elliot framework για τη δημιουργία των αρχικών προτάσεων από τους base αλγορίθμους συστημάτων συστάσεων.

```
experiment:
  dataset: movielens_1m
  data_config:
    strategy: fixed
    train_path: data/movielens_1m/splitting/train.tsv
    test_path: data/movielens_1m/splitting/test.tsv
  path_output_rec_result: Thesis/Results
  path_output_rec_performance: Thesis/Results
  splitting:
    save_on_disk: True
    save_folder: data/movielens_1m/splitting/
    test_splitting:
      strategy: random_subsampling
      test_ratio: 0.2
  top_k: 50
  evaluation:
    cutoffs: [50, 30, 20, 10]
    simple_metrics: [nDCG, Precision, Recall, ItemCoverage, EPC, Gini, HR, ARP, ACLT, APLT,
    PopREO, PopRSP]
    relevance_threshold: 0
  gpu: 1
  external_models_path: elliot-master/external/models/__init__.py
  models:
    ItemKNN:
      meta:
        verbose: True
        save_recs: True
        validation_metric: nDCG@10
      neighbors: [50, 70]
      similarity: cosine
      implementation: classical
    UserKNN:
      meta:
        verbose: True
        save_recs: True
        validation_metric: nDCG@10
      neighbors: [ 50,70 ]
      similarity: cosine
      implementation: classical
```

Κώδικας 4.1: Παράδειγμα yaml αρχείου στο Elliot

Στο Elliot framework είναι αρκετά απλή η εκτέλεση ενός πειράματος για διαφορετικούς αλγορίθμους και η αξιολόγηση από τις μετρικές που επιθυμούμε δημιουργώντας απλά ένα αρχείο τύπου .yaml. Ένα παράδειγμα τέτοιου αρχείου δίνεται στον κώδικα 4.1. Ακόμη υπάρχει και η δυνατότητα

για ρύθμιση υπερπαραμέτρων (hyperparameter tuning) επιλέγοντας ανάμεσα σε 51 στρατηγικές. Τέλος, το Elliot δίνει την δυνατότητα για στατιστικές δοκιμές.
Η εφαρμογή "Bias & fairness in Recsys" περιέχει 7 διαφορετικές σελίδες στις οποίες μπορεί να πλοηγηθεί ο χρήστης:

1. Αρχική (Home)
2. Οπτικοποίηση δεδομένων (Visualize data)
3. Δημιουργία συστημάτων συστάσεων (Build recommendation systems)
4. Αξιολόγηση αποτελεσμάτων (Bias identification)
5. Μετριασμός μεροληψίας (Bias mitigation)
6. Επεξήγηση μετρικών (Metrics explanation)
7. Μεταφόρτωση δεδομένων (Upload data)

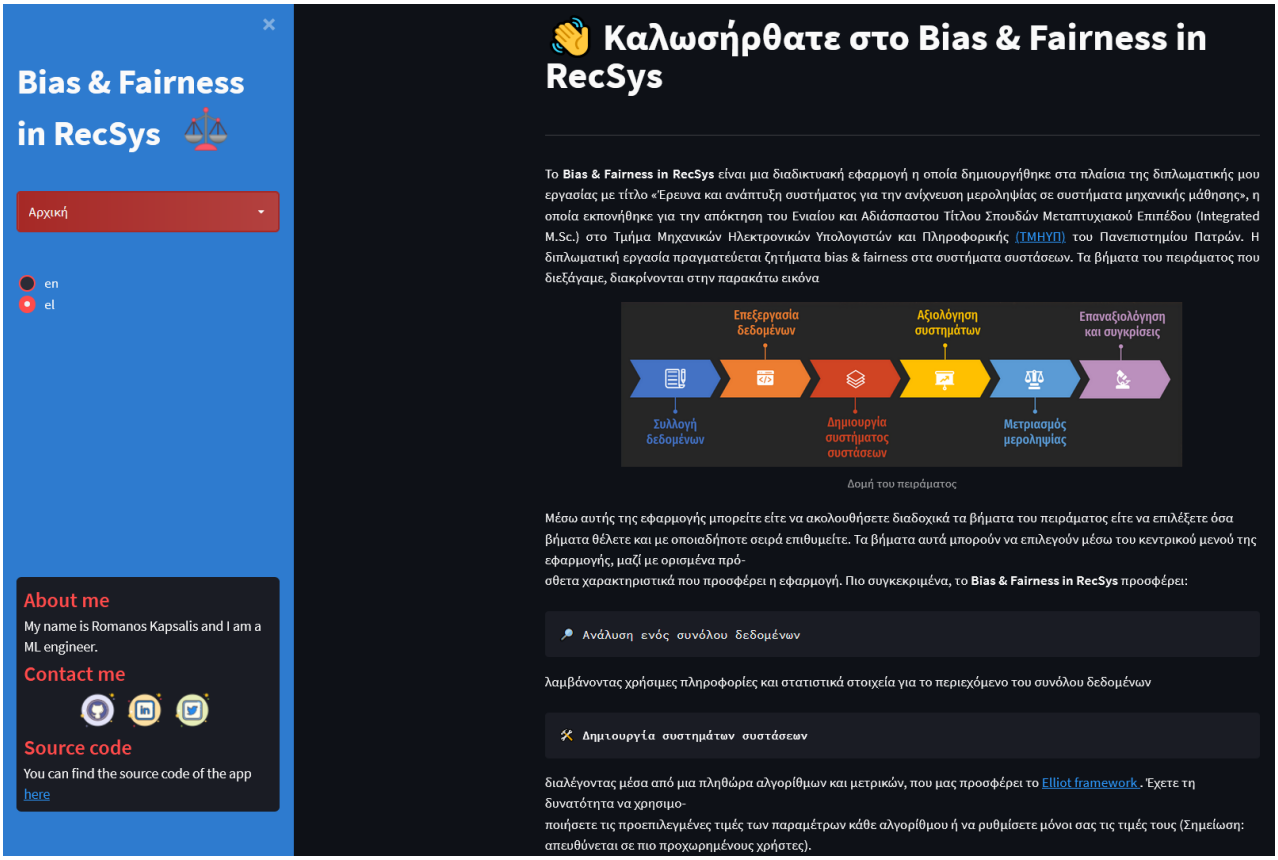
Η εφαρμογή υποστηρίζει δύο γλώσσες ελληνικά και αγγλικά, η εναλλαγή μεταξύ των οποίων είναι αρκετά εύκολη. Θα πρέπει να σημειωθεί πως για όλα τα γραφήματα που παρουσιάζονται στους χρήστες σε κάθε σελίδα της εφαρμογής υπάρχει η δυνατότητα να τα αποθηκεύσουν στο σύστημά τους ή/και να τα επεξεργαστούν (να εστιάσουν στο σημείο που θέλουν). Ακολουθεί η αναλυτική περιγραφή όλων των σελίδων της εφαρμογής, ενώ σε όλες τις περιπτώσεις η γλώσσα της εφαρμογής που θα χρησιμοποιήσουμε θα είναι η ελληνική.

4.1 Αρχική σελίδα

Στην αρχική σελίδα παρουσιάζονται ορισμένες πληροφορίες για τη δομή του πειράματος και της εφαρμογής. Επίσης, στον χρήστη δίνονται αναλυτικές οδηγίες για τη χρήση της εφαρμογής και την υλοποίηση του πειράματος μέσω αυτής. Ο χρήστης έχει τη δυνατότητα να πλοηγηθεί στις σελίδες ώστε να εκτελέσει τα βήματα του πειράματος με τη σειρά που αναφέρονται ή να ακολουθήσει ορισμένες μόνο φάσεις του πειράματος με οποιαδήποτε σειρά επιθυμεί ο ίδιος.

4.2 Οπτικοποίηση δεδομένων

Σε αυτή τη σελίδα αρχικά επιλέγουμε ένα σύνολο δεδομένων που επιθυμούμε να αναλύσουμε και ακολούθως προβάλλονται στην οθόνη ορισμένα χρήσιμα στατιστικά στοιχεία αυτού του συνόλου δεδομένων, γραφήματα και πίνακες που το περιγράφουν. Όσον αφορά τα στατιστικά στοιχεία, για όλα τα σύνολα δεδομένων προβάλλεται ο αριθμός των χρηστών, των αντικειμένων και των αξιολογήσεων, η αραιότητα του μητρώου χρηστών-αντικειμένων, το rating space, ο λόγος των χρηστών προς τα αντικείμενα, ο λόγος των αξιολογήσεων προς τα αντικείμενα και ο λόγος των αξιολογήσεων προς τους χρήστες.
Τα γραφήματα δίνουν στον χρήστη χρήσιμες πληροφορίες για ένα σύνολο δεδομένων προκειμένου να κατανοήσει καλύτερα τη δομή και την κατανομή των δεδομένων που υπάρχουν σε αυτό, δηλαδή των χρηστών, των αντικειμένων και των αξιολογήσεων στην περίπτωσή μας. Το πρώτο γράφημα που προβάλλεται παρουσιάζει τα πιο δημοφιλή προϊόντα, ως προς τον αριθμό των αξιολογήσεων

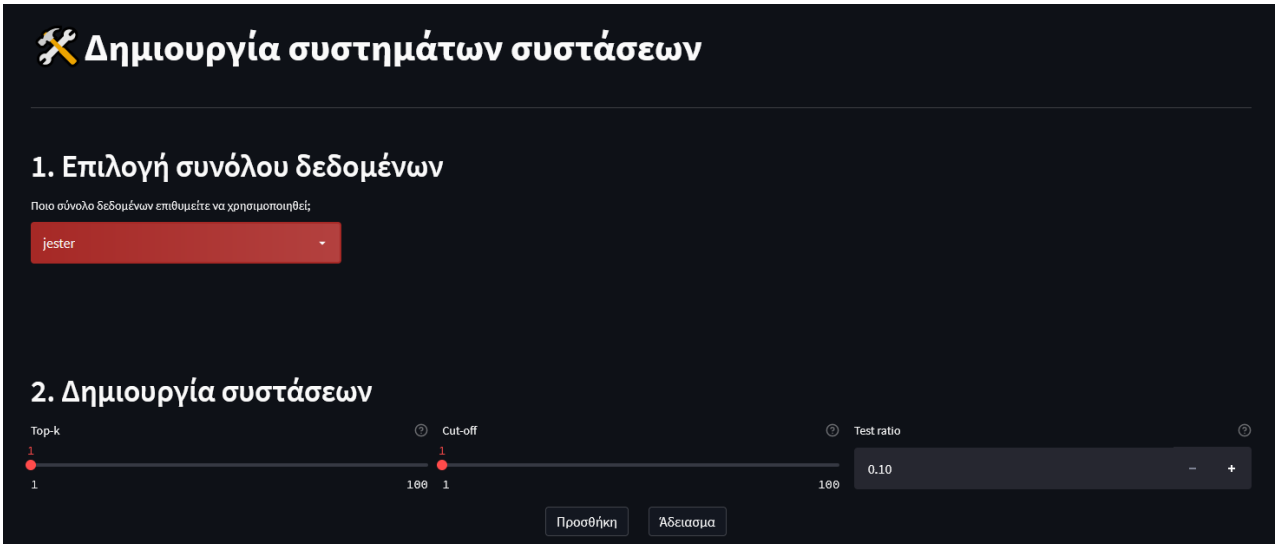


Εικόνα 4.2: Αρχική σελίδα

που έχουν λάβει, παρέχοντας μάλιστα τη δυνατότητα επιλογής του αριθμού των πιο δημοφιλών αντικειμένων που επιθυμούμε. Το επόμενο γράφημα μας δίνει τους κορυφαίους χρήστες, ως προς τον αριθμό των αξιολογήσεων που έχουν πραγματοποιήσει και εδώ όπως και στο προηγούμενο γράφημα, παρέχεται η δυνατότητα επιλογής του αριθμού των πιο δημοφιλών χρηστών που επιθυμούμε να προβάλλουμε. Ένα αρκετά χρήσιμο γράφημα, ειδικά αν θέλουμε να δημιουργήσουμε συστήματα συστάσεων, αποτελεί η κατανομή της μέσης τιμής των αξιολογήσεων για όλα τα αντικείμενα. Τέλος, θα αποτελούσε σημαντική παράλειψη αν δεν παρουσιάζαμε το γράφημα του long tail, το οποίο σχετίζεται άμεσα με το popularity bias στα συστήματα συστάσεων. Στα σύνολα δεδομένων του Movielens που είναι διαθέσιμες και κάποιες επιπλέον πληροφορίες όπως είναι τα δημογραφικά χαρακτηριστικά των χρηστών και τα είδη των ταινιών υπάρχουν επιπλέον δύο γραφήματα, ένα για την κατανομή των ηλικιακών ομάδων των χρηστών και ένα που σχετίζεται με το φύλο και δείχνει το ποσοστό των ανδρών και των γυναικών.

4.3 Δημιουργία συστημάτων συστάσεων

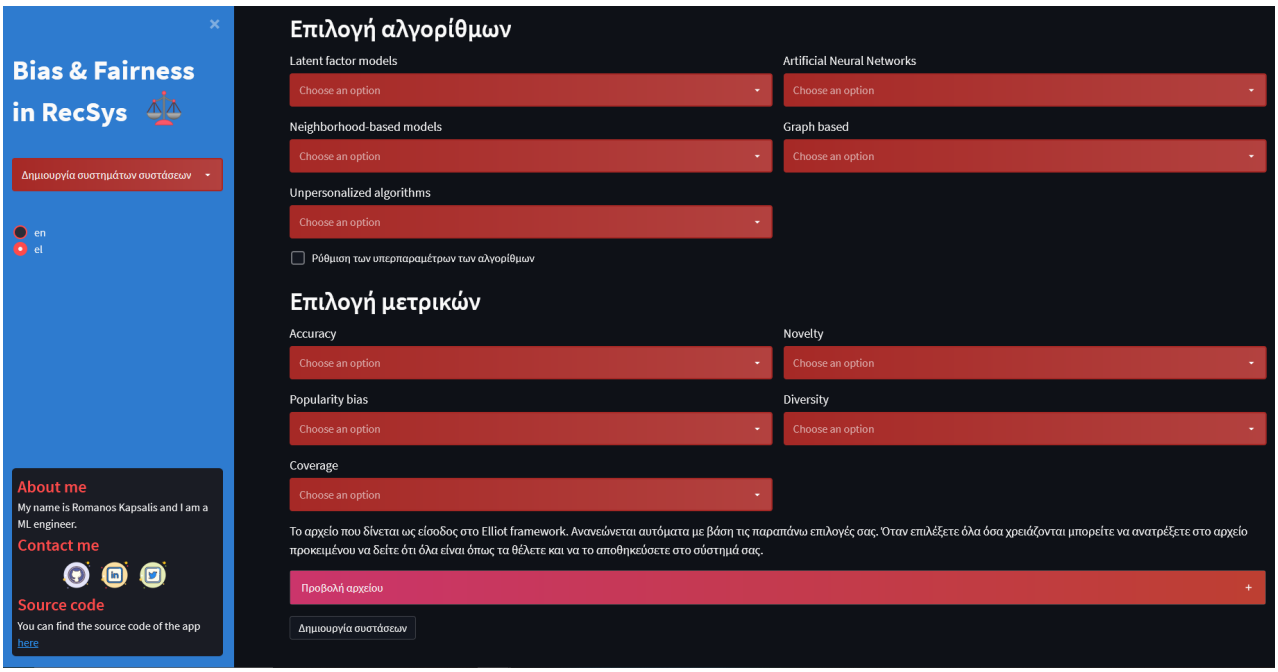
Η σελίδα «Δημιουργία συστημάτων συστάσεων» υλοποιεί ουσιαστικά το 3ο βήμα του πειράματος (Εικόνα 4.1), το οποίο είναι η δημιουργία των συστημάτων συστάσεων. Η εφαρμογή προκειμένου να δημιουργήσει τα συστήματα συστάσεων συνδέεται με το Elliot framework.



Εικόνα 4.3: Σελίδα δημιουργίας συστήματος συστάσεων

Πρώτα, όπως φαίνεται και στην Εικόνα 4.3 θα πρέπει να γίνει η επιλογή του συνόλου δεδομένων στο οποίο θα δημιουργηθούν τα συστήματα συστάσεων. Ο χρήστης μπορεί να επιλέξει ένα από τα σύνολα δεδομένων που χρησιμοποιήσαμε στο πείραμα ή κάποιο σύνολο δεδομένων που έχει ανεβάσει μέσω της σχετικής σελίδας («Μεταφόρτωση δεδομένων») που υπάρχει στην εφαρμογή μας. Επόμενο βήμα, είναι ο καθορισμός του μεγέθους των λιστών συστάσεων που θα δημιουργήσουν οι αλγόριθμοι (top-k), του μεγέθους ή των μεγεθών των λιστών συστάσεων που θα λάβει υπόψη της μια μετρική αξιολόγησης (cut-off) και του ποσοστού του συνόλου δεδομένων που θα χρησιμοποιηθεί για τη δημιουργία του συνόλου δοκιμής, κατά τη διάσπασή του σε δύο υποσύνολα, το σύνολο εκπαίδευσης και το σύνολο δοκιμής. Για τη χρήση παραπάνω από μίας τιμής cut-off, αφού επιλεγεί η τιμή προστίθεται στη λίστα των cut-off με το πάτημα του κουμπιού «Προσθήκη». Προφανώς όπως γίνεται αντιληπτό, θα πρέπει να ισχύει ότι cut-off $\leq$ top-k , σε διαφορετική περίπτωση ο χρήστης λαμβάνει προειδοποιητικό μήνυμα και καλείται να αλλάξει τις τιμές ώστε να ικανοποιείται αυτός ο περιορισμός. Επίσης, με το πάτημα του κουμπιού «Άδειασμα», διαγράφονται όλες οι τιμές που εμπεριέχονται στην λίστα των cut-off, καλύπτοντας με αυτόν τον τρόπο την περίπτωση εισαγωγής λάθος τιμής από τον χρήστη. Θα πρέπει να επισημάνουμε εδώ, πως στην περίπτωση εισαγωγής μιας τιμής στην λίστα των cut-off παρουσιάζεται μήνυμα στην οθόνη το οποίο μας ενημερώνει για την επιτυχή προσθήκη μιας τιμής και για το περιεχόμενο της λίστας cut-off. Αντίστοιχο μήνυμα προβάλλεται και κατά το άδειασμα της λίστας.
Ακολούθως, ο χρήστης καλείται να επιλέξει τους αλγορίθμους που επιθυμεί να χρησιμοποιηθούν στο πείραμα για τη δημιουργία των συστάσεων. Οι αλγόριθμοι είναι κατηγοριοποιημένοι ανά οικογένεια αλγορίθμων, όπως φαίνεται και στην Εικόνα 4.4, προς διευκόλυνση των χρηστών. Σε αυτό το σημείο θα πρέπει να επισημανθεί πως δεν έχουν χρησιμοποιηθεί όλες οι οικογένειες αλγορίθμων που υπάρ-

χουν στο Elliot, παρά μόνο όσες χρησιμοποιήσαμε στο πείραμά μας και όσες ήταν σχετικές με αυτό. Οι οικογένειες αυτές είναι: latent factor models, artificial neural networks, unpersonalized, graph based και neighborhood based. Ακριβώς κάτω από το σημείο της σελίδας στο οποίο γίνεται η επι-

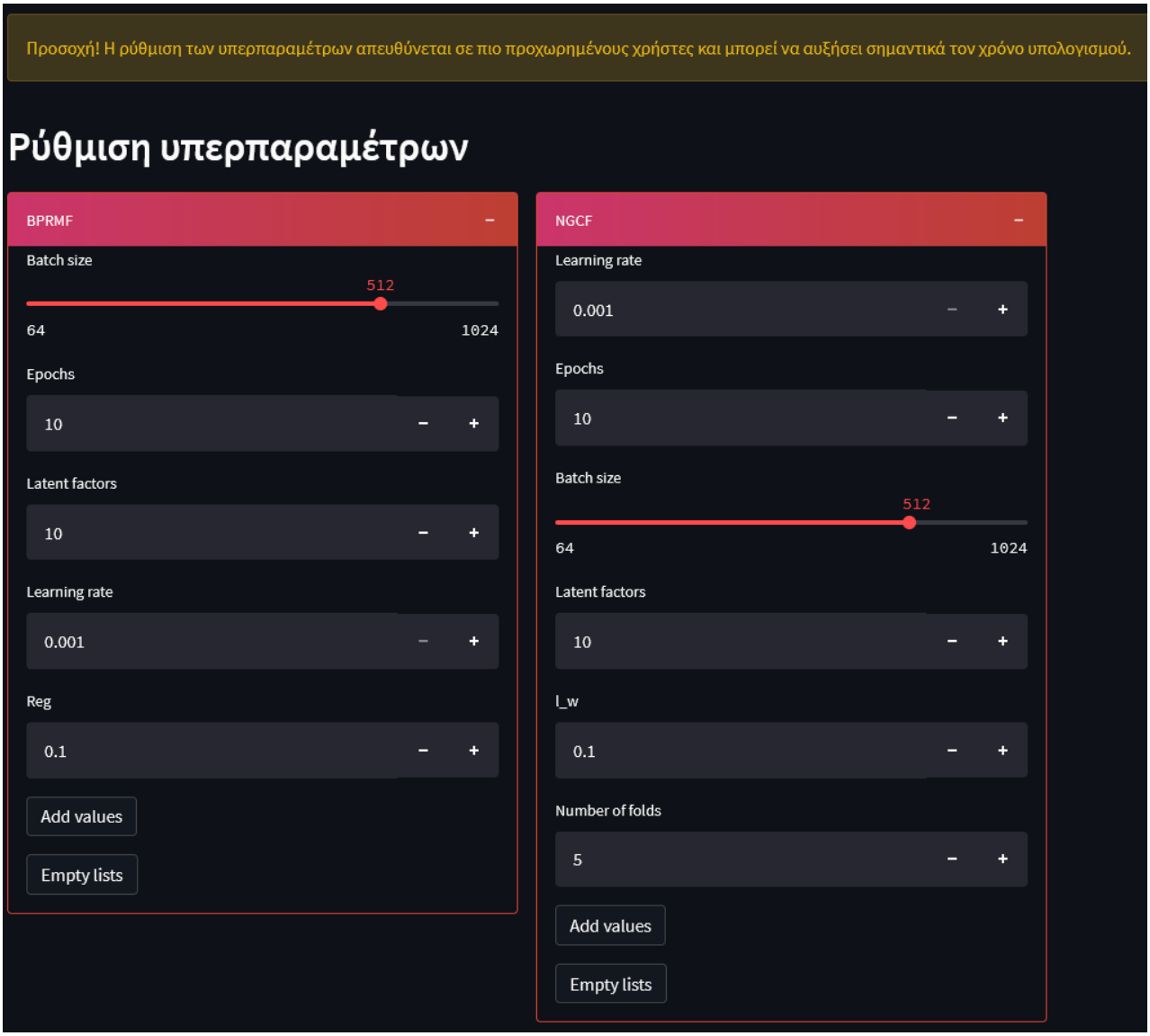


Εικόνα 4.4: Σελίδα δημιουργίας συστήματος συστάσεων

λογή των αλγορίθμων, υπάρχει το checkbox «Ρύθμιση των υπερπαραμέτρων των αλγορίθμων», το οποίο εάν πατηθεί μας επιτρέπει να ρυθμίσουμε τις παραμέτρους των αλγορίθμων (hyperparameter optimization). Πιο αναλυτικά, σε αυτή την περίπτωση δεν χρησιμοποιούνται οι προκαθορισμένες παράμετροι των αλγορίθμων, αλλά ρυθμίζονται από τον ίδιο τον χρήστη. Έτσι για κάθε αλγόριθμο που έχει επιλεχθεί, δημιουργείται ένα πλαίσιο το οποίο περιέχει τις διαθέσιμες παραμέτρους για αυτόν. Εντός αυτού του πλαισίου υπάρχουν επίσης δύο κουμπιά «Προσθήκη» και «Διαγραφή». Η ρύθμιση των παραμέτρων είναι διαθέσιμη για όλες τις οικογένειες αλγορίθμων εκτός από την unpersonalized, καθώς για τους αλγορίθμους που ανήκουν σε αυτή την οικογένεια δεν υπάρχουν παράμετροι για να ρυθμιστούν. Επειδή ωστόσο αυτή η επιλογή απαιτεί καλή γνώση των αλγορίθμων και δυνητικά μπορεί να αυξήσει σημαντικά το υπολογιστικό κόστος, κρίθηκε αναγκαίο να υπάρχει κατάλληλη προειδοποίηση. Στην Εικόνα 4.5 δίνεται ένα παράδειγμα ρύθμισης των παραμέτρων των αλγορίθμων BPRMF και NGCF.

Ένα ακόμη απαραίτητο στοιχείο για την διεξαγωγή ενός πειράματος είναι φυσικά οι μετρικές αξιολόγησης, και αυτές, όπως και οι αλγόριθμοι, είναι κατηγοριοποιημένες. Οι κατηγορίες είναι: ακρίβεια, popularity bias, diversity, novelty και coverage. Προκειμένου να διευκολύνουμε όσο περισσότερο γίνεται όλους τους χρήστες, υπάρχει η δυνατότητα προβολής του αρχείου yml που χρησιμοποιεί το Elliot, το οποίο δημιουργείται αυτόματα από την εφαρμογή μας και ανανεώνεται δυναμικά καθώς ο χρήστης εισάγει ή διαγράφει μια μετρική ή έναν αλγόριθμο και τις παραμέτρους. Μάλιστα, αφού ολοκληρωθεί η επιλογή όλων των στοιχείων του πειράματος και βεβαιωθεί ο χρήστης πως δεν έχει κάνει κάποιο λάθος, παρέχεται η δυνατότητα να κατεβάσει το συγκεκριμένο αρχείο στη συσκευή του, πατώντας το κουμπί «Λήψη αρχείου».

Τέλος, μετά τη ρύθμιση όλων των παραπάνω στοιχείων, πατώντας το κουμπί «Δημιουργία», ξεκινά-



Εικόνα 4.5: Ρύθμιση υπερπαραμέτρων των αλγορίθμων BPRMF και NGCF μέσω της εφαρμογής

ει η δημιουργία των λιστών συστάσεων από τους αλγορίθμους που έχουν επιλεγεί και η αξιολόγηση των αποτελεσμάτων από τις μετρικές αξιολόγησης, μέσω του Elliot. Τα αρχεία αποτελεσμάτων που έχουν δημιουργηθεί αποθηκεύονται προσωρινά στο σύστημα έως ότου ο χρήστης να κλείσει την εφαρμογή.

Από όλα τα παραπάνω συνάγεται το συμπέρασμα ότι το σύστημα που αναπτύξαμε, δεν αποτελεί μόνο ένα εργαλείο για τον εντοπισμό και τον μετριασμό της μεροληψίας αλλά και ένα εργαλείο για τη δημιουργία συστημάτων συστάσεων με αρκετά φιλικό τρόπο προς τους χρήστες

4.4 Αξιολόγηση αποτελεσμάτων

Σε αυτή τη σελίδα γίνεται η ανάλυση των αποτελεσμάτων, δηλαδή των λιστών συστάσεων για κάθε χρήστη, που έχουν παράξει οι αλγόριθμοι του Elliot. Η ανάλυση μπορεί να γίνει είτε επιλέγοντας τα αποτελέσματα που αφορούν ένα σύνολο δεδομένων είτε μπορεί να γίνει σύγκριση διαφορετικών συνόλων δεδομένων.

1. Επιλογή ενός συνόλου δεδομένων
Εάν επιλεγεί να γίνει η ανάλυση των αποτελεσμάτων ενός μόνο συνόλου δεδομένων, τότε αρχικά θα πρέπει να επιλέξουμε τον τύπο ή τους τύπους της ανάλυσης που επιθυμούμε. Υπάρχουν διαθέσιμοι τρεις τύποι ανάλυσης των αποτελεσμάτων: προβολή καλύτερων αποτελεσμάτων, ανάλυση υπερπαραμέτρων και ανάλυση cut-off, με τον προεπιλεγμένο τύπο ανάλυσης να είναι η προβολή των καλύτερων αποτελεσμάτων.
Στη συνέχεια, γίνεται η επιλογή ενός συνόλου δεδομένων. Για όλα τα σύνολα δεδομένων η εύρεση και η ανάγνωση των αρχείων αποτελεσμάτων γίνεται αυτόματα από το σύστημα χωρίς να απαιτείται καμία περαιτέρω ενέργεια του χρήστη. Ωστόσο, όπως γίνεται αντιληπτό για το σύνολο δεδομένων που έχει μεταφορτώσει ο χρήστης, απαραίτητη προϋπόθεση είναι να έχει δημιουργήσει πρώτα συστήματα συστάσεων μέσω της σχετικής σελίδας της εφαρμογής.
Επόμενο βήμα αποτελεί η επιλογή των μετρικών αξιολόγησης. Οι μετρικές αξιολόγησης είναι κατηγοριοποιημένες, ακριβώς όπως και στη σελίδα «Δημιουργία συστημάτων συστάσεων». Έπειτα, ανάλογα με τον τύπο ανάλυσης των αποτελεσμάτων προβάλλονται τα σχετικά γραφήματα.

Προβολή καλύτερων αποτελεσμάτων
Στην προβολή των καλύτερων αποτελεσμάτων προβάλλονται τα καλύτερα αποτελέσματα που έχουν προκύψει σε κάθε διαθέσιμη τιμή cut-off, δηλαδή ο καλύτερος συνδυασμός των τιμών όλων των υπερπαραμέτρων και ο χρήστης έχει τη δυνατότητα να επιλέξει την τιμή του cut-off που επιθυμεί.

Ανάλυση cut-off
Η διαφορά της ανάλυσης cut-off από την προβολή των καλύτερων αποτελεσμάτων είναι στο είδος των γραφημάτων που παρουσιάζονται και πως δεν χρειάζεται να επιλεγεί κάποια τιμή cut-off. Πιο συγκεκριμένα, προβάλλονται οι τιμές που έχουν προκύψει για όλες τις τιμές cut-off σε ένα γράφημα για όλους τους αλγορίθμους και για μια επιλεγμένη μετρική.

Εικόνα 4.6: Παράδειγμα προβολής καλύτερων αποτελεσμάτων και ανάλυσης cut-off

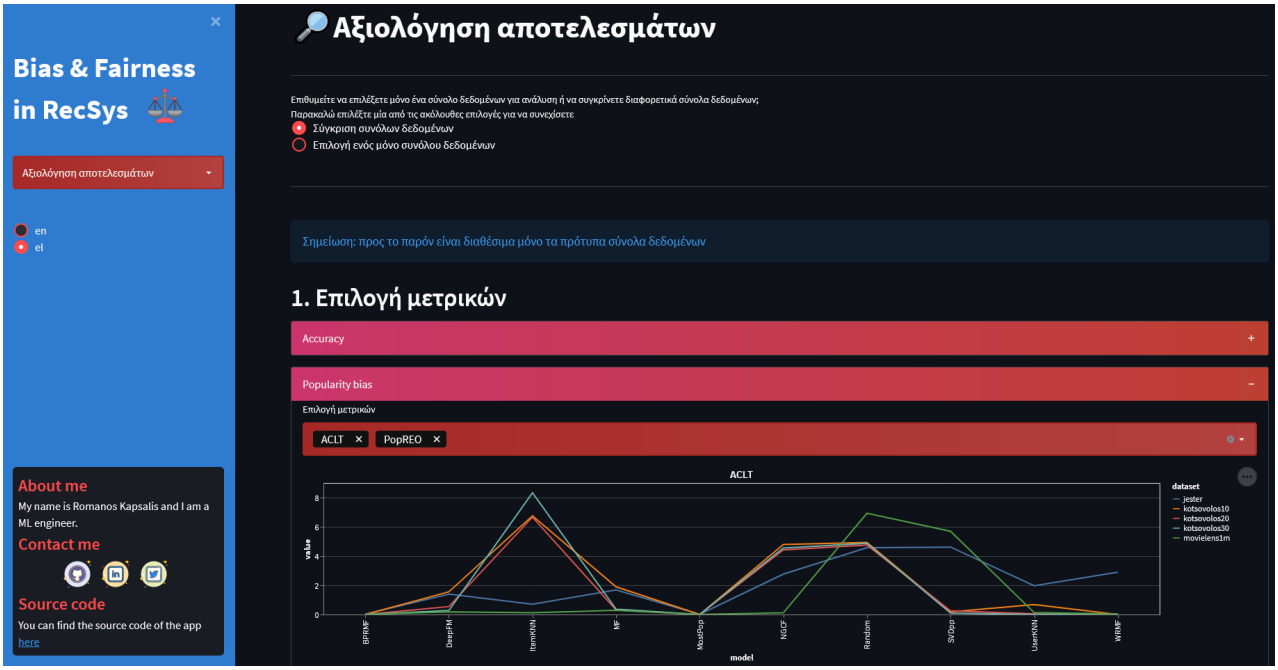
Ανάλυση υπερπαραμέτρων

Αυτός ο τύπος ανάλυσης χρειάζεται εκτός από το σύνολο δεδομένων και τις μετρικές αξιολόγησης, και την επιλογή του αλγορίθμου ή των αλγορίθμων, του οποίου ή των οποίων οι παράμετροι θα αναλυθούν. Αφού επιλεγεί ο αλγόριθμος, τότε ο χρήστης επιλέγει επίσης και τις παραμέτρους που θέλει και με τη σειρά που θέλει. Η μια παράμετρος είναι εκείνη που θα βρίσκεται στον άξονα x και η άλλη καθορίζει το πλήθος των γραμμών (με άλλα λόγια καθορίζει ποια υπερπαράμετρος θα χρησιμοποιηθεί για να κωδικοποιηθούν οι μοναδικές τιμές της μέσω διαφορετικών χρωμάτων). Εάν ο αριθμός των παραμέτρων ενός αλγορίθμου είναι μεγαλύτερος του 3, τότε υπάρχει και μια τρίτη παράμετρος που μπορεί να επιλεγεί και αφορά την μεταβλητή που θα κρατηθεί σταθερή. Δηλαδή, θα δημιουργηθούν τόσα γραφήματα, όσες και οι τιμές της παραμέτρου. Εάν επιλεγεί να γίνει σύγκριση διαφορετικών συνόλων δεδομένων, υπάρχει διαθέσιμος μόνο ένας τύπος ανάλυσης, η προβολή των καλύτερων αποτελεσμάτων. Αυτό συμβαίνει διότι η ανάλυση των υπερπαραμέτρων για έναν αλγόριθμο σε διαφορετικά σύνολα δεδομένων, είναι μια αρκετά σύνθετη και δύσκολη διαδικασία, καθώς τα σύνολα δεδομένων πολλές φορές διαφέρουν κατά πολύ μεταξύ τους, με κυριότερη διαφορά το πολύ μεγάλο εύρος τιμών cut-off που μπορούμε να συναντήσουμε. Επομένως, θα πρέπει να ληφθούν υπόψη αρκετές παράμετροι, κάτι που δεν εξετάστηκε στην παρούσα εργασία και αποτελεί αντικείμενο μελλοντικής εργασίας.

Εικόνα 4.7: Παράδειγμα ανάλυσης υπερπαραμέτρων.

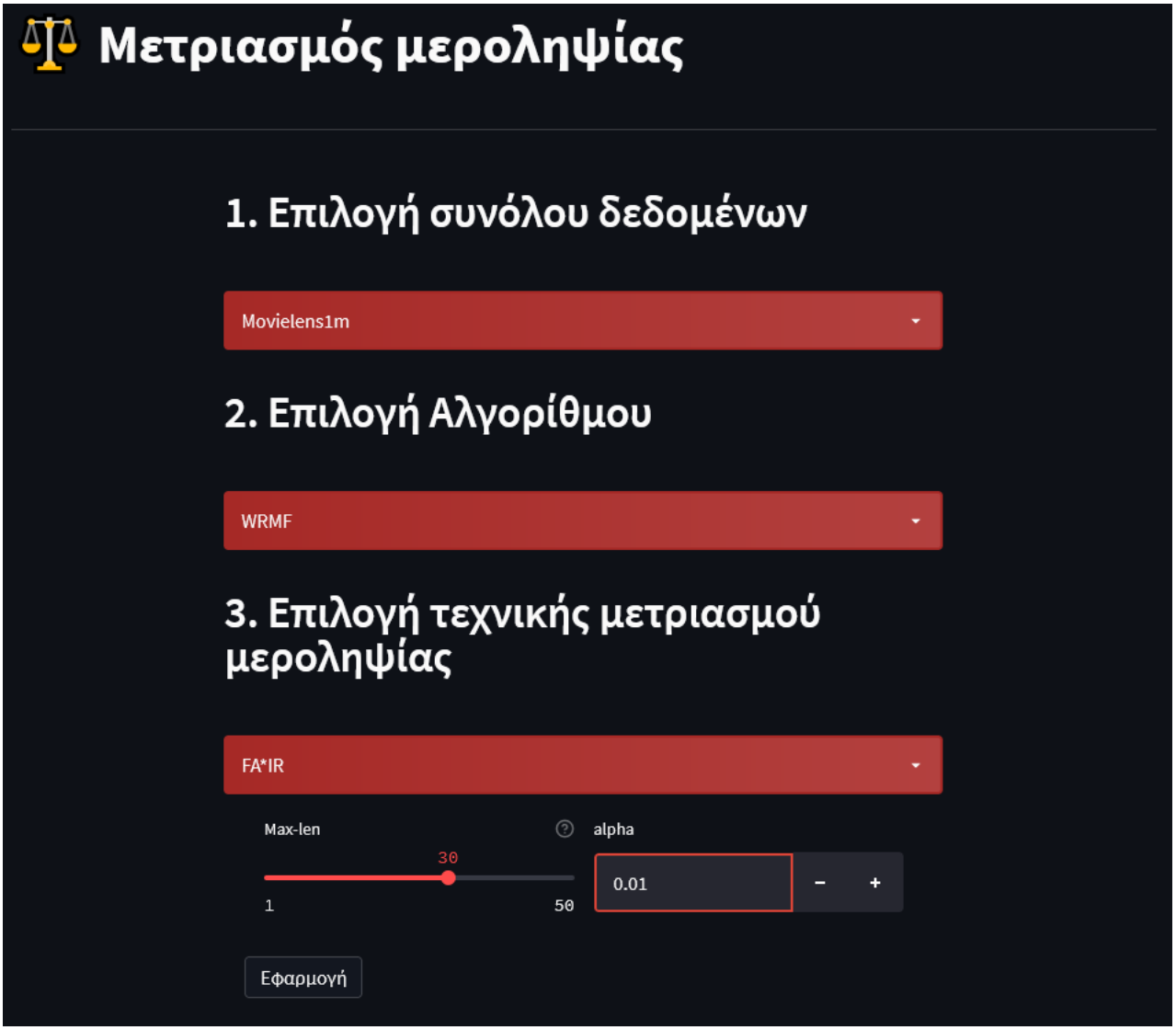
2. Σύγκριση διαφορετικών συνόλων δεδομένων

Στη σύγκριση διαφορετικών συνόλων δεδομένων (Εικόνα 4.7) ο χρήστης μπορεί να συγκρίνει τα αποτελέσματα όπως αυτά προέκυψαν από διαφορετικά σύνολα δεδομένων και έχει δύο επιλογές. Η πρώτη είναι να καθορίσει ο ίδιος τα σύνολα δεδομένων που επιθυμεί, εάν προηγουμένως έχει μεταφορτώσει τα αρχεία που εμπεριέχουν τα καλύτερα αποτελέσματα για κάθε cut-off για όλους τους αλγορίθμους, για κάποιο σύνολο δεδομένων που έχει στην κατοχή του ο χρήστης (περισσότερες λεπτομέρειες θα βρείτε στην υποενότητα «Μεταφόρτωση δεδομένων»). Ενώ η δεύτερη είναι η σύγκριση όλων των προτύπων συνόλων δεδομένων. Σε αυτή την έκδοση της εφαρμογής είναι διαθέσιμη μόνο η 2η επιλογή, οι λόγοι που οδήγησαν σε αυτόν τον περιορισμό περιγράφονται αναλυτικά στην ενότητα 6.2 του Κεφαλαίου 6. Και στις δύο περιπτώσεις, όπως και στην «Επιλογή ενός μόνο συνόλου δεδομένων», εμφανίζονται γραφήματα για κάθε μετρική που έχει προεπιλεγεί.



Εικόνα 4.8: Παράδειγμα σύγκρισης συνόλων δεδομένων.

4.5 Μετριασμός μεροληψίας



Εικόνα 4.9: Σελίδα μετριασμού μεροληψίας

Η σελίδα αυτή υλοποιεί το βήμα του πειράματος κατά το οποίο γίνεται ο μετριασμός του popularity bias που έχει εντοπιστεί σε ένα σύστημα συστάσεων. Όταν ο χρήστης επισκέπτεται αυτή τη σελίδα, τότε μπορεί είτε να διαβάσει αναλυτικές πληροφορίες για τους αλγορίθμους μετριασμού της μεροληψίας και για τη διαδικασία γενικότερα, είτε να μετριάσει κατευθείαν την μεροληψία σε έναν αλγόριθμο, εάν είναι εξοικειωμένος με τη διαδικασία.

Ο μετριασμός της μεροληψίας αποτελείται από τρία αρκετά απλά βήματα. Αρχικά, και όπως σε όλα τα βήματα που έχουμε περιγράψει έως τώρα, γίνεται η επιλογή του συνόλου δεδομένων και έπειτα η επιλογή του αλγορίθμου ο οποίος δημιούργησε τις λίστες συστάσεων στις οποίες πιθανώς εντοπίσαμε την ύπαρξη μεροληψίας μέσω της σελίδας «Ανάλυση δεδομένων». Προφανώς η λίστα με τους διαθέσιμους αλγόριθμους αλλάζει και ανανεώνεται δυναμικά, κάθε φορά που επιλέγουμε κάποιο σύνολο δεδομένων. Μετά την επιλογή του αλγορίθμου, σειρά έχει η επιλογή της τεχνικής που επιθυμούμε να εφαρμόσουμε για τον μετριασμό της μεροληψίας. Υπάρχουν τέσσερις διαθέσιμες τεχνι-

κές οι οποίες ανήκουν όλες στην κατηγορία των post-processing αλγορίθμων, και πιο συγκεκριμένα τεχνικές που αναδιατάσσουν (re-ranking) μια λίστα συστάσεων. Οι αλγόριθμοι αυτοί είναι οι FAR, PFAR, FA*IR και Calibrated Recommendations (Cali) τους οποίους έχουμε περιγράψει εκτενώς στην Ενότητα 3 και προέρχονται από το framework Librec-auto. Σε αυτούς δίνονται ως είσοδος η λίστα συστάσεων για κάθε χρήστη που έχει δημιουργήσει ο base αλγόριθμος, καθώς και ένα αρχείο τύπου csv, που περιέχει όλα τα IDs των αντικειμένων και τον χαρακτηρισμό long ή head, αν ανήκει στο long tail και στο head, αντίστοιχα για καθένα από αυτά. Ένα παράδειγμα τέτοιου αρχείου δίνεται στην Εικόνα 4.10. Για τον αλγόριθμο FA*IR υπάρχει προειδοποίηση προς τους χρήστες ότι ανάλογα με το μέγεθος του συνόλου δεδομένων ο υπολογισμός του μπορεί να είναι εξαιρετικά αργός.

