ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
“ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ”

Εφαρμογές τεχνικών μηχανικής μάθησης σε δεδομένα γράφων

Από
Σάββας Λουρίδας
ΜΕ2143

Υποβάλλεται
για την εκπλήρωση των προϋποθέσεων λήψης
Μεταπτυχιακού Διπλώματος
στην ειδίκευση “Προηγμένα Πληροφοριακά Συστήματα”
του ΠΜΣ “Πληροφοριακά Συστήματα & Υπηρεσίες”
στο
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Φεβρουάριος 2024

Επιβλέπουσα: Μαρία Χαλκίδη
Ακαδημαϊκή Θέση: Αναπληρώτρια Καθηγήτρια

Συγγραφέας: Σάββας Λουρίδας

ΣΕΛΙΔΑ ΕΓΚΥΡΟΤΗΤΑΣ

Ονοματεπώνυμο Φοιτητή: Σάββας Λουρίδας

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας: Εφαρμογές τεχνικών μηχανικής μάθησης σε δεδομένα γράφων

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών "Πληροφοριακά Συστήματα & Υπηρεσίες" του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις …… [ημερομηνία έγκρισης] από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή
Επιβλέπων/ουσα (Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς)

……..……[ονοματεπώνυμο, βαθμίδα, υπογραφή]

Μέλος Εξεταστικής Επιτροπής: ……..[ονοματεπώνυμο, βαθμίδα, υπογραφή]

Μέλος Εξεταστικής Επιτροπής: ……..[ονοματεπώνυμο, βαθμίδα, υπογραφή]

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

Ο Σάββας Λουρίδας, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Εφαρμογές τεχνικών μηχανικής μάθησης σε δεδομένα γράφων», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.
Επιπλέον *δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.*
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται τις διατάξεις που προβλέπει η Ελληνική και Κοινοτική Νομοθεσία περί πνευματικής ιδιοκτησίας.

Ο ΔΗΛΩΝ

Ονοματεπώνυμο: Σάββας Λουρίδας
Αριθμός Μητρώου: ΜΕ2143
Υπογραφή: Σάββας Λουρίδας

Κεφάλαια	Περιεχόμενο	Μέγεθος (σελίδες)
Εξώφυλλο	Το εξώφυλλο της ΜΔΕ.	1
Σελίδα πνευματικών δικαιωμάτων	Η σελίδα πνευματικών δικαιωμάτων της ΜΔΕ.	1
Σελίδα εγκυρότητας	Η σελίδα εγκυρότητας της ΜΔΕ.	1
Σελίδα δομής	Η παρούσα σελίδα επεξήγησης της δομής της ΜΔΕ.	1
Περίληψη/ Abstract (Ελληνικά/Αγγλικά)	Η περίληψη σε Ελληνικά και Αγγλικά της ΜΔΕ.	2
Ευχαριστίες	Οι ευχαριστίες της ΜΔΕ.	1
Περιεχόμενα	Τα περιεχόμενα της ΜΔΕ, κλικάρισμα μεταφέρει στην ανάλογη ενότητα.	2
Κατάλογος εικόνων	Ο κατάλογος εικόνων της ΜΔΕ, κλικάρισμα μεταφέρει στην ανάλογη ενότητα.	1
Πρόλογος	Ο πρόλογος της ΜΔΕ.	1
Κεφάλαιο 1: Εισαγωγή	Ο σκοπός της εργασίας και η διάρθρωση των κεφαλαίων που ακολουθούν.	2
Κεφάλαιο 2: Θεωρία γράφων	Η θεωρία των γράφων, δηλαδή τύποι, αποτύπωση, εφαρμογές και προβλήματα τους.	8
Κεφάλαιο 3: Μηχανική μάθηση	Η θεωρία της μηχανικής μάθησης, δηλαδή μορφές, μοντέλα, εφαρμογές και περιορισμοί της.	11
Κεφάλαιο 4: Μεθοδολογία πειραματικής μελέτης	Η μεθοδολογία του πειράματος, δηλαδή το σύνολο δεδομένων MovieLens100K, η ανάλυση του, η με κώδικα πάιθον μετατροπή του σε μη-κατευθυνόμενο γράφο και η χρήση του σε μοντέλο μηχανικής μάθησης τύπου GNN με μετρικές ‘loss’ και ‘AUC’.	17
Κεφάλαιο 5: Αποτελέσματα πειραματικής μελέτης	Τα αποτελέσματα του πειράματος, δηλαδή με παραμέτρους 5 και 50 epochs οι τιμές ‘loss’ και ‘AUC’ και η ερμηνεία τους.	6
Κεφάλαιο 6: Συμπεράσματα	Τα συμπεράσματα και μελλοντικές βελτιώσεις.	1
Πίνακας ορολογίας	Ο πίνακας ορολογίας της ΜΔΕ.	1
Συντμήσεις-Αρκτικόλεξα-Ακρωνύμια	Οι συντμήσεις-αρκτικόλεξα-ακρωνύμια της ΜΔΕ.	1
Βιβλιογραφικές αναφορές	Τα βιβλία, εργασίες και web links που χρησιμοποιήθηκαν στην ΜΔΕ.	4
	ΣΥΝΟΛΟ	**62**

