# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

**Σχολή Χρηματοοικονομικών και Στατιστικής**

**Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης**

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΣΤΗΝ

**ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ**


# ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΕΩΝ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ: ΤΕΧΝΙΚΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ


Σωτήριος Κ. Τρούσας


Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς

Δεκέμβριος 2023

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίσθηκε από τη ΓΣΕΣ του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμό 20/13 - 7 - 2022 συνεδρίασή του σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική

Τα μέλη της Επιτροπής ήταν:

- Κούτρας Μάρκος, Καθηγητής (Επιβλέπων)

- Θεοδωρίδης Ιωάννης, Καθηγητής

- Πελέκης Νικόλαος, Αναπληρωτής Καθηγητής

Η έγκριση της Διπλωματικής Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

# UNIVERSITY OF PIRAEUS

## School of Finance and Statistics

## Department of Statistics and Insurance Science

POSTGRADUATE PROGRAM IN **APPLIED STATISTICS**


# RECOMMENDATION SYSTEMS IN DATA SCIENCE: TECHNIQUES AND APPLICATIONS


Sotirios K. Trousas


MSc Dissertation submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfillment of the requirements for the degree of Master of Science in *Applied Statistics*


Piraeus, Greece

December 20223

*Στην Κατερίνα*

*και στην οικογένειά μου,*

# Ευχαριστίες

Σε αυτή την ενότητα θα ήθελα να ευχαριστήσω τόσο τα άτομα που με βοήθησαν στην εκπόνηση της παρούσας διπλωματικής εργασίας, όσο και αυτούς που με βοήθησαν κατά τη διάρκεια των σπουδών μου στο Π.Μ.Σ. Εφαρμοσμένη Στατιστική. Αρχικά, θα ήθελα να ευχαριστήσω τον Καθηγητή του τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς, κύριο Μάρκο Κούτρα, για την ανάθεση του θέματος και την πολύτιμη συνεισφορά του ώστε να λάβει η διπλωματική μου εργασία την παρούσα μορφή της. Στη συνέχεια θα ήθελα να ευχαριστήσω, τα μέλη της τριμελούς συμβουλευτικής επιτροπής, τους Καθηγητή κύριο Ιωάννη Θεοδωρίδη και Αναπληρωτή Καθηγητή κύριο Νικόλαο Πελέκη, για το χρόνο που αφιέρωσαν στη διόρθωση της διπλωματικής εργασίας. Ευχαριστώ επίσης, το συμφοιτητή μου Γεώργιο Κουνιό για τις εποικοδομητικές συζητήσεις που είχαμε κατά τη διάρκεια των σπουδών μας στο Πανεπιστήμιο Πειραιώς, τον παιδικό μου φίλο και συνάδελφο Νικόλαο Βαρελά για τη βοήθεια και τις χρήσιμες συμβουλές του σε σχέση με το αντικείμενο της παρούσας εργασίας, αλλά και την οικογένειά μου που βρίσκεται πάντα δίπλα μου και με στηρίζει σε κάθε φάση της ζωής μου. Τέλος, ευχαριστώ από καρδιάς τη σύντροφό μου Κατερίνα για την υποστήριξη που μου προσέφερε και συνεχίζει να μου προσφέρει εδώ και χρόνια, η οποία σε μεγάλο βαθμό ευθύνεται για πολλά από αυτά που έχω πετύχει μέχρι σήμερα.


Για την παρούσα διπλωματική εργασία χρησιμοποιήθηκε (και) εξοπλισμός του εργαστηρίου Στατιστικής που αποκτήθηκε με χρηματοδότηση της πράξης «Προμήθεια και Εγκατάσταση Εξειδικευμένου Ερευνητικού Εξοπλισμού στο Πανεπιστήμιο Πειραιώς» (MIS 5066760) του Περιφερειακού Επιχειρησιακού Προγράμματος «Αττική 2014-2020»

# Περίληψη

Τα Συστήματα Συστάσεων (RS) είναι εξαιρετικά δημοφιλή στη σύγχρονη εποχή. Ίσως δεν είναι υπερβολή να τα χαρακτηρίσουμε ως ένα από τα πιο ισχυρά εργαλεία της Μηχανικής Μάθησης, αφού εφαρμόζονται σήμερα σε ευρεία κλίμακα και πληθώρα τομέων. Ένας από τους χαρακτηριστικότερους τομείς είναι το ηλεκτρονικό εμπόριο, όπου τα Συστήματα Συστάσεων χρησιμοποιούνται με σκοπό να προωθήσουν και να αυξήσουν τις πωλήσεις των ηλεκτρονικών καταστημάτων.

Τα Συστήματα Συστάσεων στοχεύουν στην πρόβλεψη των ενδιαφερόντων των χρηστών και στη δημιουργία προτάσεων που είναι πιθανό να επιλέξει ο χρήστης. Τα δεδομένα που απαιτούνται για τη λειτουργία ενός τέτοιου συστήματος πηγάζουν συνήθως από αναζητήσεις στις μηχανές αναζήτησης καθώς και το ιστορικό αγορών ή από άλλες πληροφορίες σχετικά με τους ίδιους τους χρήστες και τα προϊόντα που αυτοί έχουν επιλέξει στο παρελθόν. Ιστότοποι όπως το Google, το Netflix, το Spotify, η Amazon κτλ χρησιμοποιούν τέτοια δεδομένα για τη δημιουργία συστάσεων, οι οποίες δεν είναι κοινές σε όλους τους χρήστες, αλλά εξατομικευμένες, βασιζόμενες στις προτιμήσεις του κάθε χρήστη.

Στόχος αυτής της εργασίας είναι η θεωρητική περιγραφή των Συστημάτων Συστάσεων και η εφαρμογή, στη συνέχεια, της μεθοδολογίας που θα περιγραφεί σε ένα σύνολο πραγματικών δεδομένων ταινιών. Έτσι, στα της παρούσας διπλωματικής, αφού ολοκληρώθηκε η στατιστική ανάλυση σχετικών δεδομένων, κατασκευάστηκε ένα RS που ανήκει στην κατηγορία Content Based Recommender Systems, χρησιμοποιώντας ένα μοντέλο λογιστικής παλινδρόμησης. Στη συνέχεια, κατασκευάστηκαν δύο RSs της κατηγορίας Collaborative Filtering. Το πρώτο, με τη βοήθεια της τεχνικής Singular Value Decomposition-SVD και το δεύτερο με τη βοήθεια της τεχνικής Alternating Least Squares-ALS. Η σύγκριση και των τριών RSs έγινε με χρήση των μετρικών απόδοσης Precision, Recall και F1-Score. Το σύστημα συστάσεων με την καλύτερη απόδοση ήταν αυτό της τεχνικής Singular Value Decomposition, και ως εκ τούτου ήταν αυτό που χρησιμοποιήθηκε για να παραχθούν συστάσεις.

# Abstract

Recommendation Systems are extremely popular in the modern era. It may not be an exaggeration to characterize them as one of the most powerful tools of Machine Learning, as they are applied today on a broad scale and in a plethora of domains. One of the most characteristic areas is e-commerce, where Recommendation Systems are used to promote and increase the sales of online stores. Recommendation Systems aim to predict the interests of users and create personalized recommendations based on individual's preferences. The data required for the operation of such a system usually come from search engine queries, as well as the purchase history or other information about the users themselves and the products they have chosen in the past. Websites such as Google, Netflix, Spotify, Amazon, etc., use such data to create recommendations that are not common to all users but personalized, based on each user's preferences.

The goal of this Thesis is the theoretical description of Recommendation Systems and then the application of the aforementioned methodology on a set of real movie data. We first catty over statistical analysis of the available data and then, a Recommendation System (RS) is constructed based on the Content-Based Recommender Systems category, using a logistic regression model. Then, two RSs belonging to the Collaborative Filtering category are constructed. The first one uses the Singular Value Decomposition-SVD technique, and the second one the Alternating Least Squares-ALS technique. A comparison of the three RSs is then performed using evaluation metrics like Precision, Recall and F1-Score. The best-performing RS was the one based on Singular Value Decomposition (SVD) technique; therefore, it was exploited to generate recommendations.

# Περιεχόμενα

# Κατάλογος Σχημάτων

# Κατάλογος Πινάκων

# Κατάλογος Συντομογραφιών

**RS:** Recommender System

**CF:** Collaborative Filtering

**CBF:** Content Based Filtering

**MF:** Matrix Factorization

**SVD:** Singular Value Decomposition

**ALS:** Alternating Least Squares

# ΚΕΦΑΛΑΙΟ 1
# Εισαγωγή

Τα Συστήματα Συστάσεων (Recommender Systems, RS) είναι εργαλεία και τεχνικές λογισμικού που παρέχουν προτάσεις για αντικείμενα που μπορεί να είναι χρήσιμα σε έναν χρήστη [1, 2, 3]. Οι προτάσεις αφορούν διάφορες διαδικασίες λήψης αποφάσεων, όπως ποια αντικείμενα να αγοράσει, ποια ταινία να επιλέξει να παρακολουθήσει ή ποια διαδικτυακά νέα να διαβάσει.

Ο όρος "αντικείμενο" χρησιμοποιείται γενικά για να υποδηλώσει αυτό που το σύστημα συστάσεων προτείνει στους χρήστες. Ένα RS εστιάζει συνήθως σε ένα συγκεκριμένο τύπο αντικειμένου (π.χ Movies ή Songs) και ανάλογα με τη σχεδίασή του, τη γραφική διασύνδεση του χρήστη και την κεντρική τεχνική που χρησιμοποιείται για τη δημιουργία των προτάσεων προσαρμόζεται ώστε να παρέχει χρήσιμες και αποτελεσματικές προτάσεις για αυτόν τον συγκεκριμένο τύπο αντικειμένου.

Τα RSs είναι κυρίως στραμμένα προς άτομα που δεν έχουν αρκετή προσωπική εμπειρία ή δεξιότητα για να αξιολογήσουν το ενδεχομένως προχωρημένο αριθμό εναλλακτικών αντικειμένων που ένας ιστότοπος, για παράδειγμα, μπορεί να προσφέρει [2]. Ένα παράδειγμα αποτελεί ένα σύστημα συστάσεων βιβλίων που βοηθά τους χρήστες να επιλέξουν ένα βιβλίο για να διαβάσουν. Στο δημοφιλές Amazon.com, ο ιστότοπος χρησιμοποιεί ένα σύστημα συστάσεων για να προσαρμόσει το διαδικτυακό κατάστημα για κάθε πελάτη [4]. Δεδομένου ότι οι προτάσεις είναι συνήθως εξατομικευμένες, διάφοροι χρήστες ή ομάδες χρηστών λαμβάνουν διάφορες προτάσεις. Επιπλέον, υπάρχουν και μη-εξατομικευμένες προτάσεις. Αυτές είναι απλούστερες στο να δημιουργηθούν και συνήθως παρουσιάζονται σε περιοδικά ή εφημερίδες. Τα τυπικά παραδείγματα περιλαμβάνουν τις δέκα κορυφαίες επιλογές βιβλίων, ταινιών κλπ. Παρόλο που μπορεί να είναι χρήσιμες και αποτελεσματικές σε ορισμένες καταστάσεις, αυτού του είδους οι μη-εξατομικευμένες προτάσεις συνήθως δεν αποτελούν αντικείμενο της έρευνας η οποία γίνεται στα πλαίσια των συστημάτων συστάσεων.

Στην πιο απλή τους μορφή, οι εξατομικευμένες προτάσεις προσφέρονται στη μορφή ιεραρχημένων καταλόγων από αντικείμενα. Κατά την εκτέλεση αυτής της ιεράρχησης, τα

συστήματα συστάσεων προσπαθούν να προβλέψουν ποια προϊόντα ή υπηρεσίες είναι τα πιο κατάλληλα, με βάση τις προτιμήσεις και τους περιορισμούς του χρήστη. Για να ολοκληρώσουν αυτή την υπολογιστική εργασία, τα συστήματα συστάσεων συλλέγουν από τους χρήστες τις προτιμήσεις τους, που είναι είτε ρητά εκφρασμένες, π.χ. ως αξιολογήσεις προϊόντων, είτε έχουν εξαχθεί μέσω της ερμηνείας των ενεργειών του χρήστη. Για παράδειγμα, ένα σύστημα συστάσεων μπορεί να θεωρήσει την πλοήγηση σε μια συγκεκριμένη σελίδα προϊόντος ως έμμεσο σημάδι προτίμησης για τα αντικείμενα που εμφανίζονται σε εκείνη τη σελίδα.

Η ανάπτυξη των συστημάτων συστάσεων ξεκίνησε από μια αρκετά απλή παρατήρηση: οι άνθρωποι συχνά βασίζονται σε προτάσεις που τους κάνουν άλλοι για να πάρουν αποφάσεις στην καθημερινή τους ζωή [1, 5]. Για παράδειγμα, είναι συνηθισμένο να βασιζόμαστε στις συστάσεις των φίλων μας όταν επιλέγουμε ένα βιβλίο να διαβάσουμε, οι εργοδότες βασίζονται στις συστατικές επιστολές για τις προσλήψεις προσωπικού, και όταν επιλέγουμε μία ταινία να παρακολουθήσουμε, τείνουμε να διαβάζουμε και να βασιζόμαστε στις βαθμολογίες ταινιών που έχει γράψει ένας κριτικός κινηματογράφου.

Κατά την προσπάθειά τους να μιμηθούν αυτή τη συμπεριφορά, τα πρώτα συστήματα συστάσεων εφάρμοσαν αλγόριθμους για να εκμεταλλευτούν τις προτάσεις που παρήγαγε μια κοινότητα χρηστών, προκειμένου να παρέχουν προτάσεις σε έναν ενεργό χρήστη, δηλαδή ένα χρήστη που ψάχνει για προτάσεις. Οι προτάσεις ήταν για αντικείμενα που είχαν αρέσει σε παρόμοιους χρήστες (εκείνους με παρόμοιες προτιμήσεις). Αυτή η προσέγγιση ονομάζεται συνεργατικό φιλτράρισμα (Collaborative Filtering), και η λογική της είναι ότι, εάν ο ενεργός χρήστης συμφώνησε στο παρελθόν με ορισμένους άλλους χρήστες, τότε και οι άλλες προτάσεις που προέρχονται από παρόμοιους χρήστες θα είναι σχετικές και ενδιαφέρουσες για τον ενεργό χρήστη.

Καθώς οι ιστότοποι ηλεκτρονικού εμπορίου άρχισαν να αναπτύσσονται, δημιουργήθηκε η ανάγκη για παροχή προτάσεων που προέρχονται από το φιλτράρισμα του πλήρους φάσματος των διαθέσιμων εναλλακτικών. Οι χρήστες είχαν δυσκολία να διαλέξουν τις πιο κατάλληλες επιλογές από την τεράστια ποικιλία αντικειμένων (προϊόντα και υπηρεσίες) που προσέφεραν αυτοί οι διαδικτυακοί ιστότοποι.

Η εκρηκτική αύξηση και ποικιλία των πληροφοριών που είναι διαθέσιμες στον Παγκόσμιο Ιστό και η γρήγορη εισαγωγή νέων υπηρεσιών ηλεκτρονικού εμπορίου (αγορά προϊόντων, σύγκριση προϊόντων, δημοπρασία κ.λπ.) συχνά επιβαρύνει υπερβολικά τους χρήστες, οδηγώντας τους να λαμβάνουν κακές αποφάσεις. Η μεγάλη διαθεσιμότητα επιλογών, αντί να παράγει οφέλη, άρχισε να μειώνει την ευκολία των χρηστών στο να μπορούν να επιλέξουν. Κατανοήθηκε ότι, ενώ το να έχει κάποιος επιλογές είναι καλό, το να έχει περισσότερες επιλογές δεν είναι πάντα καλύτερο. Πιο συγκεκριμένα, η υπερβολή των διαθέσιμων επιλογών, συχνά δημιουργεί σύγχυση στον καταναλωτή και αντί να του προκαλεί αίσθημα ευφορίας τον δυσαρεστεί και τον μπερδεύει [6].

Τα συστήματα συστάσεων έχουν αποδειχθεί τα τελευταία χρόνια ένα αξιόλογο μέσο για την αντιμετώπιση του προβλήματος της υπερφόρτωσης πληροφοριών. Τελικά, ένα σύστημα συστάσεων αντιμετωπίζει αυτό το φαινόμενο καθοδηγώντας έναν χρήστη προς νέα αντικείμενα στα οποία δεν έχει ακόμα εμπειρία και που ενδεχομένως να είναι σχετικά με την τρέχουσα εργασία του χρήστη. Τα αιτήματα ενός χρήστη προέρχονται από τις ανάγκες και τις προσωπικές του επιλογές, για αυτό το λόγο τα συστήματα συστάσεων δημιουργούν προτάσεις χρησιμοποιώντας διάφορους τύπους γνώσης και δεδομένων σχετικά με αυτούς, τα διαθέσιμα αντικείμενα και τις προηγούμενες συναλλαγές που αποθηκεύονται σε εξατομικευμένες βάσεις δεδομένων. Ο χρήστης μπορεί στη συνέχεια να περιηγηθεί στις προτάσεις. Μπορεί να τις αποδεχτεί ή όχι και μπορεί να παρέχει, αμέσως ή σε μια επόμενη φάση, μια σιωπηρή ή ρητή ανατροφοδότηση. Όλες αυτές οι ενέργειες και οι ανατροφοδοτήσεις των χρηστών μπορούν να αποθηκευτούν στη βάση δεδομένων του συστήματος συστάσεων και να χρησιμοποιηθούν για τη δημιουργία νέων προτάσεων στις επόμενες αλληλεπιδράσεις με το σύστημα.