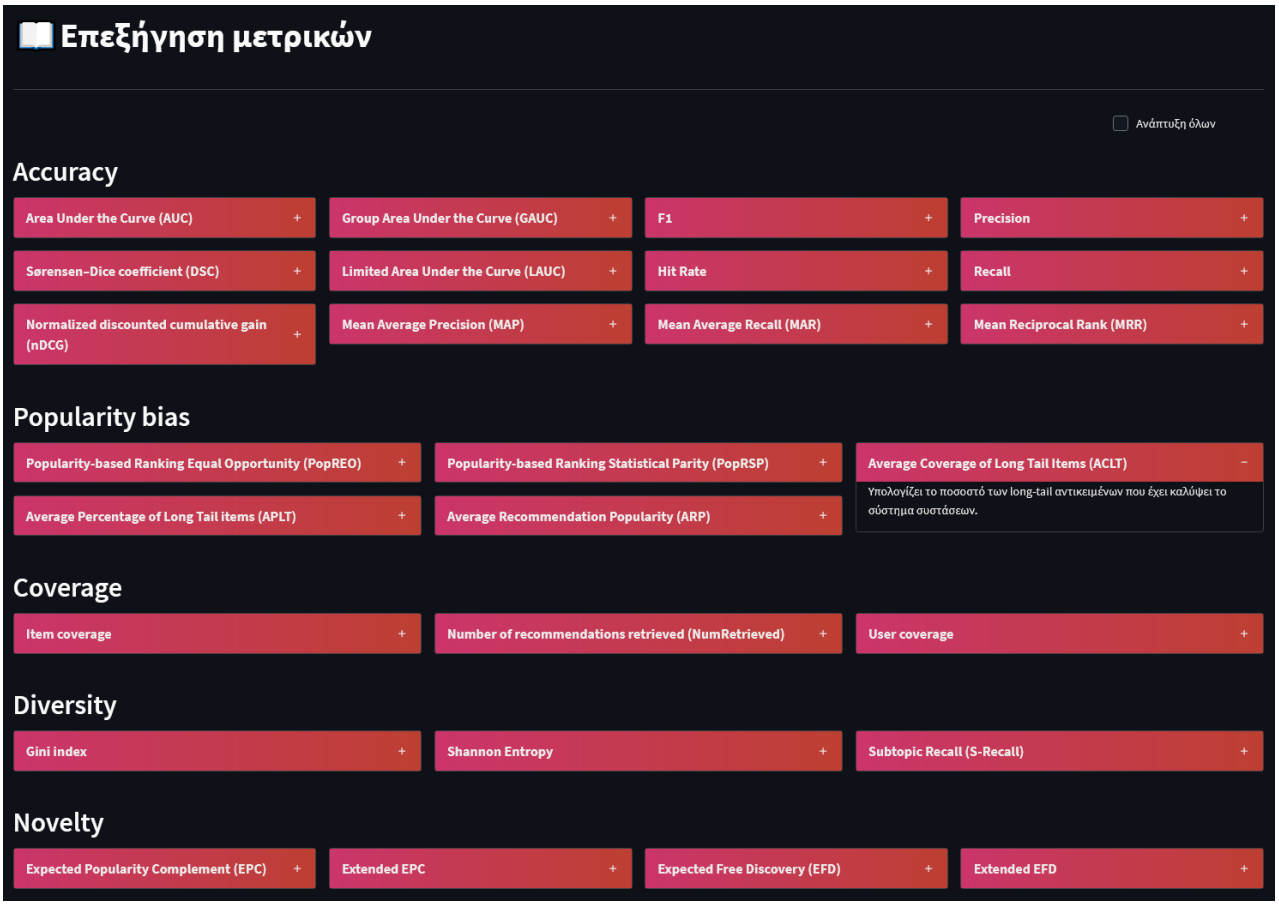
```
itemid,feature,value
50,head,0
258,head,0
904,long,1
100,head,0
1030,long,1
.    .     .
.    .     .
.    .     .
```

Εικόνα 4.10: Παράδειγμα αρχείου item features στο Movielens100k

Αξίζει να σημειωθεί ότι για την εύρεση του αρχείου που έδωσε τα καλύτερα αποτελέσματα γίνεται πρώτα η ανάγνωση ενός αρχείου τύπου json που έχει δημιουργήσει το Elliot και περιέχει τις καλύτερες παραμέτρους. Κάθε ένας από αυτούς τους αλγορίθμους περιέχει και δύο παραμέτρους τις οποίες θα πρέπει να ρυθμίσει ο χρήστης. Η μία παράμετρος είναι το μέγεθος των λιστών συστάσεων για κάθε χρήστη, αυτό θα πρέπει σε κάθε περίπτωση να είναι μικρότερο από το αρχικό μέγεθος, και η άλλη παράμετρος καθορίζει τον βαθμό ισορροπίας ανάμεσα στην ακρίβεια και στη μεροληψία. Για την εφαρμογή της επιλεγμένης τεχνικής μετριασμού της μεροληψίας αρκεί το πάτημα του κουμπιού «Εφαρμογή». Αφού ολοκληρωθεί η διαδικασία, τότε γίνεται η αξιολόγηση των νέων λιστών που έχουν παραχθεί, χρησιμοποιώντας τις μετρικές αξιολόγησης που είχαν χρησιμοποιηθεί και στο πείραμα κατά το οποίο δημιουργήθηκαν οι αρχικές λίστες συστάσεων. Ο χρήστης ενημερώνεται με σχετικά μηνύματα που προβάλλονται στην οθόνη για την πρόοδο τόσο του μετριασμού της μεροληψίας, όσο και της αξιολόγησης των αποτελεσμάτων. Τέλος, και με προϋπόθεση την επιτυχή έκβαση των δύο παραπάνω διαδικασιών, στον χρήστη παρουσιάζονται τα αποτελέσματα των μετρικών αξιολόγησης μέσω γραφημάτων.

4.6 Επεξήγηση μετρικών

Αναφερθήκαμε προηγουμένως στην αναγκαιότητα αυτή η εφαρμογή να μπορεί να χρησιμοποιηθεί και από άτομα χωρίς ιδιαίτερες τεχνικές γνώσεις. Σε αυτό το πλαίσιο, δημιουργήθηκε μια ακόμη σελίδα η οποία περιγράφει κάθε μετρική που είναι διαθέσιμη στο Elliot, σε γλώσσα απλή και κατανοητή για όλους τους χρήστες ανεξαρτήτως των γνώσεων τους. Για παράδειγμα, στην Εικόνα 4.11 ο χρήστης έχει επιλέξει να προβάλλει την επεξήγηση για την μετρική ACLT. Προς διευκόλυνση του χρήστη υπάρχει η δυνατότητα ανάπτυξης όλων των πλαισίων, μέσω απλής επιλογής στο πάνω δεξιό μέρος της οθόνης («Ανάπτυξη όλων»), ενώ οι μετρικές είναι κατηγοριοποιημένες σε 5 κατηγορίες.

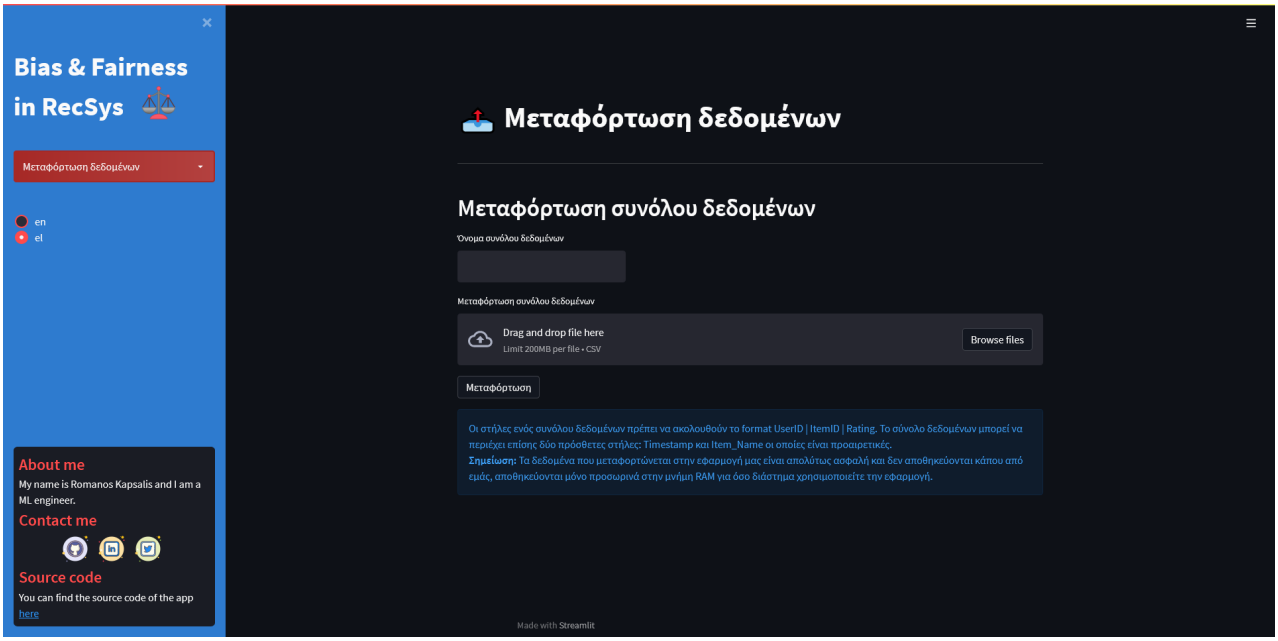


Εικόνα 4.11: Σελίδα επεξήγησης μετρικών αξιολόγησης.

Οι επεξηγήσεις των μετρικών συμβάλλουν στην επεξηγησιμότητα, στοχεύοντας σε μια μελλοντική επέκταση της εφαρμογής και προς αυτήν την κατεύθυνση.

4.7 Μεταφόρτωση δεδομένων

Εκτός από τα πρότυπα σύνολα δεδομένων τα οποία χρησιμοποιήθηκαν στο πείραμά μας και είναι διαθέσιμα για επιλογή και χρήση στην εφαρμογή, ο χρήστης μπορεί να μεταφορτώσει το δικό του σύνολο δεδομένων. Το σύνολο δεδομένων αποθηκεύεται τοπικά στην μνήμη RAM και παραμένει σε αυτή έως ότου ο χρήστης κλείσει την εφαρμογή με οποιονδήποτε τρόπο ή ανανεώσει την ιστοσελίδα της εφαρμογής. Με αυτόν τον τρόπο, διασφαλίζουμε στους χρήστες πως δεν έχουμε πρόσβαση στα δεδομένα τους, καθώς αυτά δεν αποθηκεύονται κάπου.



Εικόνα 4.12: Σελίδα μεταφόρτωσης δεδομένων.

Η εφαρμογή υποστηρίζει δύο τύπους αρχείων ".tsv" και ".csv", ενώ θα πρέπει να επισημάνουμε πως το μέγιστο μέγεθος του συνόλου δεδομένων που θα μεταφορτωθεί δεν θα πρέπει να ξεπερνάει τα 200mb, για λόγους που σχετίζονται με την ομαλή λειτουργία και απόδοση του συστήματος. Πιο συγκεκριμένα, μετριάζουμε τον κίνδυνο να υπερχειλίσει η μνήμη RAM, προκαλώντας πολυάριθμα προβλήματα. Ένας ακόμη περιορισμός που έχουμε εισάγει είναι πως το σύνολο δεδομένων θα πρέπει να περιέχει υποχρεωτικά τρεις στήλες: User ID, Item ID και Rating, ενώ προαιρετικά μπορεί να περιέχει και δύο επιπλέον στήλες, τη στήλη "Timestamp", η οποία θα περιέχει την ακριβή χρονική στιγμή που υποβλήθηκε μια αξιολόγηση για ένα προϊόν από κάποιον χρήστη και τη στήλη "Item_Name", η οποία θα περιέχει το όνομα κάθε αντικειμένου.

Κεφάλαιο 5

Πειραματική αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάζεται αρχικά η δομή των πειραμάτων που διεξήχθησαν με χρήση της εφαρμογής που υλοποιήθηκε και περιγράψαμε στο προηγούμενο κεφάλαιο και στη συνέχεια τα αποτελέσματα που προέκυψαν από αυτά. Όλα τα πειράματα διεξήχθησαν σε σύστημα με επεξεργαστή AMD Ryzen 5 3600 6-Core Processor 3.59 GHz και μνήμη RAM χωρητικότητας 16gb .

5.1 Σύνολα δεδομένων

Πίνακας 5.1: Στατιστικά στοιχεία των συνόλων δεδομένων (1/2)

Σύνολο δεδομένων	Χρήστες	Αντικείμενα	Αξιολογήσεις	Αραιότητα
Movielens1M	6.049	3.705	1.000.209	0,958
Elec_retailer10,20,30	{ 826, 414, 276 }	3.078	8.263	{ 0,996 0,993 0,990 }
Amazon (Auto)	2.928	1.835	20.473	0,996
Movielens100k	943	1.682	100.000	0,937

Πίνακας 5.2: Στατιστικά στοιχεία των συνόλων δεδομένων (2/2)

Σύνολο δεδομένων	Rating Space	UIR	RPU	RPI
Movielens1M	22.384.240	1,63	165,598	269,889
Elec_retailer10	2.542.428	0,27	9,99	2,68
Elec_retailer20	1.274.292	0,13	19,96	2,68
Elec_retailer30	849.528	0,09	29,94	2,68
Amazon (Auto)	5.372.880	1,596	6,992	11,157
Movielens100k	1.586.126	0,561	106,045	59,453

Για την διεξαγωγή των πειραμάτων χρησιμοποιήθηκαν συνολικά τέσσερα σύνολα δεδομένων. Τα δύο προέρχονται από το χώρο του ηλεκτρονικού εμπορίου και τα άλλα δύο από τον χώρο της ψυχαγωγίας.
Αναλυτικότερα, δύο σύνολα δεδομένων που χρησιμοποιήθηκαν για το πείραμα είναι το Movielens

1 Million (ML1M) και το Movielens 100K (ML100K). Το Movielens [80] είναι μια μη εμπορική υπηρεσία η οποία προσφέρει εξατομικευμένες προτάσεις ταινιών και αναπτύχθηκε από την ερευνητική ομάδα "Grouplens" στο Πανεπιστήμιο της Μινεσότα στις Η.Π.Α.. Από αυτό το σύστημα οι ερευνητές έχουν αντλήσει στοιχεία ώστε να δημιουργηθούν σύνολα δεδομένων διαφορετικών μεγεθών, εκ των οποίων επιλέχθηκαν δύο. Το πρώτο είναι το Movielens 1M το οποίο περιέχει 1.000.209 κριτικές (1 έως 5 αστέρια – ακέραιοι αριθμοί), 3952 ταινίες (εκ των οποίων έχουν λάβει αξιολογήσεις οι 3705), 18 διαφορετικών ειδών, από 6.049 διαφορετικούς χρήστες που ήταν εγγεγραμμένοι στο Movielens. Το δεύτερο είναι το Movielens 100K, το οποίο περιέχει 100.000 αξιολογήσεις (1 έως 5 αστέρια – ακέραιοι αριθμοί), 3952 ταινίες (εκ των οποίων έχουν λάβει αξιολογήσεις οι 3705), 18 διαφορετικών ειδών, από 6.049 διαφορετικούς χρήστες που ήταν εγγεγραμμένοι στο Movielens. Τόσο το Movielens 1 Million, όσο και το Movielens100K αποτελούνται από τρία αρχεία το καθένα. Το πρώτο είναι το αρχείο "movies.dat" και περιέχει πληροφορίες σχετικές με τις ταινίες. Πιο αναλυτικά, για κάθε ταινία είναι διαθέσιμες πληροφορίες όπως ένας ακέραιος αριθμός που περιγράφει το μοναδικό της χαρακτηριστικό ID, το είδος και ο τίτλος της. Το δεύτερο αρχείο είναι το "users.dat" και περιέχει, πέρα από τα IDs των χρηστών και ορισμένα δημογραφικά στοιχεία για εκείνους όπως, η ηλικία (πιο συγκεκριμένα η ηλικιακή ομάδα στην οποία ανήκουν), το επάγγελμα, το φύλο και ο ταχυδρομικός τους κώδικας. Τέλος, το αρχείο "ratings.dat" περιέχει τις αξιολογήσεις των ταινιών από τους χρήστες, δηλαδή δεδομένα που σχετίζονται με τα IDs των χρηστών, τα IDs των ταινιών, τις αξιολογήσεις των χρηστών για τις ταινίες, καθώς και τη χρονική στιγμή που υποβλήθηκε η κάθε αξιολόγηση σε Unix Epoch (ή Unix Time), δηλαδή στον αριθμό δευτερολέπτων που έχουν περάσει από τα μεσάνυχτα της 1ης Ιανουαρίου του 1970 (Ζώνη ώρας: UTC/GMT). Στο σημείο αυτό θα πρέπει να σημειωθεί πως και στα δύο σύνολα δεδομένων από το Movielens, κάθε χρήστης έχει αξιολογήσει τουλάχιστον 20 ταινίες.

Τα σύνολα δεδομένων που προέρχονται από τον χώρο του ηλεκτρονικού εμπορίου είναι το Amazon Automotive [81] (στο εξής Amazon Auto) και ένα σύνολο δεδομένων το οποίο παραχωρήθηκε από μεγάλη αλυσίδα λιανικών πωλήσεων, προϊόντων τεχνολογίας, ηλεκτρονικών και ηλεκτρικών ειδών (στο εξής θα αναφερόμαστε σε αυτό με την ονομασία "Elec_retailer"). Το Amazon Auto περιέχει 20.473 αξιολογήσεις 1.835 προϊόντων που ανήκουν στην κατηγορία Automotive της Amazon και στην οποία ανήκει οτιδήποτε σχετίζεται με το αυτοκίνητο. Από τα διαθέσιμα σύνολα δεδομένων, επιλέχθηκε το 5-core σύνολο δεδομένων, στο οποίο κάθε χρήστης έχει αξιολογήσει τουλάχιστον 5 προϊόντα. Οι στήλες που περιέχει το Amazon Auto είναι οι:

- **reviewerID:** το μοναδικό ID ενός χρήστη
- **asin:** το μοναδικό ID ενός προϊόντος
- **reviewerName:** το όνομα χρήστη (username) ενός χρήστη
- **helpful:** κατά πόσο μια αξιολόγηση ενός προϊόντος ήταν βοηθητική για άλλους χρήστες. Περιέχει τον αριθμό των χρηστών που βρήκαν την αξιολόγηση βοηθητική και τον συνολικό αριθμό των ψήφων (δηλαδή όσους την βρήκαν βοηθητική και όσους δεν την βρήκαν).
- **reviewText:** το κείμενο της αξιολόγησης
- **overall:** η αξιολόγηση του προϊόντος (σε κλίμακα από 1 έως 5 αστέρια)
- **summary:** σύνοψη της αξιολόγησης

- **unixReviewTime:** ο χρόνος της αξιολόγησης σε μορφή unix
- **reviewTime:** ο χρόνος της αξιολόγησης σε μορφή «μήνας ημέρα, έτος»

Από τις στήλες αυτές κρατήθηκαν μόνο οι στήλες “reviewerID”, “asin”, “reviewerName”, “overall” και “unixReviewTime”. Το σύνολο δεδομένων του Elec_retailer, περιείχε τις στήλες:

1. **Product ID:** το ID κάθε προϊόντος
2. **Product Name:** η ονομασία κάθε προϊόντος
3. **Review Content:** η γραπτή αξιολόγηση που έχει δοθεί σε ένα προϊόν από έναν χρήστη
4. **Review Title:** περιείχε διάφορα στοιχεία όπως (κυρίως) τον τίτλο της γραπτής αξιολόγησης που έχει δώσει ένας χρήστης σε ένα προϊόν, αν ήταν επιβεβαιωμένος αγοραστής ο χρήστης κτλ.
5. **Review Score:** αξιολογήσεις των προϊόντων σε κλίμακα από 1 έως 5
6. **Product URL:** ο υπερσύνδεσμος (URL) που οδηγούσε σε ένα προϊόν

Από τις παραπάνω στήλες κρατήθηκαν οι εξής: “Product ID”, “Product Name” και “Review Score”, καθώς οι υπόλοιπες δεν χρειάστηκαν για τα πειράματα που υλοποιήθηκαν.
Στους πίνακες 5.1 και 5.2 παρουσιάζονται συγκεντρωτικά τα σύνολα δεδομένων μαζί με τα χαρακτηριστικά τους, όπως αυτά ορίστηκαν στην υποενότητα 2.2.4 προσφέροντας πιο εύκολη σύγκρισή τους. Θα πρέπει να σημειωθεί πως ο λόγος που σε αυτούς του δύο πίνακες βλέπουμε τρία διαφορετικά σύνολα δεδομένων για το Elec_retailer είναι διότι στο συγκεκριμένο σύνολο δεδομένων απουσίαζε κάθε πληροφορία για τους χρήστες και έτσι δημιουργήσαμε 3 διαφορετικά σύνολα δεδομένων με 3 διαφορετικούς αριθμούς χρηστών τα: Elec_retailer10, Elec_retailer20 και Elec_retailer30. Περισσότερες πληροφορίες σχετικά με αυτά τα σύνολα δεδομένων και τους λόγους που μας οδήγησαν στη δημιουργία τους μπορούν να βρεθούν στην Ενότητα 5.3.

5.2 Δημιουργία συστημάτων συστάσεων

5.2.1 Αλγόριθμοι συστημάτων συστάσεων

Στο πείραμα χρησιμοποιήθηκαν πέντε διαφορετικές οικογένειες αλγορίθμων. Έγινε προσπάθεια να καλυφθούν οι πιο γνωστές οικογένειες αλγορίθμων, από τις πιο κλασικές έως τις πιο σύγχρονες προσεγγίσεις. Ακολουθεί η περιγραφή όλων των αλγορίθμων, ανά οικογένεια, μαζί με ξεχωριστή περιγραφή των υπερπαραμέτρων είτε μιας οικογένειας αλγορίθμων εφόσον όλοι αλγόριθμοί της μοιράζονται τις ίδιες υπερπαραμέτρους, είτε ανά αλγόριθμο σε περίπτωση που κάτι τέτοιο δεν συμβαίνει.

5.2.1.1 Neighborhood based αλγόριθμοι

Η πρώτη οικογένεια αλγορίθμων που εξετάσαμε είναι η neighborhood based, μια από τις παλαιότερες και πιο κλασικές προσεγγίσεις στα συστήματα συστάσεων. Από αυτή, επιλέχθηκαν δύο αλγόριθμοι ο itemKNN και ο userKNN. Οι αλγόριθμοι αυτοί περιγράφονται αναλυτικά στην υποενότητα 2.2.1. Και για τους δύο αλγορίθμους χρησιμοποιήθηκαν δύο διαφορετικές προσεγγίσεις, η κλασική και η Aiolli [82].

Υπερπαράμετροι

Αριθμός κοντινότερων γειτόνων (neighbors): ο αριθμός των γειτόνων από τους οποίες αποτελείται η γειτονιά ενός αντικειμένου ή ενός χρήστη
Μετρική ομοιότητας (similarity): μετρική η οποία υπολογίζει το βαθμό ομοιότητας δύο αντικειμένων
Υλοποίηση (implementation): η προσέγγιση που χρησιμοποιήθηκε, Aiolli ή classical

5.2.1.2 Μοντέλα λανθανόντων παραγόντων

Υπερπαράμετροι (κοινοί σε όλους τους αλγορίθμους εκτός του SLIM)

Ρυθμός μάθησης (learning rate)
στα συστήματα μάθησης σχετίζεται με το πόσο γρήγορα αυτά προσαρμόζονται στα δεδομένα που μεταβάλλονται. Υψηλός ρυθμός μάθησης σημαίνει συνήθως πως ο αλγόριθμος μαθαίνει πιο γρήγορα τα νέα δεδομένα, όμως παράλληλα ξεχνάει πιο εύκολα τα δεδομένα που έχει ήδη μάθει. Από την άλλη, όταν ορίζεται χαμηλός ρυθμός μάθησης μαθαίνει μεν πιο αργά, αλλά είναι πιο ανθεκτικός στο θόρυβο που ενδεχομένως να περιέχουν τα δεδομένα. Η υπερπαράμετρος αυτή έχει πολύ μεγάλη σημασία στα συστήματα παραγοντοποίησης μητρώου, καθώς ρυθμίζει την ακρίβεια πρόβλεψης και τον ρυθμό σύγκλισης και συνακόλουθα το υπολογιστικό κόστος και την πολυπλοκότητα του αλγορίθμου. Έρευνες έχουν δείξει πως όσο πιο μικρή η τιμή του ρυθμού μάθησης, τόσο καλύτερη η απόδοση του συστήματος. Ένα ερώτημα που τίθεται είναι τι συμβαίνει με την μεροληψία των αλγορίθμων.
Αριθμός λανθανόντων παραγόντων (factors)
ο αριθμός των κρυφών χαρακτηριστικών των αντικειμένων, καθορίζει επίσης το μέγεθος των Embeddings. Συμβολίζονται με f.
Μέγεθος πακέτου δεδομένων (batch size)
καθορίζει τον αριθμό των δειγμάτων που θα διαδοθούν μέσω του δικτύου. Έστω b το μέγεθος του πακέτου δεδομένων, σε κάθε εποχή ο αλγόριθμος χωρίζει το σύνολο δεδομένων σε πακέτα δεδομένων, μεγέθους b
Παράμετρος κανονικοποίησης λ (reg)
μέσω αυτής της παραμέτρου αποφεύγεται η υπερεκπαίδευση του μοντέλου.

Matrix Factorization (MF)
Ο πρώτος αλγόριθμος λανθανόντων παραγόντων που επιλέχθηκε είναι ο κλασικός αλγόριθμος παραγοντοποίησης μητρώου [18] που περιγράφηκε στην υποενότητα 2.2.1, ο οποίος αποτέλεσε την βάση για όλους τους αλγορίθμους παραγοντοποίησης μητρώου που αναπτύχθηκαν τα επόμενα χρόνια. Ο MF υλοποιήθηκε από το Elliot, στο Tensorflow Keras και αποτελείται από δύο embeddings, ένα για τους χρήστες και ένα για τα αντικείμενα. Η συνάρτηση βελτιστοποίησης που χρησιμοποιήθηκε για την εκμάθηση των παραμέτρων είναι η Adam (adaptive moment estimation) [83].
Ο αλγόριθμος Adam βασίζεται στην μέθοδο στοχαστικής καθόδου κλίσης (stochastic gradient descent) και προσαρμόζει την τιμή του ρυθμού μάθησης στις παραμέτρους, εκτελώντας μικρότερες ενημερώσεις, δηλαδή θέτοντας μικρότερες τιμές ρυθμού μάθησης για τις παραμέτρους που σχετίζονται με χαρακτηριστικά που εμφανίζονται συχνά, και μεγαλύτερες ενημερώσεις δηλαδή θέτει

μεγαλύτερες τιμές στον ρυθμό μάθησης για τις παραμέτρους που σχετίζονται με χαρακτηριστικά που εμφανίζονται σπάνια. Ο Adam χρησιμοποιεί εκτιμήσεις της πρώτης και της δεύτερης ροπής (moments) του gradient έχοντας ως στόχο την προσαρμογή του ρυθμού μάθησης για κάθε βάρος που υπάρχει στο νευρωνικό δίκτυο. Με τον όρο n-οστή ροπή μιας τυχαίας μεταβλητής ορίζουμε την αναμενόμενη τιμή (expected value) αυτής της μεταβλητής υψωμένη στην δύναμη n: $m_n = E\left[X^n\right]$. Η εκτίμηση της πρώτης ροπής (μέση τιμή) m_t και της δεύτερης ροπής (uncentered variance δηλαδή δεν αφαιρούμε την μέση τιμή κατά τον υπολογισμό της διασποράς) v_t δίνονται από τους τύπους:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \ , \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \end{aligned} \tag{5.1}$$

οι υπερπαράμετροι β_1 και β_2 ελέγχουν τους ρυθμούς εκθετικής μείωσης των εκτιμήσεων των ροπών, g_t είναι η κλίση της objective συνάρτησης f και $g_t^2 = g_t \odot g_t$. Η ανανέωση των τιμών των βαρών γίνεται σύμφωνα με τη σχέση:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{5.2}$$

όπου η είναι ο ρυθμός μάθησης, $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ και $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$ είναι η εκτίμηση της πρώτης και της δεύτερης ροπής αντίστοιχα, αν διορθώσουμε την μεροληψία που εισάγεται κατά την αρχικοποίησή τους.

SVD++

Υπερπαράμετροι

reg_w: όρος κανονικοποίησης για τα μητρώα παραγόντων χρηστών και αντικειμένων (ο όρος λ_2 στην εξίσωση (5.4)).
reg_b: όρος κανονικοποίησης για τα b_u και b_i (ο όρος λ_1 στην εξίσωση (5.4)).

Ο αλγόριθμος αυτός [84] αποτέλεσε μια από τις πρώτες προσπάθειες βελτίωσης του MF. Πιο συγκεκριμένα, ο αλγόριθμος Funk SVD λειτουργεί μόνο για explicit ratings ενώ ο SVD++ λειτουργεί και για implicit ratings βελτιώνοντας έτσι την ακρίβεια των προβλέψεων. Σε αυτό το πλαίσιο προσθέτουμε ένα νέο σύνολο το οποίο περιέχει λανθάνοντες παράγοντες αντικειμένου (item factors) σχετικά με implicit αξιολογήσεις και συσχετίζει ένα αντικείμενο i με ένα διάνυσμα παράγοντα (factor vector) y_i. Επιπρόσθετα, αυτός ο αλγόριθμος λαμβάνει υπόψη του και τη μεροληψία του χρήστη b_u και των αντικειμένων b_i (user and item bias). Στην πραγματικότητα οι δύο αυτές παράμετροι δείχνουν την απόκλιση του χρήστη u και του αντικειμένου i που έχουν παρατηρηθεί, από τον μέσο όρο όλων των αξιολογήσεων μ. Η πρόβλεψη της αξιολόγησης του χρήστη u για το αντικείμενο i που δεν έχει παρατηρηθεί (unobserved), δίνεται από τον παρακάτω τύπο:

$$\widehat{r_{ui}} = \mu + b_u + b_i + q_i^T \left(p_u + \left|N_u\right|^{-\frac{1}{2}} \sum_{j \in N_u} y_j \right) \tag{5.3}$$

Όπου N_u είναι το σύνολο το οποίο περιέχει όλα τα αντικείμενα για τα οποία ο χρήστης u έχει δώσει μια έμμεση προτίμηση (implicit preference), ενώ τα διανύσματα p_u και q_i είναι τα ίδια με αυτά που περιγράψαμε στην παραγοντοποίηση μητρώου. Ο χρήστης μοντελοποιείται μέσω του τύπου

$p_u + |N_u|^{-\frac{1}{2}} \sum_{j \in N_u} y_j$ και η έννοια του implicit feedback μέσω του αθροίσματος $|N_u|^{-\frac{1}{2}} \sum_{j \in N_u} y_j$. Το μεγαλύτερο μειονέκτημα αυτού του αλγορίθμου είναι το πρόβλημα cold-start, ένα πολύ κοινό πρόβλημα στα συστήματα συστάσεων, το οποίο κάνει την εμφάνισή του κατά την εισαγωγή ενός νέου χρήστη ή ενός νέου αντικειμένου. Αυτό το πρόβλημα, το περιγράψαμε αναλυτικά στην υποενότητα 2.2.4.
Ο αλγόριθμος αυτός έχει υλοποιηθεί στο Elliot με χρήση του Tensorflow framework και αποτελείται από 5 embeddings:

1. **Μητρώο παραγόντων αντικειμένων (Item factor matrix):** όπου κάθε στήλη αναπαριστά ένα αντικείμενο, το οποίο περιγράφεται από λανθάνοντες παράγοντες, με άλλα λόγια υπολογίζει το q_i.
2. **Μητρώο παραγόντων χρηστών (User factor matrix):** όπου κάθε στήλη αναπαριστά έναν χρήστη, ο οποίος περιγράφεται από λανθάνοντες παράγοντες, δηλαδή υπολογίζει το p_u.
3. **Item bias:** υπολογίζει το b_i
4. **User bias:** υπολογίζει το b_u.
5. **Παράγοντες αντικειμένων για implicit αξιολογήσεις:** αντιστοιχίζουν κάθε αντικείμενο i με ένα διάνυσμα παράγοντα y_i, μέσω αυτών των παραγόντων αντικειμένων μπορούμε να χαρακτηρίσουμε τους χρήστες με βάση τα αντικείμενα που έχουν αξιολογήσει.

Και σε αυτόν τον αλγόριθμο η συνάρτηση βελτιστοποίησης που χρησιμοποιείται είναι η Adam. Μέσω της Adam γίνεται η ελαχιστοποίηση της συνάρτησης κόστους (2.3) και με αυτόν τον τρόπο καθορίζονται οι παράμετροι της εξίσωσης. Αυτό γίνεται μέσω μιας επαναληπτικής διαδικασίας με την οποία διαπερνάμε όλες τις αξιολογήσεις που βρίσκονται στο σύνολο $K = \{(u,i) \mid r_{ui} \text{ γνωστό}\}$:

$$
\begin{aligned}
b_u &\leftarrow b_u + \gamma(e_{ui} - \lambda_1 b_u) \\
b_i &\leftarrow b_i + \gamma(e_{ui} - \lambda_1 b_i) \\
p_u &\leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda_2 p_u) \\
q_i &\leftarrow q_i + \gamma\left(e_{ui} \cdot \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j\right) - \lambda_2 \cdot q_i\right) \\
&\forall j \in N(u) : \\
y_j &\leftarrow y_j + \gamma\left(e_{ui} \cdot |N(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_2 \cdot y_j\right)
\end{aligned}
\tag{5.4}
$$

Bayesian Personalized Ranking with Matrix Factorization (BPRMF) [85]

Υπερπαράμετροι

bias_regularization: παράμετρος κανονικοποίησης για τους παράγοντες αντικειμένων
user_regularization: παράμετρος κανονικοποίησης για τους παράγοντες χρηστών
positive_item_regularization: παράμετρος κανονικοποίησης για τους παράγοντες θετικών αντικειμένων
negative_item_regularization: παράμετρος κανονικοποίησης για τους παράγοντες αρνητικών αντικειμένων

Αλγόριθμος παραγοντοποίησης μητρώου ο οποίος χρησιμοποιεί τη μέθοδο στοχαστικής καθόδου κλίσης (stochastic gradient descent), όπου η αντικειμενική συνάρτηση (objective function) είναι η Εξατομικευμένη Μπεϋζιανή Κατάταξη (Bayesian Personalized Ranking - BPR) για pair-wise προτιμήσεις ανάμεσα σε αντικείμενα που έχουν παρατηρηθεί και σε αντικείμενα που δεν έχουν παρατηρηθεί. Σύμφωνα με αυτή την προσέγγιση, αντί για ένα αντικείμενο για το οποίο προβλέπουμε τον βαθμό προτίμησης του χρήστη προς αυτό, το σύνολο εκπαίδευσης αποτελείται από ζεύγη αντικειμένων. Η δειγματοληψία εδώ γίνεται ομοιόμορφα με τυχαίο τρόπο, αυξάνοντας αρκετά τις πιθανότητες να επιλεγούν δημοφιλή αντικείμενα. Εξαιτίας αυτού, στον BPRMF, το φαινόμενο του popularity bias είναι πολύ πιο εμφανές από όσους αλγορίθμους έχουμε δει έως τώρα. Ας δούμε πιο αναλυτικά τη δομή αυτού του αλγορίθμου.

Έστω Θ η παράμετρος του μοντέλου που καθορίζει την εξατομικευμένη κατάταξη (personalized ranking). Ο στόχος του BPR είναι η μεγιστοποίηση της ακόλουθης εκ των υστέρων πιθανότητας (posterior probability) για την εύρεση της ορθής εξατομικευμένης κατάταξης για κάθε αντικείμενο i το οποίο ανήκει στο σύνολο I που περιέχει όλα αντικείμενα:

$$p(\Theta|i >_u j) \propto p(i >_u j|\Theta)p(\Theta) \tag{5.5}$$

Όπως και σε όλες τις Μπεϋζιανές προσεγγίσεις υπάρχει μια συνάρτηση πιθανοφάνειας, εδώ είναι η $p(i >_u j|\Theta)$ η οποία υπολογίζει την ατομική πιθανότητα (individual probability) ένας χρήστης να προτιμάει το αντικείμενο i έναντι του αντικειμένου j. Αυτή η πιθανότητα εξασφαλίζει την εξατομικευμένη κατάταξη και υπολογίζεται από τη λογιστική σιγμοειδή συνάρτηση σ σύμφωνα με τον τύπο:

$$p(i >_u j|\Theta) := \sigma\big(\hat{r}_{uij}(\Theta)\big), \tag{5.6}$$

$$\text{όπου } \sigma(x) = \frac{1}{1+e^{-x}} \tag{5.7}$$

όπου το $\hat{r}_{uij}(\Theta)$ συμβολίζει την σχέση ανάμεσα στον χρήστη u, στο αντικείμενο i και στο αντικείμενο j.

$$\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj} \tag{5.8}$$

Επιπροσθέτως, υπάρχει η εκ των προτέρων πιθανότητα (prior probability) $p(\Theta)$ που είναι μια κανονική κατανομή με μηδενική μέση τιμή και μητρώο διασποράς-συνδιασποράς $\Sigma(\Theta)$. Στη συνέχεια, θέτουμε $\Sigma(\Theta) = \lambda_\Theta I$. Σε αυτό το σημείο μπορούμε να δούμε τον ορισμό του κριτηρίου ελαχιστοποίησης για την εξατομικευμένη κατάταξη.

$$\begin{aligned}
\text{BPR-Opt} &= ln\big(p(\Theta|i >_u)\big) \\
&= ln\big(p(i >_u j|\Theta)p(\Theta)\big) \\
&= ln\Big(\prod_{u,i,j} p(i >_u j|\Theta)p(\Theta)\Big) \\
&= \sum_{u,i,j} ln\sigma\big(\hat{x}_{uij}\big) + lnp(\Theta) \\
&= \sum_{u,i,j} ln\sigma\big(\hat{x}_{uij}\big) - \lambda_\Theta \|\Theta\|^2
\end{aligned} \tag{5.9}$$

Weighted Regularized Matrix Factorization (WRMF)

Υπερπαράμετροι

alpha: ο βαθμός αύξησης του confidence στην εξίσωση (5.12)
reg: παράμετρος κανονικοποίησης

Ο δεύτερος αλγόριθμος για implicit δεδομένα είναι ο WRMF, ο οποίος παρουσιάζεται στο [86] και βασίζεται στην τεχνική της παραγοντοποίησης μητρώου (matrix factorization). Είδαμε σε προηγούμενη ενότητα ότι μια προσέγγιση που χρησιμοποιείται στην παραγοντοποίηση μητρώου είναι η δημιουργία μιας συνάρτησης κόστους J, μέσω της ελαχιστοποίησης του τετραγώνου της διαφοράς των προβλέψεων του αλγορίθμου και των πραγματικών αξιολογήσεων του συνόλου δεδομένων:

$$J = \min_{x*,y*} \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\|x_u\|^2 + \|y_i\|^2 \right) \tag{5.10}$$

Στον αλγόριθμο WRMF η συνάρτηση κόστους τροποποιείται ελαφρώς:

$$J_w = \min_{x*,y*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) \tag{5.11}$$

Η πρώτη διαφορά που παρατηρούμε σε σχέση με την συνάρτηση J είναι πως το μητρώο αξιολογήσεων r_{ui} έχει αντικατασταθεί από το μητρώο προτιμήσεων p_{ui}. Αν ένας χρήστης u έχει αλληλεπιδράσει με οποιονδήποτε τρόπο με το αντικείμενο i, τότε $p_{ui} = 1$, αλλιώς $p_{ui} = 0$. Επίσης, c_{ui} είναι η σιγουριά (confidence), δηλαδή πόσο σίγουροι είμαστε ότι ο χρήστης u ενδιαφέρεται πραγματικά (p_{ui}) για το αντικείμενο i.

$$c_{ui} = 1 + \alpha r_{ui} \tag{5.12}$$

όπου α είναι ο βαθμός αύξησης του confidence. Ωστόσο, για μεγάλα σύνολα δεδομένων το κόστος υπολογισμού αυτής της συνάρτησης μπορεί να γίνει πολύ μεγάλο. Προκειμένου να ξεπεραστεί αυτό το πρόβλημα, χρησιμοποιήθηκε η διαδικασία βελτιστοποίησης Εναλλασσόμενων Ελαχίστων Τετραγώνων (Alternating Least Squares - ALS), μέσω της οποίας γίνεται ο εναλλάξ επαναϋπολογισμός, με τη στοχαστική κάθοδο κλίσης, των παραγόντων χρήστη – κρατώντας το μητρώο αντικειμένου σταθερό - και των παραγόντων αντικειμένου – κρατώντας το μητρώο χρήστη σταθερό -. Από την παραπάνω διαδικασία προκύπτει η εξίσωση που ελαχιστοποιεί την συνάρτηση J_w για το $x_u : x_u = (Y^T C^u Y + \lambda \mathrm{I})^{-1} Y^T C^u p(u)$ και για το $y_i : y_i = (X^T C^i X + \lambda \mathrm{I})^{-1} X^T C^i p(i)$. Ουσιαστικά, τα δύο αυτά διανύσματα προσπαθούν να αντιστοιχήσουν χρήστες και αντικείμενα σε έναν κοινό χώρο λανθανόντων παραγόντων όπου μπορούν να συγκριθούν απευθείας.