ΠΕΡΙΛΗΨΗ

Στη σύγχρονη εποχή μας τα συστήματα συστάσεων βασίζονται σε μεγάλο βαθμό στη μηχανική μάθηση για να μαθαίνουν αυτόματα και να βελτιώνονται με την πάροδο του χρόνου με βάση τα δεδομένα και τα σχόλια των χρηστών. Πολλοί ιστότοποι και εφαρμογές όπως κοινωνικά δίκτυα, υπηρεσίες περιεχομένου, πλατφόρμες ηλεκτρονικού εμπορίου και άλλα, χρησιμοποιούν αλγόριθμους συστάσεων για να βοηθήσουν τους χρήστες να βρουν νέα αγαθά, υπηρεσίες ή πληροφορίες που μπορεί να τους ενδιαφέρουν. Προκειμένου να μοντελοποιήσουν τα γούστα και τα ενδιαφέροντα ενός χρήστη τα συστήματα συστάσεων χρησιμοποιούν αλγόριθμους μηχανικής μάθησης για την αξιολόγηση τεράστιου όγκου δεδομένων χρήστη, συμπεριλαμβανομένων αξιολογήσεων, ερωτημάτων αναζήτησης και προηγούμενων αγορών. Στη συνέχεια, με τη χρήση αυτού του μοντέλου παράγονται εξατομικευμένες συστάσεις με βάση τις απαιτήσεις και τα γούστα κάθε χρήστη.

Στην παρούσα μεταπτυχιακή διπλωματική εργασία έχουμε σκοπό να μελετήσουμε τη θεωρία των γράφων δηλαδή τους τύπους γράφων, τον τρόπο που αποτυπώνονται, που εφαρμόζονται και τα προβλήματα κατά την εφαρμογή τους. Επίσης να εξετάσουμε στη μηχανική μάθηση τις μορφές της, τα μοντέλα της, που εφαρμόζεται και τους περιορισμούς της. Μετά να προχωρήσουμε σε ένα πρακτικό πείραμα. Για το πείραμα μας μεθοδολογία μας ήταν να πάρουμε το σύνολο δεδομένων MovieLens 100K, το εξερευνήσαμε βγάζοντας χρήσιμα συμπεράσματα ανάλυσης δεδομένων σε γράφημα, το φέραμε στην κατάλληλη μορφή ετερογενή μη-κατευθυνόμενου γράφου τον οποίο τελικά χρησιμοποιήσαμε για να εκπαιδεύσουμε ένα σύστημα συστάσεων αλγορίθμου τύπου γράφου νευρωνικού δικτύου. Τελικά κάναμε δύο πειράματα αλλάζοντας τα κριτήρια εκπαίδευσης και μετρήσαμε την απόδοση του αλγορίθμου σε κάθε περίπτωση με τις μετρικές της απώλειας δυαδικής διασταυρούμενης εντροπίας με logits (loss) και της περιοχής κάτω από τη καμπύλη ROC (AUC) τις οποίες οπτικοποιήσαμε σε γραφήματα για την κάθε περίπτωση ώστε να δούμε ευκολότερα τα αποτελέσματα. Τελικά συμπεράναμε και στις δύο περιπτώσεις πως το σύστημα μας απέδωσε “εξαιρετικά”.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Δεδομένα γράφων, μηχανική μάθηση, συστήματα συστάσεων, νευρωνικά δίκτυα, πάιθον.