# ΚΕΦΑΛΑΙΟ 2

# Συστήματα Συστάσεων – Τύποι και Τεχνικές

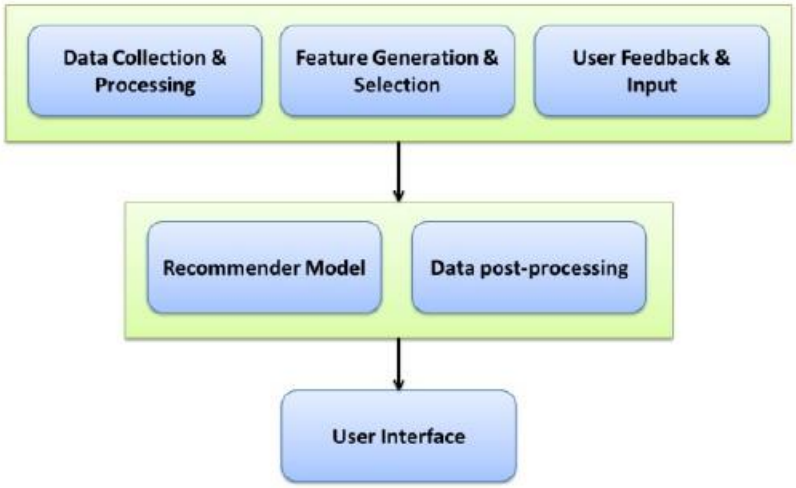
Τα συστήματα συστάσεων αποτελούν έναν από τους τομείς των συστημάτων φιλτραρίσματος πληροφοριών που έχουν προσελκύσει μεγάλο ερευνητικό ενδιαφέρον τις τελευταίες δεκαετίες και έχουν χρησιμοποιηθεί σε μια μεγάλη ποικιλία εφαρμογών, από εμπορικά ηλεκτρονικά καταστήματα (e-shops) έως κοινωνικά δίκτυα και ιστότοπους με βαθμολογίες προϊόντων. Καθώς η εφαρμοσιμότητα αυτών των εφαρμογών αυξάνεται συνεχώς, αυξάνεται επίσης το μέγεθος των γραφημάτων που αντιπροσωπεύουν τους χρήστες τους και υποστηρίζουν τη λειτουργικότητά τους. Τα τελευταία χρόνια, έχουν προταθεί διάφορες προσεγγίσεις για τον χειρισμό του προβλήματος της κλιμακούμενης απόδοσης των αλγορίθμων συστημάτων συστάσεων, ειδικά της ομάδας των αλγορίθμων CF [7–11].

Στην περίπτωση των ιστότοπων κριτικών προϊόντων ή αξιολογήσεων προϊόντων, η ανάλυση των δικτύων δύο συνιστωσών και οι πληροφορίες που μεταφέρουν έχουν προσελκύσει το ενδιαφέρον των ερευνητών στα συστήματα συστάσεων και έχουν οδηγήσει σε νέες λύσεις και αλγόριθμους. Τα συστήματα συστάσεων έχουν γίνει πολύ δημοφιλή σε ιστότοπους όπως το IMDB, το MovieLens και το Netflix, όπου οι χρήστες βαθμολογούν τις ταινίες που έχουν δει και λαμβάνουν προτάσεις για περισσότερες ταινίες που ίσως τους ενδιαφέρουν.

Τα συστήματα συστάσεων συνήθως λειτουργούν ως συστήματα που προσπαθούν να προβλέψουν τη βαθμολογία του χρήστη για οποιοδήποτε πιθανό αντικείμενο [12] ή τη γνώμη του χρήστη με βάση κάποιες πτυχές του αντικειμένου. Όπως φαίνεται στο Σχήμα 1, τα βασικά στοιχεία ενός συστήματος συστάσεων είναι τα εξής:

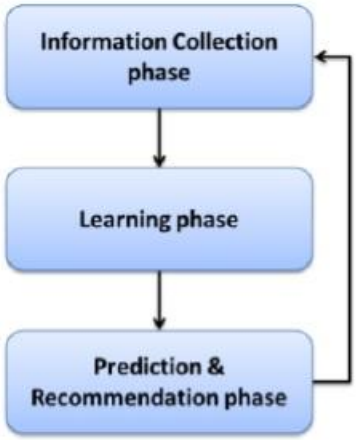
- Συλλογή & Επεξεργασία Δεδομένων: Αυτή είναι το στάδιο για τη συλλογή δεδομένων που συχνά είναι μεγάλα.
- Δημιουργία & Επιλογή Χαρακτηριστικών: Σε αυτό το στάδιο, οι αλγόριθμοι υλοποιούν τη δημιουργία και την επιλογή χαρακτηριστικών που μπορεί να γίνει είτε με προ-υπολογισμό αυτών των χαρακτηριστικών είτε δυναμικά με τη δημιουργία τους.

- Ανατροφοδότηση & Εισαγωγή Προτιμήσεων του Χρήστη: Σε αυτό το στάδιο, ο χρήστης προσκαλείται μέσω της διεπαφής του συστήματος να παρέχει βαθμολογίες για τα αντικείμενα.
- Μοντέλο Συστάσεων: Αυτό είναι το κύριο μέρος κάθε συστήματος συστάσεων που οργανώνει τον αλγόριθμο συστάσεων με όλα τα προηγούμενα δεδομένα που λαμβάνει.
- Επεξεργασία Δεδομένων μετά την Επεξεργασία: Η επεξεργασία δεδομένων μετά την επεξεργασία χρησιμοποιείται σε πολλά συστήματα για να βελτιστοποιήσει την απόδοση του RS σύμφωνα με τις καθορισμένες μετρικές και αυτό μπορεί να συμπεριλάβει τη λήψη υπόψη της ανατροφοδότησης του χρήστη.
- Διεπαφή Χρήστη: Αυτό είναι το τελευταίο βήμα κάθε συστήματος και αποτελεί το στοιχείο όπου ο χρήστης αξιοποιεί τις συστάσεις που έχουν δημιουργηθεί γι αυτόν από το σύστημα.




Σχήμα 1. Στάδια ενός συστήματος συστάσεων [13]

Από μια προοπτική που επικεντρώνεται στο σύστημα, η διαδικασία συστάσεων παίρνει ως είσοδο ένα σύνολο χρηστών και τις βαθμολογίες τους για μια συλλογή αντικειμένων και παράγει ένα εξατομικευμένο σύνολο προτεινόμενων αντικειμένων για κάθε χρήστη. Για να επιτευχθεί αυτό, ένα σύστημα συστάσεων λειτουργεί σε τρία στάδια, όπως απεικονίζεται στο Σχήμα 2 και αναλύεται παρακάτω.




Σχήμα 2. Φάσεις της διαδικασίας συστάσεων [13]

**Φάση συλλογής πληροφοριών:** Στο πρώτο στάδιο, το σύστημα συλλέγει κάθε σχετική πληροφορία για τους χρήστες προκειμένου να δημιουργήσει το προφίλ τους. Ανάλογα με τον τύπο του συστήματος, αυτές οι πληροφορίες πρέπει να έχουν ένα ελάχιστο μέγεθος και λεπτομέρειες, ώστε να ετοιμαστεί το επιθυμητό μοντέλο που θα χρησιμοποιηθεί στη φάση της σύστασης. Υπάρχουν συστήματα με συγκεκριμένα πρωτόκολλα συλλογής πληροφοριών, υπό τα οποία χρησιμοποιούν αυτές τις πληροφορίες για να καθορίσουν την τρέχουσα κατάσταση της γνώσης και υποστηρίζουν τη λήψη αποφάσεων για την εκμάθηση οποιασδήποτε διαθέσιμης πληροφορίας προφίλ χρηστών για όλους τους χρήστες που συμμετέχουν στη διαδικασία σύστασης. Οι πληροφορίες αυτές χρησιμοποιούνται ως είσοδος από τα Συστήματα Συστάσεων προκειμένου να δημιουργήσουν μια πλήρη εικόνα των χρηστών τους. Τέτοιου είδους πληροφορίες είναι, είτε υψηλής ποιότητας σαφείς ανατροφοδοτήσεις, που περιλαμβάνουν τη σαφή είσοδο των χρηστών σχετικά με το ενδιαφέρον τους για τα αντικείμενα, είτε σιωπηρές ανατροφοδοτήσεις χρησιμοποιώντας την εκτίμηση των προτιμήσεων των χρηστών έμμεσα μέσω της παρατήρησης της συμπεριφοράς των χρηστών [14].

**Φάση εκμάθησης:** Στο δεύτερο στάδιο, το σύστημα παρέχει τις συλλεγμένες πληροφορίες σε έναν αλγόριθμο μάθησης που φιλτράρει και αξιοποιεί τα χαρακτηριστικά/γνωρίσματα των χρηστών που εξυπηρετούν με τον βέλτιστο τρόπο τη φάση της σύστασης. Με άλλα λόγια, το σύστημα εξάγει τα πιο αντιπροσωπευτικά χαρακτηριστικά και εκπαιδεύει και κατασκευάζει το μοντέλο που αναγνωρίζει και ποσοτικοποιεί καλύτερα τη σχέση μεταξύ των χρηστών και των "αντικειμένων" για τα οποία το σύστημα σύστασης θα

δημιουργήσει προτάσεις, που στην πραγματικότητα αποτελεί μια αφαίρεση της σχέσης μεταξύ των αντικειμένων και των χρηστών.

**Φάση πρόβλεψης και σύστασης:** Αυτή η τελευταία φάση προβλέπει και/ή προτείνει ποιου είδους αντικείμενα μπορεί να προτιμά ο χρήστης. Αυτό μπορεί να γίνει είτε απευθείας, βασισμένο στα δεδομένα που συλλέχθηκαν στο πρώτο στάδιο συλλογής πληροφοριών, το οποίο οδηγεί σε μεθόδους μνήμης ή βασισμένες σε μοντέλο, είτε συνδυασμένο με δεδομένα σχετικά με άλλες δραστηριότητες και προτιμήσεις των χρηστών, οδηγώντας σε υβριδικές προσεγγίσεις [15]. Οι αντιδράσεις του χρήστη στα προτεινόμενα αντικείμενα καταγράφονται διαρκώς και χρησιμοποιούνται ως ανατροφοδότηση που βελτιώνει την απόδοση του συστήματος συστάσεων με την πάροδο του χρόνου.

Υπάρχουν τρεις κύριες κατηγορίες τεχνικών σύστασης οι οποίες καθορίζονται με βάση τις πληροφορίες που χρησιμοποιούνται για το φιλτράρισμα των αντικειμένων που δεν ενδιαφέρουν τον στόχο του χρήστη, την πρόβλεψη των βαθμολογιών των αντικειμένων και τη δημιουργία προτάσεων [16, 17].

i. **Τεχνικές Συνεργατικού Φιλτραρίσματος** (Collaborative Filtering, CF) που επικεντρώνονται κυρίως στις αξιολογήσεις χρηστών-αντικειμένων.
ii. **Τεχνικές Φιλτραρίσματος βασισμένες στο Περιεχόμενο** (Content Based Filtering, CBF) που χρησιμοποιούν επιπλέον περιεχόμενο για χρήστες και αντικείμενα και καθορίζουν διάφορες μετρικές ομοιότητας και μοντέλα ταιριάσματος για τη δημιουργία προτάσεων.
iii. **Τεχνικές υβριδικού τύπου** (Hybrid) που συνδυάζουν τα πλεονεκτήματα των δύο προηγούμενων τύπων.

## 2.1 Τεχνικές Συνεργατικού Φιλτραρίσματος

Το Συνεργατικό Φιλτράρισμα θεωρείται ως μία από τις κορυφαίες προσεγγίσεις για τη δημιουργία συστάσεων και γι' αυτό χρησιμοποιείται από μερικές από τις μεγαλύτερες εμπορικές πλατφόρμες. Η αναγνώριση της χρησιμότητάς της φαίνεται από το γεγονός ότι έχουν αναπτυχθεί πολλές παραλλαγές και τεχνικές για αυτόν τον σκοπό. Η βασική ιδέα πίσω από αυτήν την τεχνική είναι ότι ένας χρήστης παρέχει τις προτιμήσεις του στη μορφή βαθμολογιών για τα

διαθέσιμα αντικείμενα, είτε έμμεσα είτε άμεσα, και ένας πελάτης που φάνηκε να είχε παρόμοιες προτιμήσεις στο παρελθόν, πιθανόν να έχει ακόμα τις ίδιες. Η ομοιότητα των προτιμήσεων δύο χρηστών υπολογίζεται με βάση την ομοιότητα στο ιστορικό βαθμολογιών των χρηστών. Αυτός είναι ο λόγος που αναφέρεται το Συνεργατικό Φιλτράρισμα ως συσχέτιση ανθρώπου προς άνθρωπο. Ένα βασικό πλεονέκτημα αυτού του είδους τεχνικών είναι ότι δεν υπάρχει ή αν υπάρχει είναι περιορισμένη η ανάγκη για σημασιολογικές πληροφορίες προκειμένου να παραχθούν προτάσεις. Οι σημασιολογικές πληροφορίες περιλαμβάνουν δεδομένα ή πληροφορίες που σχετίζονται με τη σημασιολογία, δηλαδή τη σημασία ή το νόημα που κρύβεται πίσω από ένα κείμενο, ένα σύμβολο, μια έννοια ή ένα σύνολο δεδομένων. Αυτές οι πληροφορίες μπορεί να περιλαμβάνουν τον τρόπο με τον οποίο λέξεις ή φράσεις συσχετίζονται μεταξύ τους, τον τρόπο με τον οποίο ερμηνεύονται, καθώς και τις σχέσεις μεταξύ διαφορετικών στοιχείων που μπορούν να αναδειχθούν από τα δεδομένα. Αυτός είναι ο λόγος που είναι δημοφιλείς στις μεγάλες εφαρμογές που βασίζονται σε κοινωνικά δίκτυα, συμπεριλαμβανομένων των Amazon, Netflix, iTunes, IMDB κ.λπ.

Η βασική ιδέα πίσω από το CF είναι ότι οι χρήστες παρέχουν τις προτιμήσεις τους για τα διαθέσιμα αντικείμενα είτε άμεσα στη μορφή βαθμολογιών, είτε έμμεσα επιλέγοντας κάποια από αυτά στις αλληλεπιδράσεις τους και αγνοώντας τα υπόλοιπα. Με βάση αυτές τις πληροφορίες προτιμήσεων, δημιουργείται ένας πίνακας αξιολόγησης $m \times n$ που περιέχει αξιολογήσεις των $m$ χρηστών για τα $n$ αντικείμενα. Ο πίνακας αξιολόγησης χρησιμοποιείται ως βασική πληροφορία για τον προφίλ του χρήστη (ή του αντικειμένου), καθώς η σειρά των αξιολογήσεων του χρήστη (ή η στήλη του αντικειμένου) θεωρείται ενδεικτική των προτιμήσεών του. Οι αλγόριθμοι CF βασίζονται στην υπόθεση ότι οι χρήστες που έχουν συμπαθήσει (ή αντιπαθήσει) τα ίδια αντικείμενα στο παρελθόν πιθανόν να έχουν ακόμα τις ίδιες προτιμήσεις. Για να προτείνουν αντικείμενα σε έναν χρήστη, αναζητούν χρήστες με παρόμοιο πρότυπο αξιολόγησης αντικειμένου και προτείνουν όλα τα αντικείμενα που έχουν αξιολογηθεί υψηλά από αυτούς αλλά δεν έχουν αξιολογηθεί ακόμα από τον χρήστη. Οι αλγόριθμοι CF βασίζουν τη μέτρηση της ομοιότητας αντικειμένων στις αξιολογήσεις των χρηστών και προτείνουν στον χρήστη αντικείμενα που θυμίζουν (σε αξιολογήσεις) τα αντικείμενα που ο χρήστης έχει αξιολογήσει ψηλά.

Ο πίνακας αξιολόγησης στο CF είναι συνήθως αραιός, καθώς όλοι οι χρήστες δεν έχουν αξιολογήσει όλα τα αντικείμενα, και αυτό εγείρει αρκετά θέματα απόδοσης,

συμπεριλαμβανομένης της κλιμάκωσης, που σχετίζεται με την υψηλή διάσταση του πίνακα και το πρόβλημα της καθυστερημένης εκκίνησης (cold-start), που επηρεάζει τους χρήστες (ή τα αντικείμενα) χωρίς προηγούμενο ιστορικό αξιολογήσεων. Η κλιμάκωση στο CF αναφέρεται στη δυνατότητα επέκτασης ενός συστήματος συστάσεων ή πρόβλεψης ώστε να υποστηρίζει αυξημένο όγκο δεδομένων, χρηστών ή αντικειμένων. Για ένα σύστημα συστάσεων, ο cold -start των χρηστών συμβαίνει όταν ο συγκεκριμένος χρήστης είναι νέος και δεν έχει ακόμα κάνει αξιολογήσεις ή δεν έχει αλληλεπιδράσει με το σύστημα, με αποτέλεσμα να είναι δύσκολο να γίνουν συστάσεις βασισμένες στις προτιμήσεις του. [18, 19]. Για να αντιμετωπιστούν τα ζητήματα της υψηλής διάστασης και της αραιότητας του πίνακα αξιολόγησης, έχουν χρησιμοποιηθεί τεχνικές Matrix Factorization (MF) [20] για την αποσύνθεση του πίνακα αξιολόγησης χρήστη-αντικειμένου, σε γινόμενο δύο ορθογώνιων πινάκων χαμηλότερης διάστασης.

Πιο συγκεκριμένα η MF είναι μια τεχνική που χρησιμοποιείται σε αλγορίθμους συστάσεων για την ανάλυση μεγάλων πινάκων δεδομένων, όπως ο πίνακας αξιολογήσεων των χρηστών σε αντικείμενα ή προϊόντα. Η ιδέα βασίζεται στο ότι ένας μεγάλος πίνακας δεδομένων με αξιολογήσεις μπορεί να αναλυθεί σε δύο (ή περισσότερους) χαμηλότερων διαστάσεων πίνακες. Κατά τη διάσπαση αυτή, ο πίνακας αξιολογήσεων των χρηστών και των αντικειμένων αποσυνθέτεται σε πίνακες παραγόντων, όπου κάθε χρήστης και κάθε αντικείμενο αναπαρίσταται με ένα σύνολο παραγόντων. Αυτοί οι πίνακες παραγόντων μπορούν να αναπαρασταθούν ως διανύσματα χαρακτηριστικών για κάθε χρήστη και αντικείμενο. Με αυτόν τον τρόπο, η MF επιτρέπει την ανάλυση και την πρόβλεψη αξιολογήσεων που δεν υπάρχουν στον αρχικό πίνακα, αξιοποιώντας τις σχέσεις και τα πρότυπα που ανακαλύπτονται από τα χαρακτηριστικά.