Sparse Linear Method (SLIM)

Υπερπαράμετροι

alpha: Σταθερά η οποία πολλαπλασιάζει τους όρους ποινής. Για alpha = 0 είναι ισοδύναμο με μια κλασική εξίσωση ελαχίστων τετραγώνων. Είναι ο όρος α στην εξίσωση (5.15)
l1_ratio: Ο όρος ρ της εξίσωσης (5.15), όπου ισχύει: $0 \leq$ l1_ratio ≤ 1. Για l1_ratio $= 0$ η ποινή είναι η L_2. Για l1_ratio $= 1$ η ποινή είναι η L_1. Ενώ για $0 <$ l1_ratio < 1, η ποινή που εισάγεται είναι ένας συνδυασμός των L_1 και L_2.

Η μέθοδος SLIM [87] βασίζεται στην παραγοντοποίηση μητρώου και είναι μια item-item μέθοδος. Επίσης, ενδείκνυται περισσότερο για implicit δεδομένα και έχει αρκετά καλή απόδοση. Στη SLIM η βαθμολογία ενός αντικειμένου με το οποίο δεν έχει αλληλεπιδράσει ένας χρήστης u_i, υπολογίζεται από το αραιό άθροισμα των αντικειμένων με τα οποία έχει αλληλεπιδράσει: $\tilde{a}_{ij} = a_i^T w_j$, όπου w_j είναι ένα αραιό n-διάστατο μητρώο-στήλη που περιέχει τους συντελεστές άθροισης και a_i^T ένα διάνυσμα που περιέχει το ιστορικό αλληλεπίδρασης του χρήστη u_i με όλα τα αντικείμενα. Το μοντέλο που χρησιμοποιεί αυτή η μέθοδος περιγράφεται από τον τύπο: A'= AW, όπου A είναι το μητρώο χρηστών-αντικειμένων και περιέχει τις αξιολογήσεις των αντικειμένων από τους χρήστες και W είναι ένα nxn αραιό μητρώο, του οποίου κάθε γραμμή αναπαριστά τις αξιολογήσεις όλων των αντικειμένων από τον χρήστη u_i. Ο υπολογισμός του W γίνεται στην ουσία επιλύοντας ένα πρόβλημα παλινδρόμησης, καθιστώντας τον SLIM έναν από τους πρώτους αλγορίθμους για συστήματα συστάσεων που χρησιμοποίησε παλινδρόμηση για τον υπολογισμό του W. Η εκμάθηση του αραιού, μη-αρνητικού, αυτού μητρώου γίνεται μέσω της νόρμας 1 (l1-norm) του W: $||W||_1 = \sum_{i-1}^{n} \sum_{j-1}^{n} |w_{ij}|$. Για την αποφυγή της υπερεκπαίδευσης του μοντέλου, χρησιμοποιείται η νόρμα 2 (l2-norm) ή αλλιώς νόρμα Frobenius (συμβολίζεται με F στον τύπο), η οποία μετατρέπει το πρόβλημα βελτιστοποίησης σε ένα Elastic net πρόβλημα. Με τον όρο Elastic net αναφερόμαστε στη μέθοδο κανονικοποιημένης παλινδρόμησης που ορίζεται στο [88] και συνδυάζει γραμμικά τις ποινές (penalties) L_1 και L_2 των μεθόδων Lasso και Ridge, αντίστοιχα. Και οι δύο αυτές μέθοδοι, χρησιμοποιούνται για την επίλυση γραμμικών προβλημάτων της μορφής $\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p$, όπου η τιμή $\hat{y}(w, x)$ είναι γραμμικός συνδυασμός των χαρακτηριστικών $w = (w_1, ..., w_p)$. Στη μέθοδο Lasso υπάρχει μια ποινή στο μέγεθος των συντελεστών

$$\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha ||w||_1 \tag{5.13}$$

Στη μέθοδο Ridge τίθεται περιορισμός στους συντελεστές, καθώς υπάρχει μια ποινή στο τετράγωνο του μεγέθους των συντελεστών. Ο βαθμός της ποινής καθορίζεται από την παράμετρο α.

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2 \tag{5.14}$$

Ο συνδυασμός της χρήσης των L_1 και L_2 επιτρέπει την εκμάθηση ενός αραιού μοντέλου όπου μικρός αριθμός των βαρών είναι μη-μηδενικά, όπως συμβαίνει στη μέθοδο Lasso, διατηρώντας όμως παράλληλα τις ιδιότητες που προσφέρει η κανονικοποίηση που χρησιμοποιείται στη μέθοδο Ridge.

$$\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha\rho ||w||_1 + \frac{\alpha(1-\rho)}{2} ||w||_2^2 \tag{5.15}$$

Έτσι, η εκμάθηση του μητρώου W στο Slim γίνεται έχοντας ως βάση την εξίσωση (5.15) σύμφωνα

με τον τύπο:

$$\begin{aligned} \underset{W}{\textbf{minimize}} \quad & \frac{1}{2} * ||A - AW||_F^2 + \frac{\beta}{2} * ||W||_F^2 + \lambda * ||W||_1 \\ \text{Subject to} \quad & W \geq 0 \\ & \text{Diag}(W) = 0 \end{aligned} \tag{5.16}$$

Ο πρώτος όρος $\frac{1}{2} * ||A - AW||_F^2$ μετρά πόσο καλά ταιριάζει το γραμμικό μοντέλο στα δεδομένα εκπαίδευσης, ενώ οι σταθερές β και λ είναι παράμετροι κανονικοποίησης. Στο Elliot ο αλγόριθμος αυτός υλοποιήθηκε μέσω της βιβλιοθήκης scikit-learn.

5.2.1.3 Αλγόριθμος που έχει ως βάση τα γραφήματα

Neural Graph Collaborative Filtering (NGCF)

Ο αλγόριθμος αυτός [89] είναι ένα από τα πρώτα Graph Neural Network (GNN) που προτάθηκαν για τα συστήματα συστάσεων και παρέχει έναν ακόμη τρόπο για την υλοποίηση της συνεργατικής διήθησης. Αξιοποιεί την έννοια της συνεκτικότητας υψηλής τάξης (high-order connectivity) των αλληλεπιδράσεων χρηστών-αντικειμένων. Η έννοια αυτή, υποδηλώνει τη διαδρομή μέσω της οποίας μπορούμε να πάμε στην κορυφή u_1 από οποιαδήποτε κορυφή με μήκος διαδρομής l μεγαλύτερο από 1.

Ο NGCF αποτελείται από τρία κύρια στοιχεία:

1. ένα *embedding επίπεδο*, το οποίο περιέχει τα embeddings των χρηστών και των αντικειμένων τα οποία συνενώνονται σε ένα πίνακα αναζήτησης embedding:

$$\mathrm{E} = [\underbrace{e_{u_1}, ..., e_{u_N}}_{\substack{\text{embeddings} \\ \text{χρηστών}}}, \underbrace{e_{i_1}, ..., e_{i_M}}_{\substack{\text{embeddings} \\ \text{αντικειμένων}}}] \tag{5.17}$$

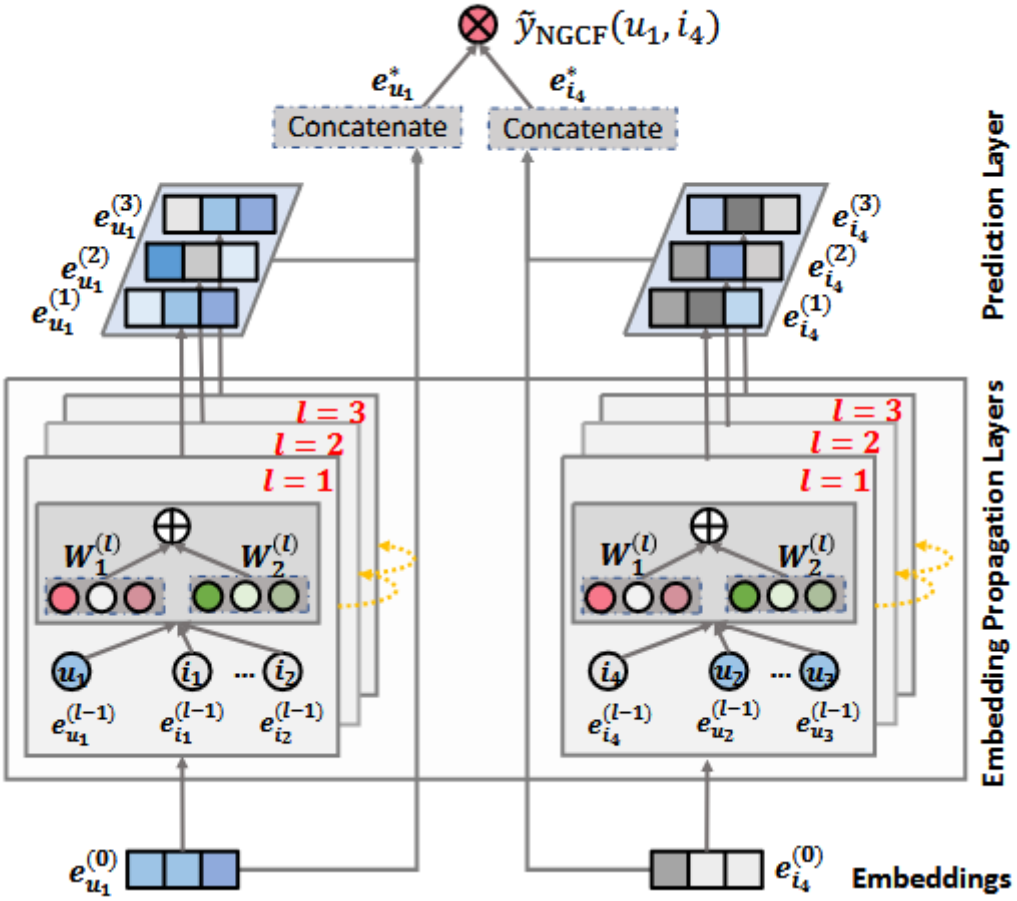
   Στη συνέχεια, ο πίνακας με τα embeddings διαδίδεται μέσω του γράφου χρήστη-αντικειμένου (μια μορφή του οποίου είδαμε στην Εικόνα 2.6), δηλαδή μέσω των επιπέδων διάδοσης embedding

2. *πολλαπλά επίπεδα διάδοσης embedding (embedding propagation layers)* τα οποία βελτιώνουν τα embeddings ενθέτοντας σε αυτά σχέσεις συνεκτικότητας υψηλής τάξης. Μπορούν να διαδοθούν μέσω αυτών τόσο αλληλεπιδράσεις χαρακτηριστικών υψηλής τάξης, όσο και χαμηλής. Στη διάδοση χαμηλής τάξης υπάρχουν δύο κύριες διαδικασίες: η *δημιουργία των μηνυμάτων* και η *άθροιση των μηνυμάτων*. Στη δημιουργία των μηνυμάτων για κάθε διασυνδεμένο ζεύγος χρήστη-αντικειμένου (u, i) το μήνυμα από το i στο u ορίζεται ως

$$m_{u \leftarrow i} = \frac{1}{\sqrt{\|N_u\| \|N_i\|}} \Big(W_1 e_i + W_2 \left(e_i \odot e_u \right) \Big) \tag{5.18}$$

   Όπου N_u και N_i οι γείτονες του χρήστη u και του αντικειμένου i, κατά την πρώτη μεταβίβαση μηνυμάτων και W_1, W_2 τα μητρώα που περιέχουν τα βάρη. Η διαδικασία της άθροισης των μηνυμάτων που έχουν δημιουργηθεί και προέρχονται από την γειτονιά ενός χρήστη u για τον επαναπροσδιορισμό της αναπαράστασης του u γίνεται σύμφωνα με την εξίσωση:

$$e_u^{(1)} = \text{LeakyReLU}\Big(m_{u \leftarrow u} + \sum_{i \in N_u} m_{u \leftarrow i} \Big) \tag{5.19}$$



Εικόνα 5.1: Αρχιτεκτονική του αλγορίθμου NGCF [Πηγή: `https://arxiv.org/pdf/1905.08108.pdf`]

όπου LeakyReLU [90] είναι η συνάρτηση ενεργοποίησης Ανορθωμένη Γραμμική Μονάδα (Rectified Linear Unit)

$$LeakyRelU(x) = \begin{cases} 0.01x, & \text{αν } x < 0 \\ x, & \text{αλλιώς} \end{cases} \tag{5.20}$$

Η διάδοση περιγράφεται σε μορφή μητρώου από τον τύπο:

$$E^{(l)} = \text{LeakyReLU}\Big((\mathscr{L} + I)\, E^{(l-1)} W_1^{(l)} + \mathscr{L} E^{(l-1)} \odot E^{(l-1)} W_2^{(l)} \Big) \tag{5.21}$$

όπου I είναι το ταυτοτικό μητρώο [1], W το μητρώο βαρών που χρησιμοποιείται κατά την εκπαίδευση για να εξαχθούν χρήσιμες πληροφορίες για την διάδοση, $E^{(l)}$ είναι ο πίνακας των embeddings μετά από l βήματα της διάδοσης και $E^{(0)}$ ο αρχικός πίνακας με τα embeddings, D είναι το διαγώνιο μητρώο, A το μητρώο γειτνίασης (adjacency matrix) και $\mathscr{L}$ είναι το Λαπλασιανό μητρώο για τον γράφο χρηστών - αντικειμένων, που περιγράφεται από τη σχέση:

$$\mathscr{L} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \; A = \begin{bmatrix} 0 & R \\ R^T & 0 \end{bmatrix}$$

[1] Ταυτοτικό μητρώο: ονομάζεται το μητρώο που περιέχει άσσους στην κύρια διαγώνιο και μηδενικά στις υπόλοιπες θέσεις

όπου $R \in \mathbb{R}^{N \times M}$ το μητρώο αλληλεπίδρασης χρηστών-αντικειμένων

3. ένα *επίπεδο πρόβλεψης (prediction layer)* το οποίο αθροίζει τα νέα embeddings, για κάθε χρήστη, τα οποία προέρχονται από διαφορετικά επίπεδα διάδοσης, ώστε να προκύψει το τελικό embedding $e_u^* = e_u^{(0)} \| \cdots \| e_u^{(L)}$, $e_i^* = e_i^{(0)} \| \cdots \| e_i^{(L)}$, με $\|$ συμβολίζεται η πράξη της άθροισης. Τέλος, εξάγει τη βαθμολογία συνάφειας ενός ζεύγους χρήστη-αντικειμένου, δηλαδή την προτίμηση ενός χρήστη u για ένα αντικείμενο i, υπολογίζοντας το εσωτερικό γινόμενο του τελικού embedding με το αντικείμενο.

$$\hat{y}_{\text{NGCF}}(u, i) = e_u^{*\top} e_i^* \tag{5.22}$$

Η εκμάθηση των παραμέτρων του μοντέλου, γίνεται μέσω της βελτιστοποίησης της συνάρτησης απώλειας BPR κατά ζεύγη (pairwise BPR loss), που περιγράφεται από την objective συνάρτηση:

$$Loss = \sum_{(u,i,j) \in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|_2^2 \tag{5.23}$$

με Θ συμβολίζονται όλες οι trainable παράμετροι [2] του μοντέλου, $\Theta = \left\{ E, \left\{ W_1^{(l)}, W_2^{(l)} \right\}_{l=1}^{L} \right\}$ και σ είναι η σιγμοειδής συνάρτηση

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5.24}$$

Οι υπόλοιπες υπερπαράμετροι για τις οποίες υπάρχει και η δυνατότητα ρύθμισης στο Elliot, περιγράφονται στο κάτωθι πλαίσιο.

Υπερπαράμετροι

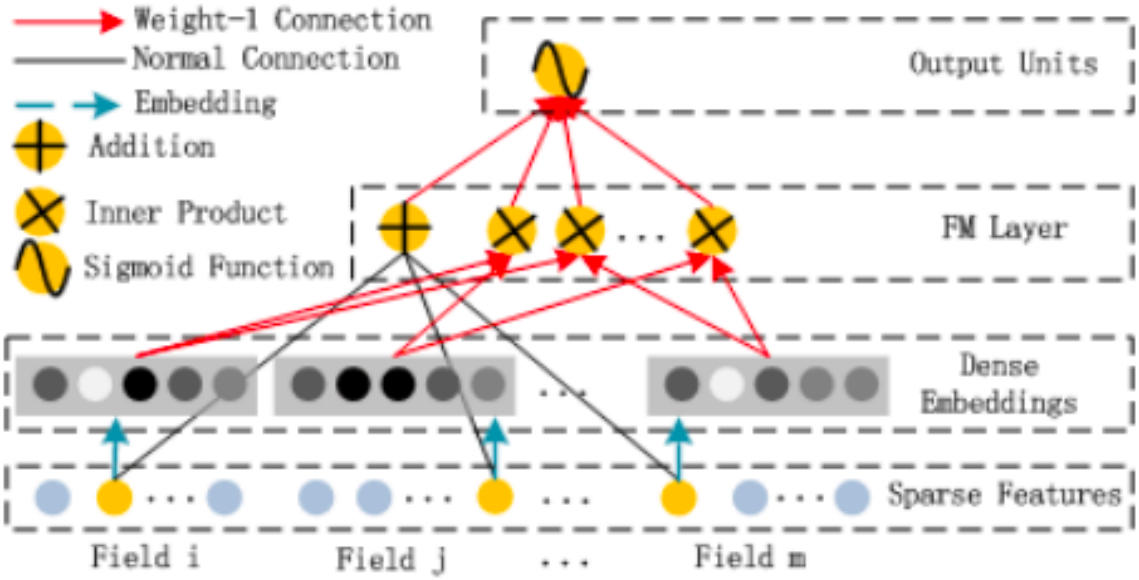
λ: στο Elliot αναφέρεται ως l_w και ελέγχει την κανονικοποίηση που εφαρμόζει η L2 στην εξίσωση (5.23), ώστε να αποφευχθεί η υπερεκπαίδευση του μοντέλου.
message_dropout: υποδεικνύει το ποσοστό των εξερχόμενων μηνυμάτων που θα απορριφθούν με τυχαίο τρόπο.
node_dropout: υποδεικνύει το ποσοστό της απόρριψης κόμβων, για την παρεμπόδιση ενός συγκεκριμένου κόμβου, ο οποίος έχει επιλεγεί με τυχαίο τρόπο και ταυτόχρονα απορρίπτει όλα τα εξερχόμενα μηνύματα.
weight_size: αριθμός μονάδων για κάθε επίπεδο διάδοσης embedding.
n_fold: αριθμός πτυχών (folds) για την διάσπαση του μητρώου γειτνίασης σε υπο-μητρώα, με σκοπό τον ευκολότερο υπολογισμό του.

[2] Με τον όρο Trainable παράμετροι αναφερόμαστε στον αριθμό των βαρών που δεν ενημερώνονται κατά την εκπαίδευση του μοντέλου στον αλγόριθμο οπισθοδιάδοσης (backpropagation).

5.2.1.4 Αλγόριθμος που έχει ως βάση τα τεχνητά νευρωνικά δίκτυα

DeepFM

Ο DeepFM [91] συνδυάζει τα βαθιά νευρωνικά δίκτυα (Deep Neural Networks - DNN) με την τεχνική των Factorization machines. Πιο συγκεκριμένα, αποτελείται από δύο βασικά στοιχεία:



Εικόνα 5.2: Αρχιτεκτονική του αλγορίθμου DeepFM [Πηγή: `https://arxiv.org/pdf/1703.04247.pdf`]

1. **Στοιχείο FM του DeepFM:** πρόκειται για την τεχνική των Factorization Machines (FM), η οποία παρουσιάστηκε για πρώτη φορά στο [92] και συνδυάζει την τεχνική της παραγοντοποίησης μητρώου με την παλινδρόμηση. Στην ενότητα 2 περιγράψαμε τους λανθάνοντες παράγοντες και την χρήση τους στα συστήματα συστάσεων. Σε αυτή την τεχνική, οι αλληλεπιδράσεις μεταξύ χρηστών-αντικειμένων αναπαριστώνται με πλειάδες (tuples) διανυσμάτων (λανθανόντων) χαρακτηριστικών (feature vectors) των οποίων οι τιμές είναι πραγματικοί αριθμοί. Έστω $\mathbf{V} \in \mathbb{R}^{d \times k}$ τα feature embeddings, τότε $\mathbf{v}_i$ είναι η γραμμή i του V, ενώ με $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ αναπαριστάται το εσωτερικό γινόμενο ανάμεσα στις μεταβλητές i και j, το οποίο μοντελοποιεί την αλληλεπίδραση κατά ζεύγη (pairwise) μεταξύ αυτών των δύο μεταβλητών. Επίσης, w_0 είναι το global bias, w_i η δύναμη (strength) της μεταβλητής i ενώ με k αναπαριστούμε τον αριθμό των διαστάσεων των λανθανόντων παραγόντων.

$$\hat{y}_{FM} = \underbrace{\mathbf{w}_0 + \sum_{i=1}^{d} \mathbf{w}_i x_i}_{\text{παλινδρόμηση}} + \underbrace{\sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j}_{\text{παραγοντοποίηση μητρώου}} \tag{5.25}$$

   Το στοιχείο αυτό είναι υπεύθυνο για την κάλυψη γραμμικών (1ης τάξης) και pairwise (2ης τάξης) αλληλεπιδράσεων.

2. **Στοιχείο Deep (learning) του DeepFM:** είναι ένα feed-forward νευρωνικό δίκτυο [93] και πιο συγκεκριμένα ένα πολυεπίπεδο Perceptron (Multilayer Perceptron - MLP), το οποίο χρησιμοποιείται για την εκμάθηση αλληλεπιδράσεων χαρακτηριστικών υψηλής τάξης. Στα

feedforward νευρωνικά δίκτυα η πληροφορία διαχέεται μόνο προς μια κατεύθυνση: προς το επόμενο επίπεδο, δηλαδή προς τα εμπρός (forward), χωρίς να δημιουργούνται κύκλοι. Αναλυτικότερα από τα επίπεδα εισόδου μεταβαίνει απευθείας στα κρυφά επίπεδα και από εκεί στα επίπεδα εξόδου. Το στοιχείο Deep του DeepFM, όπως και όλα τα MLPs, αποτελείται από τα εξής επίπεδα:

- **Επίπεδο εισόδου (input layer):** αποτελείται από δύο υποεπίπεδα, το πρώτο λαμβάνει ως είσοδο τα ανεπεξέργαστα αραιά χαρακτηριστικά και το δεύτερο embedding επίπεδο μετατρέπει αυτά τα χαρακτηριστικά σε πυκνά embeddings, χαμηλών διαστάσεων.
- **Κρυφό επίπεδο (hidden layer):** λαμβάνει ως είσοδο τα embeddings που παρήχθησαν από το embedding επίπεδο, του επιπέδου εισόδου και παράγει ως έξοδο ένα πυκνό διάνυσμα χαρακτηριστικών, του οποίου οι τιμές ανήκουν στο σύνολο των πραγματικών αριθμών.
- **Επίπεδο εξόδου (output layer):** λαμβάνει ως είσοδο το διάνυσμα που παρήγαγε το προηγούμενο επίπεδο και «τροφοδοτεί» με αυτό τη σιγμοειδή συνάρτηση:

$$\hat{y}_{DNN} = sigmoid\left(W^{|H|+1} * a^{|H|} + b^{|H|+1}\right) \tag{5.26}$$

όπου Η ο αριθμός των κρυφών επιπέδων.

Η πρόβλεψη που δίνεται από το τελευταίο επίπεδο του νευρωνικού δικτύου του αλγορίθμου υπολογίζεται από τον τύπο:

$$\hat{y} = sigmoid\left(y_{FM} + y_{DNN}\right) \tag{5.27}$$

δηλαδή από την σιγμοειδή συνάρτηση του αθροίσματος των δύο κύριων στοιχείων του αλγορίθμου, του FM και του Deep.

Στην Εικόνα 5.2 παρουσιάζεται η αρχιτεκτονική του αλγορίθμου, όπως αυτή παρουσιάστηκε στο [**?**]
.

Υπερπαράμετροι

hidden_neurons: ο αριθμός των νευρώνων σε κάθε επίπεδο, η αύξησή τους προκαλεί και αύξηση της πολυπλοκότητας.
hidden_activations: λίστα η οποία περιέχει τις συναρτήσεις ενεργοποίησης (activation functions), οι οποίες καθορίζουν την έξοδο ενός κόμβου σε ένα νευρωνικό δίκτυο.
l_w: παράμετρος κανονικοποίησης στα Embeddings χρηστών και αντικειμένων.

5.2.1.5 Μη-εξατομικευμένοι αλγόριθμοι

Στα συστήματα συστάσεων αυτού του είδους όλοι οι χρήστες λαμβάνουν τις ίδιες ακριβώς λίστες συστάσεων, εν αντιθέσει με τα εξατομικευμένα συστήματα συστάσεων (personalized recommender systems) τα οποία προτείνουν σε κάθε χρήστη αντικείμενα με βάση προηγούμενες αλληλεπιδράσεις του. Χαρακτηριστικότερο παράδειγμα μη-εξατομικευμένου συστήματος συστάσεων αποτελεί ο αλγόριθμος **Most popular (MostPop)** ο οποίος προτείνει σε κάθε χρήστη τα πιο δημοφιλή αντικείμενα. Σε αυτήν την κατηγορία ανήκει επίσης και ο αλγόριθμος **Random** ο οποίος επιλέγει

αντικείμενα με τυχαίο τρόπο από το σύνολο των διαθέσιμων αντικειμένων, ακολουθώντας την κανονική κατανομή, αποφεύγοντας τα αντικείμενα που έχουν ήδη καταναλωθεί και τα προτείνει στους χρήστες. Όπως γίνεται εύκολα αντιληπτό, αυτός ο αλγόριθμος δεν πετυχαίνει καλή ακρίβεια ειδικά αν είναι μεγάλος ο αριθμός των αντικειμένων, και η απόδοσή του εξαρτάται από την τύχη, επομένως χρησιμοποιείται μόνο για πειραματικούς σκοπούς όπως θα δείξουμε και στη συνέχεια. Οι αλγόριθμοι αυτοί χρησιμοποιούνται ως baseline (MostPop χειρότερη επίδοση και Random την καλύτερη σε ότι αφορά το popularity bias) και συγκρίνονται με τους αλγορίθμους που αναλύσαμε παραπάνω, προκειμένου να εξάγουμε πιο ασφαλή συμπεράσματα κυρίως σε ό,τι αφορά το popularity bias.

5.2.2 Μετρικές αξιολόγησης

Για την αξιολόγηση των αποτελεσμάτων, δηλαδή των λιστών συστάσεων που παρήγαγαν οι αλγόριθμοι που περιγράψαμε, χρησιμοποιήθηκαν τέσσερις μετρικές ακρίβειας, πέντε μετρικές popularity bias, μία μετρική κάλυψης αντικειμένων, μία μετρική diversity και μια μετρική novelty. Ακολουθεί η αναλυτική περιγραφή τους.

5.2.2.1 Μετρικές ακρίβειας

Normalized discounted cumulative gain (nDCG): [94], [95] ο βαθμός στον οποίο η κατάταξη που υπέβαλε ένας χρήστης συμφωνεί με την ιδανική κατάταξη, λαμβάνοντας υπόψη τη συνάφεια κάθε στοιχείου σε αυτήν τη λίστα αντικειμένων για κατάταξη. Ακολουθούν τα αναλυτικά βήματα υπολογισμού του nDCG. Κάθε πρόταση (recommendation) έχει μια βαθμολογία σχετικότητας (relevance score), η οποία ονομάζεται και κέρδος (gain). Το αθροιστικό κέρδος (cumulative gain) μας δίνει το άθροισμα όλων των βαθμολογιών σχετικότητας.

$$\text{CumulativeGain}\,(CG) = \sum_{i=1}^{n} relevance_i \tag{5.28}$$

Ωστόσο, το cumulative gain δεν λαμβάνει καθόλου υπόψη του την θέση των στοιχείων στη λίστα κατάταξης, κάτι αρκετά σημαντικό σε μια εργασία κατάταξης αντικειμένων (ranking task). Το Discounted Cumulative Gain (DCG) προσθέτει έναν λογαριθμικό παράγοντα αναγωγής (reduction factor) προκειμένου να «τιμωρήσει» (penalize) την βαθμολογία σχετικότητας αναλογικά με τη θέση του αντικειμένου

$$\text{DiscountedCumulativeGain}\,(DCG) = \sum_{i=1}^{n} \frac{relevance_i}{\log_2(i+1)} \tag{5.29}$$

Ένα πρόβλημα που προκύπτει με το DCG είναι πως στους χρήστες θα προταθεί ένας αρκετά μεταβλητός αριθμός σχετικών αντικειμένων. Οπότε θα είναι αρκετά δύσκολο να γίνουν συγκρίσεις ανάμεσα στους χρήστες. Για αυτόν τον λόγο κρίνεται απαραίτητη μια κανονικοποίηση των αποτελεσμάτων που παράγει η μετρική ώστε να βρίσκονται στο εύρος [0,1], η οποία θα μας δώσει την ιδανική κατάταξη για έναν χρήστη. Στη συνέχεια, χρησιμοποιούμε αυτή την κατάταξη ως το Ideal Discounted Cumulative Gain (IDCG)

$$\text{IdealDiscountedCumulativeGain}\,(IDCG) = \sum_{i=1}^{REL_n} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{5.30}$$

όπου rel_i η σχετικότητα του αντικειμένου στη θέση i στη λίστα συστάσεων και $2^{rel_i} = 1$ αν είναι σχετικό (hit) αλλιώς είναι ίσο με 0. Το nDCG είναι ο λόγος του DCG προς το IDCG

$$\text{Normalized Discounted Cumulative Gain } (nDCG) = \frac{DCG}{IDCG} \tag{5.31}$$

Precision ή true positive accuracy ή εμπιστοσύνη (confidence) είναι το ποσοστό των συστάσεων που ορθά προτάθηκαν από τον αλγόριθμο (True Positive) σε σχέση με το σύνολο των συστάσεων - ορθών και λανθασμένων (True positive + False positive):

$$precision = \ tpa = \ \frac{tp}{tp + fp} \tag{5.32}$$

Ανάκληση (Recall): ή true positive rate ή ευαισθησία (sensitivity). Υπολογίζει το ποσοστό των αντικειμένων που έχουν προταθεί και είναι σχετικά σε σχέση με τον συνολικό αριθμό των σχετικών αντικειμένων. Με άλλα λόγια, το ποσοστό των συστάσεων που ορθά προτάθηκαν από τον αλγόριθμο (True Positive) σε σχέση με το (ιδανικό) σύνολο όλων των συστάσεων που θα μπορούσαν να προταθούν:

$$recall = \ tpr = \ \frac{tp}{tp + fn} \tag{5.33}$$

Hit Rate (HR) [96]
Ο αριθμός των hits είναι ο αριθμός των αντικειμένων στο σύνολο δοκιμής το οποία εμφανίζονται επίσης και στα Top-n αντικείμενα που δίνονται ως συστάσεις σε κάθε χρήστη.

$$\text{HR@K} = \frac{\text{Αριθμός των Hits@}K}{\text{Αριθμός χρηστών}} \tag{5.34}$$

Εάν το hit rate είναι ίσο με 1 αυτό σημαίνει πως ο αλγόριθμος έχει εντοπίσει όλα τα «κρυφά» αντικείμενα.

5.2.2.2 Μετρικές μεροληψίας δημοφιλίας

Μέση δημοφιλία των συστάσεων - Average Recommendation Popularity (ARP): Αυτή η μετρική [97] υπολογίζει τη μέση δημοφιλία (average popularity) των αντικειμένων που προτείνονται σε κάθε λίστα.

$$ARP = \frac{1}{|U_t|} \sum_{u \in U_t} \frac{\sum_{i \in L_u} \varphi(i)}{|L_u|} \tag{5.35}$$

όπου $\boldsymbol{\varphi(i)}$ είναι ο αριθμός των φορών που το αντικείμενο (item) i έχει αξιολογηθεί στο σύνολο εκπαίδευσης, L_u είναι η λίστα των προτεινόμενων αντικειμένων για τον χρήστη u και U_t είναι ο αριθμός των χρηστών στο σύνολο δοκιμής.

Μέσο ποσοστό των long tail αντικειμένων - Average Percentage of Long Tail Items (APLT): αυτή η μετρική που ορίζεται στο [98], μετράει το μέσο ποσοστό των long tail αντικειμένων

στις λίστες συστάσεων, δηλαδή το μέσο ποσοστό των αντικειμένων στις λίστες συστάσεων, που δημιουργούνται για κάθε χρήστη, που ανήκουν στο σύνολο των long tail αντικειμένων Γ

$$APLT = \frac{1}{U_t} \sum_{u \in U_t} \frac{\left|\left\{i, i \in \left(L_u \cap \Gamma\right)\right\}\right|}{\left|L_u\right|} \tag{5.36}$$

Μέση κάλυψη των long-tail αντικειμένων - Average Coverage of Long Tail items (ACLT): Ένα πρόβλημα με την μετρική APLT είναι πως μπορεί να έχει υψηλή τιμή ακόμη και αν όλοι οι χρήστες λάβουν το ίδιο σύνολο long tail αντικειμένων. Μια λύση σε αυτό το πρόβλημα δόθηκε στο [99], μέσω της μετρικής ACLT, μια ακόμη μετρική για την αξιολόγηση της έκθεσης των long tail αντικειμένων στις λίστες συστάσεων.

$$ACLT = \frac{1}{U_t} \sum_{u \in U_t} \sum_{i \in L_u} 1\left(i \in \Gamma\right) \tag{5.37}$$

Τέλος, χρησιμοποιήθηκαν 2 μετρικές που ορίζονται στο [100]
Popularity Ranking-based Statistical Parity (PopRSP): υπολογίζει κατά πόσο η πιθανότητα να προταθούν αντικείμενα που ανήκουν στο long-tail και η πιθανότητα να προταθούν αντικείμενα που ανήκουν στο head είναι ίδια

$$RSP = \frac{std\left(P\left(R@k \mid g = g_1\right), \ldots, P\left(R@k \mid g = g_A\right)\right)}{mean\left(P\left(R@k \mid g = g_1\right), \ldots, P\left(R@k \mid g = g_A\right)\right)} \tag{5.38}$$

Popularity Ranking-based Equal Opportunity (PopREO): υπολογίζει κατά πόσο τα true positive rates των αντικειμένων που ανήκουν στο long-tail και των αντικειμένων που ανήκουν στο head είναι ίδια, λαμβάνοντας υπόψη τις προτιμήσεις των χρηστών.

$$REO = \frac{std\left(P\left(R@k \mid g = g_1, y = 1\right) \ldots P\left(R(a)k = g_A, y = 1\right)\right)}{mean\left(P\left(R@k \mid g = g_1, y = 1\right) \ldots P\left(R@k \mid g = g_A, y = 1\right)\right)} \tag{5.39}$$

5.2.2.3 Μετρικές diversity

Gini index [101]: υπολογίζει την ανισότητα που υπάρχει στην κατανομή συχνότητας των προτεινόμενων αντικειμένων και συνεπώς το diversity που υπάρχει στα αποτελέσματα. Η τιμή του gini index είναι υψηλή εάν ορισμένα αντικείμενα προτείνονται πιο συχνά σε σχέση με άλλα αντικείμενα. Επομένως ένας αλγόριθμος που ακολουθεί την κανονική κατανομή σε ότι αφορά τις συστάσεις των αντικειμένων και προτείνει κάθε αντικείμενο ίδιο αριθμό φορών, θα έχει gini index ίσο με 0. Στην αντίθετη περίπτωση, όπου ο αριθμός των φορών που προτείνονται τα διαφορετικά αντικείμενα είναι αρκετά άνισος, τότε η μετρική αυτή θα είναι ίση με 1. Το gini index ορίζεται ως εξής:

$$GiniIndex = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p\left(i_j\right) \tag{5.40}$$

Σε αυτό το σημείο κρίνεται αναγκαίο να επισημάνουμε πως στο Elliot η μετρική αυτή διαφοροποιείται και υπολογίζεται το $1 - Gini$, ώστε υψηλή τιμή του Gini index να σημαίνει καλύτερο αποτέλεσμα διευκολύνοντας έτσι τον αναγνώστη.

5.2.2.4 Μετρικές novelty

Expected Popularity Complement (EPC): [102] αξιολογεί τον αναμενόμενο αριθμό σχετικών αντικειμένων που δεν είχε δει προηγουμένως ο χρήστης. Ένα σύστημα συστάσεων λαμβάνει υψηλότερη τιμή EPC όταν όχι μόνο προτείνει αντικείμενα από το long tail, αλλά τα κατατάσσει επίσης ψηλά στις λίστες συστάσεων. Εάν το σύστημα συστάσεων προτείνει αντικείμενα με χαμηλή εκτιμώμενη έκθεση, τότε το EPC θα είναι κοντά στο ένα.

$$EPC = C \sum_{i_k \in R} \text{disc}(k) p\left(rel \mid i_k, u\right)\left(1 - p\left(\text{seen} \mid i_k\right)\right) \tag{5.41}$$

Όπου

- $p\left(seen \mid i_k\right)$ είναι η πιθανότητα ένας χρήστης να δει το αντικείμενο i
- $p\left(rel \mid i_k, u\right)$ δηλώνει την πιθανότητα το αντικείμενο i που βρίσκεται στη θέση k της κατάταξης των αντικειμένων στη λίστα συστάσεων, να είναι σχετικό για τον χρήστη u.
- C είναι μια σταθερά κανονικοποίησης, της οποίας ο σκοπός είναι να σταθεροποιήσει την μετρική και να μην εισαχθεί κάποια ανεπιθύμητη μεροληψία σε αυτή.

$$C = \frac{1}{\sum_{i_k \in R} \text{disc}(k)}$$

- $disc\,(k)$ έστω ότι ένας χρήστης περιηγείται τα αντικείμενα μιας λίστας με τη σειρά κατάταξής τους, έως ότου σταματήσει. Σε κάθε θέση k της κατάταξης, ο χρήστης λαμβάνει την απόφαση για το αν θα συνεχίσει ή όχι. Στόχος είναι να βρούμε την πιθανότητα ένας χρήστης u να συνεχίσει μειώνοντας όσο γίνεται περισσότερο την υπολογιστική πολυπλοκότητα είναι. Προκειμένου να συμβεί αυτό χρησιμοποιούμε μια συνάρτηση έκπτωσης (discount function) η οποία εκφράζει το γεγονός ότι όσο πιο χαμηλά είναι ένα αντικείμενο στη λίστα τόσο λιγότερες είναι οι πιθανότητες να το δει ένας χρήστης.

5.2.2.5 Μετρικές coverage

Item Coverage: [103] υπολογίζει τον συνολικό αριθμό των (μοναδικών) αντικειμένων που έχουν προταθεί από έναν αλγόριθμο σε όλους τους χρήστες ενός συνόλου δεδομένων.

$$\text{Item coverage} = |\cup_{u \in U}\ L_n(u)| \tag{5.42}$$

όπου $L_n(u) = \left\{i_1 \cdots \imath_N\right\}$ η λίστα των N αντικειμένων που έχουν προταθεί στον χρήστη u.