ABSTRACT

In our modern age recommender systems rely heavily on machine learning to automatically learn and improve over time based on user data and feedback. Many websites and applications such as social networks, content services, e-commerce platforms and others use recommendation algorithms to help users find new goods, services or information that may be of interest to them. In order to model a user's tastes and interests, recommender systems use machine learning algorithms to evaluate vast amounts of user data, including ratings, search queries and past purchases. Then, using this model, personalized recommendations are generated based on each user's requirements and tastes.

In this master's thesis we aim to study the theory of graphs, meaning the types of graphs, the way they are visualized, applied and the problems during their application. Also we examine in machine learning its forms, its models, its applications and its limitations. Afterwards we proceeded to a practical experiment. For our experiment our methodology was to take the MovieLens 100K dataset, explore it by making useful graph data analysis inferences, bring it into the appropriate heterogeneous bidirectional graph format which we eventually used to train a graph neural network algorithm recommendation system. Finally we did two experiments changing the training criteria and measured the performance of the algorithm in each case with the metrics binary cross entropy loss with logits (loss) and area under the ROC curve (AUC) which we visualized in graphs for each case to make it easier to see the results. In the end we concluded in both cases that our system performed "outstanding".

SUBJECT AREA: Machine learning

KEYWORDS: Graph data, machine learning, recommendation systems, neural networks, python.

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη συμβολή στην παρούσα μεταπτυχιακή διπλωματική εργασία θα ήθελα να ευχαριστήσω την επιβλέπουσα Αναπληρώτρια Καθηγήτρια κυρία Χαλκίδη Μαρία για την ανταπόκριση της στις απορίες μου και τη συνεργασία της από την αρχή έως την ολοκλήρωση αυτής της εργασίας. Αντίστοιχα θα ήθελα να ευχαριστήσω συνολικά το προσωπικό του Πανεπιστημίου Πειραιώς για την ευκαιρία που μου έδωσαν και τις γνώσεις που απέκτησα. Ευχαριστώ τους συμφοιτητές και πλεόν φίλους του τμήματος για όσες εμπειρίες αποκομίσαμε και τη φοίτηση που είχαμε. Κυρίως όμως ευχαριστώ τον πατέρα μου Δημήτρη, τη μητέρα μου Ζαφειρία και την αδερφή μου Ζωή για τη συνολική στήριξη και εμπιστοσύνη που μου έχουν δείξει καθ' όλη τη διάρκεια της ζωής μου και έτσι μου δίνουν σταθερά δύναμη.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΠΡΟΛΟΓΟΣ

Ως φοιτητής του Πανεπιστημίου Πειραιώς, της Σχολής Τεχνολογιών Πληροφορικής και Επικοινωνιών, του τμήματος Ψηφιακών Συστημάτων, του Π.Μ.Σ. Πληροφοριακά Συστήματα και Υπηρεσίες, με ειδίκευση Προηγμένα Πληροφοριακά Συστήματα, κάνω αυτή τη μεταπτυχιακή διπλωματική εργασία με θέμα ‘Εφαρμογές τεχνικών μηχανικής μάθησης σε δεδομένα γράφων’ με σκοπό την αποφοίτηση μου. Διάλεξα το συγκεκριμένο θέμα καθώς γενικά με την εξέλιξη της τεχνολογίας στην πάροδο του χρόνου και ειδικά το 2024 τα συστήματα συστάσεων παρουσιάζουν μεγάλο ενδιαφέρον λόγω της όλο και πιο εκτεταμένης χρήσης της μηχανικής μάθησης σε διάφορους καθημερινούς τομείς της ζωής μας.

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Σκοπός της διπλωματικής

Οι αλγόριθμοι στα συστήματα συστάσεων κάνουν προσαρμοσμένες προτάσεις για προϊόντα με βάση αυτό που κάθε χρήστης θεωρεί πιο σχετικό. Οι χρήστες έχουν πλέον πληθώρα επιλογών λόγω της ταχείας επέκτασης των διαδικτυακών πληροφοριών. Εξαιτίας αυτού οι πλατφόρμες ιστού πρέπει να παρέχουν εξατομικευμένες προτάσεις προϊόντων σε κάθε χρήστη, προκειμένου να τονώσουν την ικανοποίηση και την αφοσίωση των χρηστών.