Αυτό που μπορεί να ενδιαφέρει έναν χρήστη, δε σημαίνει απαραίτητα ότι είναι η καλύτερη σύσταση ανά πάσα στιγμή. Όσον αφορά τις συστάσεις βασισμένες σε μικρές στιγμές (micro-moments), είναι σημαντικό να είναι το αντικείμενο που συστήνεται στον χρήστη κατάλληλο για τη στιγμή, τον τόπο ή οποιονδήποτε άλλο τρέχον πλαίσιο για το οποίο προτείνεται. Αυτή η παρατήρηση οδήγησε στην έννοια των συστημάτων σύστασης βασισμένων στο πλαίσιο [21], η οποία έχει μελετηθεί εκτενώς τα τελευταία χρόνια, ιδιαίτερα λόγω της αυξανόμενης δημοφιλίας των κοινωνικών δικτύων που βασίζονται στην τοποθεσία [22, 23].

Τα συστήματα CF μπορούν να κατηγοριοποιηθούν σε δύο βασικές κατηγορίες: τα βασισμένα στη μνήμη (memory-based) που αναζητούν τους κορυφαίους *k* παρόμοιους χρήστες

(δηλαδή χρήστες με παρόμοιες αξιολογήσεις για τα αντικείμενα που αξιολογούνται κοινά) ή αντικείμενα (δηλαδή αντικείμενα που έχουν αξιολογηθεί παρόμοια με τα προτιμώμενα αντικείμενα από τον χρήστη) και χρησιμοποιούν μόνο αυτές τις πληροφορίες για τη δημιουργία των συστάσεών τους, και τα βασισμένα στο μοντέλο (model-based), τα οποία χρησιμοποιούν όλες τις αξιολογήσεις για να εκπαιδεύσουν ένα μοντέλο για την πρόβλεψη των προτιμήσεων του χρήστη για ένα αντικείμενο.

## 2.2 Τεχνικές Φιλτραρίσματος Βασισμένες στο Περιεχόμενο

Το CBF αναφέρεται επίσης ως γνωστικό φιλτράρισμα. Στις τεχνικές βασισμένες στο περιεχόμενο, ένα αντικείμενο συστήνεται σε έναν χρήστη όταν η ομοιότητα μεταξύ αυτού του αντικειμένου και των αντικειμένων στα οποία ο χρήστης έχει ήδη εκφράσει τις προτιμήσεις του στο παρελθόν είναι υψηλή [24, 25]. Συγκεκριμένα, η σύσταση γίνεται με τη σύγκριση της περιγραφής του περιεχομένου των αντικειμένων. Το περιεχόμενο κάθε αντικειμένου αναπαρίσταται ως ένα σύνολο περιγραφών ή λέξεων-κλειδιών, και το προφίλ του χρήστη αναπαρίσταται με τις ίδιες λέξεις-κλειδιά και δημιουργείται εξετάζοντας το περιεχόμενο των αντικειμένων που έχει δει στο παρελθόν. Για να χρησιμοποιηθεί το φιλτράρισμα βασισμένο στο περιεχόμενο, υπολογίζονται οι ομοιότητες για όλα τα αντικείμενα, τα αντικείμενα κατατάσσονται βάσει αυτών των ομοιοτήτων, και τα κορυφαία-$N$ αντικείμενα κατατάσσονται τελικά και προτείνονται στον στόχο χρήστη. Η ομοιότητα των αντικειμένων υπολογίζεται με βάση τα χαρακτηριστικά που σχετίζονται με τα συγκρινόμενα αντικείμενα. Για παράδειγμα, αν ένας χρήστης έχει αξιολογήσει θετικά ένα φιλμ που ανήκει στην κατηγορία της κωμωδίας, τότε το σύστημα μπορεί να μάθει να προτείνει άλλες ταινίες από την ίδια κατηγορία. Οι κλασικές τεχνικές συστάσεων βασισμένες στο περιεχόμενο στοχεύουν στην ταύτιση των χαρακτηριστικών του προφίλ του χρήστη με τα χαρακτηριστικά των αντικειμένων. Στις περισσότερες περιπτώσεις, τα χαρακτηριστικά των αντικειμένων είναι απλώς λέξεις-κλειδιά που εξάγονται από τις περιγραφές των αντικειμένων.

## 2.3 Τεχνικές Υβριδικού Τύπου

Οι υβριδικές τεχνικές συνδυάζουν περισσότερες από μία στρατηγικές φιλτραρίσματος, οι οποίες εφαρμόζονται ως υποσυστατικά του συστήματος συστάσεων. Αυτού του είδους οι τεχνικές προσπαθούν να ενσωματώσουν χαρακτηριστικά από τεχνικές CF και CBF με τελικό στόχο τη βελτίωση της ποιότητας των συστάσεων και την αντιμετώπιση των αδυναμιών των μεμονωμένων τεχνικών φιλτραρίσματος. Οι υβριδικές τεχνικές χωρίζονται σε έξι κατηγορίες: (i) μικτές υβριδικές (mixed hybrid), (ii) βαρυτικές υβριδικές (weighted hybrid), (iii) εναλλασσόμενες υβριδικές (switching hybrid), (iv) συνενοποιημένες υβριδικές (cascaded hybrid), (v) υβριδικές με συνδυασμό χαρακτηριστικών (featured-combination hybrid) και (vi) υβριδικές σε μετα-επίπεδο (meta-level hybrid). Μία απλή προσέγγιση, που προτείνεται είναι να δημιουργηθούν δύο σύνολα κατάταξης συστάσεων (ένα με CF και ένα με CBF) και στη συνέχεια να συνδυαστούν για να παράγουν μια τελική λίστα.

Η επιτυχία των δικτύων βαθιάς μάθησης (Deep-learning Networks) σε διάφορους τομείς εφαρμογών άνοιξε επίσης ένα νέο πεδίο έρευνας για τα υβριδικά συστήματα συστάσεων, τα οποία τροφοδοτούν τα δίκτυα βαθιάς μάθησης με πληροφορίες τόσο από τη βαθμολόγηση όσο και από το περιεχόμενο σχετικά με τους χρήστες και τα αντικείμενα και βελτιώνουν την ποιότητα των συστάσεων [26]. Παρόλο που λύνουν καλύτερα πολλά γνωστά προβλήματα που αντιμετωπίζουν τα συστήματα συστάσεων, όπως το πρόβλημα της αρχικής κατάστασης [19, 27] ή την αραιότητα του πίνακα βαθμολογήσεων [28], παραμένουν διάφορα προβλήματα κλιμάκωσης [29].

# ΚΕΦΑΛΑΙΟ 3
# Μετρικές Απόδοσης

Η μοντελοποίηση ενός συστήματος συστάσεων που προσαρμόζεται στις ανάγκες μιας επιχείρησης περιλαμβάνει μια φάση αξιολόγησης που δοκιμάζει τις δυνατότητες του συστήματος συστάσεων στα άκρα. Ένα σύστημα συστάσεων προτείνει στους χρήστες αντικείμενα (ή ενέργειες), βασισμένο στις αναμενόμενες προτιμήσεις τους.

Οι μετρικές απόδοσης χρησιμοποιούνται για να αξιολογήσουν ποσοτικά και ποιοτικά την απόδοση των υπολογιστικών προσεγγίσεων. Είναι εξαιρετικά σημαντικές για την επικύρωση ερευνητικών υποθέσεων και τη σύγκριση διαφορετικών τεχνικών [30]. Με αυτόν τον τρόπο καθορίζεται η αποτελεσματικότητα των μεθόδων ενώ αξίζει να σημειωθεί ότι ανάλογα με το επιστημονικό αντικείμενο χρησιμοποιούνται διαφορετικές μετρικές απόδοσης [31]. Στην παρούσα εργασία θα μελετηθούν η ακρίβεια (precision), η ανάκληση και το F1-σκορ.

Η ακρίβεια (accuracy) και η ακρίβεια (precision) παρόλο που αποδίδονται με τον ίδιο όρο αποτελούν διαφορετικές μετρικές απόδοσης, ωστόσο συχνά συγχέονται. Η ακρίβεια (accuracy) μετρά το ποσοστό των δεδομένων που κατηγοριοποιούνται σωστά από την εκάστοτε τεχνική, ενώ η ακρίβεια (precision) μετρά τη δυνατότητα μιας τεχνικής να λαμβάνει μια τιμή εντός ενός δεδομένου εύρους σε ένα μεγάλο αριθμό παρατηρήσεων [32, 33]. Παρότι στα κλασικά προβλήματα κατηγοριοποίησης (classification), χρησιμοποιούμε σα μετρική απόδοσης το accuracy, δε θα μπορούσαμε να κάνουμε το ίδιο και σ' ένα σύστημα συστάσεων. Στα συστήματα συστάσεων, ο στόχος είναι να παρέχονται εξατομικευμένες προτάσεις σε κάθε χρήστη. Το accuracy αγνοεί αυτήν την εξατομίκευση και εστιάζει στο συνολικό ποσοστό σωστών/λάθος προβλέψεων, χωρίς να λαμβάνει υπόψη την ποιότητα των εξατομικευμένων προτάσεων.

Γίνεται σαφής ο λόγος ύπαρξης πολλών διαφορετικών μετρικών που επιτρέπουν την αξιολόγηση της απόδοσης του συστήματος στο να προβλέπει και την παροχή λογικών συστάσεων που ταιριάζουν σε ένα δεδομένο σενάριο με βάση την εκάστοτε επιστημονική εφαρμογή. Στη συνέχεια θα παρατεθούν οι ορισμοί των μετρικών που χρησιμοποιούνται στην παρούσα εργασία και τυπικά παραδείγματα εφαρμογών ώστε να γίνει κατανοητή η χρήση τους.

## 3.1 Ακρίβεια

Η ακρίβεια είναι μια μετρική απόδοσης που χρησιμοποιείται για να μετρήσει την ποιότητα ενός στατιστικού μοντέλου ή ενός μοντέλου μηχανικής μάθησης. Πιο συγκεκριμένα μετράει πόσο συχνά ένα μοντέλο προβλέπει σωστά την κλάση των δεδομένων. Δεδομένου ότι οι αξιολογήσεις των στοιχείων είναι σε κλίμακα 1-5, χρησιμοποιείται μια κατωφλική τιμή *T* για να μετατρέψει μια απόλυτη αξιολόγηση σε δυαδική πρόβλεψη που κατατάσσει εάν το στοιχείο είναι σχετικό ή όχι [31, 32]. Η *T* ορίζεται από τον χρήστη και χρησιμοποιείται για να αποφασιστεί αν ένα μοντέλο είναι αρκετά καλό για την επίλυση ενός προβλήματος. Επιπλέον, με βάση την τιμή Τ μπορεί να καθοριστεί αν ένα δείγμα ανήκει στην εκάστοτε κλάση ή όχι. Αν η πιθανότητα που υπολογίζεται είναι μεγαλύτερη από την τιμή *T*, τότε το δείγμα θεωρείται ότι ανήκει στην κλάση, αλλιώς θεωρείται ότι δεν ανήκει.

Η ακρίβεια ορίζεται ως ο λόγος των αληθώς θετικών ενδείξεων (*TP*) προς το άθροισμα των αληθώς θετικών (*TP*) και των ψευδώς θετικών ενδείξεων *(FP).*

$$\boldsymbol{Precision} = \frac{TP}{TP+FP}$$

Στο πλαίσιο των συστημάτων συστάσεων, η ένδειξη *TP* υποδεικνύει ότι η σύσταση είναι σχετική με το πλαίσιο του χρήστη, ή έχει γίνει αποδεκτή από αυτόν, ενώ η ένδειξη *FP* σημαίνει ότι η σύσταση δεν είναι αληθής. Με άλλα λόγια, η ακρίβεια μετρά το ποσοστό των αληθώς θετικών προβλέψεων μεταξύ όλων των θετικών προβλέψεων που έκανε το μοντέλο. Ένα υψηλό σκορ ακρίβειας υποδηλώνει ότι το μοντέλο κάνει λιγότερες ψευδείς θετικές προβλέψεις και είναι πιο ακριβές στην αναγνώριση των αληθώς θετικών, γεγονός που υποδεικνύει την σημαντικότητά της στα συστήματα συστάσεων [34].

## 3.2 Ανάκληση

Ένας άλλος τρόπος για να αξιολογηθεί η ποιότητα ενός στατιστικού μοντέλου ή ενός μοντέλου μηχανικής μάθησης είναι η ανάκληση και ορίζεται:

$$\boldsymbol{Recall} = \frac{TP}{TP+FN}$$

όπου *FN* η ψευδώς αρνητική ένδειξη.

Η *FN* υποδεικνύει ότι το σύστημα συστάσεων απέτυχε να κάνει σύσταση που τελικά πραγματοποιήθηκε από τον χρήστη, Η διαφορά της ακρίβειας με την ανάκληση είναι ότι η μεν πρώτη εστιάζει στην ακρίβεια των προβλέψεων του μοντέλου, ενώ η δεύτερη εστιάζει στην ικανότητα του μοντέλου να εντοπίζει όλα τα δείγματα της κλάσης [31,33].

Για να γίνει αντιληπτή η διαφορά, παρατίθεται ένα παράδειγμα με υψηλή ακρίβεια και χαμηλή ανάκληση, δηλαδή ένα μοντέλο που κάνει πολύ λίγες ψευδείς θετικές προβλέψεις, αλλά χάνει πολλές από τις πραγματικές θετικές περιπτώσεις στο σύνολο δεδομένων. Ας υποθέσουμε ότι υπάρχει ένα μοντέλο που προβλέπει εάν ένας ασθενής έχει καρκίνο ή όχι. Εάν το μοντέλο έχει υψηλό σκορ ακρίβειας, σημαίνει ότι όταν προβλέπει ότι ένας ασθενής έχει καρκίνο, συνήθως είναι σωστό. Ωστόσο, εάν το μοντέλο έχει χαμηλό σκορ ανάκλησης, σημαίνει ότι χάνει πολλές από τις πραγματικές περιπτώσεις καρκίνου στο σύνολο δεδομένων. Με άλλα λόγια, το μοντέλο δεν αναγνωρίζει όλους τους ασθενείς που έχουν καρκίνο.

## 3.3 F1-Score

Το F1-Score είναι μια μετρική απόδοσης της μηχανικής μάθησης που συνδυάζει τις βαθμολογίες ακρίβειας και ανάκλησης ενός μοντέλου. Ορίζεται ως η αρμονική μέση της ακρίβειας και της ανάκλησης και κυμαίνεται από 0 έως 1, με το 1 να είναι η καλύτερη δυνατή βαθμολογία και δίνεται από την ακόλουθη εξίσωση:

$$\boldsymbol{F1\text{-}Score} = 2 \times \frac{Precsion \times Recall}{Precision + Recall}$$

Το F1 Score είναι ένας τρόπος να συνδυαστούν οι δυο μετρικές (ακρίβεια και ανάκληση) σε μια, παρέχοντας μια ισορροπημένη εικόνα απόδοσης ενός μοντέλου. Υψηλό F1-score συνεπάγεται ότι το μοντέλο έχει τόσο υψηλή ακρίβεια όσο και υψηλή ανάκληση, ή με άλλα

λόγια, μπορεί να προβλέψει σωστά τα περισσότερα θετικά παραδείγματα αποφεύγοντας πολλά ψευδώς θετικά [35, 36].

# ΚΕΦΑΛΑΙΟ 4

# Στατιστικές Τεχνικές

Σε αυτό το κεφάλαιο, θα εξετάσουμε τις βασικές αρχές και τις προηγμένες πρακτικές στατιστικής μοντελοποίησης και ανάλυσης δεδομένων που χρησιμοποιούνται για την εξαγωγή συστάσεων. Εξετάζεται η χρήση μεθόδων όπως η λογιστική παλινδρόμηση και η παραγοντοποίηση πινάκων, με διάφορες προσεγγίσεις της, που χρησιμοποιούνται για να αντιμετωπίσουν την πρόκληση της παροχής εξατομικευμένων προτάσεων σε χρήστες βάσει των προηγούμενων δράσεών τους αλλά και των χαρακτηριστικών τους.

## 4.1 Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση χρησιμοποιείται για την ταξινόμηση κατηγορικών μεταβλητών δοθέντος ενός συνόλου ανεξάρτητων μεταβλητών. Πολλές φορές συγκρίνεται με την διαχωριστική ανάλυση με την πρώτη συνήθως να αποφέρει καλύτερα αποτελέσματα καθώς δεν υποθέτει ότι οι επεξηγηματικές μεταβλητές ακολουθούν κανονική κατανομή.