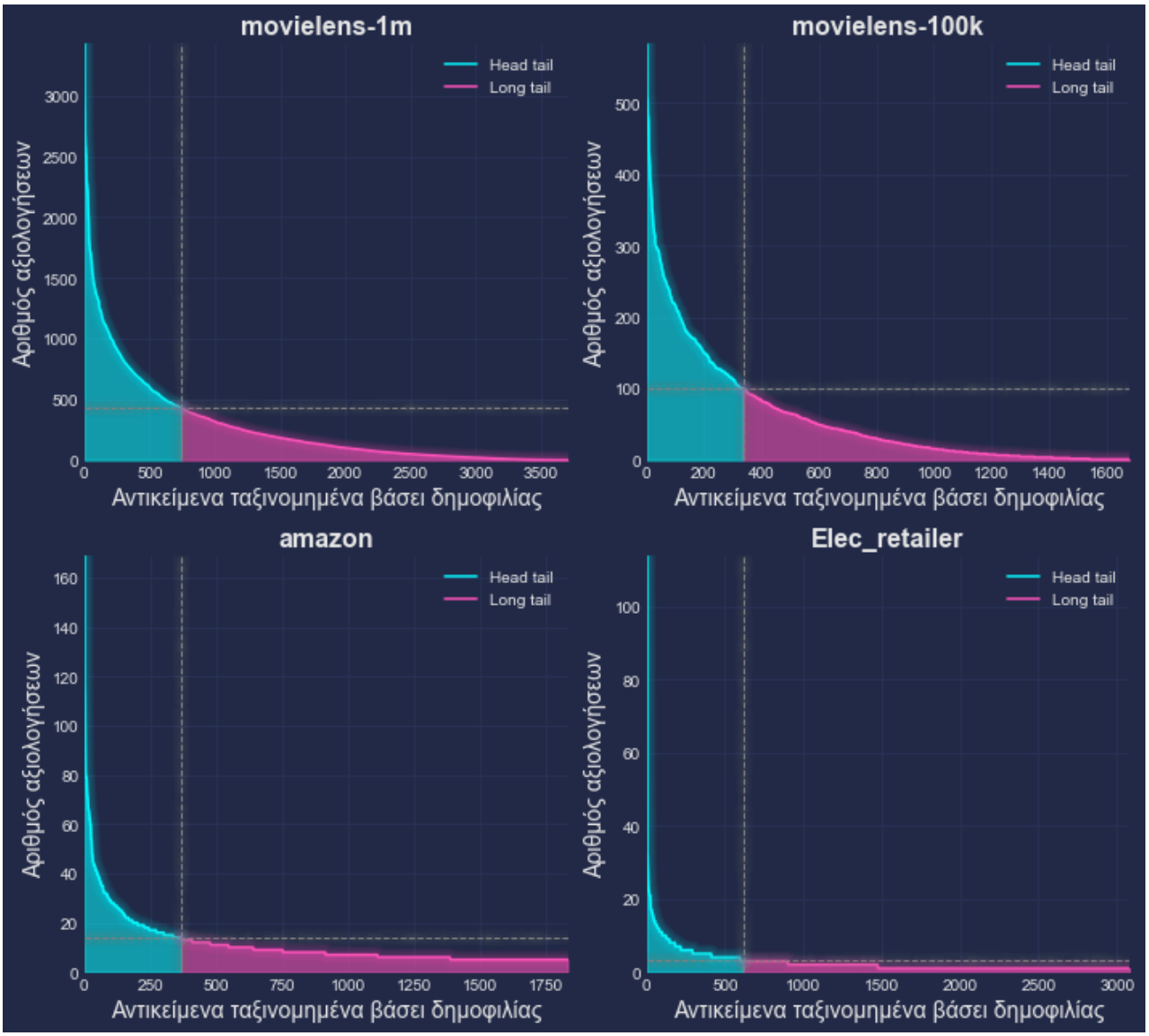
5.3 Επεξεργασία και οπτικοποίηση συνόλων δεδομένων

Ύστερα από ανάλυση των συνόλων δεδομένων που αντλήσαμε από το διαδίκτυο, προέκυψε πως δεν χρειάζονται κάποια επεξεργασία και μπορούν να χρησιμοποιηθούν άμεσα.
Το σύνολο δεδομένων στο οποίο χρειάστηκε να γίνει η μεγαλύτερη επεξεργασία είναι το Elec_-retailer. Αυτό ήταν αναμενόμενο καθώς είναι το μοναδικό από τα τέσσερα σύνολα δεδομένων που είναι «πραγματικό», δηλαδή προέρχεται κατευθείαν από την βάση δεδομένων μιας εταιρείας έχοντας υποστεί ελάχιστη ή και καθόλου επεξεργασία. Αρχικά, για πάνω από 200 αξιολογήσεις που υπήρχαν στο Elec_retailer υπήρχε μόνο ο υπερσύνδεσμος για το προϊόν και όχι το ID και χρειάστηκε να βρούμε το ID του, επισκέπτοντας τον αντίστοιχο σύνδεσμο για κάθε μία αξιολόγηση, κάτι που κόστισε αρκετό χρόνο. Σε αυτό το σημείο θα πρέπει να αναφερθεί πως υπήρχαν περίπου 700 κριτικές προϊόντων χωρίς να υπάρχει διαθέσιμο το ID του προϊόντος, η ονομασία του ή έστω κάποιος σύνδεσμος που να οδηγεί σε αυτό. Επειδή η διαδικασία εύρεσης αυτών των προϊόντων ήταν από δύσκολη έως αδύνατη σε πολλές περιπτώσεις, αλλά και αρκετά χρονοβόρα λάβαμε την απόφαση αυτές οι κριτικές να απορριφθούν. Από την παραπάνω επεξεργασία του συνόλου δεδομένων απέμειναν 3.078 προϊόντα και 8.263 αξιολογήσεις. Ένα από τα μεγαλύτερα προβλήματα που υπήρχαν εδώ, ήταν η απουσία κάθε πληροφορίας για τους χρήστες, ένα από τα τρία πιο σημαντικά στοιχεία για τα συστήματα συστάσεων που ερευνούμε (τα υπόλοιπα δύο είναι φυσικά πληροφορίες για τις αξιολογήσεις και για τα αντικείμενα). Το πρόβλημα μετατράπηκε σε μία μεγάλη και ωραία πρόκληση, καθώς μας οδήγησε σε διαφορετικά ερευνητικά μονοπάτια. Αυτό συνέβη διότι, η δημιουργία των χρηστών δεν ήταν απλή υπόθεση, δηλαδή δεν θα ήταν επιστημονικά ορθό να επιλέξουμε έναν τυχαίο αριθμό χρηστών και να τους δημιουργήσουμε. Έπρεπε να λάβουμε υπόψη μας διάφορες παραμέτρους και κυρίως το πρόβλημα της αραιότητας του μητρώου χρηστών-αντικειμένων, το οποίο εξαρτάται άμεσα και από τον αριθμό των χρηστών που υπάρχουν σε ένα σύνολο δεδομένων. Με γνώμονα όλα τα παραπάνω δημιουργήθηκαν 3 διαφορετικά σύνολα δεδομένων, με 3 διαφορετικούς αριθμούς χρηστών:

1. **Elec_retailer30:** 276 χρήστες, εκ των οποίων οι 275 έχουν αξιολογήσει 30 προϊόντα ο καθένας και ένας 13 προϊόντα.
2. **Elec_retailer20:** 414 χρήστες, εκ των οποίων οι 413 έχουν αξιολογήσει 20 προϊόντα ο καθένας και ένας 3 προϊόντα.
3. **Elec_retailer10:** 826 χρήστες, εκ των οποίων οι 825 έχουν αξιολογήσει 10 προϊόντα ο καθένας και ένας 3 προϊόντα.

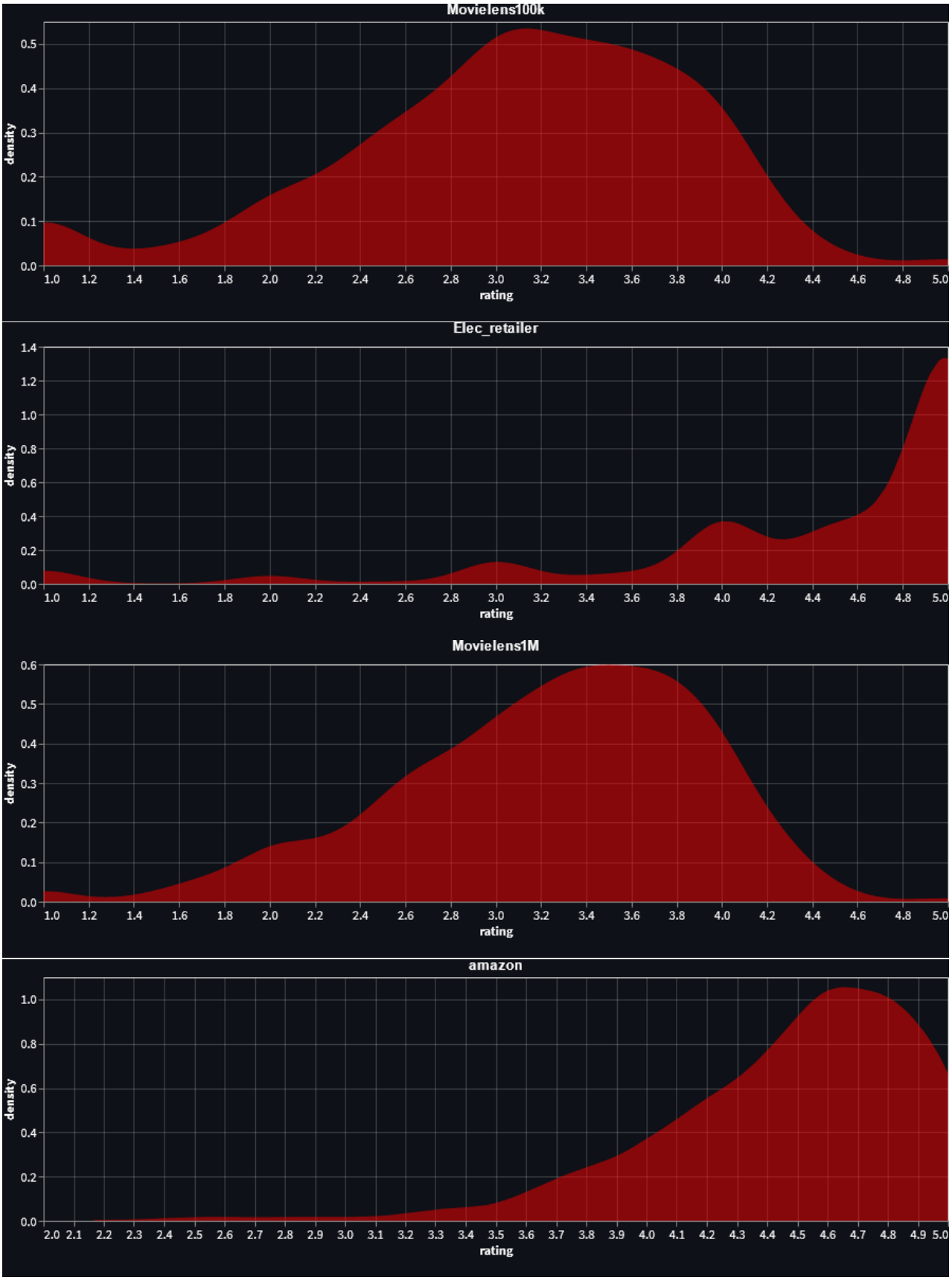
Για όλα τα σύνολα δεδομένων που χρησιμοποιήθηκαν έγινε με τυχαίο τρόπο η διάσπασή τους σε 2 μικρότερα σύνολα δεδομένων. Πιο συγκεκριμένα, το 80% αποτέλεσε το σύνολο εκπαίδευσης και το 20% το σύνολο δοκιμής.
Ύστερα από την ολοκλήρωση της επεξεργασίας των δεδομένων, σειρά έχει η οπτικοποίησή τους για την καλύτερη κατανόησή τους, η οποία θα βοηθήσει αρκετά στη συνέχεια, τόσο στη ρύθμιση των υπερπαραμέτρων των αλγορίθμων συστημάτων συστάσεων, όσο και στην αξιολόγησή τους. Αρχικά, για την παρατήρηση του φαινομένου long-tail δημιουργήθηκε το κάτωθι γράφημα, στο οποίο το 20% των αντικειμένων που έχουν λάβει τις περισσότερες αξιολογήσεις, δηλαδή τα πιο δημοφιλή αντικείμενα, ανήκει στο head, και το υπόλοιπο 80% αφορά λιγότερο δημοφιλείς ή καινούριες ταινίες. Πιο συγκεκριμένα, στον άξονα x είναι τα IDs των ταινιών διατεταγμένα κατά φθίνουσα σειρά ως προς τον αριθμό των αξιολογήσεων που έχουν λάβει και στον άξονα y ο συνολικός αριθμός των αξιολογήσεων που έλαβε κάθε ταινία. Από το διάγραμμα αυτό είναι αρκετά ευδιάκριτο το φαινόμενο

του long tail, καθώς περίπου 700 ταινίες έχουν λάβει από 500 έως και περίπου 3.500 αξιολογήσεις η κάθε μία, ενώ 3.000 ταινίες έχουν λάβει από λίγες έως ελάχιστες αξιολογήσεις.



Εικόνα 5.3: Το φαινόμενο του long-tail στα 4 σύνολα δεδομένων

Στην Εικόνα 5.4 παρουσιάζεται για κάθε σύνολο δεδομένων ένα διάγραμμα το οποίο θα μας βοηθήσει να κατανοήσουμε καλύτερα την κατανομή των αξιολογήσεων. Στον άξονα x έχουμε τον μέσο όρο των αξιολογήσεων των αντικειμένων και στον άξονα y την Εκτίμηση πυκνότητας πυρήνα (Kernel Density Estimation - KDE), όπου όλα τα σημεία του χώρου λαμβάνουν κάποια τιμή η οποία δείχνει ότι υπάρχει συγκεκριμένη πιθανότητα επίσκεψής τους. Προκειμένου να γίνει πιο κατανοητή η έννοια του μέσου όρου των αξιολογήσεων ας δούμε ένα παράδειγμα. Έστω ότι μια ταινία έχει αξιολογηθεί τέσσερις φορές και έχει λάβει τις βαθμολογίες 2, 2, 3 και 5. Τότε ο μέσος όρος αξιολόγησης αυτής της ταινίας είναι $\frac{2+2+3+5}{4} = \frac{15}{4} = 3,75$. Στα σύνολα δεδομένων από τον χώρο του ηλεκτρονικού εμπορίου η συντριπτική πλειονότητα των αντικειμένων έχει μέσο όρο αξιολογήσεων μεγαλύτερο του 3,5. Αντιθέτως, στα σύνολα δεδομένων του Movielens υπάρχει πολύ μεγαλύτερη ισορροπία και μάλιστα η κατανομή τους είναι αρκετά στην κανονική κατανομή, κάτι που είναι ευδιάκριτο από τα διαγράμματα.



Εικόνα 5.4: Κατανομή μέσης αξιολόγησης ανά αντικείμενο στα 4 σύνολα δεδομένων

5.4 Αξιολόγηση αποτελεσμάτων

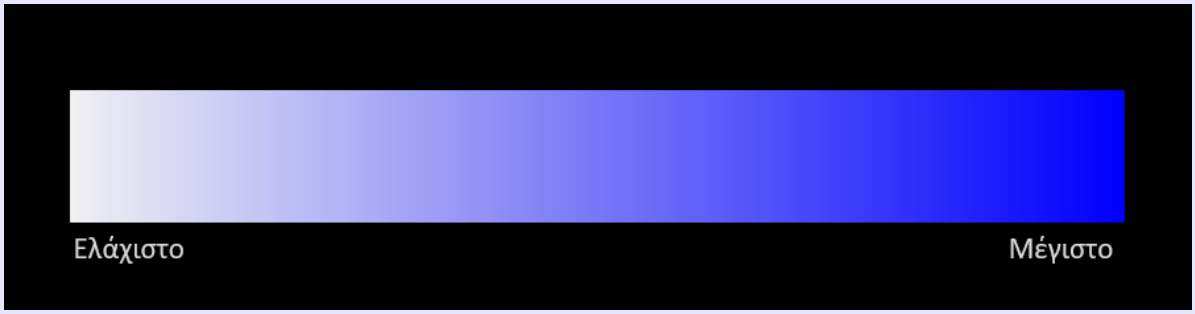
Αφού δημιουργήσαμε τις λίστες συστάσεων στο προηγούμενο βήμα στην ενότητα αυτή παρουσιάζουμε και αναλύουμε τα αποτελέσματα της αξιολόγησης αυτών των συστάσεων ως προς την ακρίβεια, το popularity bias, το diversity, το novelty και την κάλυψη των αντικειμένων, όπως αυτά προέκυψαν από τις μετρικές αξιολόγησης που επιλέξαμε. Στις υποενότητες που ακολουθούν θα βρείτε δύο τύπους ανάλυσης την «Ανάλυση υπερπαραμέτρων» και την «Ανάλυση καλύτερων αποτελεσμάτων». Γενικότερα στην παρούσα ενότητα, θα επιχειρήσουμε να απαντήσουμε τα εξής ερωτήματα:

1. Εκτός της ακρίβειας, επηρεάζουν οι διαφορετικές τιμές των υπερπαραμέτρων το popularity bias, το diversity, το novelty και την κάλυψη των αντικειμένων, και αν ναι σε ποιον βαθμό;
2. Ποια είναι η συμπεριφορά των οικογενειών αλγορίθμων ειδικότερα και των διάφορων αλγορίθμων γενικότερα, απέναντι σύνολα δεδομένων που χρησιμοποιήθηκαν; Είναι αυτό εφικτό να επιτευχθεί σε όλες τις περιπτώσεις;
3. Ποιες οι βέλτιστες τιμές των υπερπαραμέτρων ώστε να επιτευχθεί το αντιστάθμισμα μεροληψίας-ακρίβειας;
4. Επηρεάζουν τα χαρακτηριστικά των δεδομένων τα αποτελέσματα;
5. Ποιες είναι οι επιπτώσεις από τη λάθος επιλογή ενός αλγορίθμου ή/και των υπερπαραμέτρων του;

Προκειμένου να εξάγουμε όσο το δυνατόν πιο ασφαλή συμπεράσματα, στην «Ανάλυση υπερπαραμέτρων» και στον «Μετριασμό της μεροληψίας» δεν χρησιμοποιήθηκε το σύνολο δεδομένων Elec_retailer στο οποίο έχουμε δημιουργήσει μόνοι μας τους χρήστες, όπως προαναφέρθηκε, και επομένως είναι υπαρκτός ο κίνδυνος να έχουμε εισάγει κάποιο είδος μεροληψίας άθελά μας.

Σημείωση

Όλοι οι πίνακες που δημιουργήθηκαν στην παρούσα ενότητα χρησιμοποιούν την κάτωθι χρωματική διαβάθμιση, προκειμένου να διευκολύνουν τον αναγνώστη να διαχωρίσει τα καλύτερα από τα χειρότερα αποτελέσματα. Εδώ απαιτείται ιδιαίτερη προσοχή, καθώς στις μετρικές PopREO, PopRSP και ARP όσο μεγαλύτερη είναι η τιμή τους, τόσο μεγαλύτερη είναι η μεροληψία που εισάγεται.



5.4.1 Ανάλυση υπερπαραμέτρων

Προτού προχωρήσουμε όμως στην ανάλυση των υπερπαραμέτρων κρίνεται αναγκαίο να δοθούν ορισμένες διευκρινίσεις. Αρχικά, στον πίνακα 5.3 παρουσιάζονται όλοι οι αλγόριθμοι που χρησιμοποιήθηκαν στα πειράματα - εκτός από τους δύο μη-εξατομικευμένους - , μαζί με όλες τις τιμές των υπερπαραμέτρων που επιλέχθηκαν. Σε παρένθεση βρίσκεται η συντομογραφία της ονομασίας των υπερπαραμέτρων που χρησιμοποιείται στους πίνακες μέσω των οποίων παρουσιάζονται τα αποτελέσματα, χάριν εξοικονόμησης χώρου.
Στο σημείο αυτό θα πρέπει να σημειωθεί ότι εξαιτίας του μεγάλου όγκου των δεδομένων που προέκυψαν από την εκτέλεση κάθε αλγορίθμου για τρία διαφορετικά σύνολα δεδομένων και διάφορες τιμές των υπερπαραμέτρων και την αξιολόγηση από δώδεκα διαφορετικές μετρικές αξιολόγησης, στην ανάλυση που ακολουθεί δεν παρουσιάζονται όλες οι μετρικές, προς διευκόλυνση του αναγνώστη. Από τις μετρικές ακρίβειας έχει επιλεγεί η nDCG και από τις μετρικές μεροληψίας δημοφιλίας οι ARP, APLT, PopREO, PopRSP, μαζί με τις μετρικές EPC, Gini index και Item coverage.
Επιπροσθέτως, οι τιμές του μεγέθους των λιστών συστάσεων που θα ληφθούν υπόψη κατά τον υπολογισμό των μετρικών αξιολόγησης (cut-off) είναι: [10, 30, 50, 70, 100], ενώ η τιμή του cut-off που επιλέχθηκε για την ανάλυση των υπερπαραμέτρων είναι η 100, καθώς χρησιμοποιείται αρκετά συχνά στην βιβλιογραφία και είναι κατάλληλη σύμφωνα με τα χαρακτηριστικά των δεδομένων των συνόλων δεδομένων που χρησιμοποιήθηκαν. Τέλος, η μετρική ομοιότητας “dot” στους αλγορίθμους ItemKNN και UserKNN, είχε πολύ κακή επίδοση για όλα τα σύνολα δεδομένων και όλες τις μετρικές αξιολόγησης και για τον λόγο αυτό δεν παρουσιάζεται στην ανάλυση που ακολουθεί, ώστε να μην μπερδέψει τον αναγνώστη.

Πίνακας 5.3: Αλγόριθμοι και οι υπερπαράμετροί τους

Αλγόριθμος	**Υπερπαράμετροι**
ItemKNN	neighbors (**nn**): [30, 50, 70] similarity: βλέπε πίνακα 5.4
UserKNN	implementation (**imp**): [classical, aiolli]
SVD++	Batch size (**bs**): [256, 512] Factors: [50, 70] Learning rate: [0.001, 0.01, 0.1] reg_b: [0.1, 0.5, 0.7]
MF	batch_size: [256, 512] factors (**f**): [20, 50, 70] Learning rate (**lr**): [0.001, 0.01, 0.1] reg: [0.01, 0.1, 0.5, 0.7]
BPRMF	factors: [20, 50, 70] Learning rate: [0.001, 0.01, 0.1] reg: 0.01, 0.1, 0.5, 0.7
WRMF	factors: [50] alpha: [10] reg: [0.1, 0.5, 0.7]
SLIM	l1_ratio: 0.1, 0.5, 1 alpha: [0.0001, 0.001, 0.01, 0.1, 1]
DeepFM	epochs: 10 batch_size: 512 factors: [30, 50, 70, 100] Learning rate: [0.001, 0.01, 0.1] l_w: 0.0001 hidden_neurons: (64,32) hidden_activations: ('relu','relu')
NGCF	Learning rate: [0.001, 0.01, 0.1] epochs: 50 batch_size: 512 factors: 64 batch_size: 256 l_w: [0.001, 0.1, 1]

5.4.1.1 Αλγόριθμοι γειτνίασης

Στους δύο αλγορίθμους που επιλέχθηκαν από αυτή την οικογένεια δοκιμάστηκαν 22 διαφορετικές μετρικές ομοιότητας. Τα αποτελέσματα μπορούμε να τα δούμε αναλυτικά στο παράρτημα, ενώ για όλες τις μετρικές που δοκιμάστηκαν έχουν επιλεγεί 30 κοντινότεροι γείτονες και υλοποίηση classical.

Πίνακας 5.4: Μετρικές ομοιότητας στους neighborhood-based αλγορίθμους

Μετρική	Τύπος
braycurtis	$\frac{\sum \lvert u_i - v_i \rvert}{\sum \lvert u_i + v_i \rvert}$
canberra	$\sum_i \frac{\lvert u_i - v_i \rvert}{\lvert u_i \rvert + \lvert v_i \rvert}$
chebyshev	$\max_i \lvert u_i - v_i \rvert$
corellation	$1 - \frac{\left(r_{i,u} - \overline{r_{i,u}}\right) \cdot \left(r_{j,u} - \overline{r_{j,u}}\right)}{\| \left(r_{i,u} - \overline{r_{i,u}}\right) \|_2 \| \left(r_{j,u} - \overline{r_{j,u}}\right) \|_2}$
cosine	$1 - \frac{\sum_u r_{i,u} r_{j,u}}{\sqrt{\sum_u r_{i,u}^2} \sqrt{\sum_u r_{j,u}^2}}$
dice	$1 - \frac{2 \sum_u r_{i,u} r_{j,u}}{\sum_u r_{i,u}^2 + \sum_u r_{j,u}^2}$
euclidean	$\sqrt{\sum_i \lvert r_{i,u} - r_{j,u} \rvert^2}$
hamming	$\frac{c_{01} + c_{10}}{n}$
jaccard	$\frac{\sum_u \left(r_{i,u} - r_{j,u}\right)^2}{\sum_u r_{i,u}^2 + \sum_u r_{j,u}^2 - \sum_u r_{i,u} r_{j,u}}$
⋮	⋮

Μετρική	Τύπος
⋮	⋮
kulsinski	$\frac{\sum_{i=1} \lvert r_{i,u} - r_{j,u\lvert}}{\sum_{i=1} \max\left(r_{i,u}, r_{i,u}\right)}$
manhattan	$\sum_i \left\lvert u_i - v_i \right\rvert$
minkowski	$\left(\sum \left(\lvert w_i (u_i - v_i) \rvert^p \right)\right)^{1/p}$
rogerstanimoto	$\frac{R}{c_{TT} + c_{FF} + R}$
russullrao	$\frac{n - c_{TT}}{n}$
Standarized euclidean (seuclidean)	$\sqrt{\sum_i \left(\frac{r_{i,u} - r_{j,u}}{\sigma_i} \right)^2}$
sokalmichener	$\frac{R}{c_{FF} + c_{TT} + R}$
sokalsneath	$\frac{R}{c_{TT} + R}$
sqeuclidean	$\sum \left(w_i \| \left(u_i - v_i\right) \|^2 \right)$
yule	$\frac{R}{c_{TT} * c_{FF} + \frac{R}{2}}$

όπου c_{ij} είναι ο αριθμός των εμφανίσεων για το $\mathtt{u[k]} = i$ και $\mathtt{v[k]} = j$ για $k < n$ και $R = 2(c_{TF} + c_{FT})$,

$$\begin{aligned} not_u &= u \\ not_v &= v \\ c_{FF} &= \sum (not_u \& not_v) \\ c_{FT} &= \sum (not_u \& v) \\ c_{TF} &= \sum (u \& not_v) \\ c_{TT} &= \sum (u \& v) \end{aligned} \tag{5.43}$$

Αλγόριθμος itemKNN

Πίνακας 5.5: Ανάλυση υπερπαραμέτρων στον αλγόριθμο itemKNN

dataset	nn	imp	nDCG	IC	EPC	Gini	ARP	APLT	PopREO	PopRSP
ml1m	30	aiolli	0.3881	2600	0.1456	0.1625	901.3742	0.0423	0.8096	0.9621
	50	aiolli	0.3949	2522	0.1469	0.1543	926.2476	0.0377	0.8287	0.9664
	70	aiolli	0.3963	2466	0.1470	0.1495	939.7414	0.0352	0.8399	0.9686
	30	classical	0.3876	2605	0.1454	0.1629	900.3087	0.0424	0.8085	0.9620
	50	classical	0.3948	2527	0.1469	0.1545	925.8837	0.0377	0.8276	0.9663
	70	classical	0.3965	2473	0.1471	0.1497	938.7630	0.0352	0.8401	0.9686
amazon	30	aiolli	0.0979	1834	0.0071	0.8007	8.4341	0.4173	0.2035	0.0154
	50	aiolli	0.1026	1834	0.0074	0.7811	9.1116	0.3678	0.2704	0.1187
	70	aiolli	0.1051	1834	0.0076	0.7628	9.5636	0.3509	0.2947	0.1547
	30	classical	0.0975	1834	0.0070	0.7987	8.3823	0.4227	0.1852	0.0044
	50	classical	0.1023	1834	0.0074	0.7824	9.0860	0.3688	0.2648	0.1166
	70	classical	0.1049	1834	0.0076	0.7634	9.5461	0.3515	0.2960	0.1537
ml10ok	30	aiolli	0.4629	1174	0.1330	0.2525	143.1330	0.1014	0.7122	0.9085
	50	aiolli	0.4620	1087	0.1326	0.2428	145.1824	0.0966	0.7329	0.9130
	70	aiolli	0.4608	1058	0.1319	0.2386	146.2705	0.0956	0.7460	0.9140
	30	classical	0.4631	1177	0.1330	0.2533	142.8456	0.1018	0.7103	0.9081
	50	classical	0.4630	1093	0.1327	0.2436	144.9271	0.0969	0.7315	0.9127
	70	classical	0.4605	1062	0.1321	0.2386	146.2758	0.0957	0.7390	0.9140

Αυτό που παρατηρούμε στον Πίνακα 5.5 είναι ότι σε όλα τα σύνολα δεδομένων ο αλγόριθμος έχει αρκετά παρόμοια συμπεριφορά. Αυτό μάλιστα ισχύει για όλες τις μετρικές αξιολόγησης. Εξαίρεση αποτελεί η μετρική ItemCoverage στο Amazon, η οποία χρησιμοποιεί όλα τα διαθέσιμα αντικείμενα για να τα προτείνει στους χρήστες, ανεξαρτήτως υλοποίησης ή τιμής του αριθμού των κοντινότερων γειτόνων. Ύστερα από όσα αναφέραμε, μπορούμε να εξετάσουμε αναλυτικά τα αποτελέσματα που προέκυψαν.

Αρχικά, όσον αφορά την τιμή του αριθμού των κοντινότερων γειτόνων (nn) και της υλοποίησης που μας δίνει τα καλύτερα αποτελέσματα, θα πρέπει να επισημάνουμε πως οι διαφορές στις τιμές των μετρικών είναι αρκετά μικρές. Παρατηρούμε ότι μεγαλύτερες τιμές της υπερπαραμέτρου nn μας δίνουν καλύτερη ακρίβεια, με τη διαφορά να είναι αρκετά πιο εμφανής στο nDCG, από ότι στο precision και στο recall (βλέπε παράρτημα). Αυτό είναι αρκετά λογικό μιας και όσο περισσότερους γείτονες έχουμε, τόση περισσότερη πληροφορία παίρνουμε για έναν χρήστη και συνεπώς μπορούμε να βγάλουμε πιο ασφαλή συμπεράσματα σχετικά με το ποια ταινία ή ποιο προϊόν θα ταιριάζει καλύτερα στις προτιμήσεις του. Γενικότερα σε όλα τα σύνολα δεδομένων και ανεξαρτήτως των τιμών των υπερπαραμέτρων ο itemKNN φαίνεται να κάνει συστάσεις αρκετά σχετικές με τα ενδιαφέροντα των χρηστών, κάτι που μαρτυράει η αρκετά υψηλή τιμή της ακρίβειας. Ωστόσο, τα πράγματα είναι αρκετά διαφορετικά στις μετρικές που αφορούν το popularity bias και το diversity. Πιο συγκεκριμένα, στα δύο σύνολα δεδομένων του Movielens, αλλά και στο Amazon (παρόλο που εκεί το φαινόμενο είναι αρκετά πιο περιορισμένο) έχουμε πάρα πολύ μεγάλη εισαγωγή μεροληψίας για μεγαλύτερες τιμές του nn. Το θετικό είναι πως στην μετρική PopREO έχουμε λίγο καλύτερα

αποτελέσματα, κάτι το οποίο σημαίνει πως εάν λάβουμε υπόψη μας τις προτιμήσεις των χρηστών τα αποτελέσματα είναι αρκετά πιο δίκαια. Συμπληρωματικά, κάτι που ίσως φανεί λίγο παράδοξο κοιτάζοντας τον πίνακα 5.5 είναι ότι στα σύνολα του Movielens μικρότερη τιμή του nn συνεπάγεται περισσότερα αντικείμενα που καλύπτει ο αλγόριθμος. Μια εξήγηση που μπορεί να δοθεί εδώ είναι ότι πολλές φορές ο αριθμός των κοντινότερων γειτόνων ενός αντικειμένου που ζητάμε μπορεί να μην υπάρχει και αυτό σε μεγάλες τιμές του nn μπορεί να οδηγήσει στο να προτείνονται αρκετές φορές τα ίδια αντικείμενα, μειώνοντας την συνολική κάλυψη των αντικειμένων που προτείνονται. Αναφορικά με τις δύο διαφορετικές υλοποιήσεις που εξετάστηκαν, η κλασική υλοποίηση δίνει ελαφρώς καλύτερα αποτελέσματα όσον αφορά την κάλυψη των αντικειμένων, αλλά και την ακρίβεια στις περισσότερες των περιπτώσεων. Παρόμοια συμπεράσματα προκύπτουν και για τις μετρικές που σχετίζονται με το popularity bias και το diversity. Τέλος, μια αρκετά ενδιαφέρουσα παρατήρηση εδώ είναι πως όσο αυξάνεται η τιμή του nn, τόσο καλύτερο γίνεται το novelty, εν αντιθέσει με το diversity.

Αλγόριθμος userKNN

Πίνακας 5.6: Ανάλυση υπερπαραμέτρων στον αλγόριθμο userKNN

dataset	nn	imp	nDCG	IC	EPC	Gini	ARP	APLT	PopREO	PopRSP
ml1m	30	aiolli	0.3688	3351	0.1359	0.2589	759.2337	0.1342	0.7176	0.8731
	50	aiolli	0.3958	3250	0.1439	0.2307	810.0231	0.1030	0.7485	0.9044
	70	aiolli	0.4095	3166	0.1476	0.2119	839.8731	0.0865	0.7713	0.9206
	30	classical	0.3663	3355	0.1352	0.2607	755.4560	0.1365	0.7152	0.8707
	50	classical	0.3952	3253	0.1437	0.2319	808.0108	0.1040	0.7471	0.9035
	70	classical	0.4092	3168	0.1475	0.2127	838.5108	0.0872	0.7691	0.9199
amazon	30	aiolli	0.1053	1834	0.0073	0.6174	12.0819	0.2860	0.3984	0.2968
	50	aiolli	0.1224	1834	0.0085	0.6469	13.4386	0.2394	0.4417	0.4025
	70	aiolli	0.1297	1834	0.0091	0.6189	14.3601	0.2189	0.4989	0.4499
	30	classical	0.1038	1834	0.0072	0.6079	11.9933	0.2901	0.3912	0.2876
	50	classical	0.1215	1834	0.0085	0.6479	13.3875	0.2407	0.4442	0.3995
	70	classical	0.1291	1834	0.0091	0.6200	14.3234	0.2199	0.5025	0.4475
ml100k	30	aiolli	0.4443	1289	0.1262	0.3150	130.9487	0.1886	0.6662	0.8202
	50	aiolli	0.4669	1224	0.1302	0.2806	139.3764	0.1481	0.7153	0.8624
	70	aiolli	0.4790	1182	0.1314	0.2575	144.3764	0.1255	0.7491	0.8851
	30	classical	0.4430	1293	0.1260	0.3173	130.4369	0.1907	0.6602	0.8179
	50	classical	0.4665	1227	0.1301	0.2821	139.0493	0.1498	0.7114	0.8607
	70	classical	0.4790	1171	0.1315	0.2583	144.2045	0.1261	0.7485	0.8845

Ο δεύτερος αλγόριθμος αυτής της κατηγορίας είναι ο userKNN. Σε αυτόν τον αλγόριθμο ισχύουν όσα αναφέραμε και στον itemKNN, δηλαδή όσο πιο μεγάλη η τιμή του nn τόσο καλύτερη είναι η ακρίβεια. Αυτό που διαφέρει εδώ είναι πως υπάρχουν πολύ μικρές αποκλίσεις όσον αφορά τις μετρικές στις διαφορετικές τιμές του nn, ενώ γενικότερα έχουμε ελαφρώς χειρότερες τιμές όσον αφορά το popularity bias, το diversity και το coverage. Επομένως, σε όλα τα σύνολα δεδομένων ο userKNN είναι ελαφρώς καλύτερος του itemKNN σε σχεδόν όλους τους τομείς. Εξαίρεση φαίνεται να αποτε-

λεί το Amazon, στο οποίο το popularity bias και το diversity είναι αρκετά καλύτερα στον itemKNN. Ενώ στο συγκεκριμένο σύνολο δεδομένων όπως και πριν καλύπτονται όλα τα διαθέσιμα αντικείμενα ανεξαρτήτως των τιμών των υπερπαραμέτρων που θα θέσουμε.
Με άλλα λόγια, φαίνεται πως στην γενική περίπτωση και όταν το μητρώο χρηστών-αξιολογήσεων δεν είναι πολύ αραιό, είναι πιο αποτελεσματικό να αναζητούμε παρόμοιους χρήστες, αντί για παρόμοια αντικείμενα με αυτά που κάποιος χρήστης έχει αξιολογήσει, ενώ συνήθως εισάγει και λιγότερη μεροληψία. Αυτό είναι αρκετά λογικό μιας που αν κοιτάμε μόνο για παρόμοια αντικείμενα, μπορεί μεν να βρούμε καλές προτάσεις προς τους χρήστες, ωστόσο δεν θα είναι ούτε πρωτότυπες σε σχέση με όσα έχει δει ο χρήστης, ούτε και θα καλύπτουν πολλά αντικείμενα από το long tail, μιας που αυτά εμφανίζονται λίγες φορές. Η διαφορά στα χαρακτηριστικά του Amazon από τα σύνολα του Movielens είναι στην αραιότητα των δεδομένων, δηλαδή εκεί όπου τα αντικείμενα έχουν λάβει λίγες αξιολογήσεις το καθένα και τις περισσότερες φορές οι χρήστες έχουν αξιολογήσει λίγα αντικείμενα. Άρα, φαίνεται πως όταν έχουμε τόση μεγάλη αραιότητα στα δεδομένα μας θα πρέπει να επιλέγεται ο itemKNN.
Συμπέρασμα: και στους δύο neighborhood based αλγορίθμους όσο αυξάνει το nn αυξάνει και η ακρίβεια, όμως αυξάνει και το popularity bias, ενώ μειώνεται το diversity. Μάλιστα αυτό φαίνεται να ισχύει πάντα ανεξαρτήτως των χαρακτηριστικών των δεδομένων. Σε αυτή την οικογένεια αλγορίθμων επομένως είναι υπαρκτό το αντιστάθμισμα ανάμεσα σε popularity bias και ακρίβεια. Στον itemKNN αλγόριθμο η αξιολόγηση που δίνεται σε ένα αντικείμενο, βασίζεται σε αξιολογήσεις που έχουν δοθεί σε παρόμοια αντικείμενα. Αυτό έχει ως αποτέλεσμα να προτείνονται σε έναν χρήστη αντικείμενα που σχετίζονται με όσα αρέσουν συνήθως σε αυτόν ή αντικείμενα που ανήκουν στο ίδιο είδος με αυτά, κάτι που όπως γίνεται αντιληπτό επηρεάζει αρνητικά το novelty και το diversity. Αντίθετα ο userKNN βρίσκει παρόμοιους χρήστες και ενισχύει το diversity για χαμηλό αριθμό από γείτονες.

5.4.1.2 Αλγόριθμοι λανθανόντων παραγόντων

Αλγόριθμος Matrix Factorization (MF)

Πίνακας 5.7: Ανάλυση υπερπαραμέτρων στον αλγόριθμο MF

dataset	lr	bs	f	nDCG	IC	EPC	Gini	ARP	APLT	PREO	PRSP
ml1m	0.001	256	20	0.3100	2245	0.1287	0.2015	829.8737	0.0665	0.7829	0.9396
		256	50	0.3151	2290	0.1283	0.1940	828.3989	0.0634	0.7826	0.9425
		256	70	0.3147	2433	0.1278	0.2036	818.4174	0.0727	0.7557	0.9338
		512	20	0.3186	2162	0.1295	0.1832	856.2911	0.0519	0.8290	0.9533
		512	50	0.3285	2316	0.1336	0.1986	832.1122	0.0644	0.7787	0.9416
		512	70	0.3265	2232	0.1305	0.1809	855.8623	0.0557	0.8093	0.9498
	0.01	256	20	0.1608	2636	0.0705	0.2477	686.4904	0.1393	0.6872	0.8679
		256	50	0.0904	3535	0.0438	0.3994	510.4456	0.2902	0.5724	0.6969
		256	70	0.0602	3668	0.0306	0.5123	406.4007	0.4100	0.4621	0.5342
		512	20	0.2122	2539	0.0894	0.2322	731.0409	0.1167	0.6928	0.8909
		512	50	0.1529	3151	0.0668	0.3101	630.2776	0.1936	0.6090	0.8101
		512	70	0.1239	3531	0.0562	0.3676	576.8589	0.2465	0.5986	0.7498
ml100k	0.001	256	20	0.3736	701	0.1092	0.1896	154.8985	0.0895	0.8323	0.9198
		256	50	0.4013	820	0.1191	0.2272	145.4408	0.1040	0.7858	0.9060
		256	70	0.4012	934	0.1210	0.2445	141.2763	0.1206	0.7315	0.8899
		512	20	0.3649	569	0.1044	0.1643	162.6094	0.0814	0.8558	0.9274
		512	50	0.4015	788	0.1188	0.2169	148.7627	0.0977	0.8022	0.9120
		512	70	0.3993	811	0.1189	0.2177	148.0750	0.1003	0.7932	0.9095
	0.01	256	20	0.3328	1134	0.1026	0.2998	126.2460	0.1891	0.5896	0.8196
		256	50	0.2930	1380	0.0893	0.3529	119.7980	0.2307	0.5614	0.7739
		256	70	0.2703	1525	0.0820	0.3854	114.0224	0.2625	0.5464	0.7372
		512	20	0.3491	1115	0.1076	0.2959	128.9323	0.1824	0.5856	0.8268
		512	50	0.3233	1312	0.0982	0.3378	123.8617	0.2166	0.5355	0.7898
		512	70	0.3016	1430	0.0917	0.3687	119.1233	0.2406	0.5287	0.7627
amazon	0.001	256	20	0.0413	1786	0.0029	0.5404	13.2596	0.2424	0.6493	0.3956
		256	50	0.0582	1811	0.0043	0.5428	14.0761	0.2276	0.6462	0.4296
		256	70	0.0577	1821	0.0042	0.5665	13.6048	0.2359	0.6557	0.4104
		512	20	0.0285	1815	0.0020	0.6000	11.8534	0.2861	0.5418	0.2966
		512	50	0.0393	1826	0.0029	0.6172	12.4528	0.2789	0.5398	0.3126
		512	70	0.0399	1830	0.0029	0.6309	12.2083	0.2793	0.5379	0.3119
	0.01	256	20	0.0481	1824	0.0035	0.6469	12.1286	0.2513	0.5800	0.3752
		256	50	0.0495	1834	0.0037	0.6991	11.4855	0.3004	0.5676	0.2648
		256	70	0.0510	1834	0.0038	0.7011	10.8732	0.3337	0.5294	0.1919
		512	20	0.0451	1822	0.0033	0.6552	11.9743	0.2516	0.5100	0.3744
		512	50	0.0520	1832	0.0040	0.6870	11.9473	0.2697	0.5635	0.3333
		512	70	0.0546	1833	0.0041	0.7026	11.4637	0.2922	0.5372	0.2829

Σε ό,τι έχει να κάνει με την ακρίβεια, αυτή εξαρτάται και από τις τρεις υπερπαραμέτρους, δηλαδή τον ρυθμό μάθησης (lr), τον αριθμό των λανθανόντων παραγόντων (f) και το batch size. Πιο αναλυτικά, σε όλα τα σύνολα δεδομένων χαμηλότερη τιμή του batch size συνεπάγεται χαμηλότερη απόδοση

του αλγορίθμου, ειδικά αν αυτό συνοδεύεται από υψηλότερη τιμή του ρυθμού μάθησης. Αυτό είναι απολύτως λογικό και σύμφωνο με σχετικές έρευνες που έχουν δημοσιευθεί. Πιο συγκεκριμένα, όπως αναφέρεται στο [104] έχει αποδειχθεί ότι πολύ μεγάλες τιμές του batch size οδηγούν σε καλύτερη απόδοση του μοντέλου και πιο ταχείς υπολογισμούς, καθώς αυξάνει τον όγκο των δεδομένων που δίνονται ως είσοδος σε κάθε επανάληψη. Αυτός ο αυξημένος όγκος δεδομένων όμως μπορεί να διανεμηθεί αποτελεσματικά, υλοποιώντας κατ᾽ αυτόν τον τρόπο ένα είδος παραλληλοποίησης. Ωστόσο χρειάζεται ιδιαίτερη προσοχή καθώς ένα πάρα πολύ μεγάλο batch size μπορεί να οδηγήσει σε χαμηλή ακρίβεια λόγω απώλειας της γενικότητας.
Ο αριθμός των λανθανόντων παραγόντων επίσης θα πρέπει να επιλεγεί προσεκτικά, καθώς ρυθμίζει την υπερεκπαίδευση του μοντέλου. Αυτό εξαρτάται από τα χαρακτηριστικά των δεδομένων του εκάστοτε συνόλου δεδομένων και από την τιμή του ρυθμού μάθησης. Στα σύνολα δεδομένων του Movielens για lr=0.001 η ακρίβεια βελτιώνεται όσο αυξάνει ο αριθμός των λανθανόντων παραγόντων έως ότου φτάσει στην τιμή 70 όπου και αρχίζει να γίνεται υπερεκπαίδευση του μοντέλου μας, ενώ η ακρίβεια μειώνεται κατά πολύ όταν αυξάνουμε την τιμή του lr. Για lr=0.001 η υπερεκπαίδευση του μοντέλου γίνεται ορατή για τιμή των λανθανόντων παραγόντων μεγαλύτερη του 20. Από την άλλη στο Amazon, το οποίο χαρακτηρίζεται από πολύ μεγάλη αραιότητα και οι λόγοι αξιολογήσεων προς χρήστες (RPU) και αξιολογήσεων προς αντικείμενα (RPI) είναι πάρα πολύ χαμηλοί σε σχέση με τα άλλα δύο σύνολα δεδομένων, η ακρίβεια αυξάνεται όσο αυξάνεται και ο αριθμός των παραγόντων εκτός της περίπτωσης όπου φαίνεται να έχουμε επιτύχει την μέγιστη ακρίβεια (f=50, bs=256, lr=0.001). Ως εκ τούτου δεν φαίνεται να ισχύει κάποιος γενικός κανόνας.
Στις μετρικές που σχετίζονται με το popularity bias και το diversity, ο ρυθμός μάθησης φαίνεται πως αποτελεί καθοριστικό παράγοντα, ειδικά για χαμηλότερες τιμές του bs (bs=256). Επίσης, όσο μεγαλύτερος ο αριθμός των λανθανόντων παραγόντων τόσο καλύτερα τα αποτελέσματα. Μάλιστα η υπερεκπαίδευση του μοντέλου που παρατηρήθηκε όταν έχουμε 70 παράγοντες, δεν φαίνεται να επηρεάζει ιδιαίτερα. Με άλλα λόγια, το πόσο δίκαιες είναι οι αποφάσεις του συγκεκριμένου αλγορίθμου εξαρτάται πολύ περισσότερο από τον ρυθμό μάθησης και το batch size. Προκειμένου να ενισχύσουμε την επιχειρηματολογία μας ας δούμε ένα παράδειγμα από το σύνολο του Movielens1M. Με ίδιο αριθμό factors (=70), για ρυθμό μάθησης = 0.001 το ARP φτάνει να είναι μέχρι και διπλάσιο σε σχέση με lr = 0.01. Βέβαια αυτό συνοδεύεται από το ανάλογο τίμημα που καλούμαστε να πληρώσουμε, το οποίο δεν είναι άλλο από την ακρίβεια, και μάλιστα είναι αρκετά μεγάλο. Για bs = 256, factors = 70, lr = 0.01, δηλαδή τον συνδυασμό των υπερπαραμέτρων που βρήκαμε πριν, έχουμε nDCG = 0.06, τη στιγμή που το μέγιστο nDCG που έχουμε είναι περίπου 0.33.
Στην κάλυψη των αντικειμένων οι διαφορές ανάμεσα σε διαφορετικές τιμές λανθανόντων παραγόντων και batch size είναι αρκετά πιο ευδιάκριτες. Μεγάλες τιμές του batch size φαίνεται να δίνουν σε αρκετές περιπτώσεις λιγότερα αντικείμενα, χωρίς όμως αυτό να ισχύει πάντα. Αντίθετα ο χαμηλός ρυθμός μάθησης δίνει σε όλες τις περιπτώσεις χαμηλότερες τιμές. Αναφέραμε παραπάνω πως αυξημένος αριθμός παραγόντων δίνει πιο εξατομικευμένες προτάσεις προς τους χρήστες. Αυτό επιβεβαιώνεται και στον αλγόριθμο MF και περισσότερο για bs=256 και lr=0.01. Χαρακτηριστικό είναι ότι για f=70 και μικρό batch size, ο αλγόριθμος καλύπτει σχεδόν όλες τα διαθέσιμα αντικείμενα, ενώ και το popularity bias μειώνεται αρκετά, κάτι το οποίο συμβαίνει σε όλα τα σύνολα δεδομένων. Σε μικρότερα σύνολα δεδομένων με πολύ μεγάλη αραιότητα, όπως το Amazon, ωστόσο δεν ισχύουν όλα όσα αναφέρθηκαν. Πιο συγκεκριμένα, στα σύνολα δεδομένων του Movielens στη γενική περίπτωση το diversity, το coverage και το popularity bias βελτιώνονται όσο αυξάνεται η τιμή του ρυθμού μάθησης, ενώ στο Amazon αυτό ισχύει μόνο για το popularity bias.