Με τις παραδοσιακές τεχνικές προτάσεων έχουμε κάποια προβλήματα, εάν ο χρήστης δεν έχει βαθμολογήσει ήδη κάποια προτίμηση του πώς θα του γίνει πρόταση; Ή εάν επιλέξουμε προτάσεις μόνο βάση παρόμοιων προιόντων στο τέλος ο κύκλος προτάσεων θα περιοριστεί πολύ. Λύση σε αυτά τα προβλήματα έρχονται να μας δώσουν τα συστήματα συστάσεων βασισμένα σε γράφους. Ένα τέτοιο σύστημα αποθηκεύει αξιολογημένα δεδομένα του περιεχομένου του χρήστη μέσα στη δομή του γράφου, σε συνδυασμό με αλγόριθμους γράφων και διάφορες τεχνικές συστάσεων. Σε σύγκριση με τα προυπάρχοντα σύστηματα συστάσεων, το σύστημα συστάσεων που βασίζεται σε γράφο έχει δύο κύρια πλεονεκτήματα, τα οποία είναι η επεκτασιμότητα και η ποικιλομορφία της μοντελοποίησης σχέσεων. [1]

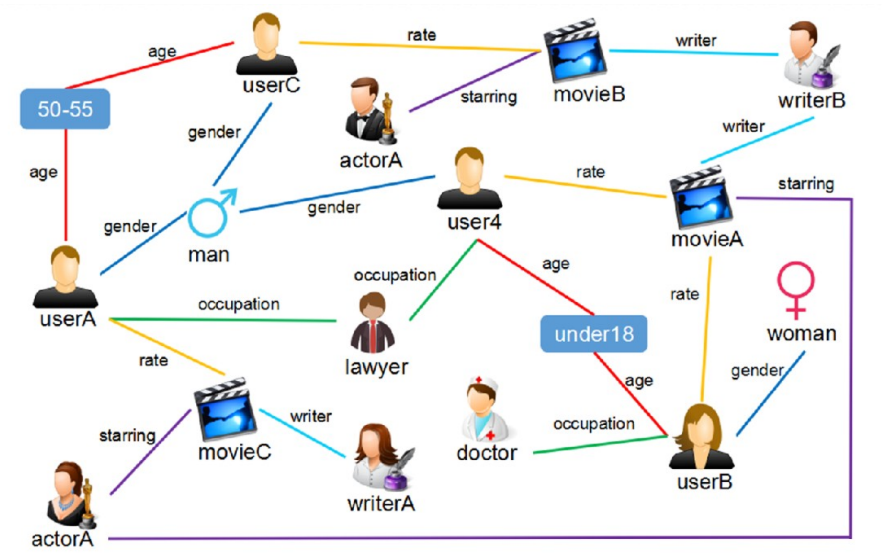
Τα συστήματα συστάσεων που βασίζονται στη μηχανική μάθηση είναι ισχυρές μηχανές που χρησιμοποιούν αλγόριθμους μηχανικής μάθησης για να τμηματοποιούν τους πελάτες με βάση δεδομένα χρήστη και μοτίβα συμπεριφοράς όπως για παράδειγμα το ιστορικό αγορών και περιήγησης, επισημάνσεις "μου αρέσει" ή κριτικές και να τους στοχεύουν με εξατομικευμένες προτάσεις προϊόντων ή περιεχομένου. [2]

Στην παρούσα μεταπτυχιακή διπλωματική εργασία πρώτα ορίσαμε το πρόβλημα, δηλαδή πώς δημιουργούμε και παρακολουθούμε ένα πετυχημένο σύστημα συστάσεων. Κατόπιν επιλέξαμε το MovieLens 100K που είναι ένα γνωστό σύνολο δεδομένων που περιέχει χρήστες, ταινίες και βαθμολογίες για το οποίο ορίσαμε τις σχέσεις μεταξύ των δεδομένων αυτών. Κατόπιν γράψαμε κώδικα python ενός συστήματος συστάσεων μηχανικής μάθησης που αξιοποιεί το σύνολο δεδομένων. Συγκεκριμένα πρώτα κατασκευάσαμε έναν ετερογενή μη-κατευθυνόμενο γράφο για πέρασμα πληροφοριών, τον οποίο αργότερα αξιοποιήσαμε στη μηχανική μάθηση εκπαιδεύοντας ένα μοντέλο γράφου νευρωνικού δικτύου για συστάσεις τύπου συνεργατικού φιλτραρίσματος. Για να αξιολογήσουμε το σύστημα μας χρησιμοποιήσαμε τις κατάλληλες μετρικές τύπου ‘loss’ και ‘AUC’ και είδαμε αλλάζοντας κριτήρια εκπαίδευσης πως ανταποκρίνεται το σύστημα μας. Αυτά θα τα δούμε αναλυτικότερα στα επόμενα κεφάλαια αφού πρώτα εξετάσουμε τις θεωρίες σχετικές με το θέμα που επιλέξαμε. Καθώς τα συστήματα συστάσεων είναι πλεόν ένα εκτεταμένο κομμάτι της ηλεκτρονικής πραγματικότητας γύρω μας αυτό τα κάνει πολύ ενδιαφέροντα για να τα εξετάσουμε, ενώ η πρόκληση είναι πώς θα κάνουμε πετυχημένες και γρήγορες προτάσεις παρά αντίστοιχα την έλλειψη ορισμένων πληροφοριών ή τον όγκο των δεδομένων.