Στο μοντέλο της λογιστικής παλινδρόμησης η μεταβλητή απόκρισης παλινδρομείται σε ένα σύνολο *m* ανεξάρτητων μεταβλητών $X_1, X_2, \ldots, X_m$ [37]. Το σύνολο τιμών της μεταβλητής απόκρισης είναι το (0,1). Παλινδρομώντας τις ανεξάρτητες μεταβλητές οι εκτιμήσεις που παίρνουμε έχουν πεδίο τιμών το $(-\infty, +\infty)$. Για το λόγο αυτό χρησιμοποιείται η συνάρτηση $logit(\pi)$ η οποία έχει πεδίο ορισμού το (0,1) και πεδίο τιμών όλο το $\mathbb{R}$. Για $\pi \in (0,1)$ έχουμε ότι:

$$logit(\hat{\pi}) = log(\frac{\hat{\pi}}{1-\hat{\pi}})$$

Το μοντέλο της λογιστικής παλινδρόμησης δίνεται από τον τύπο:

$$logit(\hat{\pi}) = \widehat{\beta_0} + \widehat{\beta_1} X_1 + \widehat{\beta_2} X_2 + \ldots + \widehat{\beta_m} X_m$$

και η εκτιμώμενη πιθανότητα $\pi$ υπολογίζεται ως:

$$\hat{\pi} = \frac{exp\ (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_m X_m)}{1 + exp\ (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_m X_m)}$$

Η μόνη υπόθεση που γίνεται είναι πως όλες οι παρατηρήσεις είναι ανεξάρτητες μεταξύ τους και ακολουθούν κατανομή Bernoulli(π). Οι εκτιμήσεις των παραμέτρων β γίνονται από την μεγιστοποίηση της πιθανοφάνειας της κατανομής Bernoulli.

$$L(\boldsymbol{\beta} / \boldsymbol{X}) = \prod_{i=1}^{N} Pr(Y \mid X_i\ ; \beta) =$$

$$\prod_{i=1}^{N} \pi^{y_i}\ (1 - \pi)^{1 - y_i}$$

Υπολογίζοντας το λογάριθμο της πιθανοφάνειας, έχουμε ότι:

$$log(L(\boldsymbol{\beta} / \boldsymbol{X})) = log(\prod_{i=1}^{N} \hat{\pi}^{y_i}\ (1 - \hat{\pi})^{1 - y_i}) =$$
$$= \sum_{i=1}^{N} log\ (\hat{\pi}^{y_i}(1 - \hat{\pi})^{y_i}) = \sum_{i=1}^{N} (y_i log\hat{\pi} + (1 - y_i) log\ (1 - \hat{\pi}))$$

Δεν υπάρχει κλειστός τύπος για τον υπολογισμό των παραμέτρων που μεγιστοποιούν την παραπάνω ποσότητα. Η μέθοδος με την οποία μεγιστοποιείται η παραπάνω συνάρτηση είναι αυτή των Newton-Raphson. Συγκεκριμένα, ο τύπος για την ενημέρωση των παραμέτρων σε κάθε επανάληψη της μεθόδου μπορεί να εκφραστεί με την παρακάτω εξίσωση:

$$\beta_{n+1} = \beta_n - (\nabla^2 f(\beta_n))^{-1} \nabla f(\beta_n),$$ όπου

- $\beta_n$ είναι οι τρέχουσες τιμές των παραμέτρων στην επανάληψη *n*.
- $\nabla f(\beta_n)$, είναι ο πίνακας Gradient (πίνακας πρώτων παραγώγων) της συνάρτησης πιθανοφάνειας στο σημείο $\beta_n$.
- $(\nabla^2 f(\beta_n))^{-1}$, είναι ο αντίστροφος του Εσιανού πίνακα (πίνακας δεύτερων παραγώγων) της συνάρτησης πιθανοφάνειας στο σημείο $\beta_n$, χρησιμοποιείται για τον υπολογισμό του επόμενου βήματος.

Ο τύπος αυτός ανανεώνει τις παραμέτρους $\beta_n$ σε κάθε επανάληψη της μεθόδου, υπολογίζοντας τη διαφορά με βάση τον αντίστροφο του Εσιανού και τον Gradient.

## 4.2 Matrix Factorization

Η παραγοντοποίηση πινάκων είναι μια τεχνική που αποσυνθέτει έναν πίνακα σε ένα γινόμενο δύο ή περισσότερων πινάκων, συνήθως με μικρότερες διαστάσεις. Μπορεί να χρησιμοποιηθεί για την ανίχνευση της κρυφής δομής στα δεδομένα, όπως τα κρυφά χαρακτηριστικά ή προτιμήσεις των χρηστών και των αντικειμένων σε ένα σύστημα συστάσεων. Η παραγοντοποίηση πινάκων μπορεί επίσης να βοηθήσει στη μείωση της υπολογιστικής πολυπλοκότητας και των απαιτήσεων μνήμης για την επίλυση γραμμικών εξισώσεων ή προβλημάτων ιδιοτιμών [38].

Υπάρχουν διαφορετικοί τύποι παραγοντοποίησης πινάκων, ανάλογα με τις ιδιότητες και τις εφαρμογές του αρχικού πίνακα. Παρακάτω θα γίνει λεπτομερής αναφορά στους τύπους παραγοντοποίησης πινάκων που χρησιμοποιούνται στην παρούσα εργασία.

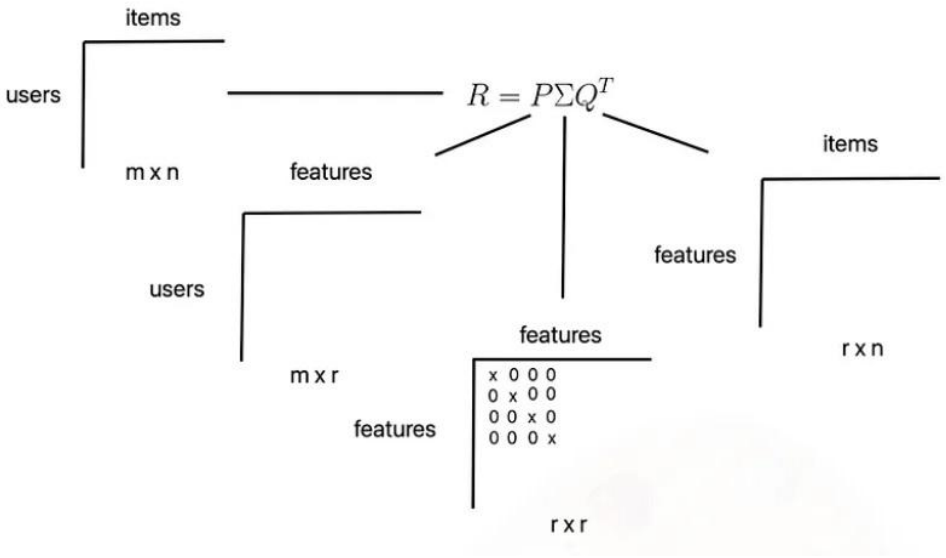
### 4.2.1 Singular Value Decomposition

Η Singular Value Decomposition είναι μια μέθοδος που χρησιμοποιείται για τη μείωση των διαστάσεων κάτω από την υπόθεση ότι τα δεδομένα μας μπορούν να αναπαρασταθούν ικανοποιητικά από πίνακα μικρότερης διάστασης. Είναι σύνηθες να χρησιμοποιείται σε αλγόριθμους που προσπαθούν να προβλέψουν τις προτιμήσεις ανθρώπων, όπως για παράδειγμα σε προτιμήσεις τους για ταινίες τις ποιές έχουν δει και έχουν βαθμολογήσει.

Η SVD είναι ένας τρόπος να αποσυνθέσουμε οποιονδήποτε πίνακα σε τρεις απλούστερους πίνακες που έχουν μερικές ωραίες ιδιότητες. Η ιδέα είναι να βρούμε ποιες είναι οι κύριες κατευθύνσεις ή συνιστώσες με τις οποίες μπορεί να εκφραστεί ο πίνακας. Για παράδειγμα, αν έχουμε έναν πίνακα που αναπαριστά κάποια σημεία δεδομένων σε ένα δισδιάστατο επίπεδο, μπορούμε να σκεφτούμε την SVD ως τον τρόπο να βρούμε την καλύτερη ευθεία που ταιριάζει σε εκείνα τα σημεία. Στη συνέχεια βρίσκουμε μια άλλη ευθεία που είναι κάθετη στην πρώτη. Αυτές οι δύο ευθείες ονομάζονται οι κύριες συνιστώσες των δεδομένων και αιχμαλωτίζουν τη μεγαλύτερη ποσότητα ή πληροφορία των δεδομένων.

Υποθέτουμε ότι έχουμε έναν πίνακα αξιολογήσεων *R* με *m* χρήστες και *n* αντικείμενα. Μπορούμε να διαχωρίσουμε τον πίνακα σε άλλους δύο πίνακες, *P* και *Q*, και ένα διαγώνιο πίνακα *Σ*. Η SVD εκφράζει τον *R* ως το γινόμενο τριών πινάκων: $R = P\Sigma Q^T$,
όπου:

- Ο πίνακας *P* είναι ορθογώνιος και διαστάσεων *m x r*.
- Ο πίνακας *Σ* είναι διαγώνιος και διαστάσεων *r x r*, με τα στοιχεία της διαγωνίου να αποτελούν τις σιγματικές τιμές (singular values) του *R*.
- Ο πίνακας *Q* είναι ορθογώνιος και διαστάσεων *r x n*.




Σχήμα 3. Διαγραμματική απεικόνιση της SVD [13]

Με μαθηματικούς όρους, η SVD επιτυγχάνεται μέσω της αποσύνθεσης του πίνακα *R*, ως εξής:

- Υπολογισμός των ιδιοτιμών και ιδιοδιανυσμάτων του $RR^T$ (για να βρεθούν οι πίνακες *P* και *Q*).
- Τα στοιχεία της διαγωνίου του πίνακα *Σ* προκύπτουν από τις τετραγωνικές ρίζες των μη μηδενικών ιδιοτιμών του $RR^T$ ή του $R^TR$.

Το κύριο στοιχείο πληροφορίας εδώ, είναι ότι κάθε ένα από τα στοιχεία της διαγωνίου του πίνακα *Σ* αντιπροσωπεύει πόσο συνεισφέρει η κάθε κύρια συνιστώσα στον αρχικού πίνακα αξιολογήσεων *R*. Εάν ταξινομήσουμε αυτές τις τιμές με φθίνουσα σειρά και εξαιρέσουμε όλες

εκτός από τις κορυφαίες *k* (καθορισμένες από τον χρήστη), μπορούμε να λάβουμε την καλύτερη προσέγγιση του πίνακα αξιολογήσεων πολλαπλασιάζοντας τους περικομμένους πίνακες ξανά μαζί. Αυτή είναι η αρχή πίσω από τη μείωση της διαστατικότητας της SVD.

Βέβαια, η SVD έχει και τα ακόλουθα μειονεκτήματα: α. χρειάζεται αρκετό υπολογιστικό χρόνο, β. δε μπορεί να χειριστεί τα missing values (τα δεδομένα πρέπει να συμπληρωθούν χρησιμοποιώντας το μέσο).

Γι' αυτούς τους λόγους, χρησιμοποιούμε άλλες τεχνικές μηχανικής μάθησης (ειδικότερα, την gradient descent και την alternating least squares) για να αναζητήσουμε την καλύτερη προσέγγιση του πίνακα αξιολογήσεων. Κατά τη διάρκεια της εκπαίδευσης των δεδομένων, λαμβάνουμε υπόψη μόνο τα στοιχεία με βαθμολογία και αγνοούμε εκείνα που δεν έχουν. Έτσι, απαλείφεται η ανάγκη συμπλήρωσης των missing values.

Στη δεύτερη τεχνική (ALS) θα αναφερθούμε εκτενώς στην επόμενη ενότητα της παρούσας εργασίας.

### 4.2.2 Alternating Least Squares

Η ALS είναι μια τεχνική που μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων παραγοντοποίησης πινάκων, όπως η εύρεση των κρυφών παραγόντων των χρηστών και των αντικειμένων σε ένα σύστημα συστάσεων. Ως κρυφοί παράγοντες (latent factors) ορίζονται τα αόρατα ή κρυμμένα χαρακτηριστικά ή οι παράμετροι που περιγράφουν τις ιδιότητες ή τις προτιμήσεις των χρηστών και των αντικειμένων (π.χ. προϊόντα, ταινίες, μουσική) σε ένα σύστημα συστάσεων.

Η παραγοντοποίηση πινάκων είναι ένας τρόπος να διασπάσουμε έναν μεγάλο και αραιό πίνακα σε ένα γινόμενο δύο μικρότερων και πυκνότερων πινάκων, οι οποίοι μπορούν να αποτυπώσουν την κρυφή δομή και τα πρότυπα (patterns) στα δεδομένα.

Έστω ένας πίνακα *R* που αναπαριστά τις αξιολογήσεις που έχουν δώσει *m* χρήστες σε *n* αντικείμενα, όπου κάθε καταχώριση $r_{ui}$ είναι η αξιολόγηση του χρήστη *u* στο αντικείμενο *i*, ή μηδέν αν ο χρήστης δεν έχει αξιολογήσει το αντικείμενο. Θέλουμε να βρούμε δύο πίνακες *U* και *V*, τέτοιους ώστε ο *U* να έχει *m* γραμμές και *k* στήλες, ο *V* να έχει *n* γραμμές και *k* στήλες, και ο *R* να είναι περίπου ίσος με το γινόμενο του *U* με τον ανάστροφο του *V*. Εδώ, το *k* είναι μια παράμετρος που καθορίζει τον αριθμό των κρυφών παραγόντων, που είναι τα χαρακτηριστικά

που επηρεάζουν τις αξιολογήσεις. Κάθε γραμμή του *U* αντιπροσωπεύει τις προτιμήσεις ενός χρήστη για κάθε παράγοντα, και κάθε γραμμή του *V* αντιπροσωπεύει τη σημασία ενός αντικειμένου για κάθε παράγοντα [39].

Το πρόβλημα είναι να βρεθούν οι πίνακες *U* και *V* που ελαχιστοποιούν το τετραγωνικό σφάλμα μεταξύ του *R* και του γινομένου του *U* με τον ανάστροφο του *V*, μόνο για τις καταχωρήσεις του *R* που δεν είναι μηδενικές. Αυτό σημαίνει ότι ενδιαφερόμαστε μόνο για τις αξιολογήσεις που παρατηρούνται και αγνοούμε τις απουσιάζουσες. Μαθηματικά, η συνάρτηση στόχου μπορεί να γραφτεί ως:

$$\min_{U,V} \sum_{(\boldsymbol{u},\boldsymbol{i})\epsilon\,\Omega} (r_{ui} - \boldsymbol{u_i^T v_i})^2$$

όπου *Ω*, είναι το σύνολο δεικτών των μη μηδενικών καταχωρίσεων του *R*, και $\boldsymbol{u_i}$ και $\boldsymbol{v_i}$ είναι οι *i-οστές* γραμμές των *U* και *V* αντίστοιχα.

Η μέθοδος ALS λειτουργεί διαδοχικά μεταξύ δύο βημάτων: κρατώντας σταθερό τον *U* και επιλύοντας ως προς τον *V*, και στη συνέχεια, κρατώντας σταθερό τον *V* και επιλύοντας ως προς τον *U*. Σε κάθε βήμα, η συνάρτηση στόχου γίνεται μια τετραγωνική μορφή ενός πίνακα, η οποία μπορεί να λυθεί για τα $\boldsymbol{v_i}$ ή τα $\boldsymbol{u_i}$, θέτοντας την παράγωγό της ίση με μηδέν. Για παράδειγμα, για να λύσουμε για τον *V*, μπορούμε να πάρουμε την παράγωγο της συνάρτησης στόχου ως προς κάθε γραμμή του *V* ίση με το μηδέν [40]. Αυτό μας δίνει την ακόλουθη εξίσωση για κάθε *i*:

$$\sum_{u\,\epsilon\,\Omega_i} \boldsymbol{u_i u_i^T v_i} = \sum_{u\,\epsilon\,\Omega_i} r_{ui}\boldsymbol{u_i}$$

όπου $\Omega_i$ είναι το σύνολο των χρηστών που έχουν αξιολογήσει το αντικείμενο *i*. Αυτό είναι ένα γραμμικό σύστημα εξισώσεων που μπορεί να λυθεί ως προς $\boldsymbol{v_i}$.

Παρόμοια, για να λύσουμε ως προς *U*, μπορούμε να πάρουμε την παράγωγο της συνάρτησης στόχου ως προς κάθε γραμμή του *U* ίση με το μηδέν. Αυτό μας δίνει την ακόλουθη εξίσωση για κάθε *u*:

$$\sum_{i\,\epsilon\,\Omega_u} \boldsymbol{v_i v_i^T u_i} = \sum_{i\,\epsilon\,\Omega_u} \boldsymbol{r_{ui} v_i}$$

όπου, $\Omega_u$ είναι το σύνολο των αντικειμένων που έχουν αξιολογηθεί από το χρήστη $u$. Αυτό είναι ένα ακόμα γραμμικό σύστημα εξισώσεων που μπορεί να λυθεί ως προς $\boldsymbol{u_i}$. Η μέθοδος ALS επαναλαμβάνει αυτά τα δύο βήματα μέχρι να συγκλίνει ή μέχρι να πετύχει ένα μέγιστο αριθμό επαναλήψεων.

Η ALS έχει ορισμένα πλεονεκτήματα και μειονεκτήματα. Μερικά από τα πλεονεκτήματα είναι:

- Είναι εύκολη στην υλοποίηση και την παραλληλοποίηση, αφού κάθε γραμμή του $U$ και του $V$ μπορεί να λυθεί ανεξάρτητα.
- Μπορεί να χειριστεί απουσιάζουσες τιμές και βαρυσήμαντα σφάλματα, προσαρμόζοντας αντίστοιχα τη συνάρτηση στόχου και τα γραμμικά συστήματα.
- Μπορεί να ενσωματώσει όρους κανονικοποίησης, όπως η $L_2$ νόρμα ή η $L_1$ νόρμα, για να αποτρέψει το overfitting και να βελτιώσει τη γενίκευση.

Μερικά από τα μειονεκτήματα είναι:

- Μπορεί να είναι αργή στη σύγκλιση, ειδικά για μεγάλους και αραιούς πίνακες.
- Μπορεί να κολλήσει σε τοπικά ελάχιστα, ανάλογα με την αρχικοποίηση και τον αριθμό των παραγόντων.
- Δε μπορεί να αντιμετωπίζει προβλήματα κλιμάκωσης, αφού απαιτεί αποθήκευση και αντιστροφή μεγάλων πινάκων.

## 4.3 Ridge Regression

Η τεχνική Ridge Regression δημιουργήθηκε με σκοπό την αντιμετώπιση του φαινομένου της πολυσυγγραμικότητας. Το φαινόμενο της πολυσυγγραμικότητας εμφανίζεται σε σύνολα δεδομένων όπου οι επεξηγηματικές μεταβλητές παρουσιάζουν υψηλή γραμμική συσχέτιση. Στο μοντέλο της γραμμικής παλινδρόμησης, έχοντας επεξηγηματικές μεταβλητές υψηλά συσχετισμένες, οδηγούμαστε συνήθως σε υψηλά τυπικά σφάλματα για τις εκτιμήτριες ελαχίστων τετραγώνων.