Συμπέρασμα: εδώ η ισορροπία απέναντι σε υψηλές τιμές της ακρίβειας και έναν πιο δίκαιο αλγόριθμο στις προβλέψεις του, θα πρέπει να αναζητηθεί στην κατάλληλη ρύθμιση του ρυθμού μάθησης και του batch size, ενώ ταυτόχρονα θα πρέπει να ρυθμίσουμε κατάλληλα την τιμή του αριθμού των λανθανόντων παραγόντων ώστε αφενός να μην γίνει η υπερεκπαίδευση του μοντέλου και αφετέρου να έχουμε πιο εξατομικευμένες συστάσεις προς τους χρήστες. Παράλληλα, ιδιαίτερη προσοχή απαιτείται στα σύνολα δεδομένων με πολύ μεγάλη αραιότητα.

Αλγόριθμος SVD++

Πίνακας 5.8: Ανάλυση υπερπαραμέτρων στον αλγόριθμο SVD++

dataset	lr	bs	f	nDCG	IC	EPC	Gini	ARP	APLT	PREO	PRSP
ml1m	0.001	256	50	0.1097	2528	0.0347	0.1803	475.7883	0.4135	0.6076	0.5291
		256	70	0.1100	2555	0.0355	0.1786	473.9877	0.4048	0.6211	0.5419
		512	50	0.0991	1964	0.0284	0.1155	422.3330	0.5079	0.6161	0.3784
		512	70	0.1070	2130	0.0325	0.1442	450.8142	0.4569	0.6035	0.4624
	0.01	256	50	0.0546	3394	0.0203	0.3635	344.7017	0.5452	0.3942	0.3125
		256	70	0.0418	3441	0.0162	0.3907	305.0175	0.5917	0.3060	0.2246
		512	50	0.0672	3384	0.0248	0.3516	366.7941	0.5079	0.4257	0.3784
		512	70	0.0610	3455	0.0224	0.3636	364.2431	0.5151	0.3857	0.3660
	0.1	256	50	0.0026	2926	0.0023	0.2672	61.8344	0.9053	0.6093	0.6137
		256	70	0.0022	2967	0.0019	0.2517	51.8655	0.9279	0.6222	0.6978
		512	50	0.0038	2157	0.0023	0.1511	62.2276	0.9005	0.1512	0.5962
		512	70	0.0075	3007	0.0058	0.3349	143.2406	0.7195	0.2338	0.0570
ml100k	0.001	256	50	0.2740	283	0.0700	0.0782	196.2112	0.0002	0.9979	0.9999
		256	70	0.2829	290	0.0726	0.0797	191.2422	0.0004	0.9980	0.9997
		512	50	0.2763	278	0.0693	0.0767	190.9509	0.0017	0.9926	0.9986
		512	70	0.2859	276	0.0729	0.0791	194.1394	0.0019	0.9919	0.9984
	0.01	256	50	0.1194	1182	0.0347	0.2652	84.2399	0.4161	0.5087	0.5351
		256	70	0.1249	1420	0.0368	0.3427	88.2081	0.3901	0.4637	0.5726
		512	50	0.1309	1123	0.0369	0.2473	84.6969	0.4238	0.5069	0.5237
		512	70	0.1420	1131	0.0397	0.2542	86.2853	0.4180	0.5070	0.5322
	0.1	256	50	0.0110	1369	0.0067	0.4014	23.7195	0.8648	0.6339	0.4620
		256	70	0.0108	1368	0.0069	0.4181	25.0680	0.8497	0.5926	0.4123
		512	50	0.0111	1413	0.0070	0.4158	26.0127	0.8442	0.5568	0.3944
		512	70	0.0112	1415	0.0068	0.4286	25.8171	0.8468	0.5059	0.4027
amazon	0.001	256	50	0.1229	1129	0.008768	0.1533	24.84	0.0146	0.9368	0.9607
		256	70	0.1241	1297	0.008851	0.1817	23.33	0.0265	0.9081	0.9289
		512	50	0.1205	1297	0.008563	0.1799	23.09	0.0280	0.9013	0.9248
		512	70	0.1232	1481	0.008781	0.2183	21.49	0.0438	0.8716	0.8831
	0.01	256	50	0.1050	1333	0.007443	0.1958	22.50	0.0492	0.9391	0.8691
		256	70	0.1067	1493	0.007640	0.2504	20.56	0.0723	0.8725	0.8092
		512	50	0.1041	1483	0.007444	0.2406	20.08	0.0626	0.8647	0.8343
		512	70	0.1000	1649	0.007229	0.3279	17.45	0.1013	0.7761	0.7351
	0.1	256	50	0.0140	1602	0.001005	0.3295	7.32	0.4658	0.0507	0.0828
		256	70	0.0140	1739	0.001007	0.5147	7.35	0.4641	0.0200	0.0793
		512	50	0.0167	1713	0.001257	0.4850	8.078	0.4216	0.0292	0.0067
		512	70	0.0180	1716	0.001296	0.4881	7.94	0.4424	0.0115	0.0356

Ο SVD++ έχει χειρότερη απόδοση από τον MF όσον αφορά την ακρίβεια, κάτι που επιβεβαιώνεται και από τις μετρικές μας. Για όλα τα σύνολα δεδομένων ισχύει ότι όσο πιο χαμηλή είναι η τιμή του ρυθμού μάθησης, τόσο καλύτερη η ακρίβεια. Όσον αφορά το batch size, αυτό επηρεάζει ελάχιστα την ακρίβεια. Εν αντιθέσει με την ακρίβεια όμως, μια χαμηλή τιμή του ρυθμού μάθησης έχει ως συνέπεια την κάλυψη λιγότερων αντικειμένων από όσα είναι διαθέσιμα σε ένα σύνολο δεδομέ-

νων. Χαρακτηριστικό είναι πως στο Movielens 1M για lr=0.001, bs=512 και factors=50 ο αλγόριθμος καλύπτει 1964 αντικείμενα, ενώ για lr=0.01 και διατηρώντας τις τιμές των υπόλοιπων υπερπαραμέτρων ίδιες, ο αλγόριθμος καλύπτει 3394 αντικείμενα! Αυτό που παρατηρούμε από τον πίνακα είναι πως αν η τιμή του ρυθμού μάθησης δεν είναι πολύ χαμηλή τότε προτείνονται σχεδόν όλα τα διαθέσιμα αντικείμενα του συνόλου δεδομένων. Γενικότερα για lr = 0.01 έχουμε τις πιο καλές τιμές. Όσον αφορά την υπερπαράμετρο bs, η τιμή 256 είναι η καλύτερη, κάτι που είναι αρκετά ευδιάκριτο όταν έχουμε χαμηλό ρυθμό μάθησης και η διαφορά είναι περίπου 400 αντικείμενα στη μια περίπτωση και περίπου 500 στην άλλη. Τέλος, ο αριθμός των λανθανόντων παραγόντων δεν επηρεάζει πολύ τα αποτελέσματα με την τιμή 70 να δίνει ελαφρώς καλύτερα αποτελέσματα.
Παρόμοια είναι η κατάσταση και στις μετρικές που σχετίζονται με το popularity bias. Στη μετρική ARP οι διαφορές δεν είναι πολύ μεγάλες στο Movielens1M, όπου όλες οι τιμές βρίσκονται στο εύρος [305-473], όμως δεν ισχύει το ίδιο και στα υπόλοιπα σύνολα δεδομένων. Στον πίνακα 5.8, παρατηρούμε ότι στο ml100k η τιμή πέφτει στο μισό, και άρα βελτιώνεται καθώς αυξάνεται ο ρυθμός μάθησης, ενώ και σε αυτή την περίπτωση για lr = 0.01 και bs = 256 έχουμε τις πιο καλές τιμές. Παρόμοια συμπεράσματα προκύπτουν και για το Amazon. Αυτό γίνεται ακόμη πιο αντιληπτό και από τις μετρικές APLT, PopRSP, PopREO. Τέλος, ο αριθμός των λανθανόντων παραγόντων δεν επηρεάζει πολύ τα αποτελέσματα με την τιμή 70 να δίνει ελαφρώς καλύτερα αποτελέσματα. Στη μετρική APLT τα πράγματα διαφοροποιούνται ελαφρώς, για bs=512, f=50 η τιμή παραμένει η ίδια ανεξαρτήτως της τιμής του learning rate και μειώνεται για f=70 και lr=0.001, για bs=256 έχουμε τα αναμενόμενα αποτελέσματα (πτώση κατά 0.15 με 0.2 περίπου για lr=0.001).
Συμπέρασμα: Από όλα τα παραπάνω, καταλαβαίνουμε πως και σε αυτή την περίπτωση η ισορροπία απέναντι σε υψηλές τιμές της ακρίβειας και έναν πιο δίκαιο αλγόριθμο στις προβλέψεις του, θα πρέπει να αναζητηθεί στην κατάλληλη ρύθμιση του ρυθμού μάθησης (κυρίως) και του batch size. Η τιμή του ρυθμού μάθησης που δίνει την καλύτερη ακρίβεια σε αυτόν τον αλγόριθμο εισάγει πάρα πολύ μεγάλη μεροληψία στα αποτελέσματα και η βέλτιστη τιμή θα πρέπει να αναζητηθεί στο εύρος [0.001, 0.01].

Αλγόριθμος BPRMF

Πίνακας 5.9: Ανάλυση υπερπαραμέτρων στον αλγόριθμο BPRMF

dataset	f	lr	bias_reg	nDCG	IC	EPC	Gini	ARP	APLT	PopREO	PopRSP
ml1m	10	0.001	0.01	0.2251	369	0.0761	0.0364	1295.3387	0.0000	1.0000	1.0000
	10	0.001	0.5	0.2189	454	0.0748	0.0391	1284.8317	0.0000	1.0000	1.0000
	50	0.001	0.01	0.2237	387	0.0759	0.0373	1292.7072	0.0000	1.0000	1.0000
	50	0.001	0.5	0.2041	926	0.0716	0.0494	1244.9211	0.0002	0.9998	0.9998
	70	0.001	0.01	0.2232	408	0.0756	0.0377	1291.5389	0.0000	1.0000	1.0000
	70	0.001	0.5	0.1973	1304	0.0693	0.0553	1224.0925	0.0008	0.9982	0.9993
	10	0.01	0.01	0.2273	374	0.0773	0.0371	1292.7171	0.0000	1.0000	1.0000
	10	0.01	0.5	0.2587	1170	0.0877	0.0449	1257.5215	0.0011	0.9986	0.9991
	50	0.01	0.01	0.2474	454	0.0839	0.0414	1277.8439	0.0000	1.0000	1.0000
	50	0.01	0.5	0.3077	1321	0.1057	0.0621	1183.8039	0.0020	0.9899	0.9982
	70	0.01	0.01	0.2537	481	0.0855	0.0428	1273.1544	0.0000	1.0000	1.0000
	70	0.01	0.5	0.3143	1363	0.1081	0.0659	1169.6796	0.0026	0.9858	0.9978
ml100k	20	0.001	0.1	0.2724	293	0.0694	0.0791	197.6606	0.0001	0.9990	0.9999
	20	0.001	0.5	0.2653	427	0.0681	0.0895	194.0480	0.0019	0.9926	0.9984
	50	0.001	0.1	0.2715	315	0.0690	0.0818	197.1850	0.0002	0.9979	0.9998
	50	0.001	0.5	0.2532	605	0.0655	0.1010	190.2631	0.0059	0.9848	0.9950
	70	0.001	0.1	0.2680	315	0.0691	0.0831	197.0399	0.0003	0.9990	0.9998
	70	0.001	0.5	0.2453	802	0.0645	0.1115	187.1938	0.0122	0.9834	0.9896
	20	0.01	0.1	0.3836	485	0.1054	0.1490	166.0932	0.0626	0.8691	0.9448
	20	0.01	0.5	0.3901	595	0.1060	0.1432	171.4156	0.0643	0.8497	0.9432
	50	0.01	0.1	0.3961	549	0.1098	0.1658	162.5153	0.0605	0.8724	0.9467
	50	0.01	0.5	0.4067	631	0.1106	0.1540	169.5547	0.0614	0.8566	0.9459
	70	0.01	0.1	0.4018	587	0.1124	0.1728	160.6813	0.0546	0.8808	0.9521
	70	0.01	0.5	0.4125	647	0.1131	0.1562	169.5984	0.0532	0.8648	0.9533
amazon	20	0.001	0.1	0.0664	860	0.0047	0.0727	33.6145	0.0013	1.0000	0.9964
	20	0.001	0.5	0.0656	1049	0.0046	0.0820	33.0403	0.0032	1.0000	0.9915
	50	0.001	0.1	0.0629	1729	0.0045	0.1382	31.1811	0.0227	0.9901	0.9390
	50	0.001	0.5	0.0594	1786	0.0042	0.1721	29.9246	0.0351	0.9847	0.9061
	70	0.001	0.1	0.0610	1817	0.0043	0.1915	29.7195	0.0436	0.9543	0.8837
	70	0.001	0.5	0.0558	1829	0.0040	0.2359	28.1845	0.0609	0.9406	0.8386
	20	0.01	0.1	0.0686	184	0.0049	0.0562	34.7828	0.0000	1.0000	1.0000
	20	0.01	0.5	0.0656	405	0.0047	0.0631	33.8204	0.0000	1.0000	0.9999
	50	0.01	0.1	0.0675	285	0.0048	0.0589	34.4908	0.0000	1.0000	1.0000
	50	0.01	0.5	0.0625	1022	0.0044	0.0834	32.5422	0.0025	0.9953	0.9933
	70	0.01	0.1	0.0676	353	0.0048	0.0614	34.2429	0.0000	1.0000	1.0000
	70	0.01	0.5	0.0605	828	0.0043	0.0794	32.7178	0.0006	1.0000	0.9983

Τα αποτελέσματα που προέκυψαν από τις μετρικές ακρίβειας είναι αρκετά αναμενόμενα με βάση όσα αναφέρονται στη βιβλιογραφία. Πιο αναλυτικά, μεγαλύτερος ρυθμός μάθησης, μεγαλύτερη τιμή της παραμέτρου λ και μεγαλύτερος αριθμός λανθανόντων παραγόντων συνεπάγεται καλύτερη ακρίβεια, καθώς αποτελεί τον βέλτιστο συνδυασμό απόκτησης πληροφοριών του αλγορίθμου, που

θα τον βοηθήσουν να μάθει καλύτερα και να αποκτήσει καλύτερη γενίκευση. Ωστόσο στο Amazon ισχύουν τα τελείως αντίθετα από όσα αναφέρθηκαν, κάτι που αποτελεί μια σοβαρή ένδειξη πως τα χαρακτηριστικά των συνόλων δεδομένων και κυρίως η αραιότητα επηρεάζουν τα αποτελέσματα. Βέβαια κάτι που ισχύει σε όλα τα σύνολα δεδομένων είναι πως εάν ο ρυθμός μάθησης είναι αρκετά χαμηλός τότε η ακρίβεια είναι αντιστρόφως ανάλογη του αριθμού των λανθανόντων παραγόντων.
Παρόλο όμως που ο BPRMF έχει σχετικά καλή ακρίβεια, δεν συμβαίνει το ίδιο με το diversity, το novelty και το coverage. Σε ότι έχει να κάνει με το coverage, καταλυτικός φαίνεται πως είναι ο ρόλος της παραμέτρου λ, η αύξηση της οποίας οδηγεί σε έως και περίπου 3.5 φορές, όπως παρατηρούμε στο Movielens1M, περισσότερα αντικείμενα να καλύπτονται από τον αλγόριθμο. Αυτό είναι αρκετά λογικό καθώς αν θυμηθούμε τον τύπο της συνάρτησης ελαχιστοποίησης (5.10), στο BPR-OPT το $p(\theta)$ είναι μια κανονική κατανομή με μητρώο διασποράς-συνδιασποράς, όπου είναι όλα τα αντικείμενα του συνόλου δοκιμής. Σε γενικές γραμμές βέβαια, το coverage θεωρείται αρκετά χαμηλό. Σε όλες τις περιπτώσεις η αύξηση του lr κάνει πιο αισθητή τη διαφορά στις τιμές των μετρικών μόνο εάν αυτό συνοδεύεται από μια σχετικά μεγάλη τιμή του λ.
Προηγουμένως είδαμε ότι η σύγκριση των αντικειμένων γίνεται ανά ζεύγη. Σε ένα σύνολο δεδομένων όμως ο αριθμός των ζευγών μπορεί να είναι πολύ μεγάλος. Για τον λόγο αυτό τα ζεύγη δειγματοληπτούνται ομοιόμορφα με τυχαίο τρόπο και είναι αρκετά απίθανο να βρεθούνε rank inversions σε συχνή βάση εκτός αν πρόκειται για πολύ δημοφιλή αντικείμενα, αυξάνοντας αρκετά τις πιθανότητες να επιλεγούν δημοφιλή αντικείμενα μιας που κατατάσσονται πιο ψηλά από τα υπόλοιπα τις περισσότερες φορές. Εξαιτίας αυτού στον BPRMF, το φαινόμενο του popularity bias είναι πολύ πιο εμφανές από όσους αλγορίθμους έχουμε δει έως τώρα. Χαρακτηριστικά αναφέρουμε πως η τιμή της μετρικής APLT είναι στο διάστημα [0, 0.0025] στο Movielens1M και στο [0, 0.0643] στα άλλα δύο σύνολα δεδομένων. Το popularity bias μπορεί να μειωθεί ελαφρώς μέσω κατάλληλης ρύθμισης της παραμέτρου κανονικοποίησης λ (Regularization coefficient for the bias), ενώ σε αυτή την κατεύθυνση συμβάλει και ο ρυθμός μάθησης. Μεγάλη τιμή του ρυθμού μάθησης σε συνδυασμό με μεγάλη τιμή του bias_reg και του αριθμού των λανθανόντων παραγόντων μπορεί να βελτιώσει ελαφρώς το popularity bias. Μάλιστα αυτό ισχύει για όλα τα σύνολα δεδομένων που εξετάσαμε.
Συμπέρασμα: Το συμπέρασμα που προκύπτει εδώ είναι αρχικά πως σε αυτόν τον αλγόριθμο η εισαγωγή μεροληψίας φτάνει στον μέγιστο βαθμό σύμφωνα με τις μετρικές μας Επιπροσθέτως, το αντιστάθμισμα ανάμεσα στην ακρίβεια και στην μεροληψία, εξαρτάται από την παράμετρο λ και το f, και όχι από τον ρυθμό μάθησης. Τέλος, αυτό που προκαλεί ιδιαίτερη εντύπωση, είναι ότι, στη γενική περίπτωση, εν αντιθέσει με όσα έχουμε δει έως τώρα βελτίωση της ακρίβειας συνεπάγεται βελτίωση και όχι επιδείνωση του popularity bias.

Αλγόριθμος WRMF

Στον αλγόριθμο αυτόν οι συνδυασμοί των διαφορετικών τιμών των υπερπαραμέτρων που εξετάστηκαν ήταν πάρα πολλοί, επομένως η παρουσίαση τους σε έναν πίνακα, δεν είναι εφικτή εδώ. Αντ'αυτού στο παράρτημα μπορείτε να δείτε αναλυτικά γραφήματα για την συμπεριφορά των υπερπαραμέτρων στον αλγόριθμο WRMF, κρατώντας σταθερή μόνο την τιμή του αριθμού των λανθανόντων παραγόντων ($f = 50$).

Στον WRMF, όσο μεγαλύτερο το λ τόσο καλύτερη η ακρίβεια. Σχετικά με το α, στη γενική περίπτωση η συμπεριφορά του είναι απολύτως αναμενόμενη από όσα γνωρίζουμε από τη θεωρία, όσο λιγότερο σίγουροι είμαστε για το τι πραγματικά ενδιαφέρει τους χρήστες, τόσο χειρότερη ακρίβεια έχουμε. Η παράμετρος αυτή είναι άμεσα εξαρτώμενη από τα χαρακτηριστικά των δεδομένων και χρειάζεται πάντα να δοκιμάσουμε αρκετές τιμές, προκειμένου να βρούμε εκείνη που ταιριάζει καλύτερα στα δεδομένα μας. Το ίδιο ισχύει και για την παράμετρο λ, η οποία μας βοηθάει να αντιμετωπίσουμε την υπερεκπαίδευση του μοντέλου, για αυτό και στις περισσότερες περιπτώσεις, όσο πιο υψηλή η τιμή της, τόσο καλύτερη η ακρίβεια. Το κοινό χαρακτηριστικό είναι πως για $\alpha = 20$ έχουμε πολύ υψηλή ακρίβεια και για $\alpha = 70$ αρκετά χαμηλή. Ένα πολύ χαμηλό επίπεδο confidence όπως αυτό που έχουμε εάν θέσουμε το α ίσο με 0.1, φαίνεται πως εξαρτάται από τα δεδομένα και τα χαρακτηριστικά τους, το πόσο θα επηρεάσει τα χαρακτηριστικά. Αναφέρουμε χαρακτηριστικά, πως στο ml1m έχει την 2η καλύτερη απόδοση, ενώ στο ml100k την 2η χειρότερη! Ανατρέχοντας στον πίνακα 5.2, παρατηρούμε ότι η διαφορά στην αραιότητα των δεδομένων των δύο συνόλων είναι αρκετά μικρή και δεν μπορεί να δικαιολογήσει αυτή τη συμπεριφορά της παραμέτρου α. Η μεγαλύτερη διαφορά τους είναι ο ότι στο ml100k τα αντικείμενα είναι περισσότερα από τους χρήστες, ενώ στο ml1m ισχύει το αντίθετο. Άρα εξαρτάται άμεσα από τα δεδομένα, απλά για βέλτιστα αποτελέσματα θα πρέπει να αποφεύγεται να τίθεται σε ακραίες τιμές (πολύ χαμηλές ή πολύ υψηλές). Ακόμη θα πρέπει να υπενθυμίσουμε πως ο συγκεκριμένος αλγόριθμος *δεν* υποθέτει πως εάν ένας χρήστης δεν έχει αλληλεπιδράσει με ένα αντικείμενο τότε δεν του αρέσει το συγκεκριμένο αντικείμενο. Απεναντίας, υποθέτει πως υπάρχει αρνητική προτίμηση σε αυτό και ο βαθμός του confidence για αυτή την υπόθεση καθορίζεται από την παράμετρο c της εξίσωσης (5.11).

Τα συμπεράσματα που προκύπτουν για το diversity και την κάλυψη των αντικειμένων είναι πως μεγαλύτερη τιμή του α και μεγαλύτερη τιμή του reg σε συνδυασμό με τη βέλτιστη τιμή των factors πριν γίνει η υπερεκπαίδευση, συνεπάγονται καλύτερο diversity και coverage. Ωστόσο, στην κάλυψη των αντικειμένων στις πολύ μικρές τιμές του reg, μπορεί να συμβαίνει και το αντίθετο δηλαδή πολύ μικρή τιμή του α να έχει καλύτερα αποτελέσματα. Στο diversity, εν αντιθέσει με το coverage για $\alpha = 0.1$ η τιμή του είναι σχεδόν μηδενική για $\lambda \leq 0.1$ για όλα τα f, ενώ για $\lambda \geq 0.5$ η τιμή για $f = 0.5$ ξεχωρίζει από αυτές των υπόλοιπων τιμών, αν και γενικά είναι αρκετά χαμηλή. Για α=1, δεν υπάρχει σχεδόν καμία διαφορά για $\lambda \leq 0.1$, ενώ για $\lambda \geq 0.5$ το Gini index παραμένει μηδενικό για $f = 70$, ενώ το καλύτερο αποτέλεσμα είναι για $f = 50$ και $\lambda = 0.5$. Η μηδενική τιμή του $f = 70$ πιθανόν έχει ως αιτιολογία την υπερεκπαίδευση του μοντέλου, κατά την οποία το μοντέλο μαθαίνει ακριβώς τις προτιμήσεις του χρήστη, όμως δεν έχει τη δυνατότητα να του προτείνει είδη ταινιών για τα οποία δεν έχει εκδηλώσει κάποια προτίμηση. Αυτό μπορεί να δικαιολογήσει και την χαμηλή τιμή του EPC, και το φαινόμενο του popularity bias, που παρουσιάζεται αρκετά οξυμένο. Μια ενδιαφέρουσα παρατήρηση εδώ είναι πως, ενώ η υπερεκπαίδευση του μοντέλου για $f = 70$, δεν επηρεάζει πάρα πολύ την ακρίβεια του μοντέλου, φαίνεται πως ενισχύει σε πάρα πολύ μεγάλο βαθμό το φαινόμενο του popularity bias, μειώνει το novelty και σχεδόν μηδενίζει το diversity, ανεξαρτήτως των τιμών του λ και του α.

Αλγόριθμος Slim

Στον αλγόριθμο SLIM θα εξετάσουμε αρχικά τη συμπεριφορά του στα σύνολα δεδομένων του Movielens, όπου προκύπτουν παρόμοια συμπεράσματα και στη συνέχεια στο Amazon, στο οποίο απαιτείται περισσότερη ανάλυση.

Ας δούμε πρώτα πως επηρεάζουν οι διάφορες τιμές της υπερπαραμέτρου l_1 τα αποτελέσματα. Για $l_1 = 1$ έχουμε την χειρότερη απόδοση, καθώς σε αυτή την περίπτωση ο όρος της εξίσωσης (5.16) που αντιστοιχεί στο L2, εξαρτάται μόνο από τη σταθερά β και συνεπώς το μοντέλο έχει υπερεκπαιδευτεί. Όπως έχουμε προαναφέρει για $l_1 = 0$ η ποινή είναι η L_2 και για $l_1 = 1$ η ποινή είναι η L_1. Επίσης, σε κάθε περίπτωση εκτός από όταν $l_1 = 1$, το α θα πρέπει να είναι μεγαλύτερο από 0.1 αν θέλουμε να έχουμε καλή ακρίβεια. Γνωρίζουμε από τη θεωρία ότι όσο μεγαλύτερη είναι η τιμή της l1 νόρμας, τόσο πιο αραιό είναι το μητρώο W. Το κοινό σε όλα τα σύνολα δεδομένων είναι ότι για $\alpha > 0.001$, έχουμε αρκετά αυξημένο το popularity bias, η τιμή του Gini index πέφτει κατακόρυφα, μέχρι να μηδενιστεί για α = 1, ενώ παρόμοια είναι η εικόνα και στον αριθμό των αντικειμένων που καλύπτει ο αλγόριθμος. Μια εξήγηση σε αυτό θα μπορούσε να είναι η υπερεκπαίδευση του μοντέλου σε συνδυασμό με την αραιότητα του W, η οποία αυξάνει όσο αυξάνει η τιμή του α. Σε αυτόν τον αλγόριθμο ωστόσο βλέπουμε κάτι που δεν είδαμε σε κανέναν άλλον αλγόριθμο που εξετάσαμε. Στο Amazon η υπερπαράμετρος l_1 αυξάνει κατά πολύ την ακρίβεια, την κάλυψη των αντικειμένων, το diversity (τα οποία σχεδόν φτάνει ακόμη και να είναι 6 φορές μεγαλύτερη) και μειώνει ακόμη και την ελάχιστη μεροληψία που υπάρχει όταν είναι ίση με 0.1. Όμως για κάθε άλλη τιμή της τα αποτελέσματα παραμένουν αμετάβλητα. Η υπερπαράμετρος alpha εδώ επηρεάζει ελάχιστα τα αποτελέσματα, εκτός από όταν έχουμε μια πάρα πολύ μικρή τιμή, όπου για αυξημένη τιμή του l1 δεν έχουμε τόσο χαμηλή ακρίβεια όπως στις υπόλοιπες τιμές. Συνεπώς, ο SLIM έχει αρκετά περίεργη συμπεριφορά στο λιγότερο πυκνό σύνολο δεδομένων που δοκιμάστηκε. Μάλιστα με βάση όσα γνωρίζουμε για την αρχιτεκτονική του και από τον πολύ χαμηλό λόγο των αξιολογήσεων προς αντικείμενα (rpi). Από τα παραπάνω καταλήγουμε στο ότι το l_1 θα πρέπει να έχει μια τιμή λίγο μεγαλύτερη από 0.5, καθώς θα μας δώσει ένα μοντέλο το οποίο δεν θα εισάγει αρκετή μεροληψία, θα καλύπτει αρκετά από τα διαθέσιμα αντικείμενα, θα έχει πολύ υψηλό diversity, χωρίς να έχουμε σημαντικές απώλειες στην ακρίβεια.

Συμπέρασμα: Η απόφαση που θα πρέπει να λάβουμε εδώ και η οποία θα καθορίσει το αντιστάθμισμα ανάμεσα στη μεροληψία και στην ακρίβεια, αφορά την επιλογή της τιμής του α, καθώς υψηλή τιμή του α ($\alpha \geq 0.1$) μας παρέχει μεν την καλύτερη απόδοση που έχουμε δει από όλους τους αλγορίθμους έως τώρα, ωστόσο έχει πολύ μεγάλες επιπτώσεις στην μεροληψία που εισάγει ο SLIM. Από τον πίνακα 5.10 που παρουσιάζεται, προκύπτει πως μια τιμή του α ανάμεσα σε 0.01 και 0.1 αποτελεί τη βέλτιστη επιλογή.

Πίνακας 5.10: Ανάλυση υπερπαραμέτρων στον αλγόριθμο Slim

dataset	l1	alpha	nDCG	IC	EPC	Gini	ARP	APLT	PREO	PRSP
MLlM	0.1	0.0001	0.1307	3650	0.0479	0.5996	352.9123	0.5548	0.0686	0.2950
	0.4462	0.0001	0.1080	3470	0.0419	0.3559	375.5098	0.4855	0.2479	0.4162
	1.0	0.0001	0.0912	2630	0.0373	0.2890	373.5911	0.4722	0.1481	0.4379
	0.1	0.001	0.1056	3675	0.0382	0.7076	233.2001	0.6795	0.2351	0.0383
	0.4462	0.001	0.0564	3391	0.0222	0.4017	173.7575	0.7740	0.1423	0.1989
	1.0	0.001	0.0764	1913	0.0367	0.2184	315.5139	0.4964	0.2474	0.3979
	0.1	0.01	0.1021	3650	0.0432	0.7246	210.5273	0.6958	0.3219	0.0004
	0.4462	0.01	0.0361	3157	0.0189	0.3889	125.1206	0.8337	0.5980	0.3730
	1.0	0.01	0.0755	1042	0.0353	0.1232	307.7161	0.5658	0.5329	0.2743
	0.1	0.1	0.1908	3146	0.0873	0.5304	333.9968	0.4811	0.0821	0.4234
	0.4462	0.1	0.0673	2160	0.0423	0.2459	327.3186	0.3945	0.0004	0.5568
	1.0	0.1	0.0211	590	0.0147	0.0514	295.4697	0.5269	0.0651	0.3454
	0.1	1.0	0.3340	1829	0.1325	0.2150	717.3738	0.0463	0.8549	0.9584
	0.4462	1.0	0.1219	736	0.0647	0.0798	729.0053	0.1180	0.9759	0.8895
	1.0	1.0	0.0313	412	0.0159	0.0318	339.3944	0.4640	0.4252	0.4512
MLlook	0.1	0.0001	0.0226	1488	0.0129	0.1917	36.8668	0.7458	0.5919	0.1098
	0.4462	0.0001	0.0241	1493	0.0134	0.1993	36.2885	0.7502	0.5948	0.1213
	1.0	0.0001	0.0218	1564	0.0125	0.2215	33.6496	0.7712	0.6404	0.1778
	0.1	0.001	0.0256	1522	0.0140	0.2016	37.1106	0.7444	0.5765	0.1062
	0.4462	0.001	0.0265	1597	0.0140	0.2266	35.1715	0.7604	0.6233	0.1484
	1.0	0.001	0.0262	1611	0.0142	0.2814	30.7331	0.7943	0.6505	0.2428
	0.1	0.01	0.0593	1648	0.0251	0.2965	38.0234	0.7359	0.4307	0.0843
	0.4462	0.01	0.1223	1655	0.0426	0.4366	39.6579	0.7341	0.2099	0.0796
	1.0	0.01	0.2039	1596	0.0696	0.6060	53.6996	0.6396	0.0983	0.1401
	0.1	0.1	0.3384	1594	0.1029	0.5323	86.3270	0.4290	0.3572	0.5160
	0.4462	0.1	0.4292	1154	0.1248	0.3557	114.6668	0.2337	0.4887	0.7705
	1.0	0.1	0.4568	916	0.1307	0.2876	129.5690	0.1309	0.6769	0.8797
	0.1	1.0	0.4667	880	0.1313	0.2651	135.1113	0.1125	0.7185	0.8978
	0.4462	1.0	0.4418	592	0.1180	0.1801	162.3961	0.0357	0.9956	0.9690
	1.0	1.0	0.3762	512	0.0984	0.1475	165.0419	0.1343	0.9895	0.8763
Amazon	0.1	0.0001	0.0606	1813	0.0045	0.3119	8.4835	0.4287	0.1577	0.0078
	0.4462	0.0001	0.0271	641	0.0021	0.0625	8.0104	0.4524	0.0465	0.0559
	1.0	0.0001	0.0262	690	0.0019	0.0634	8.2022	0.4365	0.1157	0.0238
	0.1	0.001	0.0605	1807	0.0045	0.3068	8.3588	0.4322	0.1352	0.0149
	0.4462	0.001	0.0203	360	0.0015	0.0562	7.9628	0.4555	0.1857	0.0620
	1.0	0.001	0.0135	163	0.0010	0.0543	8.1593	0.4457	0.0050	0.0423
	0.1	0.01	0.0635	1785	0.0047	0.3169	8.2326	0.4334	0.1174	0.0173
	0.4462	0.01	0.0143	165	0.0011	0.0543	8.1394	0.4470	0.0516	0.0450
	1.0	0.01	0.0133	121	0.0010	0.0542	8.1681	0.4452	0.0050	0.0413
	0.1	0.1	0.0504	481	0.0037	0.0597	8.3326	0.4330	0.3519	0.0166
	0.4462	0.1	0.0133	121	0.0010	0.0542	8.1681	0.4452	0.0050	0.0413
	1.0	0.1	0.0133	121	0.0010	0.0542	8.1681	0.4452	0.0050	0.0413
	0.1	1.0	0.0133	121	0.0010	0.0542	8.1681	0.4452	0.0050	0.0413
	0.4462	1.0	0.0133	121	0.0010	0.0542	8.1681	0.4452	0.0050	0.0413
	1.0	1.0	0.0133	121	0.0010	0.0542	8.1681	0.4452	0.0050	0.0413

5.4.1.3 Αλγόριθμοι που βασίζονται στα τεχνητά νευρωνικά δίκτυα

Αλγόριθμος DeepFM

Πίνακας 5.11: Ανάλυση υπερπαραμέτρων στον αλγόριθμο DeepfM

dataset	f	lr	nDCG	IC	EPC	Gini	ARP	APLT	PREO	PRSP
ml1m	30	0.0001	0.2291	708	0.0776	0.0365	1298.92	0.0005	0.9995	0.9995
	50	0.0001	0.2289	666	0.0778	0.0366	1297.99	0.0005	0.9998	0.9996
	70	0.0001	0.2282	637	0.0775	0.0364	1298.16	0.0004	0.9989	0.9996
	100	0.0001	0.2362	777	0.0804	0.0373	1296.50	0.0005	0.9984	0.9996
	30	0.001	0.3588	2313	0.1377	0.1646	914.05	0.0506	0.7528	0.9545
	50	0.001	0.3724	2557	0.1442	0.1885	874.03	0.0656	0.6895	0.9405
	70	0.001	0.3763	2702	0.1469	0.2026	854.55	0.0748	0.6688	0.9317
	100	0.001	0.3770	2798	0.1476	0.2118	840.50	0.0820	0.6508	0.9249
	30	0.01	0.2542	3356	0.1019	0.3026	698.21	0.1839	0.5352	0.8207
	50	0.01	0.2334	3455	0.0941	0.3305	664.70	0.2095	0.4936	0.7925
	70	0.01	0.2191	3510	0.0880	0.3622	629.70	0.2357	0.4489	0.7625
	100	0.01	0.2087	3469	0.0853	0.3763	604.92	0.2464	0.4149	0.7500
ml100	30	0.0001	0.0164	467	0.0084	0.1097	32.79	0.7635	0.2058	0.1568
	50	0.0001	0.0506	761	0.0153	0.1511	49.18	0.6913	0.0652	0.0247
	70	0.0001	0.0653	832	0.0194	0.1624	65.47	0.5766	0.2863	0.2668
	100	0.0001	0.0802	943	0.0220	0.1966	69.37	0.5678	0.3075	0.2835
	30	0.001	0.3799	549	0.1034	0.1235	184.45	0.0292	0.9091	0.9748
	50	0.001	0.3957	599	0.1087	0.1411	177.04	0.0376	0.8857	0.9673
	70	0.001	0.3991	650	0.1113	0.1584	170.12	0.0563	0.8576	0.9506
	100	0.001	0.4095	695	0.1141	0.1685	166.33	0.0653	0.8369	0.9423
	30	0.01	0.4039	1169	0.1177	0.2779	135.81	0.1605	0.6052	0.8497
	50	0.01	0.3979	1252	0.1161	0.2905	134.22	0.1685	0.5714	0.8415
	70	0.01	0.3878	1268	0.1141	0.3018	132.84	0.1771	0.5419	0.8324
	100	0.01	0.3836	1305	0.1115	0.2940	133.95	0.1662	0.5517	0.8438
amazon	30	0.0001	0.0123	373	0.0009	0.0810	7.8900	0.3960	0.0065	0.0596
	50	0.0001	0.0117	454	0.0009	0.0904	8.1093	0.4047	0.0005	0.0414
	70	0.0001	0.0131	487	0.0009	0.0963	8.0021	0.4083	0.0196	0.0340
	100	0.0001	0.0118	557	0.0008	0.1066	7.5489	0.4117	0.1363	0.0271
	30	0.001	0.0154	1230	0.0011	0.1660	8.6061	0.4075	0.2077	0.0356
	50	0.001	0.0263	1450	0.0019	0.2336	10.3587	0.3680	0.3407	0.1184
	70	0.001	0.0293	1577	0.0020	0.2738	11.9005	0.3018	0.5414	0.2618
	100	0.001	0.0364	1670	0.0027	0.3225	12.5061	0.3098	0.4843	0.2441
	30	0.01	0.0623	1816	0.0046	0.5544	14.3751	0.1930	0.6919	0.5109
	50	0.01	0.0715	1831	0.0052	0.5882	14.0710	0.2080	0.6408	0.4754
	70	0.01	0.0662	1833	0.0049	0.6250	13.5796	0.2265	0.6782	0.4322
	100	0.01	0.0720	1834	0.0053	0.6282	13.7071	0.2328	0.6195	0.4178

Σε όλα τα σύνολα δεδομένων για ρυθμό μάθησης ίσο με 0.001, όσο μεγαλύτερος είναι ο αριθμός των λανθανόντων παραγόντων, τόσο καλύτερη είναι η ακρίβεια. Η επιρροή του ρυθμού μάθησης εξαρτάται κατά πολύ από τα χαρακτηριστικά των δεδομένων. Αρχικά σε όλα τα σύνολα δεδομένων

η χαμηλότερη τιμή του ρυθμού μάθησης που δοκιμάσαμε, μας δίνει πολύ μικρή ακρίβεια, παρόλο που έχει την καλύτερη επίδοση όσον αφορά το popularity bias και το diversity, συνεπώς όπως γίνεται κατανοητό δεν μπορεί να χρησιμοποιηθεί. Στα σύνολα δεδομένων του Movielens μεγαλύτερη ακρίβεια έχουμε για ρυθμό μάθησης ίσο με 0.001, ενώ λιγότερο popularity bias εισάγεται για ρυθμό μάθησης ίσο με 0.01. Θα πρέπει να σημειωθεί πως στο Movielens100k δεν λάβαμε υπόψη την περίπτωση που έχουμε ρυθμό μάθησης ίσο με 0.0001, διότι παρόλο που έχουμε την λιγότερη εισαγωγή μεροληψίας, η ακρίβεια είναι πάρα πολύ χαμηλή, όπως και το diversity, το novelty αλλά και η κάλυψη των αντικειμένων. Επίσης, στη γενική περίπτωση για όλα τα σύνολα δεδομένων ισχύει ότι όσο μεγαλύτερος είναι ο ρυθμός μάθησης τόσο καλύτερο το diversity. Συμπληρωματικά, για ρυθμό μάθησης ≥ 0.001 ο αλγόριθμος καλύπτει πάρα πολλά αντικείμενα στο Movielens1M και στο Amazon, όπως και στο Movielens100k όπου μπορεί η επίδοση να είναι χειρότερη για ρυθμό μάθησης ίσο με 0.001 χωρίς ωστόσο να είναι ιδιαίτερα κακή. Από όλα όσα προαναφέρθηκαν, μπορούμε να συμπεράνουμε ότι η βέλτιστη τιμή του ρυθμού μάθησης βρίσκεται στο διάστημα [0.01, 0.1]. Όσον αφορά μια άλλη υπερπαράμετρο, τον αριθμό των λανθανόντων παραγόντων δεν παίζει τόσο σημαντικό ρόλο στη διαμόρφωση των αποτελεσμάτων όσο ο ρυθμός μάθησης, ωστόσο σε καμία περίπτωση δεν θα πρέπει να θεωρηθεί ήσσονος σημασίας.