1.2 Διάρθρωση της διπλωματικής

Στην παρούσα μεταπτυχιακή διπλωματική εργασία μετά από το εισαγωγικό "ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ", ακολουθεί η εξής διάρθρωση:

- *ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΙΑ ΓΡΑΦΩΝ* στο οποίο επεξηγούμε τη θεωρία των γράφων, δηλαδή τους βασικούς τύπους τους, την αποτύπωση τους, την εφαρμογή τους, όπως επίσης και τα προβλήματα τους.
- *ΚΕΦΑΛΑΙΟ 3: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ* στο οποίο επεξηγούμε τη θεωρία της μηχανικής μάθησης, δηλαδή τις βασικές μορφές της, τα μοντέλα της, τις εφαρμογές της, όπως επίσης και τους περιορισμούς της.
- *ΚΕΦΑΛΑΙΟ 4: ΜΕΘΟΔΟΛΟΓΙΑ ΠΕΙΡΑΜΑΤΙΚΗΣ ΜΕΛΕΤΗΣ* στο οποίο επεξηγούμε τη μεθοδολογία του πειράματος μας, δηλαδή τι είναι το σύνολο δεδομένων μας, πως το αναλύσαμε, πως το μετατρέψαμε σε ετερογενή μη-κατευθυνόμενο γράφο, πώς το χρησιμοποιήσουμε σε αλγόριθμο μηχανικής μάθησης τύπου νευρωνικού δικτύου γράφου με δύο διαφορετικά κριτήρια εκπαίδευσης 'loss' και 'AUC'.
- *ΚΕΦΑΛΑΙΟ 5: ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΕΙΡΑΜΑΤΙΚΗΣ ΜΕΛΕΤΗΣ* στο οποίο ορίζουμε τις παραμέτρους και τις μετρικές και τελικά βλέπουμε και ερμηνεύουμε τα αποτελέσματα.
- *ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ* όπου τελικά καταλήγουμε στα συμπεράσματα αυτής της εργασίας.



Εικόνα 1: Ετερογενής γράφος συστήματος συστάσεων. [55]

Στην 'Εικόνα 1' βλέπουμε ένα παράδειγμα πως αποκτά γνώση ένας ετερογενής γράφος συστήματος συστάσεων, με τους χρήστες να έχουν διάφορες διακριτές προτιμήσεις που τους συνδέουν με τα χαρακτηριστικά των ταινιών και τελικά μεταξύ τους ώστε να γίνει η επιλογή από το σύστημα πρότασης μέσω μηχανικής μάθησης.

ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΙΑ ΓΡΑΦΩΝ

2.1 Εισαγωγή στη θεωρία γράφων

Η μελέτη των γράφων, οι οποίοι είναι μαθηματικές δομές που χρησιμοποιούνται για την αναπαράσταση αλληλεπιδράσεων ανά ζεύγη μεταξύ αντικειμένων, είναι γνωστή ως θεωρία των γράφων στα μαθηματικά. Με αυτή την έννοια, ένας γράφος (graph) αποτελείται από κορυφές (vertices) γνωστές και ως κόμβους (nodes), που συνδέονται με ακμές (edges).