Οι εκτιμήτριες ελαχίστων τετραγώνων των $\boldsymbol{\beta}$ είναι:

$$\widehat{\boldsymbol{\beta}}^{ols} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Για τον παραπάνω υπολογισμό, χρειάζεται ο υπολογισμός του $(X^T X)^{-1}$, χρησιμοποιώντας τον τύπο:

$$A^{-1} = \frac{1}{det\ (A)}\ adj(A).$$

Στην περίπτωση της πολυσυγραμμικότητας η ορίζουσα του πίνακα $X^T X$ είναι πολύ κοντά στο μηδέν έχοντας ως αποτέλεσμα την αστάθεια του υπολογισμού του αντιστρόφου του πίνακα $X^T X$ και τη δυσκολία στην ακριβή επίλυση των γραμμικών εξισώσεων, που απαιτούνται για την εκτέλεση ορισμένων μεθόδων ή προσεγγιστικών αλγορίθμων.

Αυτό που πρότειναν οι Hoerl και Kennard ήταν τη συρρίκνωση των παραμέτρων, προσθέτοντας έναν όρο στη συνάρτηση ζημιάς, ώστε να τιμωρούνται οι μεγάλες τιμές των παραμέτρων [41]. Η συνάρτηση ζημιάς χρησιμοποιώντας ως επεξηγηματικές τις κανονικοποιημένες μεταβλητές (η κανονικοποίηση μεταβλητών αναφέρεται στη διαδικασία με την οποία αφαιρούμε τη μέση τιμή από κάθε παρατήρηση και διαιρούμε με την τυπική απόκλισή της) έχει τον εξής τύπο:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{k=1}^{p} {\beta_k}^2$$

υπό τον περιορισμό:

$$\sum_{k=1}^{p} {\beta_k}^2 \leq t$$

Οι παράμετροι της Ridge παλινδρόμησης εκτιμούνται ως:

$$\widehat{\boldsymbol{\beta}}^{ridge} = (X^t X + \lambda I)^{-1} X^T \boldsymbol{y}$$

όπου ο πίνακας $I$ είναι ο μοναδιαίος πίνακας $pxp$. Ουσιαστικά προστίθεται μία θετική σταθερά στα διαγώνια στοιχεία του πίνακα $X'X$ πριν την αντιστροφή του. Το $\lambda$ (lambda) αντιπροσωπεύει

την παράμετρο κανονικοποίησης, η οποία ελέγχει το επίπεδο της κανονικοποίησης που εφαρμόζεται στο μοντέλο. Η επιλογή του $\lambda$ είναι σημαντική για την απόδοση του μοντέλου.

Συνήθως, ο $\lambda$ επιλέγεται μέσω διαδικασιών όπως η διασταυρούμενη επικύρωση (cross-validation) ή η χρήση κριτηρίων όπως του σφάλματος ελαχίστων τετραγώνων (mean squared error) σε ένα σύνολο επικύρωσης (validation set). Κατά την διαδικασία διασταυρούμενης επικύρωσης, διαφορετικές τιμές $\lambda$ δοκιμάζονται σε διαφορετικά υποσύνολα των δεδομένων εκπαίδευσης και η τιμή που οδηγεί στην καλύτερη απόδοση στα δεδομένα επικύρωσης επιλέγεται ως τελική τιμή για το $\lambda$. Οι πρακτικές για την εκτίμηση του $\lambda$ μπορεί να διαφέρουν ανάλογα με το πρόβλημα και το μέγεθος του συνόλου δεδομένων, αλλά η συνήθης πρακτική είναι να δοκιμάζονται διαφορετικές τιμές $\lambda$ και να επιλέγεται αυτή που οδηγεί στην καλύτερη απόδοση στα δεδομένα επικύρωσης.

Στην περίπτωση όπου τα διανύσματα των μεταβλητών είναι ορθογώνια, έχουμε ότι:

$$\widehat{\boldsymbol{\beta}}^{ols} = X^T \boldsymbol{y}, \text{ αφού } X^T X = I.$$

Επομένως, οι συντελεστές της παλινδρόμησης Ridge είναι ίσοι με:

$$\widehat{\boldsymbol{\beta}}^{ridge} = (X'X + \lambda I)^{-1} X'\boldsymbol{y} \Rightarrow \hat{\beta}^{ridge} = (I + \lambda I)^{-1} \widehat{\boldsymbol{\beta}}^{ols} .$$

Από την παραπάνω σχέση, προκύπτει:

$$\widehat{\boldsymbol{\beta}}^{ridge} = \frac{1}{1+\lambda} \widehat{\boldsymbol{\beta}}^{ols}$$

όπου, $\widehat{\boldsymbol{\beta}}^{ols}$ είναι οι εκτιμητές των $\boldsymbol{\beta}$ που προκύπτουν από τη μέθοδο των ελαχίστων τετραγώνων.

Εύκολα αποδεικνύεται ότι οι εκτιμητές Ridge δεν είναι αμερόληπτοι εκτιμητές καθώς:

$$E[\hat{\beta}^{ridge}] = \frac{1}{1+\lambda} E[\hat{\beta}^{ols}] = \frac{1}{1+\lambda} \beta .$$

Επίσης, η διακύμανση των εκτιμητών Ridge είναι μικρότερη από ότι των εκτιμητών ελαχίστων τετραγώνων.

$$V[\widehat{\boldsymbol{\beta}}^{ridge}] = (\frac{1}{1+\lambda})^2 \; V[\widehat{\boldsymbol{\beta}}^{ols}] < V[\widehat{\boldsymbol{\beta}}^{ols}].$$

Ουσιαστικά η παράμετρος $\lambda$ είναι μια παράμετρος που ρυθμίζει το αντιστάθμισμα μεταξύ μεροληψίας και διακύμανσης των εκτιμητριών μας. Μεγάλες τιμές του $\lambda$ οδηγούν σε εκτιμητές με μικρή διακύμανση αλλά μεγάλη μεροληψία. Για μικρές τιμές του $\lambda$ οδηγούμαστε σε εκτιμήτριες με μεγάλη διακύμανση και μικρή μεροληψία. Τέλος, για $\lambda = 0$, οι εκτιμήτριες που προκύπτουν είναι αυτές των ελαχίστων τετραγώνων. Συνήθεις τιμές του $\lambda$ είναι μεταξύ 0 και 1.

# ΚΕΦΑΛΑΙΟ 5
# Αριθμητική Εφαρμογή σε Πραγματικά Δεδομένα

## 5.1 Περιγραφή του συνόλου δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα εργασία ανήκει στη συλλογή συνόλων δεδομένων MovieLens. Τα σύνολα δεδομένων του MovieLens χρησιμοποιούνται ευρέως στην εκπαίδευση, την έρευνα και τη βιομηχανία. Εκατοντάδες χιλιάδες λήψεις των συγκεκριμένων συνόλων πραγματοποιούνται κάθε χρόνο, αντικατοπτρίζοντας τη χρήση τους σε δημοφιλή βιβλία προγραμματισμού, παραδοσιακά και online μαθήματα [42].

Αυτά τα σύνολα δεδομένων αποτελούν προϊόν της δραστηριότητας των μελών της ερευνητικής ομάδας GroupLens. Η GroupLens είναι μία ερευνητική ομάδα του Department of Computer Science and Engineering, του Πανεπιστημίου της Minnesota, Twin Cities των ΗΠΑ, που εξειδικεύεται σε συστήματα συστάσεων, online κοινότητες, κινητές και ασύρματες τεχνολογίες, ψηφιακές βιβλιοθήκες και τοπικά γεωγραφικά συστήματα πληροφοριών. Η GroupLens Research έχει συλλέξει και διαθέσει σύνολα δεδομένων με αξιολογήσεις από τον ιστότοπο του MovieLens (https://movielens.org/).

Σκοπός της παρούσας εφαρμογής είναι η δημιουργία συστάσεων (Recommendations) σε χρήστες που έχουν δει και αξιολογήσει κάποιες ταινίες, για άλλες που δεν έχουν παρακολουθήσει ακόμα. Κάθε χρήστης έχει παρακολουθήσει συγκεκριμένο αριθμό ταινιών και έχει δώσει αντίστοιχες βαθμολογίες για αυτές. Χρησιμοποιώντας αυτά τα δεδομένα, δημιουργούμε μοντέλα προβλέψεων των βαθμολογιών που δεν έχουν καταχωρηθεί ακόμα. Στη συνέχεια οι αλγόριθμοι προτείνουν τις ταινίες με τις υψηλότερες εκτιμώμενες βαθμολογίες στον εκάστοτε χρήστη.

Το σύνολο δεδομένων που χρησιμοποιείται για την παρακάτω εφαρμογή, είναι αυτό των 100 χιλιάδων γραμμών. Ο λόγος που επιλέχθηκε το μικρότερο εκ των διαθέσιμων συνόλων, ήταν για την ελαχιστοποίηση του υπολογιστικού χρόνου.

Κατά τη λήψη του συγκεκριμένου συνόλου δεδομένων, παρατηρήθηκε ότι αποτελείται από δύο ξεχωριστά υποσύνολα. Το πρώτο, αφορά βαθμολογίες ταινιών (ratings), ενώ το δεύτερο ταινίες (movies). Το πρώτο υποσύνολο (ratings dataset), περιέχει 4 μεταβλητές και 100.000

γραμμές. Οι μεταβλητές του είναι: ταυτότητα χρήστη (userID), ταυτότητα αντικειμένου (itemID), βαθμολογία (Rating) και χρονική στιγμή που καταχωρήθηκε η βαθμολογία (TimeStamp). Το δεύτερο υποσύνολο (movies dataset), περιέχει 21 μεταβλητές και 1664 γραμμές. Οι μεταβλητές του είναι: ταυτότητα αντικειμένου (itemID), τίτλος ταινίας (Title) και άλλες 19 δίτιμες μεταβλητές (λαμβάνουν τιμές 0 ή 1) που αναφέρονται στις κατηγορίες των ταινιών. Έτσι, όταν μία ταινία εμπίπτει στην κατηγορία της εκάστοτε στήλης, το αντίστοιχο κελί του πίνακα παίρνει την τιμή 1 ή 0 διαφορετικά. Ο παρακάτω πίνακας (Πίνακας 5.1) παρουσιάζει αναλυτικά τις κατηγορίες των ταινιών που έχουμε στο δεύτερο υποσύνολο.

| **Κατηγορία Ταινίας** | **Επεξήγηση** |
|---|---|
| UnknownCat | Άγνωστη Κατηγορία Ταινίας |
| Action | Ταινία Δράσης |
| Adventure | Περιπέτεια |
| Animation | Κινούμενα Σχέδια |
| Children's | Ταινία για Παιδιά |
| Comedy | Κωμωδία |
| Crime | Εγκληματική Ταινία |
| Documentary | Ντοκιμαντέρ |
| Drama | Δράμα |
| Fantasy | Ταινία Φαντασίας |
| Film-Noir | Φίλμ Νουάρ |
| Horror | Ταινία Τρόμου |
| Musical | Μιούζικαλ |
| Mystery | Ταινία Μυστηρίου |
| Romance | Ρομαντική Ταινία |
| Sci-Fi | Ταινία Επιστημονικής Φαντασίας |
| Thriller | Ταινία Θρίλερ |
| War | Πολεμική Ταινία |
| Western | Ταινία Γουέστερν |

Πίνακας 5.1 Επεξήγηση Κατηγοριών Ταινιών

Έχουμε, λοιπόν, 100.000 βαθμολογίες για ταινίες (ratings) από 943 διαφορετικούς χρήστες (users). Αυτές οι βαθμολογίες αφορούν 1664 διαφορετικές ταινίες (items), σε διαφορετικές χρονικές στιγμές (TimeStamps). Πολλοί χρήστες μπορεί να έχουν βαθμολογήσει παραπάνω από μία ταινίες και προφανώς κανένας χρήστης δεν έχει βαθμολογήσει το σύνολο των ταινιών.

Για τη συγκεκριμένη εφαρμογή, χρειάζεται ένα ενιαίο σύνολο δεδομένων. Άρα, συγχωνεύθηκαν (merge) τα δύο υποσύνολα (dataframes) με βάση το κοινό τους χαρακτηριστικό.

Η κοινή μεταβλητή-χαρακτηριστικό των δύο υποσυνόλων είναι η στήλη itemID, που αναφέρεται στην ταυτότητα της κάθε ταινίας. Έτσι πλέον, δημιουργήθηκε ένα μεγάλο σύνολο δεδομένων (dataset), με 100.000 χιλιάδες γραμμές και 24 στήλες-μεταβλητές. Οι μεταβλητές του είναι: ταυτότητα χρήστη (userID), ταυτότητα αντικειμένου (itemID), βαθμολογία (Rating), χρονική στιγμή που έγινε η βαθμολογία (TimeStamp), τίτλος ταινίας (Title) και άλλες 19 δίτιμες μεταβλητές (λαμβάνουν τιμές 0 ή 1) που αναφέρονται στις παραπάνω κατηγορίες των ταινιών.

## 5.2 Προκαταρτική Επεξεργασία των Δεδομένων και Περιγραφική Στατιστική

### 5.2.1 Προκαταρτική Επεξεργασία

Η πρώτη ενέργεια που πραγματοποιήθηκε, ήταν η ερεύνα για ελλείπουσες τιμές. Παρατηρήθηκε ότι καμία απ' τις μεταβλητές που αναφέρθηκαν παραπάνω δεν είχε ελλείπουσες τιμές, όπως φαίνεται αναλυτικά στον παρακάτω πίνακα.

| **Μεταβλητή** | **Πλήθος ελλειπουσών τιμών** |
|---|---|
| userID | 0 |
| itemID | 0 |
| Rating | 0 |
| TimeStamp | 0 |
| Title | 0 |
| Unknown Cat | 0 |
| Action | 0 |
| Adventure | 0 |
| Animation | 0 |
| Children's | 0 |
| Comedy | 0 |
| Crime | 0 |
| Documentary | 0 |
| Drama | 0 |
| Fantasy | 0 |
| Film-Noir | 0 |
| Horror | 0 |
| Musical | 0 |
| Mystery | 0 |
| Romance | 0 |
| Sci-Fi | 0 |

| | |
|---|---|
| Thriller | 0 |
| War | 0 |
| Western | 0 |

Πίνακας 5.2 Πλήθος ελλειπουσών τιμών ανά μεταβλητή

Στη συνέχεια, έπρεπε να διαπιστωθεί εάν όλες αυτές οι κατηγορίες ταινιών παίζουν στατιστικά σημαντικό ρόλο στην βαθμολογία (Rating) που θα δώσει ο χρήστης. Για να γίνει αυτό, πραγματοποιήθηκε ο στατιστικός έλεγχος υπόθεσης *t*-test 19 φορές (όσες και οι κατηγορίες των ταινιών). Ουσιαστικά αυτό που θέλαμε να ελέγξουμε είναι εάν υπάρχει στατιστικά σημαντική διαφορά στο γενικό μέσο όρο βαθμολογίας και το μέσο όρο βαθμολογίας κάθε κατηγορίας (πχ Action). Παρακάτω φαίνονται τα *p*-values των ελέγχων που πραγματοποιήθηκαν.

| **Κατηγορία Ταινίας** | **p-value** |
|---|---|
| Action | $2.964 \cdot 10^{-16}$ |
| Adventure | 0.0031 |
| Animation | 0.0109 |
| Children's | $2.284 \cdot 10^{-43}$ |
| Comedy | $5.696 \cdot 10^{-137}$ |
| Crime | $1.631 \cdot 10^{-17}$ |
| Documentary | 0.0004 |
| Drama | $1.845 \cdot 10^{-286}$ |
| Fantasy | $4.189 \cdot 10^{-25}$ |
| Film-Noir | $2.117 \cdot 10^{-48}$ |
| Horror | $2.767 \cdot 10^{-57}$ |
| Musical | **0.587** |
| Mystery | $8.251 \cdot 10^{-13}$ |
| Romance | $6.890 \cdot 10^{-37}$ |
| Sci-Fi | 0.0009 |
| Thriller | 0.0019 |
| War | $4.484 \cdot 10^{-148}$ |
| Western | 0.0012 |
| Unknown Cat | **0.354** |

Πίνακας 5.3 *p*-values στατιστικού ελέγχου *t*

Όπως γίνεται σαφές, δύο κατηγορίες ταινιών δεν παίζουν στατιστικά σημαντικό ρόλο στην βαθμολογία (Rating) που θα δώσει ο χρήστης, σε επίπεδο στατιστικής σημαντικότητας α = 5%. Αυτές είναι: η κατηγορία Musical και η άγνωστη κατηγορία ταινίας Unknown Cat καθώς ο

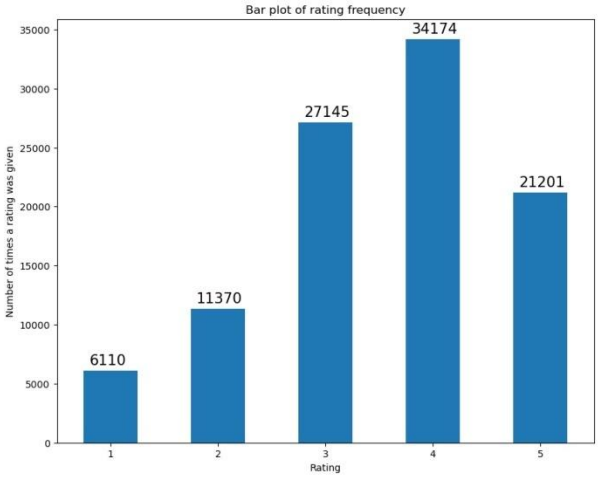
στατιστικός έλεγχος *t* που τις αφορά δίνει p-value > 0.05. Οι συγκεκριμένες δύο κατηγορίες αφαιρούνται (Drop) από το σύνολο δεδομένων και πλέον το σύνολο έχει 22 μεταβλητές συνολικά.

Η μεταβλητή TimeStamp, υποδηλώνει το χρονικό αποτύπωμα της βαθμολογίας που έκανε κάποιος χρήστης. Στο συγκεκριμένο σύνολο δεδομένων το TimeStamp έχει τη μορφή ενός 9φήφιου κωδικού. Απαραίτητο, λοιπόν, κρίθηκε να μετατραπεί (Convert) η κωδικοποιημένη μεταβλητή TimeStamp σε μία σαφή ημερομηνία, που ονομάστηκε Date. Κατόπιν αυτής της ενέργειας το σύνολο δεδομένων είχε 23 μεταβλητές.