Θα πρέπει να σημειωθεί εδώ πως για να λειτουργήσει καλύτερα και πιο αποτελεσματικά ο DeepFM μπορούμε να δώσουμε ως είσοδο και επιπλέον χαρακτηριστικά πέραν της τριάδας {id αντικειμένου - id χρήστη - αξιολόγηση}, όπως για παράδειγμα τα είδη των ταινιών ή οι κατηγορίες στις οποίες ανήκουν τα αντικείμενα, κάτι που δεν εντάσσεται στα πλαίσια αυτής της εργασίας.

Συμπέρασμα: Σε αυτόν τον αλγόριθμο η ιδιαίτερα πολύπλοκη αρχιτεκτονική του, καθώς πρόκειται για ένα νευρωνικό δίκτυο με 4 επίπεδα embeddings, αρκετά πυκνά κρυφά επίπεδα και ένα επίπεδο πρόβλεψης. και εκτός αυτών θα πρέπει να συνυπολογίσουμε και το επίπεδο που υλοποιεί τα factorization machines, αποτελεί τροχοπέδη για την επεξηγησιμότητά του. Συνεπώς, θα πρέπει να αναζητηθούν ειδικές τεχνικές για την ανάλυση της επεξηγησιμότητας και της ερμηνευσιμότητας του αλγορίθμου, κάτι το οποίο όμως δεν εντάσσεται στα πλαίσια της παρούσας εργασίας.

5.4.1.4 Αλγόριθμοι που βασίζονται στα γραφήματα

Αλγόριθμος NGCF

Πίνακας 5.12: Ανάλυση υπερπαραμέτρων στον αλγόριθμο NGCF

dataset	lr	l_w	nDCG	IC	EPC	Gini	ARP	APLT	PREO	PRSP
		0.001	0.0367	1652	0.0148	0.5455	48.65	0.6726	0.0783	0.0678
	0.0001	0.1	0.0369	1651	0.0140	0.5505	46.26	0.6824	0.0570	0.0454
		1	0.0327	1652	0.0137	0.5466	45.36	0.6946	0.0263	0.0170
		0.001	0.1684	435	0.0516	0.0850	154.59	0.0433	0.9161	0.9623
ml100k	0.001	0.1	0.1905	527	0.0538	0.0947	155.82	0.0720	0.8965	0.9361
		1	0.0278	1651	0.0118	0.5669	42.48	0.7022	0.0377	0.0010
		0.001	0.1867	387	0.0550	0.0872	172.76	0.0595	0.9579	0.9476
	0.01	0.1	0.1747	585	0.0513	0.0904	158.78	0.0827	0.8985	0.9262
		1	0.0921	269	0.0344	0.0670	104.01	0.2479	0.6559	0.7542
		0.001	0.0150	1834	0.0011	0.5544	8.78	0.4151	0.0502	0.0199
	0.0001	0.1	0.0151	1834	0.0011	0.5547	8.75	0.4179	0.0523	0.0143
		1	0.0149	1834	0.0011	0.5553	8.57	0.4238	0.0344	0.0022
		0.001	0.0203	1806	0.0015	0.5234	11.68	0.3226	0.3697	0.2159
amazon	0.001	0.1	0.0205	1816	0.0015	0.5420	11.38	0.3535	0.2954	0.1493
		1	0.0146	1832	0.0011	0.5706	8.14	0.4330	0.0143	0.0166
		0.001	0.0339	448	0.0025	0.0649	19.61	0.0546	0.9689	0.8550
	0.01	0.1	0.0404	637	0.0029	0.0702	18.58	0.1246	0.8168	0.6770
		1	0.0134	1721	0.0010	0.3998	7.11	0.5043	0.2613	0.1588
		0.001	0.1664	976	0.0601	0.0408	1081.49	0.0092	0.9911	0.9919
	0.0001	0.1	0.1730	918	0.0617	0.0459	1085.46	0.0497	0.9788	0.9553
		1	0.0243	204	0.0134	0.0279	238.81	0.7018	0.0188	0.0141
		0.001	0.1596	882	0.0620	0.0392	1122.88	0.0055	0.9961	0.9951
ml1m	0.001	0.1	0.1662	686	0.0624	0.0453	1105.10	0.0316	0.9797	0.9719
		1	0.0175	316	0.0100	0.0281	241.87	0.6115	0.2167	0.1850
		0.001	0.0927	651	0.0407	0.0379	807.05	0.0699	0.9504	0.9364
	0.01	0.1	0.1239	436	0.0493	0.0398	928.90	0.0716	0.9611	0.9348
		1	0.0248	224	0.0136	0.0279	239.60	0.7002	0.0151	0.0102

Για $l\_w > 0.1$ έχουμε σχεδόν μηδενική ακρίβεια για όλα τα σύνολα δεδομένων, κάτι απόλυτα αναμενόμενο καθώς σύμφωνα με την συνάρτηση κόστους (βλέπε εξίσωση (5.23)) με $\lambda = 1$ στην ουσία δεν ρυθμίζει την υπερεκπαίδευση. Επομένως, παρόλο που εισάγει τη λιγότερη μεροληψία δεν μπορεί να χρησιμοποιηθεί. Εκείνο που προκαλεί ιδιαίτερη εντύπωση, είναι ότι ενώ στο Movielens100k και στο Amazon η ακρίβεια βελτιώνεται όσο αυξάνει η τιμή του ρυθμού μάθησης, στο Movielens1M συμβαίνει το ακριβώς αντίθετο. Συμπερασματικά, στον συγκεκριμένο αλγόριθμο φαίνεται να μην επηρεάζει η αραιότητα των δεδομένων, καθώς το Movielens100k είναι πιο πυκνό από το Movielens1M, αλλά χαρακτηριστικά όπως το rating space που είναι πολύ μεγαλύτερο στο Movielens1M από τα άλλα δύο σύνολα δεδομένων. Στο Movielens100k στην κάλυψη των αντικειμένων, στο diversity και στο popularity bias έχουμε πάρα πολύ καλή απόδοση για lr=0.0001 και πολύ κακή απόδοση για lr>=0.001. Στο Movielens1M όμως τα πράγματα είναι αρκετά διαφορετικά και χρήζουν περαιτέρω διερεύνησης. Αυτό που παρατηρούμε είναι ότι έχουμε σχεδόν μηδενικό diversity, πάρα πολύ υψηλό

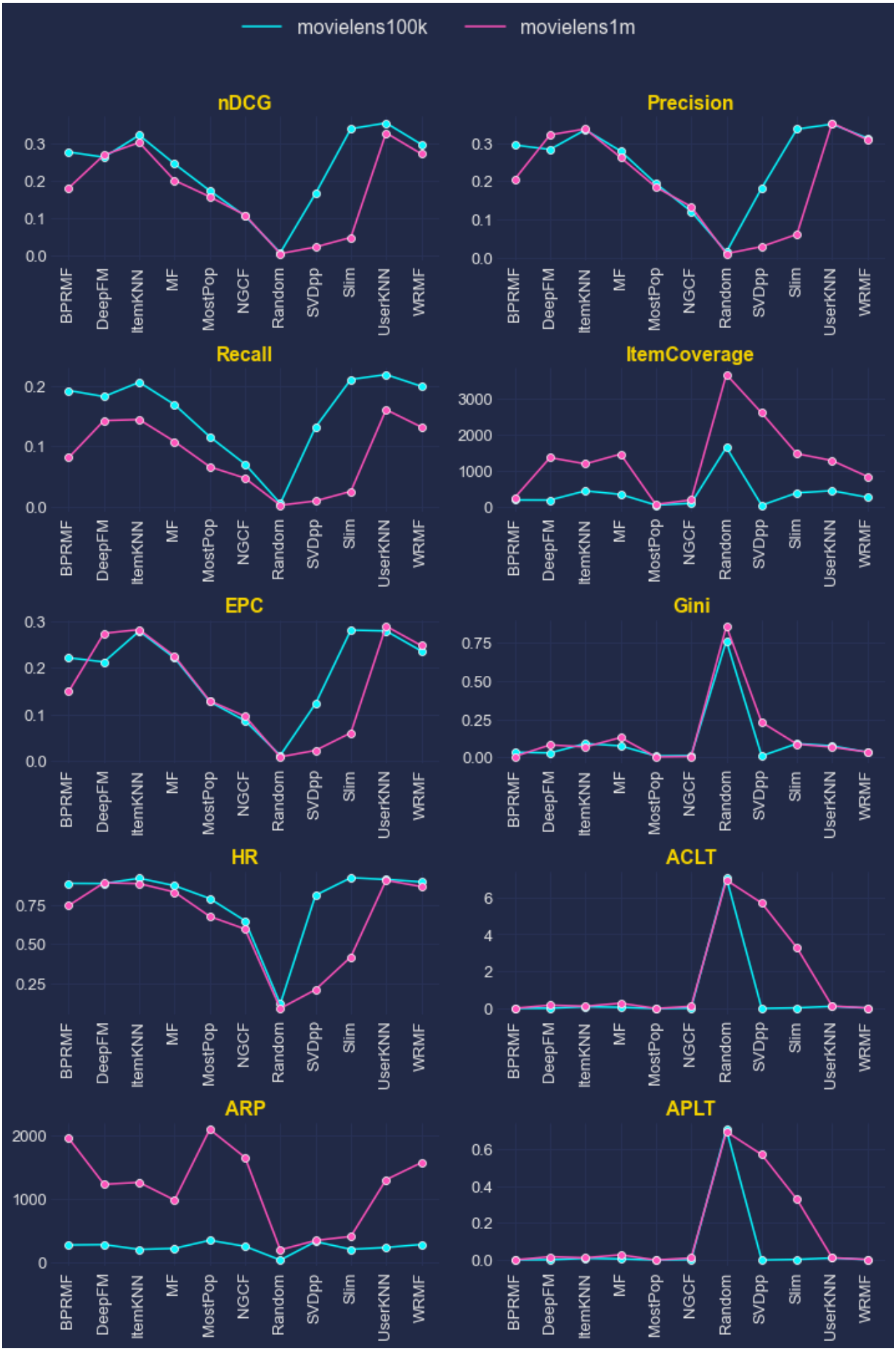
popularity bias και πολύ χαμηλή κάλυψη, με τις διαφορές ανάμεσα στις τιμές των παραμέτρων να μην είναι πολύ μεγάλες. Τέλος, στο Amazon τα αποτελέσματα φαίνεται να είναι πιο κοντά με αυτά του Movielens100k. Πιο συγκεκριμένα, η μόνη διαφορά που υπάρχει σε αυτά τα σύνολα δεδομένων είναι ότι στο Amazon οι τιμές των μετρικών παραμένουν σε σχετικά υψηλά επίπεδα έως ότου ο ρυθμός μάθησης να γίνει ίσος με 0.001 και στη συνέχει μηδενίζουν απότομα, ενώ στο Movielens100k αυτό συμβαίνει για ρυθμό μάθησης ίσο με 0.0001.
Συμπέρασμα: ο αλγόριθμος αυτός εξαρτάται αρκετά από τα χαρακτηριστικά των δεδομένων. Σε γενικές γραμμές, η ιδανική τιμή του ρυθμού μάθησης φαίνεται πως βρίσκεται στο διάστημα [0.0001, 0.001] και του l_w στο διάστημα [0.1, 0.5]. Παράλληλα, όπως και στον αλγόριθμο DeepFM η ιδιαίτερα πολύπλοκη αρχιτεκτονική του, δεν μας επιτρέπει να βγάλουμε ασφαλή συμπεράσματα.

5.5 Σύγκριση αλγορίθμων και συνόλων δεδομένων

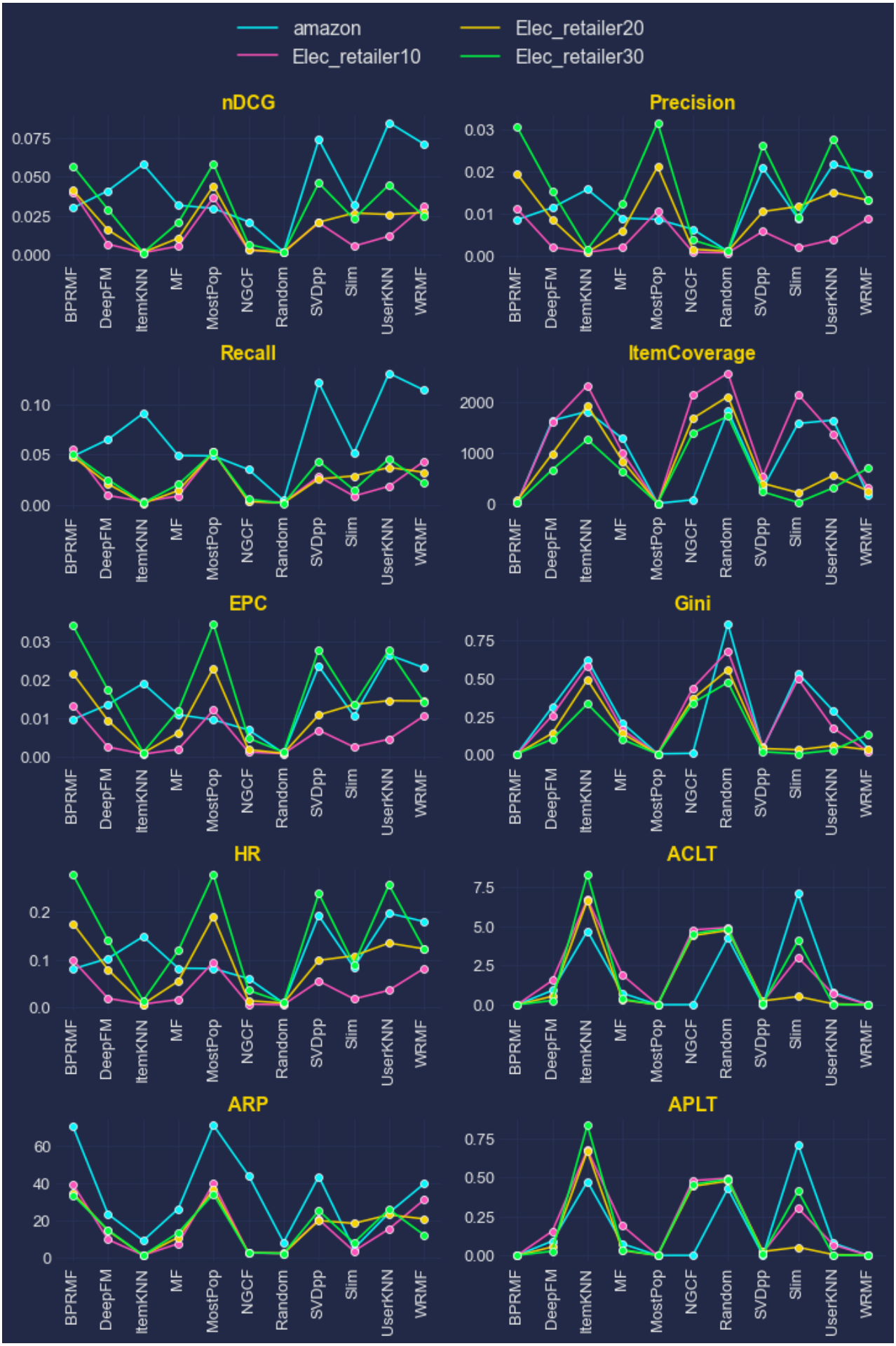
Η παρουσίαση και η σύγκριση των αποτελεσμάτων για τα διάφορα σύνολα δεδομένων και τους αλγορίθμους που χρησιμοποιήσαμε για κάθε μετρική αξιολόγησης γίνεται μέσω των Εικόνων 5.5 και 5.6. Στην Εικόνα 5.5 παρουσιάζονται τα δύο σύνολα δεδομένων από το Movielens στα ίδια γραφήματα και στην Εικόνα 5.6 τα σύνολα δεδομένων του Elec_retailer και του Amazon. Ο διαχωρισμός αυτός έγινε λαμβάνοντας υπόψη τις μεγάλες διαφορές στις τιμές των μετρικών, κατά κύριο λόγο και την αραιότητα των συνόλων δεδομένων. Ακόμη, όλα τα παρακάτω αποτελέσματα έχουν προκύψει για τιμή cut-off ίση με 10, ώστε να υπάρχει μια κοινή τιμή για όλα τα δεδομένα. Λόγω των ιδιαιτεροτήτων του συνόλου δεδομένων του Elec_retailer δεν ήταν εφικτή η επιλογή μιας μεγαλύτερης τιμής. Ακολουθεί η σύγκριση των αλγορίθμων, αρχικά ως προς την ακρίβεια και την κάλυψη των αντικειμένων και στη συνέχεια ως προς το popularity bias, το novelty και το diversity.

Ακρίβεια και κάλυψη των αντικειμένων
Στις μετρικές που επιλέχθηκαν για να μετρήσουν την ακρίβεια των αλγορίθμων, ξεχωρίζει η επίδοση των δύο neighborhood based αλγορίθμων, UserKNN και ItemKNN, για όλα τα σύνολα δεδομένων. Οι neighborhood based αλγόριθμοι εξαρτώνται σε πολύ μεγάλο βαθμό από τον αριθμό των χρηστών και των αντικειμένων, καθώς βασίζονται στον υπολογισμό ομοιοτήτων μεταξύ των αντικειμένων και των χρηστών, κάτι που είναι πολύ εμφανές στο σύνολο δεδομένων Elec_retailer. Ωστόσο, όταν ο αριθμός των χρηστών και των αντικειμένων είναι επαρκής, τότε πετυχαίνουν πολύ υψηλά ποσοστά, καλύτερα μάλιστα από όλους τους υπόλοιπους αλγορίθμους. Αρκετά καλή επίδοση έχει και ο DeepFM που συνδυάζει την τεχνική των Factorization Machines με τα βαθιά νευρωνικά δίκτυα και για τον λόγο αυτό η ακρίβεια που πετυχαίνει είναι αρκετά υψηλή. Παρόλο που μοιάζει αρκετά παράδοξο δύο από τους παλαιότερους αλγορίθμους να έχουν παρόμοια ή και καλύτερη ακρίβεια, από μια αρκετά πιο σύγχρονη προσέγγιση η οποία μάλιστα βασίζεται και στα νευρωνικά δίκτυα, αυτό έρχεται σε πλήρη συμφωνία με όσα αναφέρονται στη βιβλιογραφία [105].
Ένα άλλο σημείο που έχει ενδιαφέρον είναι η σχετικά κακή απόδοση του SVD++, η οποία είναι χειρότερη και από τον MF σε όλα τα σύνολα δεδομένων εκτός από το Amazon όπου έχει την 3η καλύτερη ακρίβεια. Μια πρώτη υπόθεση που μπορούμε να κάνουμε είναι ότι ο SVD++ δεν είναι model-based αλγόριθμος και από αυτό προκύπτουν αρκετά γνωστά και κοινά προβλήματα που εμφανίζονται στα συστήματα συστάσεων, όπως αυτό του cold-start το οποίο σημαίνει πως ο αλγόριθμος έχει χαμηλή ακρίβεια όταν έχουμε χρήστες με λίγες ή καθόλου αλληλεπιδράσεις με τα αντικείμενα (αξιολογήσεις, αγορές, προβολές κτλ.). Μια ακόμη εξήγηση είναι ότι ο SVD++

χρησιμοποιεί και implicit δεδομένα, τα οποία σε αυτό το σύνολο δεδομένων δεν είναι διαθέσιμα. Αυτός ο αλγόριθμος μαζί με τον NGCF έχουν τη χειρότερη επίδοση για όλα τα σύνολα δεδομένων. Στον NGCF αυτό που φαίνεται να συμβαίνει είναι ότι η χρήση πολλών ιεραρχικών επιπέδων για να επιτύχει την συνεκτικότητα, μειώνει τον βαθμό της ομοιότητας ανάμεσα στους χρήστες και στα κοντινότερα αντικείμενα και άρα μειώνει την ακρίβεια. Επιπρόσθετα, τρεις αλγόριθμοι που ανήκουν στην οικογένεια των λανθανόντων παραγόντων, έχουν σχετικά καλή ακρίβεια, με τον WRMF να ξεχωρίζει. Εξαίρεση αποτελεί ο Slim ο οποίος εξαρτάται πάρα πολύ από τα χαρακτηριστικά των δεδομένων, και μάλιστα όχι μόνο από την αραιότητα αλλά από έναν συνδυασμό αυτών, και έτσι η συμπεριφορά του είναι πολύ διαφορετική στα σύνολα δεδομένων, έχοντας μια από τις χαμηλότερες επιδόσεις στο Movielens100k και Amazon, μέτρια επίδοση στο Elec_retailer και την 2η καλύτερη επίδοση στο Movielens1M. Από τους μη-εξατομικευμένους αλγορίθμους ο MostPop κάνει αρκετά ακριβείς προβλέψεις, πιθανώς επειδή προτείνοντας πάντα τα πιο δημοφιλή αντικείμενα πετυχαίνει και έναν, όχι πολύ μεγάλο, αλλά ικανοποιητικό αριθμό σχετικών αντικειμένων. Αντιθέτως, ο Random προτείνει αντικείμενα ακολουθώντας την ομοιόμορφη κατανομή, μη λαμβάνοντας καθόλου υπόψη τα δεδομένου εισόδου και γι' αυτόν τον λόγο έχει σχεδόν μηδενική ακρίβεια.
Σχετικά με την κάλυψη των αντικειμένων στους αλγορίθμους itemKNN, userKNN, DeepFM, NGCF, MF έχουμε από καλά έως μέτρια αποτελέσματα, ειδικά για μικρό αριθμό χρηστών. Από την άλλη στον BPRMF, SVD++ και WRMF τα αντικείμενα που καλύπτονται είναι πολύ λίγα.
Όσον αφορά τα σύνολα δεδομένων, παρατηρούμε μεγάλες διαφορές στα τρία σύνολα δεδομένων που χρησιμοποιήσαμε. Τα χαρακτηριστικά των δεδομένων φαίνεται πως επηρεάζουν αρκετά την ακρίβεια. Πιο συγκεκριμένα, όσο πιο μεγάλη είναι η αραιότητα ενός συνόλου δεδομένων τόσο χειρότερη είναι η επίδοση, κάτι που έρχεται σε πλήρη συμφωνία με όσα γνωρίζουμε από τη θεωρία και αναφέραμε αναλυτικά στην υποενότητα 2.2.4. Αυτό είναι αρκετά ξεκάθαρο στα 4 σύνολα δεδομένων που εξετάσαμε. Στα σύνολα δεδομένων του Amazon και του Elec_retailer η πολύ μεγάλη αραιότητα οδηγεί σε πάρα πολύ μικρή έως και μηδενική ακρίβεια στο Elec_retailer η αραιότητα αγγίζει το 100%. Ωστόσο η αραιότητα, αν και το κύριο, δεν είναι το μόνο χαρακτηριστικό που επηρεάζει τα δεδομένα, κάτι που γίνεται ορατό σε ορισμένους αλγορίθμους, όπως εκείνοι που ανήκουν στην οικογένεια των neighborhood based και εξαρτώνται άμεσα από τον αριθμό των αντικειμένων και των χρηστών. Συνεπώς σε αυτούς εξίσου σημαντικά είναι χαρακτηριστικά όπως ο λόγος χρηστών προς αντικείμενα (UIR), αξιολογήσεων προς χρήστες (RPU) και αξιολογήσεων προς αντικείμενα (RPI). Προκειμένου να ενισχύσουμε την επιχειρηματολογία μας, στα σύνολα δεδομένων του Elec_retailer η μόνη παράμετρος που μεταβάλλεται από τις τρεις (αριθμός χρηστών, αριθμός αξιολογήσεων και αριθμός αντικειμένων) είναι ο αριθμός, των χρηστών, συνεπώς η διαφορά ως προς την αραιότητα είναι απειροελάχιστα μικρή. Εντούτοις, παρατηρούνται αξιοσημείωτες διαφορές, ιδιαίτερα στο Hit Rate(HR) και στο Precision και λιγότερο στο nDCG.
Στον χώρο του ηλεκτρονικού εμπορίου αυτό έχει ως συνέπεια οι προτάσεις που θα δοθούν στους χρήστες να μην είναι καθόλου σχετικές με τα ενδιαφέροντά τους. Αυτό αφενός μεν προκαλεί δυσαρέσκεια στους χρήστες, αφετέρου δε πλήττει έναν από τους βασικούς στόχους μιας επιχείρησης που είναι να κρατάει τους πελάτες της ευχαριστημένους. Ενώ και σε συστήματα που δεν σχετίζονται με το εμπόριο, προκαλεί σίγουρα την δυσαρέσκεια του χρήστη και ίσως τον ωθήσει να σταματήσει να χρησιμοποιεί τις υπηρεσίες που παρέχει το συγκεκριμένο σύστημα, αναζητώντας μια εναλλακτική επιλογή που θα ανταποκρίνεται καλύτερα στα ενδιαφέροντα και στις ανάγκες του.



Εικόνα 5.5: Σύνολα δεδομένων του Movielens, cut-off=10



Εικόνα 5.6: Σύνολα δεδομένων του Elec_retailer και του Amazon, cut-off=10

Εικόνα 5.7: Μετρικές PopREO και PopRSP.

Popularity bias, diversity, novelty

Ας εξετάσουμε πρώτα τι συμβαίνει με το novelty στα 4 σύνολα δεδομένων. Η μετρική EPC είναι εξ'ορισμού άρρηκτα συνδεδεμένη με την ακρίβεια και αυτό αποτελεί και την εξήγηση για το μηδενικό novelty του Random σε όλα τα σύνολα δεδομένων. Τα αντικείμενα που προτείνονται δεν πρέπει μόνο να μην τα έχει δει προηγουμένως ο χρήστης αλλά και να είναι σχετικά με τα ενδιαφέροντά του, κάτι που όπως είδαμε πριν στον Random δεν συμβαίνει. Το novelty στα σύνολα δεδομένων του Elec_retailer και στο Amazon είναι σχεδόν μηδενικό, ενώ στα σύνολα δεδομένων του Movielens έ-

χουμε πολύ καλύτερα αποτελέσματα, λογικά σε αυτό βοηθάει και το πολύ μεγάλο rating space. Αυτό αποτελεί μια ένδειξη πως η αραιότητα του μητρώου χρηστών-αντικειμένων δεν είναι το μόνο που επηρεάζει το novelty, αλλά και τα χαρακτηριστικά των δεδομένων. Στο novelty ξεχωρίζει η πολύ κακή απόδοση των αλγορίθμων NGCF, BPRMF και SVD++. Στους υπόλοιπους αλγόριθμους έχουμε από μέτριες έως ικανοποιητικές τιμές, με τους δύο neighborhood based και τον DeepFM να ξεχωρίζουν, κάτι το οποίο δεν ισχύει στα σύνολα δεδομένων του Elec_retailer. Κατά τη δημιουργία προτάσεων προς τους χρήστες, ένας από τους στόχους είναι να μην τους προτείνουμε αντικείμενα με τα οποία έχουν ήδη αλληλεπιδράσει, αλλά να τους οδηγήσουμε σε νέα μονοπάτια με αντικείμενα που ίσως τους ενδιαφέρουν. Χαρακτηριστικό παράδειγμα εδώ αποτελούν τα συστήματα συστάσεων μιας επιχείρησης ηλεκτρονικού εμπορίου, τα οποία έχουν την τάση να προτείνουν προϊόντα που ανήκουν στην ίδια ακριβώς κατηγορία με εκείνα που κάποιος πελάτης μόλις έχει αγοράσει. Μάλιστα συνεχίζουν να προβάλλουν τις ίδιες προτάσεις για ένα μεγάλο χρονικό διάστημα, προκαλώντας εκνευρισμό στον πελάτη και στερώντας του την ευκαιρία της ανακάλυψης προϊόντων ή και ολόκληρων κατηγοριών προϊόντων, που μπορεί και να αγνοούσε την ύπαρξή τους.

Στη συνέχεια, θα εξετάσουμε εάν κάποια αντικείμενα προτείνονται πιο συχνά σε σχέση με κάποια άλλα, και αν ναι σε τι βαθμό συμβαίνει αυτό. Αν εξαιρέσουμε τους αλγορίθμους itemKNN, NGCF και τον DeepFM που έχει σχετικά μέτρια επίδοση, στους υπόλοιπους αλγορίθμους παρατηρείται χαμηλό diversity στα σύνολα δεδομένων του Elec_retailer και του Amazon, δηλαδή σε αυτά με πολύ υψηλή αραιότητα.
Εκ πρώτης όψεως από την μετρική APLT, θα έλεγε κάποιος πως όσο πιο αραιό είναι ένα μητρώο, τόσο λιγότερο είναι το popularity bias που εισάγεται. Αληθεύει όμως αυτό; Ένα πρόβλημα με την μετρική APLT είναι πως μπορεί να έχει υψηλή τιμή ακόμη και εάν όλοι οι χρήστες έχουν λάβει το ίδιο σύνολο long-tail αντικειμένων. Σε όλους τους αλγορίθμους εκτός από τον NGCF και τον ItemKNN η μετρική PopReo είναι σχεδόν μηδενική για το Elec_retailer. Στο statistical parity τα πράγματα διαφοροποιούνται ελαφρώς κυρίως για τους αλγορίθμους MF και DeepFM, όσον αφορά τα σύνολα δεδομένων του Elec_retailer, με τον αριθμό των χρηστών (και των αντικειμένων) να φαίνεται πως παίζει πολύ μεγαλύτερο ρόλο εδώ. Άρα όταν λαμβάνουμε υπόψη και τις προτιμήσεις των χρηστών, η μεροληψία υπέρ των πιο δημοφιλών αντικειμένων είναι αρκετά πιο εμφανής. Ο μόνος αλγόριθμος που φαίνεται να επηρεάζεται από το από την αραιότητα είναι ο SVD++. Ας δούμε λίγο πιο προσεκτικά τι συμβαίνει με το popularity bias στο Elec_retailer, με τη βοήθεια της μετρικής ARP. Παρόλο που σχεδόν το 90% των προϊόντων έχει λάβει 1-2 αξιολογήσεις, παρατηρούμε πως ο μέσος αριθμός των αξιολογήσεων των προϊόντων στις λίστες συστάσεων που έχουν παράξει οι αλγόριθμοι για κάθε χρήστη, είναι αρκετά μεγαλύτερος από όσο θα περιμέναμε και θα θέλαμε. Αυτό ισχύει για όλους τους αλγόριθμους εκτός από τον ItemKNN και τον NGCF, ενώ οι MF και DeepFM έχουν μέτρια επίδοση. Πρέπει να διευκρινιστεί πως ο μέσος αριθμός αξιολογήσεων αναφέρεται στις αξιολογήσεις στο σύνολο εκπαίδευσης και αποτελεί σοβαρή ένδειξη για την ύπαρξη του popularity bias.
Παρόμοια συμπεράσματα προκύπτουν και στο Amazon, με τις διαφορές να εντοπίζονται στους αλγορίθμους NGCF, itemKNN και WRMF. Αναλυτικότερα, στον itemKNN η ακρίβεια είναι αρκετά υψηλή στο Amazon και πάρα πολύ χαμηλή στο Elec_retailer. Στον NGCF η διαφορά είναι ότι στον Amazon εισάγει αρκετή μεροληψία, ενώ στο Elec_retailer ελάχιστη έως καθόλου.

Κατά την ανάλυση του αλγορίθμου BPRMF είδαμε ότι το φαινόμενο του popularity bias είναι αρκετά έντονο σε αυτόν, τα αντικείμενα που καλύπτονται είναι ελάχιστα, ενώ το diversity είναι εξίσου χαμηλό και εξηγήσαμε τους λόγους για τους οποίους συμβαίνει αυτό. Στα παραπάνω διαγράμματα μάλιστα παρατηρούμε πως όχι απλά είναι έντονο το φαινόμενο, αλλά έχει την χειρότερη επίδοση από

όλους τους αλγορίθμους, σχεδόν ίδια με αυτή του αλγορίθμου MostPop σε όλα τα σύνολα δεδομένων. Εκτός όμως από τον BPRMF και οι περισσότεροι από τους υπόλοιπους αλγορίθμους φαίνεται πως έχουν την τάση να προτείνουν περισσότερο τα πιο δημοφιλή αντικείμενα, χωρίς να έχουν σχεδιαστεί για αυτόν τον σκοπό. Ακόμη, οι μετρικές μας δείχνουν πως οι neighborhood based αλγόριθμοι έχουν την τάση να μην προτείνουν συχνά αντικείμενα που ανήκουν στο long tail. Σε αυτό το σημείο θα πρέπει να υπενθυμίσουμε πως στον itemKNN για να δημιουργηθούν τα προτεινόμενα αντικείμενα προς τους χρήστες υπολογίζονται οι ομοιότητες των αντικειμένων με βάση τις αξιολογήσεις που έχουν λάβει από τους χρήστες, όπως φαίνεται και στον κάτωθι ψευδοκώδικα [15]:

```
For each item in product catalog, I1
    For each customer C who purchased I1
        For each item I2 purchased by customer C
            Record that a customer purchased I1 and I2
        For each item I2
            Compute the similarity between I1 and I2
```

Εικόνα 5.8: Ψευδοκώδικας αλγορίθμου itemKNN

Όπως γίνεται εύκολα αντιληπτό, τα αντικείμενα που βρίσκονται στο long tail, έχουν λιγότερες πιθανότητες να επιλεχθούν εδώ. Αυτό επιβεβαιώνεται πλήρως και από τις μετρικές, ειδικά όσο αυξάνεται ο αριθμός των γειτόνων Συνεπώς χρειάζεται λίγο πιο ενδελεχής έρευνα. Εξετάζουμε τόσο το statistical parity (PopRSP) όσο και το equal opportunity (PopReo), που μας δείχνουν αν έχουν τις ίδιες πιθανότητες να προταθούν τα αντικείμενα που ανήκουν στο long-tail και τα αντικείμενα που ανήκουν στο head ή αν υπάρχει μεροληψία υπέρ των πιο δημοφιλών αντικειμένων (όσο μεγαλύτερη η τιμή τους, τόσο μεγαλύτερη η μεροληψία που υπάρχει). Ιδιαίτερη έμφαση δόθηκε στην PopREO, καθώς λαμβάνει υπόψη της και τις προτιμήσεις των χρηστών.

Η ύπαρξη του popularity bias σε ένα σύστημα προτάσεων ταινιών όπως το Movielens και σε οποιοδήποτε σύστημα ψυχαγωγίας υπάρχει στο διαδίκτυο αρχικά έχει άμεση επίπτωση στους καλλιτέχνες και στους δημιουργούς ενός προϊόντος. Εάν οι αλγόριθμοι συνηθίζουν να μεροληπτούν υπέρ των πιο δημοφιλών ταινιών στην προκειμένη περίπτωση, τότε οι νέοι καλλιτέχνες και οι ταινίες που δεν έχουν γίνει γνωστές στο ευρύ κοινό, θα μείνουν καταδικασμένοι για πάντα στην αφάνεια, ανεξαρτήτως της ποιότητάς τους. Αυτό ισχύει και στον χώρο του ηλεκτρονικού εμπορίου με τα λιγότερο δημοφιλή προϊόντα. Αρχικά, ζημιώνεται άμεσα η επιχείρηση που τα πουλάει, καθώς ένα πολύ μεγάλο ποσοστό των προϊόντων που μπορεί να αγγίζει και το 80%, δηλαδή τα αντικείμενα που ανήκουν στο long-tail έχουν ελάχιστες πιθανότητες προβολής από τους χρήστες και συνεπώς αγοράς τους από αυτούς. Επίσης ζημιώνεται έμμεσα ο κατασκευαστής των προϊόντων που ανήκουν στο long-tail, διότι θα πουληθούν ελάχιστα κομμάτια και ενδεχομένως το κατάστημα να μην ξαναγοράσει τα συγκεκριμένα προϊόντα.

5.6 Μετριασμός μεροληψίας

Για τον μετριασμό της μεροληψίας χρησιμοποιήθηκε ένας αλγόριθμος που ανήκει στην in-processing κατηγορία τεχνικών μετριασμού της μεροληψίας και τρεις αλγόριθμοι που ανήκουν στην post-processing κατηγορία. Ο in-processing αλγόριθμος που χρησιμοποιήθηκε είναι ο pairwise_reg, ενώ οι post-processing είναι οι FAR, PFAR και Calibrated recommendations (στο εξής θα αναφέρεται ως "cali") οι οποίες περιγράφονται αναλυτικά στην υποενότητα 3.2.2. Συμπληρωματικά, σε όλους τους πίνακες ως base αναφέρεται ο αρχικός αλγόριθμος, στα αποτελέσματα του οποίου εφαρμόσαμε τις τεχνικές μετριασμού τις μεροληψίας. Στο σημείο αυτό, θα αποτελούσε παράλειψη εάν δεν αναφέραμε πως κατά τη διάρκεια των πειραμάτων δοκιμάσαμε και έναν επιπλέον αλγόριθμο τον FA*IR. Ωστόσο ο χρόνος υπολογισμού των νέων λιστών συστάσεων σε όλα τα σύνολα δεδομένων, ήταν απαγορευτικά πολύ μεγάλος, καθώς ανάλογα με το σύνολο δεδομένων μπορεί να άγγιζε ή και να ξεπερνούσε την μία ώρα, ενώ και τα αποτελέσματα ήταν παρόμοια ή ακόμη και χειρότερα με τους υπόλοιπους αλγορίθμους που δοκιμάσαμε. Παρ'όλα αυτά, ο FA*IR είναι διαθέσιμος στην εφαρμογή μας, με την σύσταση να χρησιμοποιείται σε μικρά σύνολα δεδομένων, παρέχοντας σαφή προειδοποίηση προς τους χρήστες για το ζήτημα που υπάρχει.
Η in-processing τεχνική επιλέχθηκε καθώς από την ανάλυση των αποτελεσμάτων φαίνεται πως το πρόβλημα στον BPRMF είναι αρκετά μεγάλο με τιμές που πλησιάζουν αυτές του MostPop, συνεπώς το πρόβλημα είναι στη δομή του αλγορίθμου και οι post-processing τεχνικές δεν θα βοηθήσουν. Στους post-processing αλγόριθμους δόθηκαν ως είσοδος οι λίστες συστάσεων που λάβαμε στο τρίτο βήμα του πειράματος από τους base αλγορίθμους συστάσεων, μεγέθους 100 για κάθε χρήστη. Οι αλγόριθμοι αυτοί λαμβάνουν επιπλέον ως είσοδο και μια λίστα όλων των αντικειμένων που υπάρχουν στο σύνολο δεδομένων, στην οποία έχουν επισημανθεί ποια από αυτά ανήκουν στο long tail και ποια ανήκουν στο head. Συνεπώς, δημιουργήθηκε για κάθε σύνολο δεδομένων ένα αρχείο το οποίο περιείχε τρεις στήλες:

1. **"itemid":** περιέχει τα IDs όλων των αντικειμένων που υπάρχουν
2. **"feature":** η στήλη αυτή για 20% των αντικειμένων που ανήκουν στο head, περιέχει την λέξη "head" και για τα υπόλοιπα αντικείμενα την λέξη "long"
3. **"value":** περιέχει την τιμή 0 αν το αντικείμενο ανήκει σε προστατευόμενη ομάδα και την τιμή 1 διαφορετικά.