2.2 Τύποι γράφων

Μπορούμε να χωρίσουμε τους γράφους σε δύο κύριους τύπους

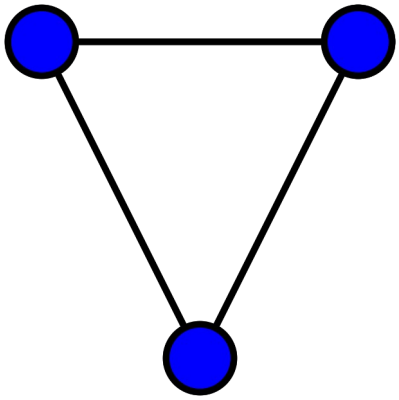
- Τους *μη-κατευθυνόμενους γράφους*, οι μη-κατευθυνόμενοι γράφοι χαρακτηρίζονται από ακμές που συνδέουν δύο κορυφές συμμετρικά.
- Τους *κατευθυνόμενους γράφους*, οι κατευθυνόμενοι γράφοι χαρακτηρίζονται από ακμές που συνδέουν δύο κορυφές ασύμμετρα.

2.2.1 Μη-κατευθυνόμενοι γράφοι

Σε μια συγκεκριμένη αλλά ευρέως χρησιμοποιούμενη ερμηνεία του όρου [3] γράφος G είναι ένα διατεταγμένο ζεύγος G=(V,E) για το οποίο:

- *V* είναι μια συλλογή από κορυφές, γνωστές και ως κόμβοι.
- *E* είναι μη-ταξινομημένα ζεύγη κορυφών ή μια συλλογή ακμών όπου κάθε ακμή συνδέεται με δύο διαφορετικές κορυφές.

Αυτό το είδος είναι ακριβώς γνωστό ως μη-κατευθυνόμενος απλός γράφος.



Εικόνα 2: Μη-κατευθυνόμενος γράφος [56]

Στην ‘Εικόνα 2’ βλέπουμε ένα μη-κατευθυνόμενο γράφο με τρείς κορυφές (μπλέ κύκλοι) και τρείς ακμές (μαύρες γραμμές) και αντίστοιχα με μία γενικότερη έννοια του όρου που επιτρέπει πολλαπλές άκμές [4], γράφος είναι ένα διατεταγμένο τριπλό $G=(V,E,\varphi)$ για τον οποίο:

- *V* είναι μια συλλογή από κορυφές, γνωστές και ως κόμβοι.
- *E* είναι μη-ταξινομημένα ζεύγη κορυφών ή μια συλλογή ακμών.
- *φ:E* είναι μια συνάρτηση πρόσπτωσης που συσχετίζει κάθε ακμή με ένα μη-ταξινομημένο ζεύγος κορυφών, δηλαδή κάθε ακμή έχει δύο διαφορετικές κορυφές που σχετίζονται με αυτήν.

Αυτό το είδος αντικειμένου είναι γνωστό ώς μη-κατευθυνόμενος πολυγράφος.

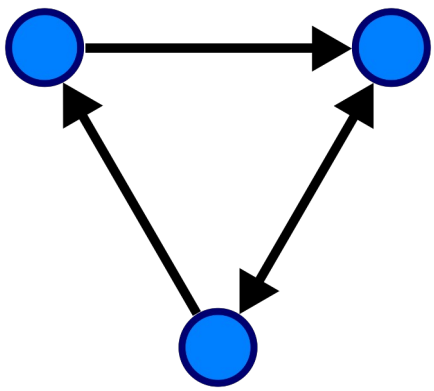
Μια ακμή που συνδέει μια κορυφή με τον εαυτό της ονομάζεται βρόχος (loop). Οι βρόχοι δεν επιτρέπονται στους γράφους σύμφωνα με τους δύο προηγούμενους ορισμούς μας και για να μπορέσουμε να χρησιμοποιήσουμε βρόχους έχουμε τους μη-κατευθυνόμενους απλούς γράφους επιτρέποντες βρόχους και τους μη-κατευθυνόμενους πολυγράφους επιτρέποντες βρόχους.

2.2.2 Κατευθυνόμενοι γράφοι

Σε ένα κατευθυνόμενο γράφο κάθε ακμή έχει έναν συγκεκριμένο προσανατολισμό. Ένας κατευθυνόμενο γράφος σε μια στενή αλλά ευρέως χρησιμοποιούμενη ερμηνεία [5] είναι ένα διατεταγμένο ζεύγος $G=(V,E)$ για τον οποίο:

- *V* είναι μια συλλογή από κορυφές, γνωστές και ως κόμβοι.
- *E* είναι ένα διατεταγμένο ζεύγος κορυφών που συνθέτουν ένα σύνολο ακμών, γνωστές επίσης ως κατευθυνόμενες ακμές, δηλαδή κάθε ακμή συνδέεται με δύο διαφορετικές κορυφές.