Όπως με κάθε εφαρμογή που έγκειται στα πλαίσια της Μηχανικής Μάθησης, έτσι κι εδώ, τα δεδομένα έπρεπε να χωριστούν σε 3 επιμέρους υποσύνολα (Training set, Validation set, Test set). Όπως έχει ήδη αναφερθεί, σκοπός της παρούσας εφαρμογής είναι η δημιουργία προτάσεων (Recommendations) σε χρήστες που έχουν δει και αξιολογήσει κάποιες ταινίες, για άλλες που δεν έχουν παρακολουθήσει ακόμα. Δε θα ήταν λογικό οι αλγόριθμοι που θα τρέξουμε παρακάτω να εκπαιδευτούν (Train) σε πρόσφατα δεδομένα κριτικών και να δώσουν προβλέψεις (Predictions) με βάση πιο παλιά. Η εκπαίδευση πρέπει να γίνει στα παλαιότερα διαθέσιμα δεδομένα που υπάρχουν. Είναι γνωστό πλέον, μέσω της μεταβλητής Date, ότι η παλαιότερη βαθμολογία έγινε την 20/9/1997 στις 04:05:10 ενώ η πιο πρόσφατη βαθμολογία έγινε την 23/4/1998 στις 00:10:38. Αυτό είναι ένα διάστημα περίπου 7 μηνών, δηλαδή περίπου 210 ημερών. Η εκπαίδευση των δεδομένων, λοιπόν, γίνεται μέσω του υποσυνόλου των δεδομένων των πρώτων 120 ημερών (Train set). Η επαλήθευση γίνεται μέσω του υποσυνόλου των δεδομένων των επόμενων 45 ημερών (Validation set), ενώ ο έλεγχος γίνεται μέσω του υποσυνόλου των δεδομένων των τελευταίων 45 ημερών (Test set).

### 5.2.2 Περιγραφική Στατιστική

Στην παρούσα ενότητα δίνονται τα απαραίτητα περιγραφικά στατιστικά στοιχεία για την καλύτερη κατανόηση του συνόλου δεδομένων. Στο παρακάτω ιστόγραμμα παρατίθενται οι συχνότητες των βαθμολογιών (Ratings) που έχουν δώσει οι χρήστες για τις προαναφερθείσες ταινίες.




Σχήμα 4. Συχνότητες των βαθμολογιών των χρηστών

Αυτό που παρατηρείται είναι ότι οι περισσότεροι χρήστες έχουν βαθμολογήσει τις ταινίες που παρακολούθησαν με μια βαθμολογία 4 αστέρων. Ακολουθούν οι βαθμολογίες 3 αστέρων και 5 αστέρων. Ένα πρώτο σχόλιο θα μπορούσε να είναι πως οι χρήστες μπαίνουν σε διαδικασία βαθμολογίας μιας ταινίας, κυρίως όταν αυτή τους άρεσε.

Ένα άλλο μέτρο που θα μπορούσε να μας δώσει στοιχεία είναι η μέση βαθμολογία (Mean Rating) που έχει λάβει κάποια ταινία. Παρακάτω παρατίθεται ο πίνακας με τις κορυφαίες είκοσι ταινίες με γνώμονα τη μέση βαθμολογία που έχουν λάβει.

| **Τίτλος Ταινίας** | **Μέση Βαθμολογία** |
|---|---|
| They Made Me a Criminal (1939) | 5.00 |
| Marlene Dietrich: Shadow and Light (1996) | 5.00 |
| Saint of Fort Washington, The (1993) | 5.00 |
| Someone Else's America (1995) | 5.00 |
| Star Kid (1997) | 5.00 |
| Great Day in Harlem, A (1994) | 5.00 |
| Aiqing wansui (1994) | 5.00 |
| Santa with Muscles (1996) | 5.00 |
| Prefontaine (1997) | 5.00 |
| Entertaining Angels: The Dorothy Day Story (1996) | 5.00 |
| Pather Panchali (1955) | 4.63 |
| Some Mother's Son (1996) | 4.50 |
| Maya Lin: A Strong Clear Vision (1994) | 4.50 |
| Anna (1996) | 4.50 |
| Everest (1998) | 4.50 |
| Close Shave, A (1995) | 4.49 |

| | |
|---|---|
| Schindler's List (1993) | 4.47 |
| Wrong Trousers, The (1993) | 4.47 |
| Casablanca (1942) | 4.46 |
| Wallace & Gromit: The Best of Aardman Animation (1996) | 4.45 |

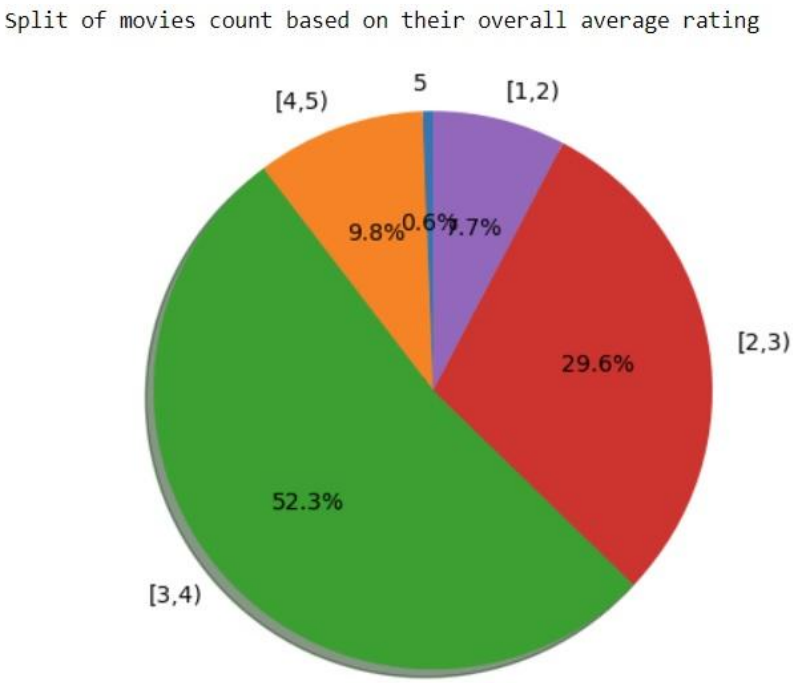
Πίνακας 5.4 Οι 20 ταινίες με τις υψηλότερες μέσες βαθμολογίες

Αυτό που παρατηρείται είναι πως μόνο 10 ταινίες έχουν λάβει την απόλυτη βαθμολογία κατά μέσο όρο, δηλαδή 5 στα 5. Από το σύνολο το 1664 ταινιών, 163 έχουν λάβει μέση βαθμολογία μικρότερη από 5 αλλά μεγαλύτερη ή ίση με 4 και 871 ταινίες με μέση βαθμολογία μικρότερη από 4 αλλά μεγαλύτερη ή ίση με 3. Αναλυτικά, στον παρακάτω πίνακα φαίνεται το πλήθος των ταινιών ανά διάστημα μέσης βαθμολογίας.

| **Μέση Βαθμολογία** | |
|---|---|
| 5 | 10 |
| [4,5) | 163 |
| [3,4) | 871 |
| [2,3) | 492 |
| [1,2) | 128 |
| < 1 | 0 |
| **Sum** | **1664** |

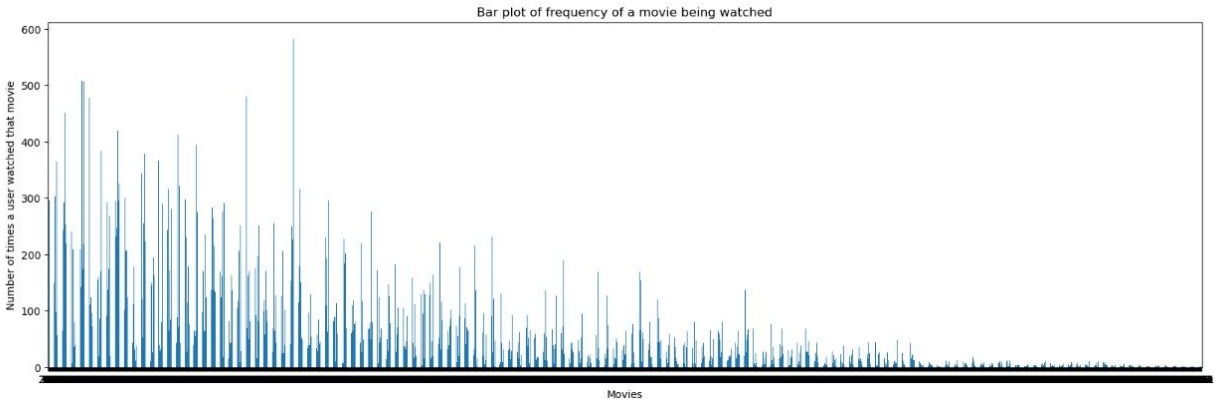
Πίνακας 5.5 Πλήθος ταινιών ανά διάστημα μέσης βαθμολογίας

Η πλειοψηφία των ταινιών, λοιπόν, παίρνουν μέση βαθμολογία από 3 μέχρι 4 αστέρια (52.3%) και αμέσως μετά από 2 μέχρι 3 αστέρια (29.6%). Διαγραμματικά αυτό μπορούμε να το δούμε από το επόμενο διάγραμμα πίτας.




Σχήμα 5. Ποσοστό συχνότητας ταινίας ανά διάστημα μέσης βαθμολογίας

Παρακάτω, δημιουργήθηκε ένα ιστόγραμμα με το πλήθος των φορών που κάποιος χρήστης έχει δει μια συγκεκριμένη ταινία.



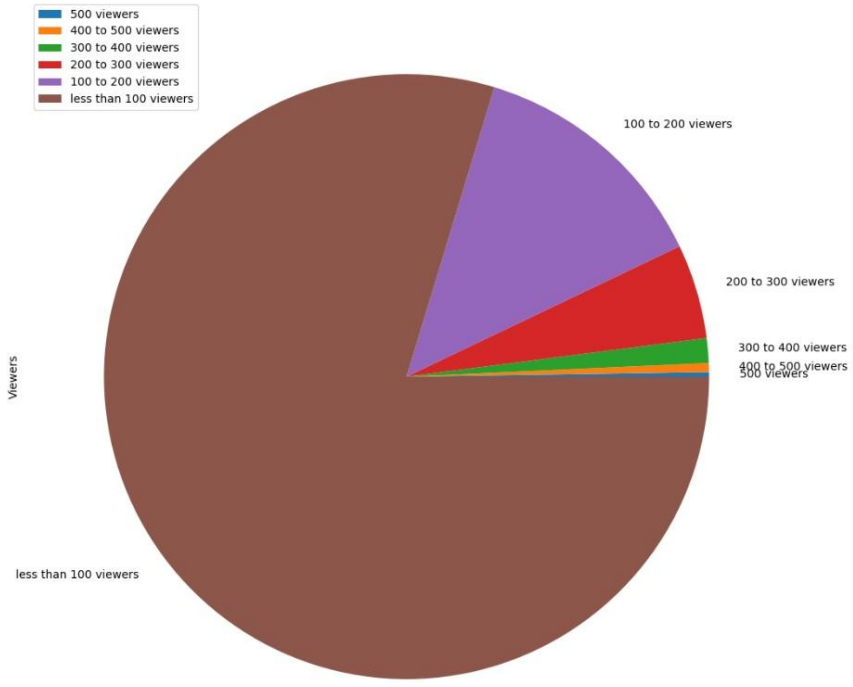

Σχήμα 6. Αριθμός χρηστών που έχουν δει μια συγκεκριμένη ταινία

Γίνεται αντιληπτό ότι πολύ λίγες ταινίες έχουν παρακολουθηθεί από περισσότερους από 100 από τους 943 συνολικά χρήστες. Για να γίνει σαφές το πόσες φορές έχει παρακολουθηθεί μία ταινία δίνεται ο παρακάτω πίνακας.

| | |
|---|---|
| Αριθμός ταινιών με περισσότερους από 500 τηλεθεατές | 4 |
| Αριθμός ταινιών με περισσότερους από 400 και λιγότερους από 500 τηλεθεατές | 8 |
| Αριθμός ταινιών με περισσότερους από 300 και λιγότερους από 400 τηλεθεατές | 22 |
| Αριθμός ταινιών με περισσότερους από 200 και λιγότερους από 300 τηλεθεατές | 84 |
| Αριθμός ταινιών με περισσότερους από 100 και λιγότερους από 200 τηλεθεατές | 220 |
| Αριθμός ταινιών με λιγότερους από100 τηλεθεατές | 1326 |

Πίνακας 5.6 Αριθμός ταινιών ανά διάστημα προβολών

Οι 1326 ταινίες από τις συνολικά 1664, έχουν λιγότερες από 100 προβολές και 220 ταινίες έχουν από 100 έως 200 προβολές. Οι ταινίες στα διαστήματα προβολών (200,300], (300,400], (400,500] και (500, ∞) είναι συνολικά 118.




Σχήμα 7. Γραφική απεικόνιση πλήθους ταινιών ανά διάστημα προβολών

Παρακάτω, παρατίθενται οι τίτλοι των πιο δημοφιλών ταινιών, δηλαδή αυτών με πάνω από 400 κριτικές. Όπως έχει ήδη αναφερθεί αυτές είναι μόνο 12.

| Τίτλος Ταινίας | Αριθμός κριτικών |
|---|---|
| 1. Star Wars (1977) | 583 |
| 2. Contact (1997) | 509 |
| 3. Fargo (1996) | 508 |
| 4. Return of Jedi (1983) | 507 |
| 5. Liar Liar (1997) | 485 |
| 6. English Patient, The (1996) | 481 |
| 7. Scream (1996) | 478 |
| 8. Toy Story (1995) | 452 |
| 9. Air Force One (1997) | 431 |
| 10. Independence Day (ID4) (1996) | 429 |
| 11. Raiders of the Lost Ark (1981) | 420 |
| 12. Godfather, The (1972) | 413 |

Πίνακας 5.7 Αριθμός κριτικών δημοφιλών ταινιών

Ως δημοφιλής, όμως, μπορεί να ονομαστεί μία ταινία και με βάση το μέσο Rating που έχει λάβει.

| Τίτλος Ταινίας | Μέσο Rating | Αριθμός Κριτικών |
|---|---|---|
| 1. They Made Me a Criminal (1939) | 5.00 | 1 |
| 2. Marlene Dietrich: Shadow and Light (1996) | 5.00 | 1 |
| 3. Saint of Fort Washington, The (1993) | 5.00 | 2 |
| 4. Someone Else's America (1995) | 5.00 | 1 |
| 5. Star Kid (1997) | 5.00 | 3 |
| 6. Great Day in Harlem, A (1994) | 5.00 | 1 |
| 7. Aiqing wansui (1994) | 5.00 | 1 |
| 8. Santa with Muscles (1996) | 5.00 | 2 |
| 9. Prefontaine (1997) | 5.00 | 3 |
| 10. Entertaining Angels: The Dorothy Day Story (1996) | 5.00 | 1 |

Πίνακας 5.8 Δημοφιλείς ταινίες με βάση τη μέση βαθμολογία

Οι δέκα ταινίες με το μεγαλύτερο μέσο Rating συγκεντρώνουν βαθμολογία 5.0/5.0. Αυτό δε μπορεί να είναι αντικειμενικό αν δούμε τον αριθμό κριτικών που έχει η κάθε μία. Οι δέκα αυτές ταινίες έχουν παρακολουθηθεί το πολύ τρεις φορές άρα η μέση βαθμολογία φαίνεται να μην

είναι αντικειμενική. Ένας πιο σωστός τρόπος αξιολόγησης για το αν μια ταινία είναι δημοφιλής ή όχι θα ήταν ο συνδυασμός της μέσης βαθμολογίας με τον αριθμό των κριτικών, όπως φαίνεται παρακάτω.

| **Τίτλος Ταινίας** | **Μέσο Rating** | **Αριθμός Κριτικών** |
|---|---|---|
| 23. Star Wars (1977) | 4.36 | 583 |
| 34. Godfather, The (1972) | 4.28 | 413 |
| 40. Raiders of the Lost Ark (1981) | 4.25 | 420 |
| 64. Fargo (1996) | 4.16 | 508 |
| 129. Return of Jedi (1983) | 4.01 | 507 |
| 236. Toy Story (1995) | 3.89 | 452 |
| 292. Contact (1997) | 3.80 | 509 |
| 412. English Patient, The (1996) | 3.66 | 481 |
| 428. Air Force One (1997) | 3.63 | 431 |
| 597. Scream (1996) | 3.44 | 478 |
| 598. Independence Day (ID4) (1996) | 3.44 | 429 |
| 837. Liar Liar (1997) | 3.16 | 485 |

Πίνακας 5.9 Μέσο Rating των δημοφιλέστερων ταινιών με βάση τις κριτικές

Γίνεται σαφές ότι μέσω του συγκεκριμένου πίνακα, προκύπτει μια πιο αντικειμενική άποψη για το αν μια ταινία πρέπει να προταθεί ή όχι. Για παράδειγμα, η ταινία Star Wars (1977) βρίσκεται στη θέση 23 με βάση το μέσο Rating. Παρ' όλα αυτά συγκεντρώνει βαθμολογία περίπου 4.36/5.0 μετά από 583 προβολές. Ακόμα και εμπειρικά, αυτό είναι πολύ πιο αξιόπιστο κριτήριο σύστασης σε σχέση με μία ταινία που συγκεντρώνει μέση βαθμολογία 5.0/5.0 μετά από μία ή δύο προβολές.

## 5.3 Εφαρμογή των μοντέλων

### 5.3.1 Λογιστική Παλινδρόμηση

Για να παραχθούν συστάσεις μέσω της λογιστικής παλινδρόμησης χρειάζεται να δημιουργηθεί μία δίτιμη μεταβλητή απόκρισης για το συγκεκριμένο μοντέλο. Κατασκευάζεται, λοιπόν, η μεταβλητή Label που παίρνει τιμή 1 εάν η βαθμολογία που έχει λάβει μια ταινία από ένα συγκεκριμένο χρήστη είναι μεγαλύτερη του 3 (για ratings ίσα με 4 ή 5). Διαφορετικά, η μεταβλητή Label παίρνει την τιμή 0 (για ratings ίσα με 1,2 ή 3). Ορίζεται, δηλαδή, σαν επιτυχία («κάνε σύσταση») για το μοντέλο, όταν η βαθμολογία (Rating) υπερβαίνει το 3 και σαν αποτυχία για το μοντέλο («μην κάνεις σύσταση»), όταν η βαθμολογία (Rating) είναι από 3 και κάτω.