Για τον μετριασμό της μεροληψίας, επιλέχθηκαν οι λίστες συστάσεων από τους αλγορίθμους συστάσεων που διαπιστώθηκε στο προηγούμενο βήμα πως έλαβαν τις χαμηλότερες τιμές στις μετρικές που σχετίζονται με το popularity bias και το diversity. Στη συνέχεια ρυθμίστηκαν οι υπερπαράμετροι των αλγορίθμων μετριασμού μεροληψίας, δηλαδή ο αριθμός των αντικειμένων που θα περιέχει κάθε λίστα χρηστών στην οποία έχει γίνει ανακατάταξη και η παράμετρος κανονικοποίησης λ. Οι αλγόριθμοι εκτελέστηκαν για διάφορες τιμές του λ, πιο συγκεκριμένα λ={0.1 , 0.2 , 0.5 , 0.7}. Θα πρέπει να σημειωθεί ωστόσο πως στους πίνακες που ακολουθούν το λ είναι ίσο με 0.5 καθώς η τιμή αυτή εξασφαλίζει τη βέλτιστη ισορροπία ανάμεσα στην ακρίβεια και στην μεροληψία, όπως διαπιστώθηκε από δοκιμές που πραγματοποιήσαμε μέσω της εφαρμογής. Εάν κάποιος δεν ενδιαφέρεται τόσο για την ακρίβεια, μπορεί να θέσει την τιμή του λ στο εύρος [0.6 , 0.7]. Αφού δόθηκαν ως είσοδος οι δύο προαναφερθείσες λίστες στους postprocessing αλγορίθμους, εκείνοι παρήγαγαν ως έξοδο τις νέες re-ranked λίστες συστάσεων μεγέθους 30 για κάθε χρήστη. Επίσης, σε όλες τις μετρικές έχει γίνει στρογγυλοποίηση στα τρία δυαδικά ψηφία, εκτός από τις μετρικές

Recall, EPC που έγινε στα τέσσερα, την μετρική ARP στα 2 και φυσικά την μετρική ItemCoverage που είναι ακέραιος αριθμός. Τέλος, διατηρήθηκαν τα ίδια σύνολα εκπαίδευσης και δοκιμής που χρησιμοποιήθηκαν και στο προηγούμενο βήμα. Στις παραγράφους που ακολουθούν περιγράφεται η επίδραση των αλγορίθμων μετριασμού μεροληψίας στα διάφορα σύνολα δεδομένων, δίνοντας ιδιαίτερη έμφαση στον Cali, καθώς όπως θα εξηγήσουμε πετυχαίνει (σχεδόν) πάντα την μεγαλύτερη μείωση της μεροληψίας. Αναφερόμαστε αρχικά αρκετά αναλυτικά στην επίδραση που έχει ο αλγόριθμος στα δύο σύνολα δεδομένων του Movielens και στο τέλος αναφερόμαστε στην επίδραση που έχει στο σύνολο του Amazon. Αυτό έγινε διότι όπως θα εξηγήσουμε σε επόμενη παράγραφο, στο σύνολο δεδομένων έχουμε πάρα πολύ χαμηλή ακρίβεια και ενδεχομένως τα συμπεράσματα που θα προκύψουν να μην ασφαλή.

Πίνακας 5.13: Μετριασμός μεροληψίας στον αλγόριθμο WRMF

	Movielens100k				Movielens1M				Amazon			
	base	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**
nDCG	**0.339**	0.333	0.336	0.322	**0.298**	0.286	0.286	0.282	**0.094**	0.089	0.091	0.073
Prec.	0.207	**0.208**	0.207	**0.208**	**0.220**	0.219	0.219	0.219	**0.012**	0.011	0.011	0.010
Recall	0.3616	0.3617	0.3625	**0.3631**	**0.2576**	0.2560	0.2560	0.2572	**0.1996**	0.1887	0.1935	0.1779
IC	505	571	560	**607**	1445	1517	1517	**1594**	349	654	631	**704**
EPC	0.1790	0.1789	**0.1792**	0.1775	**0.1960**	0.1930	0.1933	0.192	**0.0150**	0.0146	0.0149	0.0119
Gini	0.071	0.078	0.077	**0.080**	0.064	0.072	0.072	**0.075**	0.076	0.109	0.102	**0.123**
HR	**0.976**	**0.976**	**0.976**	0.975	0.956	**0.957**	**0.957**	**0.957**	**0.300**	0.286	0.292	0.273
ACLT	0.164	0.348	0.298	**0.407**	0.381	0.413	0.414	**0.558**	0.034	0.543	0.387	**0.986**
ARP	237.52	231.19	232.61	**230.41**	1331.21	1302.38	1302.49	**1297.07**	31.89	28.02	29.02	**26.08**
APLT	0.005	0.012	0.010	**0.014**	0.013	0.014	0.014	**0.019**	0.001	0.018	0.013	**0.033**
PREO	0.969	0.952	0.956	**0.942**	0.944	0.937	0.936	**0.904**	0.991	0.964	0.959	**0.930**
PRSP	0.995	0.990	0.991	**0.989**	0.989	0.988	0.988	**0.984**	0.997	0.951	0.965	**0.912**

Ο πρώτος αλγόριθμος στον οποίο δοκιμάσαμε να μετριάσουμε την μεροληψίας είναι ο WRMF. Με τον αλγόριθμο cali έχουμε αύξηση 20% στο Movielens100k και 10% στο Movielens1M στην κάλυψη των αντικειμένων. Στις μετρικές popularity bias, παρατηρείται μια μείωση της τάξης του 3% και 2.5% στο ARP, στο ACLT έχουμε πάρα πολύ μεγάλη αύξηση 148% και 46.5% και στο popREO μία αρκετά μικρή βελτίωση 3% 4%, στα ml100k και ml1m αντίστοιχα. στο Gini 11% και 17% ml100k και ml1m αντίστοιχα. Στις μετρικές ακρίβειας, το precision, το HR και το recall μένουν σχεδόν αμετάβλητα, ενώ στο nDCG έχουμε μια μείωση της τάξης του 5% και στα 2 σύνολα δεδομένων. Επομένως με μια σχετικά μικρή μείωση μόνο στο nDCG, βελτιώνουμε όλες τις υπόλοιπες μετρικές.

Ακολουθεί η ανάλυση για τον αλγόριθμο SLIM, για όλες τις μετρικές το πρώτο ποσοστό που αναφέρεται αντιστοιχεί στο ml100k και το δεύτερο στο ml1m. Στον αλγόριθμο Slim παρατηρούμε μια αρκετά μεγαλύτερη μείωση 17% και 11.5% στην τιμή του nDCG. Μικρότερη αλλά όχι αμελητέα είναι η μείωση στο precision (6.5% και 6%), ενώ ακόμα πιο μικρή είναι στο Recall (4% και στα δύο σύνολα δεδομένων). Παράλληλα, ο συγκεκριμένος αλγόριθμος μετριασμού της μεροληψίας παράγει κατά 17% και κατά 12% περισσότερα αντικείμενα, στα ml100k και ml1m αντίστοιχα. Τα ποσοστά αυτά είναι ελαφρώς μεγαλύτερα από τον WRMF, όπου ο base αλγόριθμος κάλυπτε λιγότερα αντικείμενα από τον Slim. Σε ό,τι έχει να κάνει με το popularity bias η βελτίωση που

έχουμε αντικατοπτρίζεται με την μείωση 10% και 9.4% στη μετρική ARP, με την τεράστια αύξηση, 195% και 86% στην APLT, και την μείωση 13% και 8% στην popREO. Ωστόσο, στην popRSP όπου το πρόβλημα είναι πολύ μεγαλύτερο, η βελτίωση είναι αρκετά μικρότερη (-3.6% και -2%). Τέλος, μικρή αύξηση της τάξης του 26.6% και 17.2% έχουμε στη μετρική Gini.

Πίνακας 5.14: Μετριασμός μεροληψίας στον αλγόριθμο Slim

	Movielens100k				Movielens1M				Amazon			
	base	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**
nDCG	**0.386**	0.370	0.384	0.319	**0.261**	0.247	0.253	0.231	0.049	**0.064**	0.063	0.059
Prec.	**0.2277**	0.2197	0.2265	0.2130	**0.2040**	0.1940	0.1970	0.1912	0.0065	**0.0080**	0.0079	**0.0080**
Recall	**0.3949**	0.3810	0.3929	0.3780	**0.2404**	0.2283	0.2314	0.2302	0.1179	0.1377	0.1370	0.1373
IC	609	690	653	**711**	1396	1504	1484	**1563**	1780	**1784**	**1784**	**1784**
EPC	**0.2111**	0.2045	0.2108	0.1877	**0.1952**	0.1864	0.1900	0.1772	0.0080	**0.0107**	0.0105	0.0099
Gini	0.150	0.181	0.163	**0.190**	0.1449	0.1667	0.1622	**0.1697**	**0.7106**	0.5506	0.5694	0.5490
HR	**0.986**	0.979	0.983	0.981	**0.949**	0.943	0.944	0.944	0.182	**0.215**	0.213	**0.215**
ACLT	0.602	1.460	0.920	**1.771**	0.649	1.046	0.940	**1.221**	**14.777**	10.876	11.229	10.804
ARP	176.98	162.89	171.54	**158.59**	817.31	750.68	766.75	**740.25**	**6.98**	10.43	10.13	10.46
APLT	0.020	0.049	0.031	**0.059**	0.022	0.035	0.031	**0.041**	**0.493**	0.363	0.374	0.360
PREO	0.930	0.864	0.908	**0.806**	0.908	0.882	0.890	**0.841**	**0.056**	0.178	0.157	0.180
PRSP	0.983	0.957	0.973	**0.948**	0.981	0.969	0.972	**0.964**	0.136	0.130	**0.105**	0.135

Μια ενδιαφέρουσα παρατήρηση εδώ είναι πως εάν επιχειρήσουμε σε ένα σύνολο δεδομένων με πολύ μεγάλη αραιότητα, όπου ο base αλγόριθμος μας έχει δώσει πολύ χαμηλή ακρίβεια και ελάχιστη μεροληψία, τότε όλοι οι αλγόριθμοι re-ranking έχουν αρκετά διαφορετική επίδραση από τα σύνολα δεδομένων του Movielens και από ότι ενδεχομένως θα αναμέναμε. Πιο συγκεκριμένα, βελτιώνουν την ακρίβεια και εισάγουν λίγη περισσότερη μεροληψία, αυξάνοντας κατά πολύ λίγο και τα αντικείμενα που καλύπτονται. Μια υπόθεση που μπορούμε να κάνουμε εδώ είναι πως οι re-ranking αλγόριθμοι προσπαθούν να βρουν το καλύτερο δυνατό αντιστάθμισμα ανάμεσα στην ακρίβεια και στην μεροληψία και για να γίνει αυτό εδώ θα πρέπει να «θυσιαστεί» η μεροληψία και όχι η ακρίβεια, καθότι εκεί έχουμε πάρα πολύ καλή επίδοση και δεν επηρεάζει πολύ τα αποτελέσματα μια ελαφρά εισαγωγή επιπλέον μεροληψίας. Ωστόσο ένα σημαντικό θέμα που φαίνεται να προκύπτει εδώ, είναι η σημαντική πτώση που παρατηρείται στην τιμή του diversity. Αυτή είναι η μία περίπτωση στο Amazon για τον αλγόριθμο Slim. Η άλλη περίπτωση, είναι να λειτουργήσει κλασικά δηλαδή να μειώσει την μεροληψία, μειώνοντας όμως ταυτόχρονα και την ακρίβεια, όπως συμβαίνει στον αλγόριθμο DeepFM και στον MF που έχει λίγο καλύτερη ακρίβεια από τον Slim. Στον αλγόριθμο DeepFM έχουμε μια αύξηση στο nDCG της τάξης του 10%, 13% και 20% στα σύνολα δεδομένων με την σειρά που εμφανίζονται στον πίνακα. Στην κάλυψη των αντικειμένων, τα αντικείμενα που προτείνονται αυξάνονται κατά 24% στο ML1M και 8.4% στο ml100k, ενώ στο Amazon έχουμε την κάλυψη 10 επιπλέον αντικειμένων. Επιπρόσθετα, στην μετρική Gini την μεγαλύτερη αύξηση την παρατηρούμε στο ML1M (30%), ενώ στα υπόλοιπα δύο τα νούμερα είναι αρκετά μικρότερη, 11% και 12% αντίστοιχα. Η μείωση στο ARP είναι αρκετά μικρότερη από το Slim (4%, 9%). Στο σύνολο Amazon δεν αυξάνει την μεροληψία όπως στον Slim, αλλά την μειώνει επιτυγχάνοντας παρόμοια νούμερα με εκείνα που είδαμε στο WRMF ως προς το ποσοστό μείωσης. Τέλος, στον αλγόριθμο MF προκύπτουν παρόμοια συμπεράσματα αν μελετήσουμε προσεκτικά τον πίνακα 5.16.

Πίνακας 5.15: Μετριασμός μεροληψίας στον αλγόριθμο DeepFM

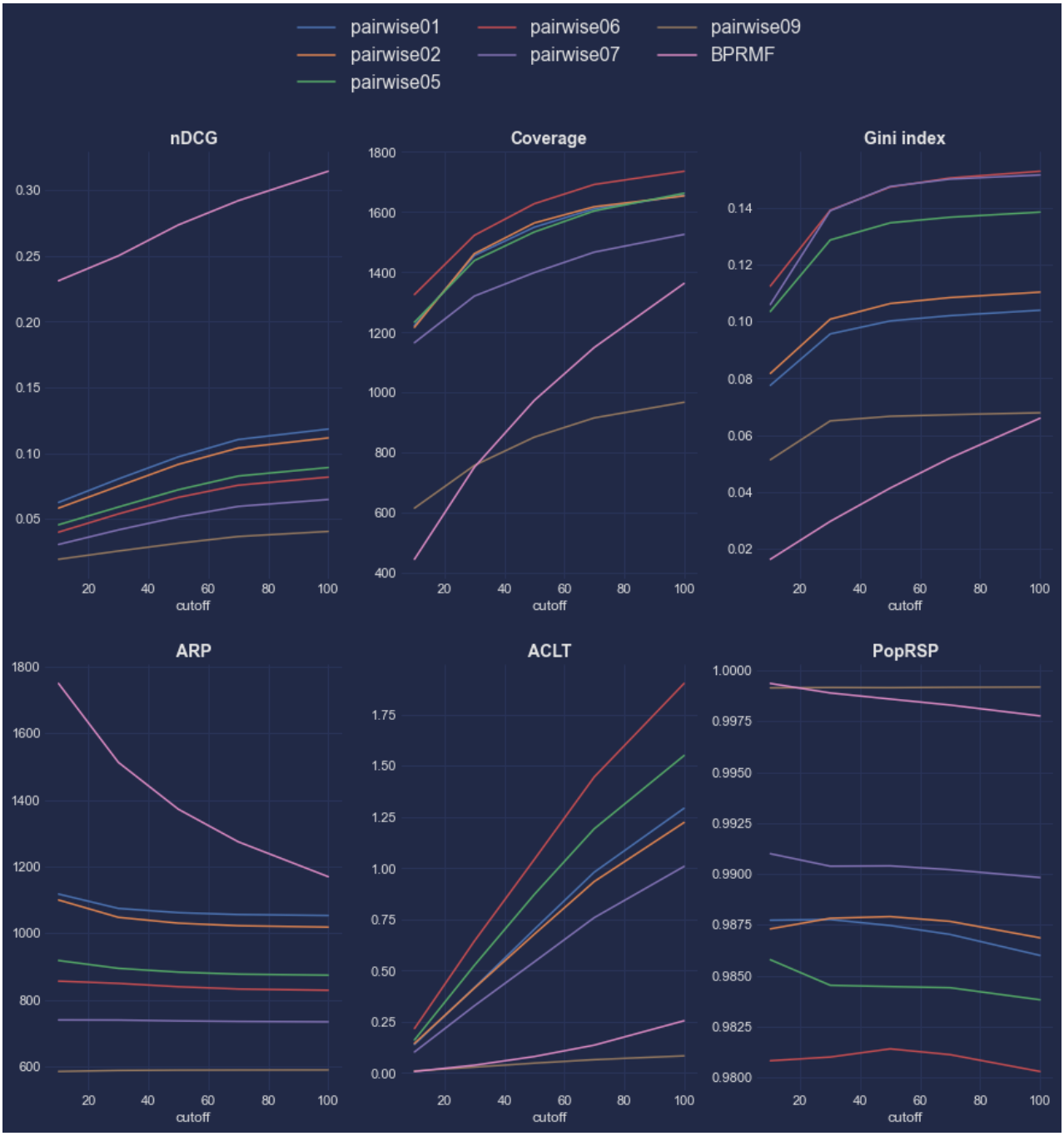
	Movielens100k				Movielens1M				Amazon			
	base	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**
nDCG	**0.318**	0.306	0.308	0.285	**0.294**	0.275	0.278	0.256	**0.055**	0.051	0.053	0.044
Prec.	**0.2048**	0.1988	0.2005	0.1983	**0.2330**	0.2221	0.2239	0.2208	**0.0068**	0.0065	0.0066	0.0064
Recall	**0.3651**	0.3549	0.3573	0.3529	**0.2850**	0.2741	0.2764	0.2747	**0.1174**	0.1120	0.1138	0.1110
IC	356	380	373	**443**	2124	2165	2152	**2303**	1812	1818	1817	**1822**
EPC	**0.1723**	0.1666	0.1680	0.1624	**0.2168**	0.2045	0.2064	0.1971	**0.0091**	0.0084	0.0087	0.0073
Gini	0.071	0.073	0.072	**0.079**	0.144	0.176	0.170	**0.187**	0.451	0.486	0.476	**0.504**
HR	0.979	0.982	**0.983**	0.979	**0.975**	0.974	0.973	0.972	**0.184**	0.174	0.178	0.173
ACLT	0.093	0.128	0.119	**0.386**	1.31	1.93	1.77	**2.37**	4.42	4.86	4.72	**5.18**
ARP	227.34	223.48	224.57	**216.54**	1016.21	942.10	957.98	**924.42**	18.30	17.34	17.62	**16.91**
APLT	0.0031	0.0043	0.0040	**0.0129**	0.044	0.064	0.059	**0.079**	0.147	0.162	0.157	**0.173**
PREO	0.970	0.964	0.965	**0.920**	0.794	0.763	0.770	**0.687**	0.793	0.757	**0.755**	0.760
PRSP	0.997	0.996	0.997	**0.989**	0.961	0.942	0.947	**0.928**	0.621	0.585	0.597	**0.560**

Πίνακας 5.16: Μετριασμός μεροληψίας στον αλγόριθμο MF

	Movielens100k				Movielens1M				Amazon			
	base	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**	**base**	**FAR**	**PFAR**	**cali**
nDCG	**0.305**	0.298	0.300	0.268	**0.241**	0.231	0.236	0.222	**0.043**	0.037	0.038	0.032
Prec.	**0.2090**	0.2046	0.2060	0.1947	**0.2076**	0.2000	0.2029	0.1968	**0.0054**	0.0051	0.0051	0.0051
Recall	**0.3610**	0.3545	0.3567	0.3447	**0.2425**	0.2341	0.2375	0.2343	**0.0906**	0.0856	0.0858	0.0854
IC	495	510	504	**597**	1818	1833	1825	**1926**	1656	1707	1703	**1713**
EPC	**0.1797**	0.1757	0.1769	0.1633	**0.1872**	0.1807	0.1837	0.1748	**0.0073**	0.0063	0.0065	0.0054
Gini	0.118	0.121	0.121	**0.143**	0.154	0.175	0.169	**0.185**	0.346	0.398	0.392	**0.404**
HR	0.973	0.977	**0.978**	0.966	**0.953**	0.951	0.951	0.950	**0.147**	0.141	0.141	0.141
ACLT	0.524	0.584	0.573	**1.047**	1.16	1.45	1.34	**1.74**	4.12	5.06	4.95	**5.28**
ARP	194.47	191.15	192.06	**177.87**	940.00	883.24	901.49	**862.31**	19.42	17.63	17.85	**17.41**
APLT	0.017	0.019	0.019	**0.035**	0.039	0.048	0.045	**0.058**	0.137	0.169	0.165	**0.176**
PopREO	0.951	0.949	0.951	**0.888**	0.866	0.846	0.855	**0.795**	0.809	0.793	0.813	**0.791**
PopRSP	0.985	0.983	0.984	**0.970**	0.965	0.957	0.960	**0.948**	0.645	0.570	0.578	**0.552**

Δοκιμάσαμε να εφαρμόσουμε και τις τρεις τεχνικές και στο σύνολο του Amazon, παρόλο που εκεί το μεγάλο πρόβλημα είναι η πάρα πολύ χαμηλή ακρίβεια και όχι το popularity bias, προκειμένου να εξετάσουμε την επίδραση που θα έχουν και σε μια τέτοια περίπτωση. Το βασικό συμπέρασμα που προκύπτει ύστερα από την εφαρμογή τριών post-processing τεχνικών είναι πως για τους αλγορίθμους όπου δεν είναι αρκετά έντονο το πρόβλημα του popularity bias είναι αρκετά αποτελεσματικές και οι τρεις τεχνικές. Ωστόσο σε διαφορετική περίπτωση υπάρχει μεν βελτίωση αλλά όχι τόσο μεγάλη ώστε να θεωρούμε πως αποτελούν μια καλή λύση για τον μετριασμό της μεροληψίας. Στην περίπτωση αυτή θα πρέπει να αναζητηθούν διαφορετικού τύπου τεχνικές, με τις in-processing που επιδρούν στη δομή του αλγορίθμου προσπαθώντας να επιτύχουν ένα βέλτιστο αντιστάθμισμα ανάμεσα στην μεροληψία και στην ακρίβεια ή και ένα συνδυασμό είδους τεχνικών in-processing, post-processing και pre-processing να αποτελούν δύο πολύ καλές επιλογές. Από τους τρεις αλγορίθμους που δοκιμάσαμε για όλους τους αλγορίθμους και όλα τα σύνολα δεδομένων είχαμε το ίδιο μοτίβο συμπεριφοράς.

Μεγαλύτερη μείωση της μεροληψίας παρατηρήθηκε στον Calibrated recommendations, με τον far να ακολουθεί και τον pfar να έχει την χειρότερη επίδοση. Αυτό φυσικά ισχύει και για την κάλυψη των αντικειμένων και στο diversity. Ενώ όπως ήταν αναμενόμενο το τελείως αντίθετο συμβαίνει στην ακρίβεια, αν και στις περισσότερες περιπτώσεις η μείωση στο Precision, στο Recall και στο HR είναι σχεδόν αμελητέα και είναι αρκετά πιο αισθητή στο nDCG.
Η in-processing τεχνική που δοκιμάσαμε στο σύνολο δεδομένων Movielens1M αποτελεί μέρος ενός ευρύτερου αλγορίθμου που συνδυάζει 3 τεχνικές μια pre-processing, μια in-processing (αυτή που δοκιμάσαμε) και μια post-processing. Η ακρίβεια από ότι παρατηρούμε μειώνεται πάρα πολύ ωστόσο εάν εφαρμόσουμε την in-processing τεχνική που προτείνεται, ενδεχομένως να βελτιώνεται αισθητά, χωρίς μάλιστα να επηρεάζει την μεροληψία που εισάγεται. Αυτό δεν εντάσσεται στα πλαίσια της παρούσας εργασίας και αφήνεται για μελλοντική εργασία. Εάν μάλιστα τον συγκρίνουμε με τον αλγόριθμο BPRMF που είναι και εκείνος pairwise και αποτελεί την βάση του αλγορίθμου που βελτίωσε αυτή η τεχνική, τότε είναι ξεκάθαρο πως ναι μεν το ζήτημα της μεροληψίας παραμένει σχετικά υψηλό, εντούτοις η βελτίωση είναι κατά πολύ μεγαλύτερη από όλες τις post-processing τεχνικές που δοκιμάσαμε σε οποιονδήποτε από τους αλγορίθμους. Η διαφορά μάλιστα γίνεται ακόμα πιο αισθητή σε μικρά cut-off κάτι ιδιαίτερα σημαντικό για μία πληθώρα συστημάτων που παρέχουν μικρό αριθμό συστάσεων προς τους χρήστες.



Εικόνα 5.9: Αλγόριθμος pairwise_reg και σύγκριση με τον BPRMF

Κεφάλαιο 6

Συμπεράσματα και μελλοντικές προεκτάσεις

6.1 Συμπεράσματα

Στη βιβλιογραφία τα τελευταία χρόνια γίνεται μεγάλη συζήτηση για τα ηθικά ζητήματα στη μηχανική μάθηση. Ωστόσο, η έρευνα που έχει γίνει στα συστήματα συστάσεων δεν είναι ανάλογη με αυτή που έχει γίνει σε άλλους τομείς, όπως στην κατηγοριοποίηση, την υπολογιστική όραση (computer vision) και την παλινδρόμηση. Λαμβάνοντας υπόψη μας αυτό, αυτή η διπλωματική στοχεύει στο να:

- συμβάλλει στην έρευνα γύρω από αυτό το ζήτημα
- βοηθήσει τους ερευνητές, τους διαχειριστές συστημάτων και τους χρήστες - ακόμη και εκείνους χωρίς ιδιαίτερες τεχνικές γνώσεις - , να εντοπίσουν και να μετριάσουν την μεροληψία ενός συστήματος συστάσεων, παρέχοντάς τους ταυτόχρονα την δυνατότητα να το δημιουργήσουν, όπως εκείνοι επιθυμούν, ενισχύοντας την διαφάνεια και την ελεγξιμότητα.
- αποτελέσει μια από τις πρώτες προσπάθειες ελέγχου της μεροληψίας σε ένα πραγματικό σύνολο δεδομένων
- γνωστοποιήσει στο ευρύ κοινό ζητήματα μεροληψίας στα συστήματα συστάσεων, δίνοντας ιδιαίτερη έμφαση στο popularity bias.

Στα πλαίσια αυτής της εργασίας αναπτύχθηκε μια διαδικτυακή εφαρμογή η οποία αφενός αποτέλεσε το εργαλείο για τη δημιουργία και την προβολή των αναλύσεων και των συγκρίσεων του πειράματος που διεξήχθη στα πλαίσια αυτής της εργασίας και αφετέρου αποτελεί ένα πολύτιμο εργαλείο για τον απλό χρήστη, τον ερευνητή και τον διαχειριστή ενός συστήματος. Η εφαρμογή αυτή, παρέχει στον χρήστη τις ακόλουθες δυνατότητες.
Αρχικά του επιτρέπει να μεταφορτώσει ένα σύνολο δεδομένων της αρεσκείας του και να ελέγξει για τυχόν ύπαρξη μεροληψίας σε αυτό μέσω αρκετά εύκολα κατανοητών γραφημάτων και πινάκων. Στη συνέχεια, να δημιουργήσει τα συστήματα συστάσεων όπως εκείνος επιθυμεί επιλέγοντας μέσα από μια πληθώρα αλγορίθμων και με δυνατότητα να ρυθμίσει κατάλληλα τις υπερ-παραμέτρους τους. Σε επόμενο βήμα, και αφού ο χρήστης έχει δημιουργήσει το σύστημα συστάσεων, μπορεί να ελέγξει για τυχόν ύπαρξη popularity bias, για το πόσα αντικείμενα καλύπτονται, για το diversity και το novelty. Ο έλεγχος αυτός περιλαμβάνει την ανάλυση υπερ-παραμέτρων, τη σύγκριση διαφορετικών συνόλων δεδομένων μεταξύ τους και την σύγκριση των αποτελεσμάτων για διαφορετικές τιμές μεγέθους

λιστών συστάσεων ανά χρήστη. Ενώ τέλος, παρέχει και τη δυνατότητα για μετριασμό ενδεχόμενης μεροληψίας που εντοπίστηκε στο προηγούμενο βήμα, μέσω 4 διαφορετικών αλγορίθμων (FAR, PFAR, Calibrated recommendations, FA*IR) που όλοι ανήκουν στην κατηγορία των post-processing αλγορίθμων. Στην εφαρμογή επίσης παρέχονται εξηγήσεις για κάθε διαθέσιμη μετρική αξιολόγησης σε γλώσσα απλή και κατανοητή για όλους τους χρήστες.
Αξίζει να σημειωθεί, πως με την χρήση της εφαρμογής ο χρήστης έχει άμεση πρόσβαση στα δύο κύρια σημεία εισαγωγής της μεροληψίας: στα δεδομένα και στον αλγόριθμο.

Με την ολοκλήρωση της ανάπτυξης της εφαρμογής διεξήχθησαν μέσω αυτής ορισμένα πειράματα που σχετίζονται με την εύρεση και τον μετριασμό της μεροληψίας των συστημάτων συστάσεων σε τέσσερα διαφορετικά σύνολα δεδομένων, ένα εκ των οποίων παραχωρήθηκε από μεγάλη εταιρεία λιανικών πωλήσεων. Αρχικά δημιουργήθηκαν συστήματα συστάσεων με χρήση 11 αλγορίθμων από 4 διαφορετικές οικογένειες αλγορίθμων. Η αξιολόγηση των συστημάτων συστάσεων έγινε με τη χρήση 12 διαφορετικών μετρικών κάλυψης αντικειμένων, ακρίβειας, μεροληψίας δημοφιλίας, diversity και novelty. Στο πρώτο μέρος του πειράματος εξετάστηκε το κατά εάν και πόσο επηρεάζουν οι υπερ-παράμετροι κάθε αλγορίθμου την μεροληψία και την ακρίβεια. Διαπιστώθηκε πως σε όλους τους αλγορίθμους *η ρύθμιση των υπερ-παραμέτρων επηρεάζει πολύ το ποσοστό εισαγωγής της μεροληψίας* και μάλιστα επειδή στην πλειονότητα των περιπτώσεων η ακρίβεια είναι αντιστρόφως ανάλογη της μεροληψίας, μπορεί αρκετά εύκολα να εισαχθεί αρκετή μεροληψία εάν κατά την ρύθμιση των υπερπαραμέτρων μας ενδιαφέρει περισσότερο η ακρίβεια ή ακόμη χειρότερα αγνοήσουμε τελείως την μεροληψία. Ένα ακόμη μέρος του πειράματος αποτέλεσε η σύγκριση των συνόλων δεδομένων. Η αραιότητα πέρα από την ακρίβεια επηρεάζει αρκετά, σε σχεδόν όλες τις περιπτώσεις, το novelty και το diversity, ενώ σε πολλές περιπτώσεις επιδρά αρνητικά και στο popularity bias. Το συμπέρασμα που προέκυψε εδώ είναι πως *εκτός από την αραιότητα των δεδομένων, που έχει αναφερθεί και εκτενώς στη βιβλιογραφία, ένας αρκετά σημαντικός παράγοντας για τον οποίο δεν έχει γίνει μεγάλη αναφορά στην βιβλιογραφία είναι τα χαρακτηριστικά των δεδομένων,* καθώς παρατηρήθηκε το φαινόμενο σύνολα δεδομένων με παρόμοια αραιότητα να έχουν αρκετά διαφορετική συμπεριφορά. Μάλιστα στην βιβλιογραφία δεν κατορθώσαμε να βρούμε κάτι που να εξετάζει και τα χαρακτηριστικά των δεδομένων και τις υπερπαραμέτρους.
Θα πρέπει να τονίσουμε πως παρόμοια συμπεράσματα προέκυψαν και για το πραγματικό σύνολο δεδομένων που χρησιμοποιήσαμε, όπου τα δημοφιλή προϊόντα είναι λίγα. Αυτό το φαινόμενο όπως γίνεται εύκολα κατανοητό, έχει επιπτώσεις τόσο στους χρήστες, όσο και στην εταιρία και (έμμεσα) στους κατασκευαστές των προϊόντων. Επιπρόσθετα, η αραιότητα και τα χαρακτηριστικά των δεδομένων μπορούν να επηρεάσουν πάρα πολύ την ακρίβεια των αλγορίθμων και επομένως να δίνονται προτάσεις στους χρήστες τελείως άσχετες με τα ενδιαφέροντά τους και το προφίλ τους.
Στο [99] γίνεται μια αναφορά στο το πως η αραιότητα των δεδομένων επηρεάζει και το popularity bias, πέρα από την ακρίβεια, χωρίς ωστόσο το ζήτημα να εξετάζεται εκτενώς. Επιπρόσθετα, κάτι που παρατηρήθηκε κατά την ανάλυσή μας είναι πως *υπάρχουν αλγόριθμοι οι οποίοι εισάγουν (πολύ) περισσότερη μεροληψία σε σχέση με άλλους*, όπως ο BPRMF, ο SVD++ και άλλοι όπως οι NGCF και ItemKNN που είναι πιο δίκαιοι στις προβλέψεις τους, κάτι που έρχεται σε πλήρη συμφωνία με διάφορες μελέτες που έχουν γίνει. Ωστόσο, για όσους έχουμε καλά αποτελέσματα θα πρέπει να γίνει ρύθμιση υπερπαραμέτρων, προκειμένου να βρεθεί μια ισορροπία με την ακρίβεια.
Δοκιμάσαμε επίσης συνολικά τέσσερις αλγορίθμους μετριασμού οι οποίοι ανήκουν σε δύο κατηγορίες τεχνικών. Τρεις που ανήκουν στην κατηγορία του post-processing που εφαρμόζουν την τεχνική του re-ranking και έναν που ανήκει στην κατηγορία του in-processing. Στην γενική περίπτωση στην

post-processing κατηγορία ο αλγόριθμος Cali πετυχαίνει την καλύτερη μείωση μεροληψίας, ωστόσο καλύτερο αντιστάθμισμα μεροληψίας-ακρίβειας επιτυγχάνει ο αλγόριθμος FAR. *Η κατηγορία post-processing αποδείχθηκε μέσω των πειραμάτων ότι δεν ενδείκνυται για σύνολα δεδομένων με πολύ μεγάλη αραιότητα, ενώ δεν είναι αρκετά αποτελεσματική σε περίπτωση όπου έχουμε πολύ μεγάλη εισαγωγή μεροληψίας*, κάτι που είναι γνωστό και από την βιβλιογραφία. Σε αυτή την περίπτωση θα πρέπει να χρησιμοποιήσουμε μια in-processing τεχνική είτε συνδυασμό τεχνικών (στις περισσότερες περιπτώσεις).

Ύστερα και από έρευνα που διεξήγαμε δεν βρήκαμε κάτι παρόμοιο με την εφαρμογή που υλοποιήθηκε στα πλαίσια της παρούσας εργασίας. Ενώ αξίζει να σημειωθεί αποτελεί μια από τις πρώτες προσπάθειες διερεύνησης του φαινομένου της εισαγωγής μεροληψίας στα συστήματα συστάσεων που χρησιμοποιεί ένα πραγματικό σύνολο δεδομένων (έστω και αν αυτό έχει τροποποιηθεί ελαφρώς). Κλείνοντας, η συνεισφορά της εργασίας σε κοινωνικό και ερευνητικό επίπεδο και στον επιχειρηματικό τομέα κρίνουμε πως είναι η εξής:

- Στον τομέα του ηλεκτρονικού εμπορίου αποδείχθηκε, ύστερα και από την εξέταση ενός πραγματικού συνόλου δεδομένων, πως το φαινόμενο της μεροληψίας δημοφιλίας είναι αρκετά έντονο, ενώ αρκετά αντικείμενα μπορεί να μην υπάρχουν σε καμία λίστα συστάσεων που δίνονται από τους αλγορίθμους στους χρήστες, μειώνοντας κατά πολύ τις πιθανότητες να τα δουν οι χρήστες και επομένως να πωληθούν.
- Σε κοινωνικό επίπεδο, η εφαρμογή που αναπτύχθηκε αποτελεί ένα εργαλείο για τη δημιουργία ενός συστήματος συστάσεων, εύκολα και γρήγορα για εύκολο και γρήγορο έλεγχο των συστημάτων συστάσεων και μετριασμό του popularity bias σε κάθε σύνολο δεδομένων που περιέχει explicit δεδομένα ανεξαρτήτως τομέα.
- Σε επιστημονικό επίπεδο, η εφαρμογή μπορεί να χρησιμοποιηθεί για δημιουργία πειραμάτων στα συστήματα συστάσεων είτε αυτά αφορούν ηθικά ζητήματα, όπως η μεροληψία και η δικαιοσύνη, είτε όχι.

6.2 Περιορισμοί

Αρχικά, στο διαδίκτυο υπάρχουν πολύ λίγα σύνολα δεδομένων συστημάτων συστάσεων, που να πληρούν τα κριτήρια που είχαμε θέσει. Μια ακόμη δυσκολία που κληθήκαμε να αντιμετωπίσουμε είναι πως στο πραγματικό σύνολο δεδομένων δεν υπήρχε καμία πληροφορία σχετική με τους χρήστες και αναγκαστήκαμε να δημιουργήσουμε τυχαίους χρήστες. Όπως γίνεται αντιληπτό, αυτό αποτέλεσε τροχοπέδη σε πολλές περιπτώσεις, καθώς μπορεί -άθελά μας- να εισάγαμε κάποιο είδος μεροληψίας, κάτι που θα έκανε τα συμπεράσματα που θα προέκυπταν από το πείραμα ιδιαίτερα επισφαλή, αναγκάζοντάς μας έτσι να μην τα χρησιμοποιήσουμε σε ορισμένα μέρη του πειράματος.

6.3 Μελλοντικές προεκτάσεις

Όσον αφορά την εφαρμογή που υλοποιήθηκε, ως επέκτασή της στο μέλλον σχεδιάζεται να προστεθούν οι εξής λειτουργίες:

1. παροχή της δυνατότητας στον χρήστη να μπορεί να μεταφορτώνει τα δικά του αρχεία αποτελεσμάτων

2. δυνατότητα δημιουργίας λογαριασμού για κάθε χρήστη, δημιουργώντας ειδική σελίδα για την εγγραφή και την σύνδεση του χρήστη.

3. προσθήκη περισσότερων αλγορίθμων μετριασμού της μεροληψίας που να ανήκουν σε διαφορετικές κατηγορίες.

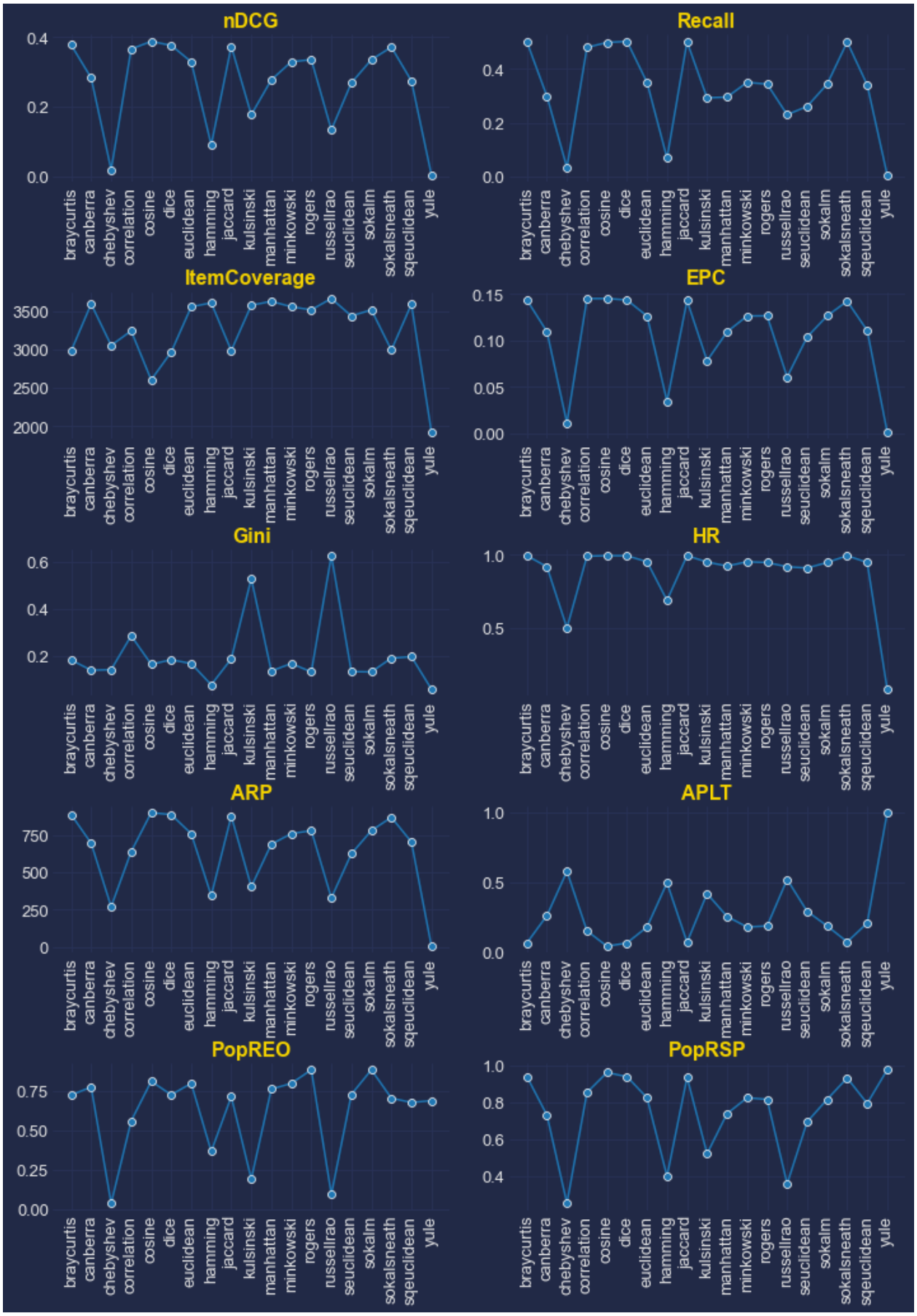
4. βελτίωση του χρόνου υπολογισμού μέσω διάφορων τεχνικών (μέσω παραλληλοποίησης, αποθήκευσης στην cache ή οποιασδήποτε άλλης τεχνικής.).

Τέλος, όσον αφορά τα πειράματα που υλοποιήθηκαν θα μπορούσε να γίνει και χρήση implicit αξιολογήσεων και σύγκριση με τα explicit ως προς την εισαγωγή της μεροληψίας και χρήση περισσότερων συνόλων δεδομένων.