Αυτό το είδος είναι ακριβώς γνωστό ως κατευθυνόμενος απλός γράφος.



Εικόνα 3: Κατευθυνόμενος γράφος.
[57]

Στην ‘Εικόνα 3’ βλέπουμε ένα κατευθυνόμενο γράφο με τρείς κορυφές (μπλέ κύκλοι) και τρείς ακμές (μαύρες γραμμές) ενώ βέλη ορίζουν την κατεύθυνση και αντίστοιχα με μια γενικότερη έννοια του όρου που επιτρέπει πολλαπλές ακμές [6], κατευθυνόμενος γράφος είναι ένα διατεταγμένο τριπλό G=(V,E,ϕ), για το οποίο:

- *V* είναι μια συλλογή από κορυφές, γνωστές και ως κόμβοι.
- *E* είναι ένα διατεταγμένο ζεύγος κορυφών που συνθέτουν ένα σύνολο ακμών, γνωστές επίσης ως κατευθυνόμενες ακμές.
- *φ:E* είναι μια συνάρτηση πρόσπτωσης που συσχετίζει κάθε ακμή με ένα ταξινομημένο ζεύγος κορυφών, δηλαδή κάθε ακμή έχει δύο διαφορετικές κορυφές που σχετίζονται με αυτήν.

Μια ακμή που συνδέει μια κορυφή με τον εαυτό της ονομάζεται όπως και στην προηγούμενη περίπτωση βρόχος. Οι βρόχοι δεν επιτρέπονται στους γράφους σύμφωνα με τους δύο προηγούμενους ορισμούς μας και για να μπορέσουμε να χρησιμοποιήσουμε βρόχους έχουμε τους κατευθυνόμενους απλούς γράφους επιτρέποντες βρόχους και τους κατευθυνόμενους πολυγράφους επιτρέποντες βρόχους.

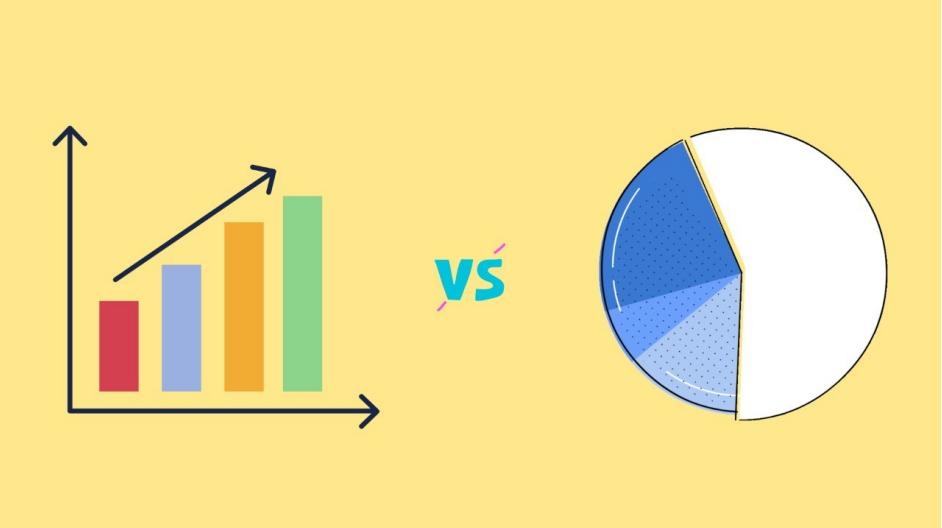
2.3 Αποτύπωση γράφων

Δεδομένου ότι ένας γράφος είναι μια αποτύπωση φυσικών σχέσεων, δεν μπορεί να συσχετιστεί με μια μόνο συγκεκριμένη αναπαράσταση. Ο βαθμός ευκολίας που προσφέρει η κάθε μια συγκεκριμένη αναπαράσταση για μια δεδομένη εφαρμογή καθορίζει τον τρόπο με τον οποίο θα αναπαρασταθεί τελικά. Βέβαια μπορούμε να πούμε πως οι δύο πιο δημοφιλείς τύποι αναπαραστάσεων είναι η πινακοειδής (tabular) αναπαράσταση και η οπτική αναπαράσταση.

- Στην *πινακοειδή (tabular) αναπαράσταση* οι σειρές ενός πίνακα μεταφέρουν πληροφορίες σχετικά με τις αλληλεπιδράσεις μεταξύ των κορυφών μέσα στο γράφο.
- Στην *οπτική αναπαράσταση* οι κορυφές απεικονίζονται και συνδέονται με ακμές.