Όπως έχει αναφερθεί και στην Ενότητα 5.2.1, το πλήρες σύνολο δεδομένων που χρησιμοποιείται έχει περικοπεί σε τρία επιμέρους υποσύνολα με χρονολογική σειρά (Training set, Validation set, Test set). Η δίτιμη μεταβλητή Label εκχωρείται και στα 3 επιμέρους υποσύνολα. Το μοντέλο λογιστικής παλινδρόμησης που χρησιμοποιείται στο υποσύνολο εκπαίδευσης (Train set) είναι το ακόλουθο:

$$logit(\hat{\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_{17} X_{17}$$

όπου οι ανεξάρτητες μεταβλητές $X_1, ..., X_{17}$ είναι οι κατηγορίες των ταινιών που υπάρχουν σε κάθε υποσύνολο των δεδομένων που χρησιμοποιούνται.

Αφού το πρώτο υποσύνολο (Train set) έχει ‘εκπαιδευθεί’, λαμβάνονται προβλέψεις (Predictions) για το υποσύνολο επικύρωσης (Validation set). Με βάση αυτές τις προβλέψεις το μοντέλο αξιολογείται μέσω των μετρικών απόδοσης που έχουν αναλυθεί στο Κεφάλαιο 3.

| Evaluation Metric | Score |
|---|---|
| Precision | 0.5355 |
| Recall | 0.1851 |
| F1-Score | 0.2751 |

Πίνακας 5.10 Σκορ μετρικών απόδοσης μοντέλου λογιστικής παλινδρόμησης

Όπως γίνεται άμεσα αντιληπτό η απόδοση του μοντέλου λογιστικής παλινδρόμησης θα μπορούσε να χαρακτηριστεί ως μέτρια. Οι ίδιες μετρικές θα χρησιμοποιηθούν και για την αξιολόγηση των παρακάτω μοντέλων ώστε να μπορέσει να επιλεχθεί το βέλτιστο μοντέλο για δημιουργία συστάσεων στα συγκεκριμένα δεδομένα.

### 5.3.2 Singular Value Decomposition

Για να παραχθούν συστάσεις μέσω της τεχνικής Singular Value Decomposition, χρειάζεται να δημιουργηθούν τρεις νέοι πίνακες, οι οποίοι όταν πολλαπλασιαστούν θα είναι ίσοι με τον αρχικό πίνακα του συνόλου δεδομένων. Εάν υποθέσουμε λοιπόν, ότι το αρχικό σύνολο δεδομένων είναι ο πίνακας $R$ διάστασης $m \times n$ ($m$=1664 ταινίες, $n$=943 χρήστες), κατασκευάζουμε τους πίνακες $P$, $\Sigma$ και $Q$, με τον πίνακα Σ να είναι διαγώνιος, ώστε $R = P\Sigma Q^T$. Έτσι παράγονται τα Predicted Ratings για κάθε χρήστη και κάθε ταινία. Στη συνέχεια, όταν η εκτιμώμενη βαθμολογία (Predicted Rating) υπερβαίνει το 3 αντικαθίσταται με τον αριθμό 1 («κάνε σύσταση»), ενώ όταν η εκτιμώμενη βαθμολογία (Predicted Rating) να είναι από 3 και κάτω αντικαθίσταται με τον αριθμό 0 («μην κάνεις σύσταση»).

Το πλήρες σύνολο δεδομένων που χρησιμοποιείται έχει περικοπεί σε τρία επιμέρους υποσύνολα με χρονολογική σειρά (Training set, Validation set, Test set). Σε πρώτο χρόνο ο αλγόριθμος SVD εκτελείται στα δεδομένα εκπαίδευσης (Train Set). Σε δεύτερο χρόνο, αφού ο αλγόριθμος έχει ‘εκπαιδευθεί’, εκτελείται και στα δεδομένα επικύρωσης (Validation Set) και παράγονται νέα Predictions. Με βάση αυτές τις προβλέψεις το μοντέλο αξιολογείται με τις ίδιες μετρικές απόδοσης που έχουν αναφερθεί παραπάνω.

| Evaluation Metric | Score |
|---|---|
| Precision | 0.7778 |
| Recall | 0.2505 |
| F1-Score | 0.3790 |

Πίνακας 5.11 Σκορ μετρικών απόδοσης τεχνικής SVD

Σε αυτή την περίπτωση, γίνεται αντιληπτό πως η απόδοση της τεχνικής SVD είναι σαφώς καλύτερη από αυτή του μοντέλου της λογιστικής παλινδρόμησης. Η ακρίβεια με την οποία κάνει

συστάσεις η συγκεκριμένη τεχνική είναι σχεδόν 78%, ποσοστό πολύ μεγαλύτερο από το 53% που παρατηρήθηκε προηγουμένως.

### 5.3.3 Alternating Least Squares

Για να παραχθούν συστάσεις μέσω της τεχνικής ALS, αναλύεται ο κύριος πίνακας της εφαρμογής (πίνακας χρηστών-βαθμολογιών), σε δύο μικρότερους πίνακες που πολλαπλασιάζονται για να προσεγγίσουν τον αρχικό. Πιο συγκεκριμένα, ο αρχικός πίνακας *R* που χρησιμοποιεί ο αλγόριθμος είναι διάστασης 943×1664 (χρήστες x ταινίες). Η διάσπασή του γίνεται σε έναν πίνακα *U* διάστασης 943 × *k* και σε έναν πίνακα *V* διάστασης 1664 × *k*.

Ο πίνακας *R* είναι περίπου ίσος με το γινόμενο του *U* με τον ανάστροφο του *V*. Το *k,* όπως έχει ήδη αναφερθεί, είναι μια παράμετρος που καθορίζει τον αριθμό των κρυφών παραγόντων, που είναι τα χαρακτηριστικά που επηρεάζουν τις αξιολογήσεις.

Με αυτόν τον τρόπο, η ALS μπορεί να παράγει μια προσέγγιση του αρχικού πίνακα αξιολογήσεων χρηστών-αντικειμένων. Στη συνέχεια, αυτή η προσέγγιση μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις σχετικά με τις προτεινόμενες ταινίες για έναν συγκεκριμένο χρήστη.

Το πλήρες σύνολο δεδομένων που χρησιμοποιείται έχει περικοπεί σε τρία επιμέρους υποσύνολα με χρονολογική σειρά (Training set, Validation set, Test set). Σε πρώτο χρόνο ο αλγόριθμος ALS εκτελείται στα δεδομένα εκπαίδευσης (Train Set). Σε δεύτερο χρόνο, αφού ο αλγόριθμος έχει ‘εκπαιδευθεί’, εκτελείται και στα δεδομένα επικύρωσης (Validation Set) και παράγονται νέα Predictions. Με βάση αυτές τις προβλέψεις το μοντέλο αξιολογείται με τις ίδιες μετρικές απόδοσης που έχουν αναφερθεί παραπάνω.

| **Evaluation Metric** | **Score** |
|---|---|
| Precision | 0.6582 |
| Recall | 0.2313 |
| F1-Score | 0.3423 |

Πίνακας 5.12 Σκορ μετρικών απόδοσης τεχνικής ALS

Σε αυτή την περίπτωση, γίνεται αντιληπτό πως η απόδοση της τεχνικής ALS είναι σαφώς καλύτερη από αυτή του μοντέλου της λογιστικής παλινδρόμησης, όμως όχι καλύτερη από αυτή

της τεχνικής SVD. Η ακρίβεια με την οποία κάνει συστάσεις η συγκεκριμένη τεχνική είναι σχεδόν 66%, ποσοστό μικρότερο από το 78% που παρατηρήθηκε προηγουμένως.

## 5.4 Αποτελέσματα

Στην παρούσα εφαρμογή δημιουργήθηκαν τρία συστήματα συστάσεων για το ίδιο σύνολο δεδομένων. Το πρώτο σύστημα εμπίπτει στην κατηγορία συστημάτων συστάσεων Content Based, ενώ τα άλλα δύο στην κατηγορία συστημάτων συστάσεων Collaborative Filtering.

Όπως γίνεται συνήθως στις εφαρμογές μηχανικής μάθησης, το πλήρες σύνολο δεδομένων χωρίστηκε σε τρία επιμέρους υποσύνολα με χρονολογική σειρά (Train Set, Validation Set, Test Set). Οι τρεις τεχνικές που χρησιμοποιήθηκαν ήταν η λογιστική παλινδρόμηση, η τεχνική SVD και η τεχνική ALS.

Στο πρώτο υποσύνολο (Train Set), 'εκπαιδεύθηκαν' και τα τρία μοντέλα. Στη συνέχεια, στο δεύτερο υποσύνολο (Validation Set), παράχθηκαν προβλέψεις και αξιολογήθηκαν. Ο σκοπός που γίνεται αυτό είναι η αποφυγή της μεροληψίας. Έχοντας εκπαιδεύσει και τα τρία μοντέλα στο ίδιο σύνολο δεδομένων (Train Set), τους ζητάτε να κάνουν προβλέψεις για ένα ξένο δεύτερο σύνολο (Validation Set). Με αυτόν τον τρόπο δημιουργούνται προβλέψεις για άγνωστα δεδομένα και για τα τρία μοντέλα, οπότε μπορούν να αξιολογηθούν σωστά.

Οι μετρικές απόδοσης που χρησιμοποιούνται στην παρούσα εργασία είναι η Ακρίβεια (Precision), η Ανάκληση (Recall) και το F1-Score που συνδυάζει τις προηγούμενες δύο. Ο παρακάτω πίνακας δίνει τα συγκεντρωτικά αποτελέσματα απόδοσης και των τριών μοντέλων.

| **Logistic Regression** | |
|---|---|
| **Evaluation Metric** | **Score** |
| Precision | 0.5355 |
| Recall | 0.1851 |
| F1-Score | 0.2751 |
| **Singular Value Decomposition** | |
| **Evaluation Metric** | **Score** |
| Precision | 0.7778 |
| Recall | 0.2505 |
| F1-Score | 0.3790 |

| Alternating Least Squares | |
|---|---|
| **Evaluation Metric** | **Score** |
| Precision | 0.6582 |
| Recall | 0.2313 |
| F1-Score | 0.3423 |

Πίνακας 5.13 Συγκεντρωτικά αποτελέσματα απόδοσης των τριών μοντέλων

Σύμφωνα με τον παραπάνω πίνακα, η τεχνική SVD υπερτερεί των άλλων δύο. Δίνει καλύτερα μέτρα απόδοσης και ως προς την ακρίβεια των συστάσεων (Precision) και ως προς την ανάκληση (Recall). Συμπεραίνεται, λοιπόν, πως για τα συγκεκριμένα δεδομένα και βάσει των τεχνικών που δοκιμάστηκαν στην παρούσα εργασία, η τεχνική SVD δημιουργεί το πιο αξιόπιστο σύστημα συστάσεων ταινιών.

Ο κύριος τρόπος για να αξιολογηθεί ένα σύστημα συστάσεων είναι η Ακρίβεια (Precision). Γι αυτό το λόγο οι τεχνικές που χρησιμοποιήθηκαν αξιολογήθηκαν με κύριο γνώμονα τα σκορ της Ακρίβειας. Ωστόσο προκειμένου να αποκτηθεί σφαιρική εικόνα για την αξιολόγηση των τεχνικών, υπολογίστηκε η Ανάκληση και το F1-Score. Η τεχνική SVD εκτός από υψηλότερη ακρίβεια, παρουσίασε και υψηλότερες τιμές Ανάκλησης και F1-Score.

Το τρίτο υποσύνολο δεδομένων που έχει κατασκευαστεί (Test Set) δεν έχει χρησιμοποιηθεί πουθενά μέχρι στιγμής. Αφού πλέον, τα τρία διαφορετικά συστήματα συστάσεων έχουν τρέξει, εκπαιδευθεί (Train Set) και έχουν αξιολογηθεί σε άγνωστα δεδομένα (Validation Set), ήρθε η ώρα το επικρατέστερο (SVD) να αξιολογηθεί μόνο του στο τρίτο υποσύνολο (Test Set). Ο λόγος που το επικρατέστερο μοντέλο δοκιμάζεται σε ένα τελείως ξένο υποσύνολο (Test Set), σε σχέση με τα προηγούμενα, είναι για να αποφευχθεί η μεροληψία. Θέλουμε, δηλαδή, το μοντέλο της SVD να δημιουργήσει συστάσεις και να αξιολογηθεί γι αυτές σε ένα σύνολο τελείως άγνωστο.

Τα αποτελέσματα δίνονται στον παρακάτω πίνακα.

| **Evaluation Metric** | **Score** |
|---|---|
| Precision | 0.8336 |
| Recall | 0.2381 |
| F1-Score | 0.3705 |

Πίνακας 5.14 Σκορ μετρικών απόδοσης τεχνικής SVD στο υποσύνολο δεδομένων Test

Παρατηρείται ότι, η τεχνική SVD αξιολογείται να έχει ακρίβεια πάνω από 83% σε ξένα δεδομένα. Το συγκεκριμένο σκορ είναι αρκετά ικανοποιητικό ώστε το μοντέλο να θεωρηθεί αξιόπιστο για παραγωγή συστάσεων σε νέα-άγνωστα δεδομένα από εδώ και στο εξής.

# ΚΕΦΑΛΑΙΟ 6

# Συμπεράσματα και συζήτηση

Στόχος της παρούσας εργασίας ήταν, να δοθεί μια περιγραφή του θεωρητικού υποβάθρου των συστημάτων συστάσεων και στη συνέχεια να δημιουργηθούν τρία συστήματα, διαφόρων κατηγοριών σε πραγματικά δεδομένα, ώστε να συγκριθούν μεταξύ τους και να επιλεγεί το πιο αξιόπιστο.

Τα δεδομένα ταινιών που χρησιμοποιήθηκαν υπάρχουν διαθέσιμα στο διαδίκτυο, δωρεάν, και η συλλογή, ανάλυση και επεξεργασία τους έχει διενεργηθεί από την ερευνητική ομάδα GroupLens του πανεπιστημίου της Minnesota των ΗΠΑ.

Τα συστήματα συστάσεων σύμφωνα με τη διεθνή βιβλιογραφία χωρίζονται σε δύο μεγάλες κατηγορίες. Αυτές είναι: Content Based Recommender Systems και Collaborative Filtering Recommender Systems. Το πρώτο σύστημα συστάσεων που δημιουργήθηκε ανήκει στην κατηγορία Content Based και χρησιμοποίησε ως στατιστική τεχνική τη λογιστική παλινδρόμηση. Τα επόμενα δύο συστήματα συστάσεων που δημιουργήθηκαν ανήκουν στην κατηγορία Collaborative Filtering. Το πρώτο εκ των δύο χρησιμοποίησε τη στατιστική τεχνική τη μέθοδο Singular Value Decomposition (SVD). Το δεύτερο χρησιμοποίησε τη μέθοδο Alternating Least Squares (ALS).

Για να συγκριθούν τα τρία συστήματα συστάσεων χρησιμοποιήθηκαν οι μετρικές αξιολόγησης Precision, Recall και F1-Score. Μεταξύ των τριών, μεγαλύτερες τιμές στα metrics έδωσε το μοντέλο της SVD και κατ’ επέκταση μεγαλύτερη αξιοπιστία για να το εμπιστευθεί κανείς για παραγωγή συστάσεων ταινιών σε νέα-άγνωστα δεδομένα.

# Παράρτημα

## Κώδικες

Για την εφαρμογή του Κεφαλαίου 5 χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python.