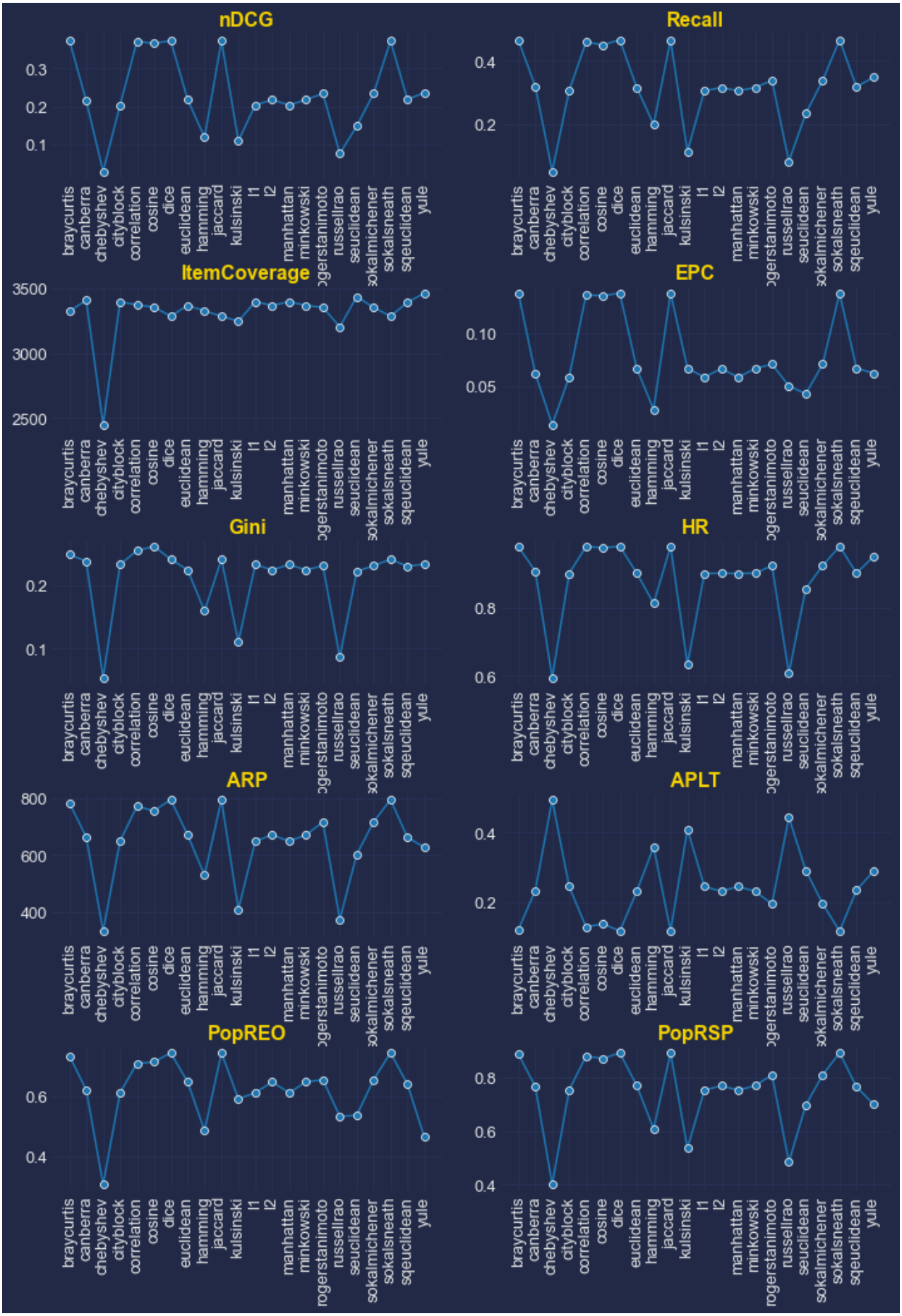
Εν κατακλείδι, οι αλγόριθμοι τεχνητής νοημοσύνης έχουν εισέλθει πλέον σε πολλές πτυχές της καθημερινότητάς μας. Ειδικότερα, οι αλγόριθμοι συστημάτων συστάσεων μπορεί να εισάγουν ή να τροποποιηθούν κατάλληλα για να εισάγουν μεροληψία, η οποία αποσαθρώνει την κοινωνία και αποτελεί σοβαρό κίνδυνο για την δημοκρατία, αφαιρώντας τη δυνατότητα κριτικής σκέψης από τον πολίτη μιας χώρας, τον πελάτη ενός ηλεκτρονικού καταστήματος και τον καταναλωτή ψυχαγωγικού περιεχόμενου. Ελπίζουμε η εργασία αυτή να αποτελέσει εφαλτήριο για περισσότερη και πιο ενδελεχή έρευνα στα συστήματα συστάσεων και να βοηθήσει όλους τους χρήστες ανεξαρτήτως των γνώσεών τους, να εντοπίσουν αλλά και να μετριάσουν την μεροληψία στα συστήματα συστάσεων.

Παράρτημα

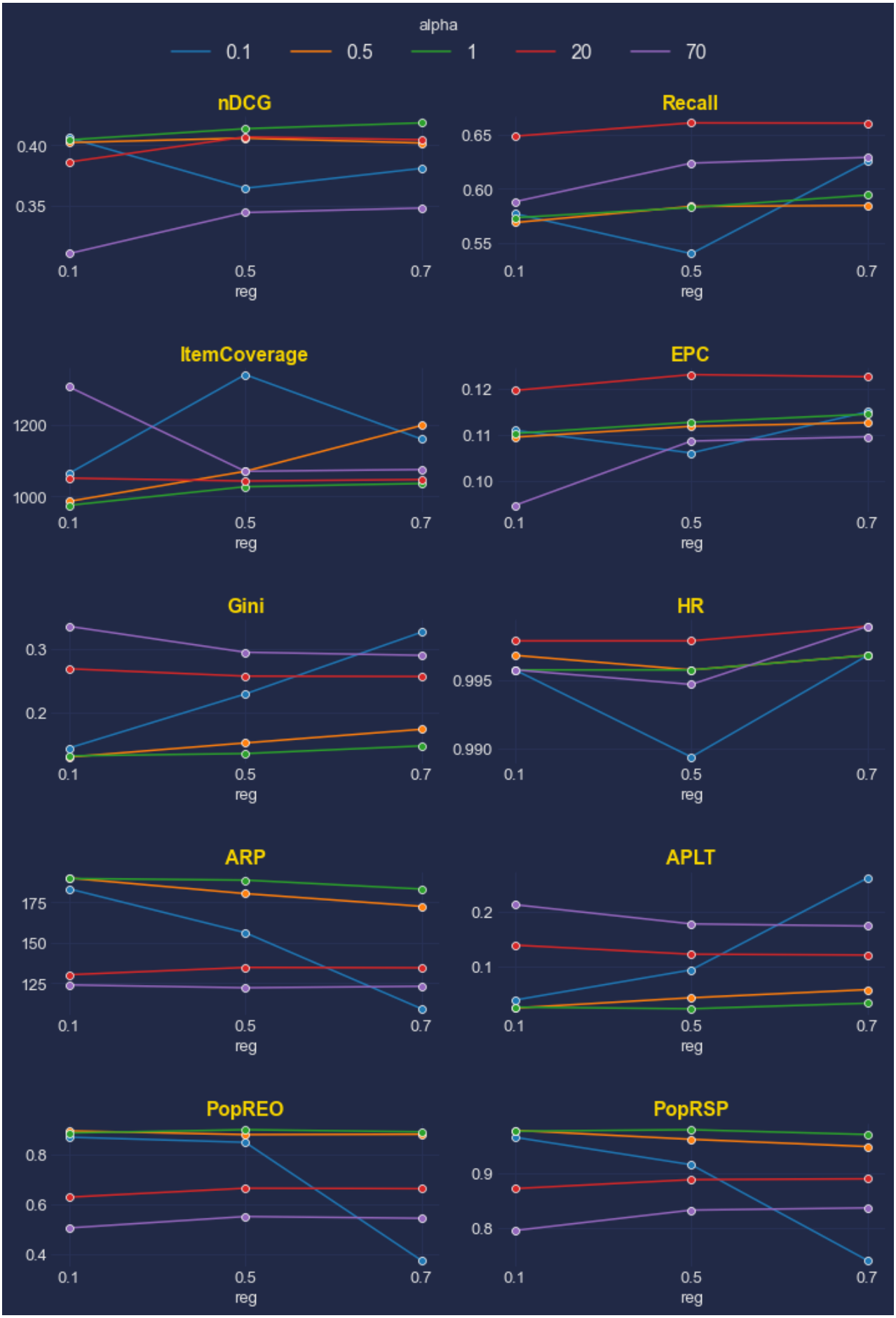
A.1 Γραφικές παραστάσεις ανάλυσης μεροληψίας



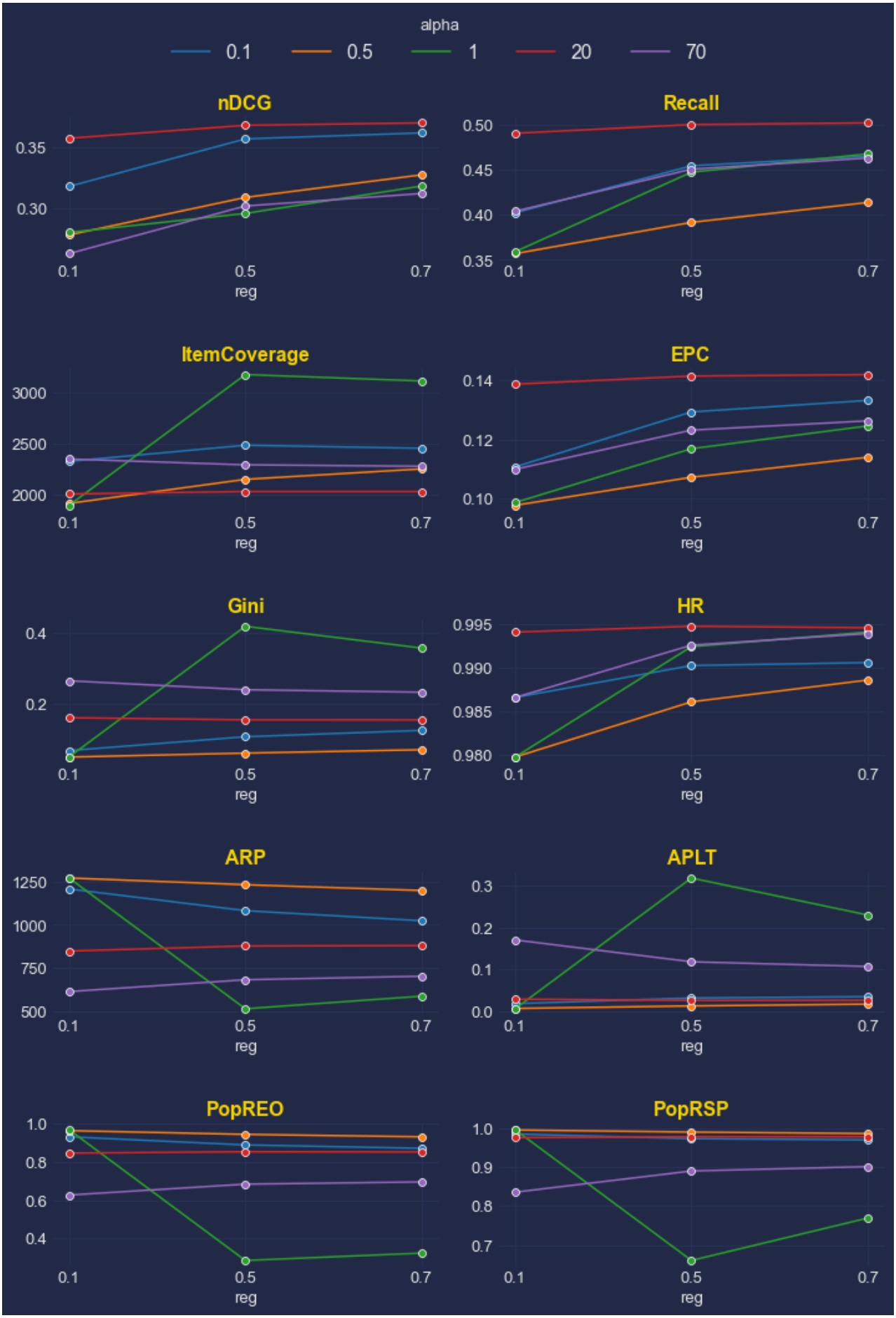
Εικόνα 1: Επίδοση μετρικών ομοιότητας στον αλγόριθμο itemKNN



Εικόνα 2: Επίδοση μετρικών ομοιότητας στον αλγόριθμο userKNN

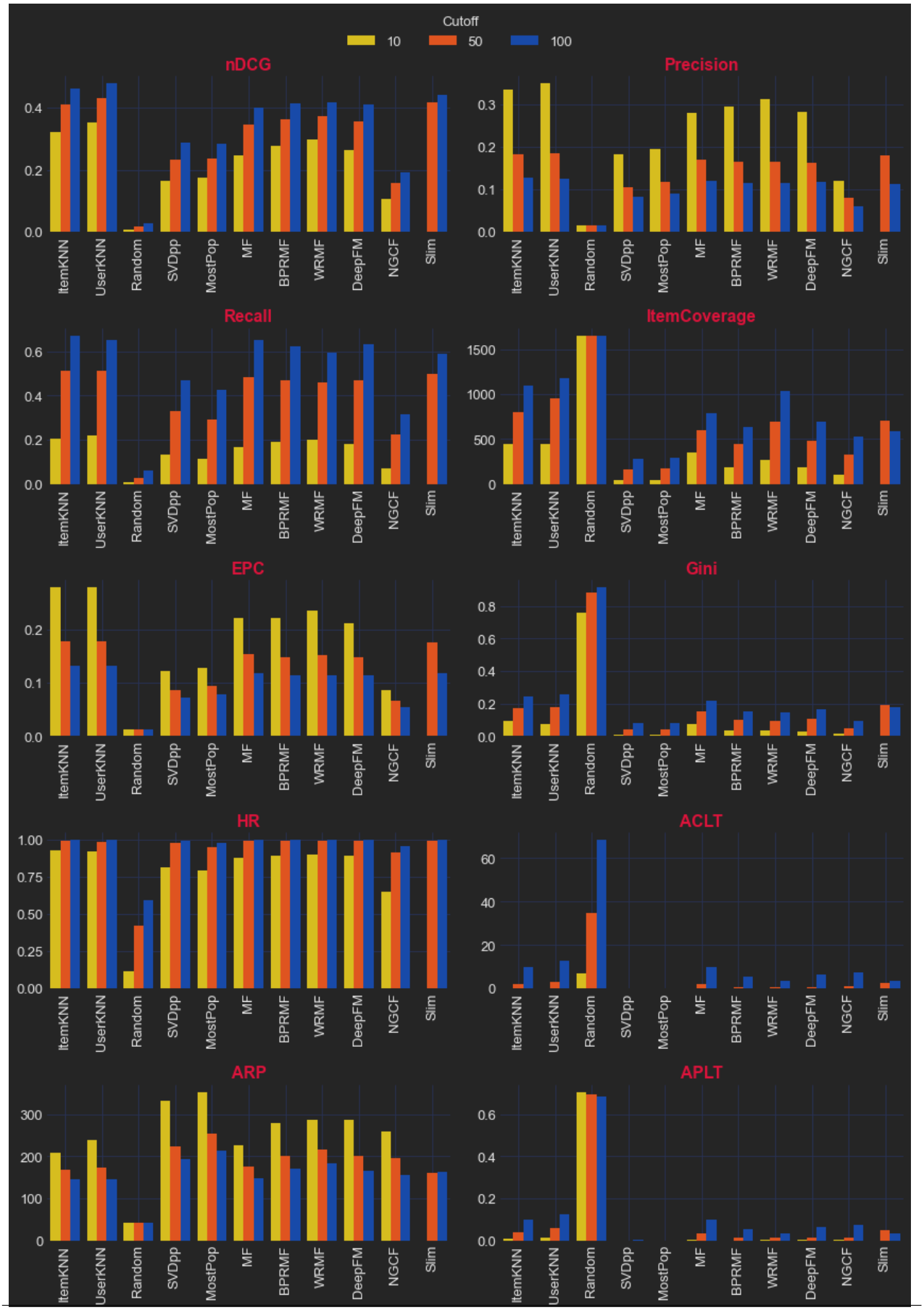


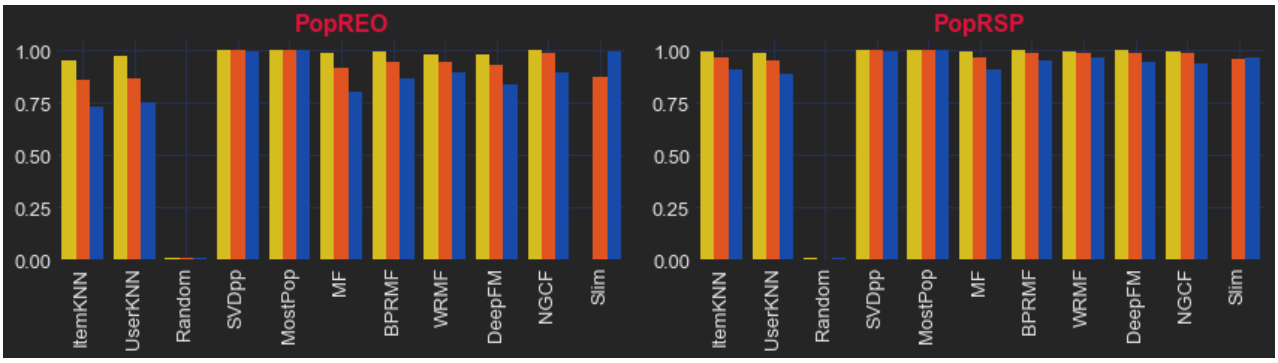
Εικόνα 3: Ανάλυση υπερπαραμέτρων στον αλγόριθμο WRMF για το σύνολο δεδομένων ML100K



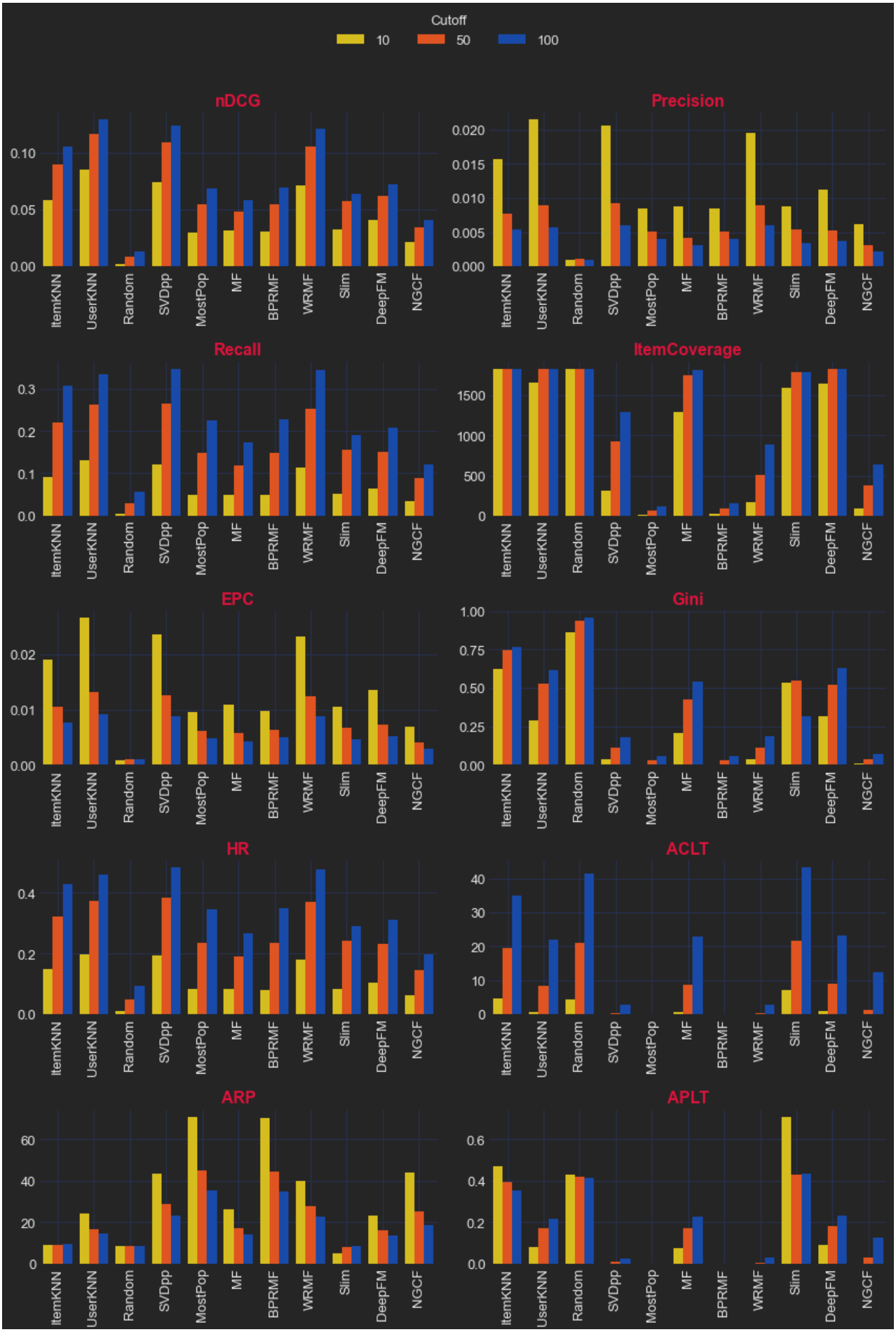
Εικόνα 4: Ανάλυση υπερπαραμέτρων στον αλγόριθμο WRMF για το σύνολο δεδομένων ML1M

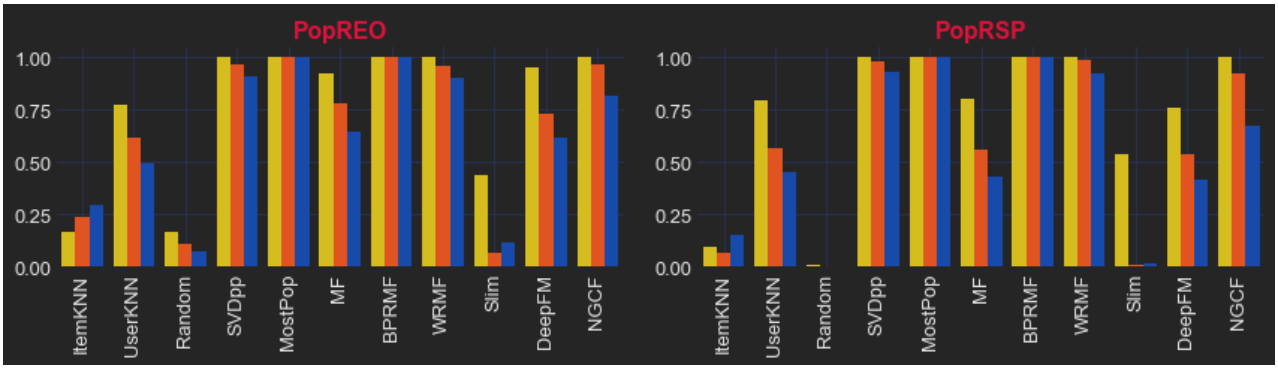
A.2 Ανάλυση υπερπαραμέτρων



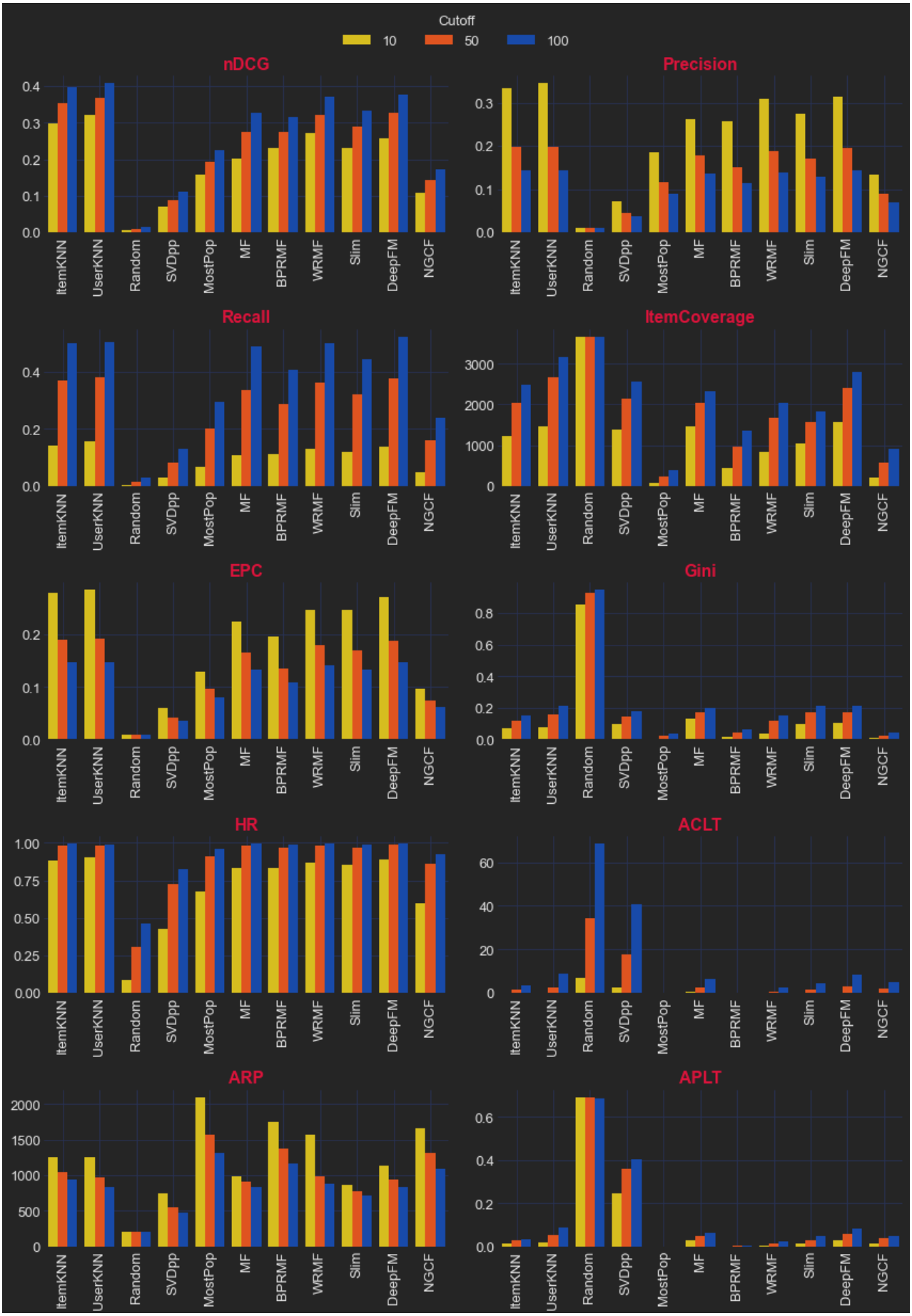


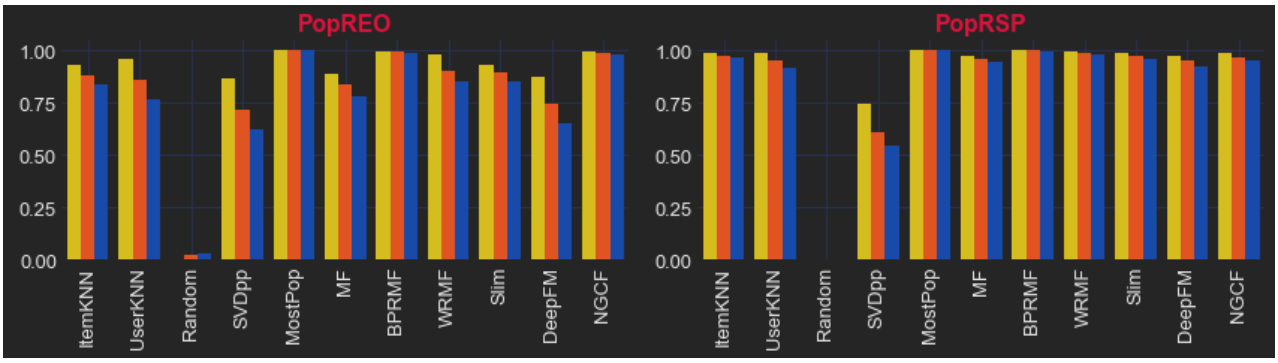
Εικόνα 5: Ανάλυση υπερπαραμέτρων στον αλγόριθμο WRMF για το σύνολο δεδομένων ml100k


Cutoff
10
50
100
nDCG
Precision
Recall
ItemCoverage
EPC
Gini
HR
ACLT
ARP
APLT
ItemKNN
UserKNN
Random
SVDpp
MostPop
MF
BPRMF
WRMF
Slim
DeepFM
NGCF



Εικόνα 6: Ανάλυση υπερπαραμέτρων στον αλγόριθμο WRMF για το σύνολο δεδομένων Amazon


Cutoff
10
50
100
nDCG
Precision
Recall
ItemCoverage
EPC
Gini
HR
ACLT
ARP
APLT
ItemKNN
UserKNN
Random
SVDpp
MostPop
MF
BPRMF
WRMF
Slim
DeepFM
NGCF



Εικόνα 7: Ανάλυση υπερπαραμέτρων στον αλγόριθμο WRMF για το σύνολο δεδομένων ML1M

Βιβλιογραφία

[1] K. Hao, "How Facebook got addicted to spreading misinformation," https://www.technologyreview.com/2021/03/11/1020600/facebook-responsible-ai-misinformation/.

[2] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[3] T. M. Mitchell, *Machine Learning*. McGraw-Hill Education, 1997.

[4] T. W. Malone, D. Rus, and R. Laubacher, "Artificial Intelligence and the Future of Work," *MIT Work of the Future*, p. 39.

[5] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics. Springer, 2009.

[7] M. H. Dunham, *Data Mining: Introductory and Advanced Topics*. USA: Prentice Hall PTR, 2002.

[8] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996.

[9] K. Falk, *Practical Recommender Systems*. Shelter Island, NY: Manning, 2019.

[10] C. C. Aggarwal, *Recommender Systems*. Cham: Springer International Publishing, 2016.

[11] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

[12] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997.

[13] D. W. Oard and J. Kim, "Implicit feedback for recommender systems," in *Proceedings of the AAAI Workshop on Recommender Systems*, vol. 83. AAAI, 1998, pp. 81–83.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work - CSCW '94*. Chapel Hill, North Carolina, United States: ACM Press, 1994, pp. 175–186.

[15] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[16] R. A. Beauregard, *A First Course in Linear Algebra; with Optional Introduction to Groups, Rings, and Fields*. Boston, Houghton Mifflin, 1973.

[17] R. Bellman, R. Bellman, and R. Corporation, *Dynamic Programming*, ser. Rand Corporation Research Study. Princeton University Press, 1957.

[18] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[19] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, Feb. 2019.

[20] K. H. Rosen, *Discrete Mathematics and Its Applications*, 5th ed. McGraw-Hill Higher Education, 2002.

[21] P. Goyal and E. Ferrara, "Graph Embedding Techniques, Applications, and Performance: A Survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, Jul. 2018.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[24] G. Adomavicius and J. Zhang, "Impact of data characteristics on recommender systems performance," *ACM Transactions on Management Information Systems*, vol. 3, no. 1, pp. 1–17, Apr. 2012.

[25] Y. Deldjoo, A. Bellogin, and T. Di Noia, "Explaining recommender systems fairness and accuracy through the lens of data characteristics," *Information Processing & Management*, vol. 58, no. 5, p. 102662, Sep. 2021.

[26] N. Idrissi and A. Zellou, "A systematic literature review of sparsity issues in recommender systems," *Social Network Analysis and Mining*, vol. 10, no. 1, p. 15, Feb. 2020.

[27] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '02. New York, NY, USA: Association for Computing Machinery, Aug. 2002, pp. 253–260.

[28] P. Brusilovsky, A. Kobsa, W. Nejdl, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, and G. Weikum, Eds., *The Adaptive Web: Methods and Strategies of Web Personalization*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321.

[29] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 115:1–115:35, Jul. 2021.

[30] S. Mattu, J. Angwin, J. Larson, and L. Kirchner, "Machine Bias," https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing?token=Gg58888u2U5db3W3CsuKrD0LD_VQJReQ, 2016.

[31] "Equality and Diversity Policy," https://www.amnesty.org.uk/equality-and-diversity-policy.

[32] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ser. ITCS '12. New York, NY, USA: Association for Computing Machinery, Jan. 2012, pp. 214–226.

[33] M. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 2564–2572.

[34] M. Hardt, E. Price, and N. Srebro, "Equality of Opportunity in Supervised Learning," *Advances in neural information processing systems,*, vol. 29, Oct. 2016.

[35] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, "Counterfactual Fairness," *Advances in neural information processing systems*, vol. 30, Mar. 2018.

[36] S. Caton and C. Haas, "Fairness in Machine Learning: A Survey," *arXiv:2010.04053 [cs, stat]*, Oct. 2020.

[37] J. A. Kroll, J. Huey, S. Barocas, E. W. Felten, J. R. Reidenberg, D. G. Robinson, and H. Yu, "Accountable Algorithms," *University of Pennsylvania Law Review*, vol. 165, p. 74.

[38] N. Diakopoulos, "Accountability in algorithmic decision making," *Communications of the ACM*, vol. 59, no. 2, pp. 56–62, Jan. 2016.

[39] F. Pasquale, *The Black Box Society: The Secret Algorithms That Control Money and Information*. Harvard University Press, Jan. 2015.

[40] R. Adams, S. Weale, and C. Barr, "A-level results: Almost 40% of teacher assessments in England downgraded," *The Guardian*, Aug. 2020.

[41] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," *arXiv:1702.08608 [cs, stat]*, Mar. 2017.

[42] D. Doran, S. Schulz, and T. R. Besold, "What Does Explainable AI Really Mean? A New Conceptualization of Perspectives," *CEUR Workshop Proceedings*, vol. 2071, Mar. 2018.

[43] C. Molnar, *Interpretable Machine Learning*, 2019.

[44] A. F. Markus, J. A. Kors, and P. R. Rijnbeek, "The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies," *Journal of Biomedical Informatics*, vol. 113, p. 103655, Jan. 2021.

[45] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI Fairness 360: An Extensible Toolkit for Detecting,

Understanding, and Mitigating Unwanted Algorithmic Bias," *arXiv:1810.01943 [cs]*, Oct. 2018.

[46] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker, "Fairlearn: A toolkit for assessing and improving fairness in AI," Microsoft, Tech. Rep. MSR-TR-2020-32, May 2020.

[47] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani, "Aequitas: A Bias and Fairness Audit Toolkit," *arXiv:1811.05577 [cs]*, Apr. 2019.

[48] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and Debias in Recommender System: A Survey and Future Directions," *arXiv:2010.03240 [cs]*, Oct. 2020.

[49] R. B. Cialdini and N. J. Goldstein, "Social influence: Compliance and conformity," *Annual Review of Psychology*, vol. 55, pp. 591–621, 2004.

[50] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling User Interest and Conformity for Recommendation with Causal Embedding," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 2980–2991.

[51] C. Lin, X. Liu, G. Xv, and H. Li, "Mitigating Sentiment Bias for Recommender Systems," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, Jul. 2021, pp. 31–40.

[52] S. Yao and B. Huang, "Beyond parity: Fairness objectives for collaborative filtering," in *Advances in Neural Information Processing Systems*, ser. NIPS'17, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 2925–2934.

[53] Y. Li, Y. Ge, and Y. Zhang, "CIKM 2021 Tutorial on Fairness of Machine Learning in Recommender Systems," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. Virtual Event Queensland Australia: ACM, Oct. 2021, pp. 4857–4860.

[54] R. Burke, "Multisided Fairness for Recommendation," *arXiv:1707.00093 [cs]*, Jul. 2017.

[55] Y. Li, H. Chen, S. Xu, Y. Ge, and Y. Zhang, "Towards Personalized Fairness based on Causal Notion," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, Jul. 2021, pp. 1054–1063.

[56] Y. Ge, S. Liu, R. Gao, Y. Xian, Y. Li, X. Zhao, C. Pei, F. Sun, J. Ge, W. Ou, and Y. Zhang, "Towards Long-term Fairness in Recommendation," *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 445–453, Mar. 2021.

[57] M. Elahi, H. Abdollahpouri, M. Mansoury, and H. Torkamaan, "Beyond Algorithmic Fairness in Recommender Systems," in *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. Utrecht Netherlands: ACM, Jun. 2021, pp. 41–46.

[58] Y.-J. Park and A. Tuzhilin, "The long tail of recommender systems and how to leverage it," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, ser. RecSys '08. New York, NY, USA: Association for Computing Machinery, Oct. 2008, pp. 11–18.

[59] C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[60] C. R. Sunstein, *Echo Chambers: Bush V. Gore, Impeachment, and Beyond*. Princeton University Press, 2001.

[61] G. Ramos, L. Boratto, and C. Caleiro, "On the negative impact of social influence in recommender systems: A study of bribery in collaborative hybrid algorithms," *Information Processing & Management*, vol. 57, no. 2, p. 102058, Mar. 2020.

[62] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, Jun. 2016.

[63] G. L. Ciampaglia, A. Nematzadeh, F. Menczer, and A. Flammini, "How algorithmic popularity bias hinders or promotes quality," *Scientific Reports*, vol. 8, no. 1, p. 15951, Dec. 2018.

[64] L. Boratto, G. Fenu, and M. Marras, "The Effect of Algorithmic Bias on Recommender Systems for Massive Open Online Courses," in *Advances in Information Retrieval*, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds. Cham: Springer International Publishing, 2019, vol. 11437, pp. 457–472.

[65] B. Rastegarpanah, K. P. Gummadi, and M. Crovella, "Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems," *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 231–239, Jan. 2019.

[66] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Recommendation Independence," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. PMLR, Jan. 2018, pp. 187–201.

[67] R. Borges and K. Stefanidis, "Enhancing Long Term Fairness in Recommendations with Variational Autoencoders," in *Proceedings of the 11th International Conference on Management of Digital EcoSystems*. Limassol Cyprus: ACM, Nov. 2019, pp. 95–102.

[68] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational Autoencoders for Collaborative Filtering," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2018, pp. 689–698.

[69] W. Liu and R. Burke, "Personalizing Fairness-aware Re-ranking," *arXiv:1809.02921 [cs]*, Sep. 2018.

[70] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: Association for Computing Machinery, Apr. 2010, pp. 881–890.

[71] H. Steck, "Calibrated recommendations," in *Proceedings of the 12th ACM Conference on Recommender Systems*. Vancouver British Columbia Canada: ACM, Sep. 2018, pp. 154–162.

[72] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "FA*IR: A Fair Top-k Ranking Algorithm," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1569–1578, Nov. 2017.

[73] L. Boratto, G. Fenu, and M. Marras, "Connecting user and item perspectives in popularity debiasing for collaborative recommendation," *Information Processing & Management*, vol. 58, no. 1, p. 102387, Jan. 2021.

[74] "Streamlit • The fastest way to build and share data apps," https://streamlit.io/.

[75] N. Hug, "Surprise: A Python library for recommender systems," *Journal of Open Source Software*, vol. 5, no. 52, p. 2174, Aug. 2020.

[76] M. Mansoury, R. Burke, A. Ordonez-Gauger, and X. Sepulveda, "Automating recommender systems experimentation with librec-auto," in *Proceedings of the 12th ACM Conference on Recommender Systems*. Vancouver British Columbia Canada: ACM, Sep. 2018, pp. 500–501.

[77] V. W. Anelli, A. Bellogin, A. Ferrara, D. Malitesta, F. A. Merra, C. Pomo, F. M. Donini, and T. Di Noia, "Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual Event Canada: ACM, Jul. 2021, pp. 2405–2414.

[78] Z. Sun, D. Yu, H. Fang, J. Yang, X. Qu, J. Zhang, and C. Geng, "Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison," in *Fourteenth ACM Conference on Recommender Systems*. Virtual Event Brazil: ACM, Sep. 2020, pp. 23–32.

[79] M. D. Ekstrand, "LensKit for Python: Next-Generation Software for Recommender System Experiments," *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2999–3006, Oct. 2020.

[80] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015.

[81] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based Recommendations on Styles and Substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 43–52.

[82] F. Aiolli, "Efficient top-n recommendation for very large scale binary rated datasets," in *Proceedings of the 7th ACM Conference on Recommender Systems*. Hong Kong China: ACM, Oct. 2013, pp. 273–280.

[83] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[84] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: Association for Computing Machinery, Aug. 2008, pp. 426–434.

[85] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, USA: AUAI Press, Jun. 2009, pp. 452–461.

[86] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," in *2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy: IEEE, Dec. 2008, pp. 263–272.

[87] X. Ning and G. Karypis, "SLIM: Sparse Linear Methods for Top-N Recommender Systems," in *2011 IEEE 11th International Conference on Data Mining*. IEEE, Dec. 2011, pp. 497–506.

[88] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

[89] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural Graph Collaborative Filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Paris France: ACM, Jul. 2019, pp. 165–174.

[90] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," p. 6.

[91] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. Melbourne, Australia: AAAI Press, Aug. 2017, pp. 1725–1731.

[92] S. Rendle, "Factorization Machines," in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 995–1000.

[93] N. Hoffmann, *Simulation Neuronaler Netze*. Wiesbaden: Vieweg+Teubner Verlag, 1992.

[94] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, Oct. 2002.

[95] ——, "IR evaluation methods for retrieving highly relevant documents," *ACM SIGIR Forum*, vol. 51, no. 2, p. 8, 2000.

[96] M. Deshpande and G. Karypis, "Item-Based Top-N Recommendation Algorithms," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, Jan. 2004.

[97] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, "Challenging the long tail recommendation," *Proceedings of the VLDB Endowment*, vol. 5, no. 9, pp. 896–907, May 2012.

[98] H. Abdollahpouri, R. Burke, and B. Mobasher, "Controlling Popularity Bias in Learning-to-Rank Recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como Italy: ACM, Aug. 2017, pp. 42–46.

[99] ——, "Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking," in *The Thirty-Second International Flairs Conference*, May 2019.

[100] Z. Zhu, J. Wang, and J. Caverlee, "Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual Event China: ACM, Jul. 2020, pp. 449–458.

[101] C. Gini, "Measurement of Inequality of Incomes," *The Economic Journal*, vol. 31, no. 121, pp. 124–126, 1921.

[102] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the Fifth ACM Conference on Recommender Systems - RecSys '11*. Chicago, Illinois, USA: ACM Press, 2011, p. 109.

[103] G. Adomavicius and YoungOk Kwon, "Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, May 2012.

[104] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," in *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France., 2017.

[105] M. F. Dacrema, S. Boglio, P. Cremonesi, and D. Jannach, "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research," *ACM Transactions on Information Systems*, vol. 39, no. 2, pp. 1–49, Mar. 2021.