### Εισαγωγή βιβλιοθηκών

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import pyspark

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score

from scipy.stats import ttest_ind

import datetime#

import sys

import scipy.sparse as sp

from scipy.sparse.linalg import svds

from sklearn.metrics import precision_score, recall_score, f1_score,
ndcg_score

from pyspark.ml.recommendation import ALS

from pyspark.sql import SparkSession
```

```
from pyspark import SparkContext

from pyspark.sql.functions import col
```

## Εισαγωγή και προπαρασκευή των δεδομένων

```
ratings_link = "C:/Users/user/Desktop/Project/ml-100k/ml-100k/u.data"

df1 = pd.read_csv(ratings_link,sep= "\t",header=
None,names=["userID","itemID","Rating","TimeStamp"],encoding ="ISO-
8859-1")

movies_link = "C:/Users/user/Desktop/Project/ml-100k/ml-100k/u.item"

df2 = pd.read_csv(movies_link,sep= "|",header=
None,index_col=False,names=["itemID","Title","Unknown
Cat","Action","Adventure",
"Animation","Children's","Comedy","Crime","Documentary","Drama","Fantas
y","Film-Noir","Horror","Musical",
          "Mystery","Romance","Sci-Fi","Thriller","War","Western"],
usecols=[0,1,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23],encoding
="ISO-8859-1")

data = pd.merge(df1,df2,left_on='itemID',right_on='itemID')
```

## Table 2

```
data.isna().sum()
```

## Table 3

```
def ttest_calculator(category):
   return ttest_ind(data[data[category] == 1]['Rating'],data[data[category] ==
0]['Rating'])

ttest_summary = {}
for col in data.columns[5:]:
   ttest_summary[col] = ttest_calculator(col)[1]

{k: v for k, v in sorted(ttest_summary.items(), key=lambda item: item[1])}
```

## Απόρριψη μη στατιστικά σημαντικών στηλών

```
data = data.drop(columns=["Unknown Cat", "Musical"])
```

## Προσθήκη μεταβλητής Date

```
data['Date'] = data['TimeStamp'].apply(lambda x :
datetime.datetime.fromtimestamp(x))
```

## Figure 4

```
sort_data = data['Rating'].value_counts(sort=False).sort_index()

sort_data.plot(kind='bar' ,figsize=(10,8), use_index = True, rot=0)
plt.title('Bar plot of rating frequency')

plt.xlabel('Rating')

plt.ylabel('Number of times a rating was given')

r4 = [1, 2, 3 , 4, 5]

for i in range(len(sort_data)):
    plt.text(x = r4[i] - 1.2, y = sort_data.iloc[i]+500, s = sort_data.iloc[i], size
=15)
```

## Table 4

```
avg_highly_rated_movies =
data.groupby(['Title']).agg({"Rating":"mean"})['Rating'].sort_values(ascending
=False)

avg_highly_rated_movies = avg_highly_rated_movies.to_frame()

avg_highly_rated_movies.head(20)
```

## Table 5

```
print("Number of movies with 5 star rating on average:
",len(avg_highly_rated_movies[avg_highly_rated_movies['Rating'] == 5.0]))

print("Number of movies with above 4 star and below 5 star rating on average:
",len(avg_highly_rated_movies[(avg_highly_rated_movies['Rating'] >= 4.0) &
(avg_highly_rated_movies['Rating'] < 5.0)]))

print("Number of movies with above 3 star and below 4 star rating on average:
",len(avg_highly_rated_movies[(avg_highly_rated_movies['Rating'] >= 3.0) &
(avg_highly_rated_movies['Rating'] < 4.0)]))

print("Number of movies with above 2 star and below 3 star rating on average:
",len(avg_highly_rated_movies[(avg_highly_rated_movies['Rating'] >= 2.0) &
(avg_highly_rated_movies['Rating'] < 3.0)]))

print("Number of movies with above 1 star and below 2 star rating on average:
",len(avg_highly_rated_movies[(avg_highly_rated_movies['Rating'] >= 1.0) &
(avg_highly_rated_movies['Rating'] < 2.0)]))

print("Number of movies with below 1 star rating on average: ",
len(avg_highly_rated_movies[(avg_highly_rated_movies['Rating'] < 1.0)]))
```

## Figure 5

```
print('Split of movies count based on their overall average rating')

labels = '5 star', '4 to 5 star', '3 to 4 star', '2 to 3 star', '1 to 2 star'

sizes = [10, 163, 871, 492, 128]

fig1, ax1 = plt.subplots()

ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)

ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```

## Figure 6

```
data['itemID'].value_counts(sort=False).plot(kind='bar' ,figsize=(20,6),
use_index = True, rot=0)

plt.title('Bar plot of frequency of a movie being watched')

plt.xlabel('Movies')

plt.ylabel('Number of times a user watched that movie')
```

## Table 6

```
popular_movies =
data.groupby(['Title']).agg({"Rating":"count"})['Rating'].sort_values(ascending
=False)

popular_movies = popular_movies.to_frame()

popular_movies.reset_index(level=0, inplace=True)

popular_movies.columns = ['Title', 'Number of Users watched']

print("Number of popular movies with more than 500 viewers:
",len(popular_movies[popular_movies['Number of Users watched'] >= 500]))

print("Number of popular movies with more than 400 and less than 500
viewers: ",len(popular_movies[(popular_movies['Number of Users watched']
>= 400) & (popular_movies['Number of Users watched'] < 500)]))

print("Number of popular movies with more than 300 and less than 400
viewers: ",len(popular_movies[(popular_movies['Number of Users watched']
>= 300) & (popular_movies['Number of Users watched'] < 400)]))

print("Number of popular movies with more than 200 and less than 300
viewers: ",len(popular_movies[(popular_movies['Number of Users watched']
>= 200) & (popular_movies['Number of Users watched'] < 300)]))

print("Number of popular movies with more than 100 and less than 200
viewers: ",len(popular_movies[(popular_movies['Number of Users watched']
>= 100) & (popular_movies['Number of Users watched'] < 200)]))

print("Number of popular movies with less than 100 viewers: ",
```

```
len(popular_movies[(popular_movies['Number of Users watched'] < 100)]))
```

**Figure 7**

```
df = pd.DataFrame({'Viewers': [4, 8, 22, 84, 220, 1326]},
                  index=['500 viewers', '400 to 500 viewers', '300 to 400 viewers',
'200 to 300 viewers', '100 to 200 viewers', 'less than 100 viewers'])

plot = df.plot.pie(y='Viewers', figsize=(8, 8))
```

**Table 7**

```
popular_movies[popular_movies['Number of Users watched'] >= 400]
```

**Table 8**

```
highly_rated_popular_movies = pd.merge(avg_highly_rated_movies,
popular_movies, how = 'inner', on='Title')

highly_rated_popular_movies.head(10)
```

**Table 9**

```
highly_rated_popular_movies[highly_rated_popular_movies['Number of Users
watched']>400]
```

## Κατασκευή των τριών υποσυνόλων δεδομένων

```
validation_date = min(data['Date']) + datetime.timedelta(days=120)

test_date = validation_date + datetime.timedelta(days=45)

data_train = data[data['Date'] <= validation_date]

data_validation = data[(data['Date'] > validation_date) & (data['Date'] <=
test_date)]
```

```
data_test = data[data['Date'] > test_date]
```

## Προσθήκη της μεταβλητής Label στα τρία υποσύνολα

```
data_train['Label'] = np.where(data_train['Rating'] > 3, 1, 0)

data_validation['Label'] = np.where(data_validation['Rating'] > 3, 1, 0)

data_test['Label'] = np.where(data_test['Rating'] > 3, 1, 0)
```

## Λογιστική Παλινδρόμηση

```
clf = LogisticRegression(random_state=0).fit(data_train.iloc[:, 6:22],
data_train['Label'])

y_pred1 = clf.predict(data_train.iloc[:, 6:22])

y_pred2 = clf.predict(data_validation.iloc[:, 6:22])
```

## Table 10

```
print("Precision:", precision_score(data_validation['Label'], y_pred2))

print("Recall:", recall_score(data_validation['Label'], y_pred2))

print("F1-score:", f1_score(data_validation['Label'], y_pred2))
```

## Singular Value Decomposition

```
pivot_df1 = data_train.pivot(index='itemID', columns='userID',
values='Rating').fillna(0)

pivot_df1

array_for_train_set = pivot_df1.values.astype(float)

array_for_train_set

u, s, vt = svds(array_for_train_set,k=10)
```

```
s_diag_matrix = np.diag(s)
X_pred1 = np.dot(np.dot(u, s_diag_matrix), vt)

mask = X_pred1 > 3

X_pred1[mask] = 1

X_pred1[~mask] = 0

X_pred1.shape

X_pred1

y_true_svd_1 = []
y_pred_svd_1 = []
pivot_matrix1 = pivot_df1.values

for i in range(pivot_matrix1.shape[0]):
    for j in range(pivot_matrix1.shape[1]):
        if pivot_matrix1[i,j] == 0:
            continue
        else:
            y_true_svd_1.append(int(pivot_matrix1[i,j]>3))
            y_pred_svd_1.append(X_pred1[i,j])

pivot_df2 = data_validation.pivot(index='itemID', columns='userID',
values='Rating').fillna(0)

pivot_df2

array_for_validation_set = pivot_df2.values.astype(float)

array_for_validation_set

u, s, vt = svds(array_for_validation_set,k=10)
s_diag_matrix = np.diag(s)
X_pred2 = np.dot(np.dot(u, s_diag_matrix), vt)

mask = X_pred2 > 3

X_pred2[mask] = 1

X_pred2[~mask] = 0

X_pred2.shape
```

```
X_pred2

y_true_svd_2 = []
y_pred_svd_2 = []
pivot_matrix2 = pivot_df2.values

for i in range(pivot_matrix2.shape[0]):
    for j in range(pivot_matrix2.shape[1]):
        if pivot_matrix2[i,j] == 0:
            continue
        else:
            y_true_svd_2.append(int(pivot_matrix2[i,j]>3))
            y_pred_svd_2.append(X_pred2[i,j])
```

## Table 11

```
print("Precision:", precision_score(y_true_svd_2,y_pred_svd_2))

print("Recall:", recall_score(y_true_svd_2, y_pred_svd_2))

print("F1-score:", f1_score(y_true_svd_2, y_pred_svd_2))
```

## Alternating Least Squares

```
def start_or_get_spark(
    app_name="Sample",
    url="local[*]",
    memory="10g",
    config=None,
    packages=None,
    jars=None,
    repositories=None,
):
    """Start Spark if not started

    Args:
        app_name (str): set name of the application
        url (str): URL for spark master
        memory (str): size of memory for spark driver. This will be ignored if
spark.driver.memory is set in config.
        config (dict): dictionary of configuration options
        packages (list): list of packages to install
```

```
        jars (list): list of jar files to add
        repositories (list): list of maven repositories

    Returns:
        object: Spark context.
    """

    submit_args = ""
    if packages is not None:
        submit_args = "--packages {} ".format(",".join(packages))
    if jars is not None:
        submit_args += "--jars {} ".format(",".join(jars))
    if repositories is not None:
        submit_args += "--repositories {}".format(",".join(repositories))
    if submit_args:
        os.environ["PYSPARK_SUBMIT_ARGS"] = "{} pyspark-
shell".format(submit_args)

    spark_opts = [
        'SparkSession.builder.appName("{}")'.format(app_name),
        'master("{}")'.format(url),
    ]

    if config is not None:
        for key, raw_value in config.items():
            value = (
                '"{}"'.format(raw_value) if isinstance(raw_value, str) else raw_value
            )
            spark_opts.append('config("{key}", {value})'.format(key=key,
value=value))

    if config is None or "spark.driver.memory" not in config:
        spark_opts.append('config("spark.driver.memory",
"{}")'.format(memory))

    # Set larger stack size
    spark_opts.append('config("spark.executor.extraJavaOptions", "-Xss4m")')
    spark_opts.append('config("spark.driver.extraJavaOptions", "-Xss4m")')

    spark_opts.append("getOrCreate()")
    return eval(".".join(spark_opts))

spark = start_or_get_spark("ALS PySpark", memory="16g")

spark.conf.set("spark.sql.analyzer.failAmbiguousSelfJoin", "false")
```

```
model1 =
ALS(rank=10,maxIter=15,implicitPrefs=False,regParam=0.05,coldStartStrateg
y='drop',
   nonnegative=False,seed=42,userCol='userID', itemCol='itemID',
ratingCol='Rating').fit(data_train)

predict1 = model1.transform(data_train)

model2 =
ALS(rank=10,maxIter=15,implicitPrefs=False,regParam=0.05,coldStartStrateg
y='drop',
   nonnegative=False,seed=42,userCol='userID', itemCol='itemID',
ratingCol='Rating').fit(data_validation)

predict2 = model2.transform(data_validation)
```

**Table 11**

```
true_positive = predict2.filter((col("Rating") >= 4) & (col("prediction") >=
4)).count()

false_positive = predict2.filter((col("Rating") < 4) & (col("prediction") >=
4)).count()

false_negative = predict2.filter((col("Rating") >= 4) & (col("prediction") <
4)).count()

precision = true_positive / (true_positive + false_positive)

recall = true_positive / (true_positive + false_negative)

f1_score = 2 * (precision * recall) / (precision + recall)

print("Precision = " + str(precision))

print("Recall = " + str(recall))

print("F1 Score = " + str(f1_score))
```

## Εφαρμογή SVD στο Test set

```
pivot_df3 = data_test.pivot(index='itemID', columns='userID',
values='Rating').fillna(0)

pivot_df3

array_for_test_set = pivot_df3.values.astype(float)

array_for_test_set

u, s, vt = svds(array_for_test_set,k=10)
s_diag_matrix = np.diag(s)
X_pred3 = np.dot(np.dot(u, s_diag_matrix), vt)

mask = X_pred3 > 3

X_pred3[mask] = 1

X_pred3[~mask] = 0

X_pred3.shape

X_pred3

y_true_svd_3 = []
y_pred_svd_3 = []
pivot_matrix3 = pivot_df3.values

for i in range(pivot_matrix3.shape[0]):
    for j in range(pivot_matrix3.shape[1]):
        if pivot_matrix3[i,j] == 0:
            continue
        else:
            y_true_svd_3.append(int(pivot_matrix3[i,j]>3))
            y_pred_svd_3.append(X_pred3[i,j])
```

**Table 14**

```
print("Precision:", precision_score(y_true_svd_3,y_pred_svd_3))

print("Recall:", recall_score(y_true_svd_3, y_pred_svd_3))

print("F1-score:", f1_score(y_true_svd_3, y_pred_svd_3))
```

# Βιβλιογραφία


1. Mahmood, T., Ricci, F., “Improving recommender systems with adaptive conversational strategies”, in C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp 73–82, ACM, 2009
2. Resnick, P., Varian, H.R.: “Recommender systems”, Communications of the ACM 40(3), pp 56–58, 1997
3. Burke, R., “Hybrid web recommender systems, in The AdaptiveWeb”, Springer Berlin-Heidelberg, pp 377–408, 2007
4. Jannach, D., “Finding preferred query relaxations in content-based recommenders”, in 3rd International IEEE Conference on Intelligent Systems, pp 355–360, 2006
5. McSherry, F., Mironov, I., “Differentially private recommender systems: building privacy into the net”, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 627–636. ACM, NY, 2009
6. Schwartz, B., “The Paradox of Choice”, ECCO, New York, 2004
7. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Recommender systems for large-scale ecommerce: Scalable neighborhood formation using clustering”, in Proceedings of the fifth international conference on computer and information technology, pp 291–324, 2002
8. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Incremental singular value decomposition algorithms for highly scalable recommender systems”, in Fifth international conference on computer and information science, pp 27–8, 2002
9. G. Takács, I. Pilászy, B. Németh, and D. Tikk, “Scalable collaborative filtering approaches for large recommender systems”, The Journal of Machine Learning Research, pp 623–656, 2009
10. M. A. Ghazanfar and A. Prugel-Bennett, “A scalable, accurate hybrid recommender system”, Third International Conference on Knowledge Discovery and Data Mining, IEEE, pp 94–98, 2010
11. O. Georgiou and N. Tsapatsoulis, “Improving the scalability of recommender systems by clustering using genetic algorithms”, in International conference on artificial neural networks, Springer, pp 442–449, 2010

12. F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook”, in Recommender systems handbook. Springer, pp 1–35, 2011
13. Sardianos Christos, “Recommender systems with real-life applications”, PhD thesis, Harokopio University of Athens, pp 50-52, 2023
14. D. W. Oard, J. Kim et al., “Implicit feedback for recommender systems”, in Proceedings of the AAAI workshop on recommender systems, pp 81-83, 1998
15. R. Burke, “Hybrid recommender systems: Survey and experiments”, User modeling and user-adapted interaction, pp 331–370, 2002
16. M. Montaner, B. López, and J. L. De La Rosa, “A taxonomy of recommender agents on the internet”, Artificial intelligence review, pp 285–330, 2003
17. P. Melville and V. Sindhwani, “Recommender systems”, Encyclopedia of machine learning, pp 829–838, 2010
18. C. Sardianos, N. Tsirakis, and I. Varlamis, “A survey on the scalability of recommender systems for social networks”, in Social Networks Science: Design, Implementation, Security, and Challenges, Springer, pp 89–110, 2018
19. J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, “Collaborative filtering and deep learning based recommendation system for cold start items”, Expert Systems with Applications, pp 29–39, 2017
20. Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems”, Computer, pp 30–37, 2009
21. A. Abbas, L. Zhang, and S. U. Khan, “A survey on context-aware recommender systems based on computational intelligence techniques”, Computing, pp 667–690, 2015
22. J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, “Recommendations in location-based social networks: a survey”, GeoInformatica, pp 525–565, 2015
23. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms”, in Proceedings of the 10th international conference on World Wide Web, pp 285–295, 2001
24. P. Lops, M. De Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends”, in Recommender systems handbook. Springer, pp 73–105, 2011
25. M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in The adaptive web, Springer, pp 325–341, 2007

26. F. Strub, R. Gaudel, and J. Mary, “Hybrid recommender system based on autoencoders”, in Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp 11–16, 2016
27. W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, and X. Li, “Connecting social media to e-commerce: Cold-start product recommendation using microblogging information”, IEEE Transactions on Knowledge and Data Engineering, pp 1147–1159, 2015
28. H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems”, in Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1235–1244, 2015
29. J. Liu and C. Wu, “Deep learning based recommendation: A survey”, in International Conference on Information Science and Applications. Springer, pp 451–458, 2017
30. Manavalan R., “Towards an intelligent approaches for cotton diseases detection: A review”, Computers and Electronics in Agriculture, pp 107-255, 2022
31. Pedro H. M. Delmondes, Fátima L. S. Nunes, “A systematic review of multi-slice and multi-frame descriptors in cardiac MRI exams”, Computer Methods and Programs in Biomedicine, pp 106-889, 2022
32. Ranjan, N. M. and R. S. Prasad, “LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features”, Applied Soft Computing, pp 994-1008, 2018
33. Parker, B. R. and A. Reisman, “Socio-Economic Considerations”, Encyclopedia of Social Measurement, K. Kempf-Leonard, New York, Elsevier, pp 547-557, 2005
34. W. Wu, L. He, and J. Yang, “Evaluating recommender systems”, in Seventh International Conference on Digital Information Management, IEEE, pp 56–61, 2012
35. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems”, ACM Transactions on Information Systems (TOIS), pp 5–53, 2004
36. Swearingen, K., Sinha, R., “Beyond algorithms: An HCI perspective on recommender systems”, in J.L. Herlocker (ed.) Recommender Systems, papers from the 2001 ACM SIGIR Workshop, New Orleans, LA, 2001
37. Πολίτης Κ., “Γενικευμένα Γραμμικά Μοντέλα”, Πανεπιστημιακές Σημειώσεις, ΠΜΣ Εφαρμοσμένη Στατιστική, Πανεπιστήμιο Πειραιώς, 2022

38. Y. Koren, R. Bell and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems”, in Computer, IEEE, pp 30-37, 2009
39. Kuroda, M., Y. Mori and M. Iizuka, “Initial Value Selection for the Alternating Least Squares Algorithm”, Advanced Studies in Classification and Data Science, Springer Singapore, pp 227-239, 2020
40. Kuroda, M., Y. Mori, M. Iizuka and M. Sakakihara, “Acceleration of the alternating least squares algorithm for principal components analysis”, Computational Statistics & Data Analysis, pp 143-153, 2011
41. Hoerl, Arthur E., and Robert W. Kennard. “Ridge Regression: Biased Estimation for Nonorthogonal Problems”, Technometrics, 55–67, 1970
42. F. Maxwell Harper, Joseph A. Konstan, “The MovieLens Datasets: History and Context”, in ACM Transactions, Interactive Intelligent Systems, pp 1-19, 2015