2.3.1 Πινακοειδής αναπαράσταση

Οι υπολογιστικές εφαρμογές επωφελούνται πολύ από την πινακοειδή (tabular) αναπαράσταση. Ένας υπολογιστής μπορεί να αποθηκεύσει γράφους με διάφορους τρόπους. Η επιλεγμένη δομή δεδομένων καθορίζεται από τη δομή του γράφου καθώς και από την τεχνική χειρισμού του γράφου. Αν και θεωρητικά ξεχωρίζουν μεταξύ τους οι δομές λίστας γειτνίασης και μήτρας γειτνίασης τελικά στις πρακτικές εφαρμογές ο συνδυασμός και των δύο είναι συχνά η βέλτιστη δομή. Επειδή οι δομές λίστας απαιτούν λιγότερη μνήμη επιλέγονται συχνά για αραιούς γράφους. Αντίθετα οι δομές μήτρας μπορούν να χρησιμοποιήσουν σημαντική ποσότητα μνήμης αλλά σε ορισμένες περιπτώσεις προσφέρουν ταχύτερη πρόσβαση. Η εφαρμογή αραιών δομών μήτρας που να είναι αποτελεσματικές σε σύγχρονες παράλληλες υπολογιστικές αρχιτεκτονικές αποτελούν αντικείμενο τωρινών ερευνών. [7]



Εικόνα 4: Δημοφιλείς πινακοειδείς αναπαραστάσεις. [58]

Στην ‘Εικόνα 4’ βλέπουμε δημοφιλείς πινακοειδείς (tabular) αναπαραστάσεις όπως πίτα και μπάρες οι οποίες συγκεντρωτικά δείχνουν πληροφορίες δεδομένων, θα χρησιμοποιήσουμε αντίστοιχο τύπο αργότερα στο πείραμα μας για να βγάλουμε συμπεράσματα σχετικά με το σύνολο δεδομένων μας.

2.3.2 Οπτική αναπαράσταση

Συνήθως η οπτική αναπαράσταση ενός γράφου αποτελείται από τη σχεδίαση ενός σημείου ή κύκλου για κάθε κορυφή καθώς και μιας γραμμής που συνδέει οποιεσδήποτε δύο κορυφές που συνδέονται με μια ακμή. Σχεδιάζεται ένα βέλος για να δείξει την κατεύθυνση του γράφου εάν είναι κατευθυνόμενος. Το βάρος προστίθεται στο βέλος εάν ο γράφος έχει βάρος. Υπάρχουν διάφορες μέθοδοι οργάνωσης ενός σχεδίου γράφου, επομένως είναι σημαντικό να το διακρίνουμε από τον ίδιο τον γράφο εννοώντας την αφηρημένη μη οπτική δομή. Δεν έχει σημασία πώς ακριβώς είναι τα πράγματα αλλά αυτό που έχει σημασία είναι ποιες κορυφές σχετίζονται με ποιες και πόσες ακμές. Στην πραγματική ζωή είναι συχνά δύσκολο να προσδιοριστεί εάν δύο σχέδια απεικονίζουν τον ίδιο γράφο. Η σχεδίαση γράφων είναι ένας τομέας των μαθηματικών και της επιστήμης των υπολογιστών που συνδυάζει μεθόδους από τη θεωρία γεωμετρικών γράφων και την οπτικοποίηση πληροφοριών για την παραγωγή δισδιάστατων απεικονίσεων γράφων που προκύπτουν από εφαρμογές όπως η ανάλυση κοινωνικών δικτύων, η χαρτογραφία, η γλωσσολογία και η βιοπληροφορική. [8]

Εικόνα 5: Απεικόνιση οπτικής αναπαράστασης γράφου. [59]

Στην ‘Εικόνα 5’ βλέπουμε πως απεικονίζεται η οπτική αναπαράσταση ενός γράφου, με κάθε κορυφή και ακμή να συνδέονται μεταξύ τους και να δημιουργούν ένα πολύπλοκο σχήμα στο οποίο ξεχωρίζουν πυκνότερα οι σχετικότερες μεταξύ τους συστάδες δεδομένων.

2.4 Εφαρμογές θεωρίας και τεχνικές μηχανικής μάθησης γράφων

Εδώ θα εξετάσουμε τις εφαρμογές των θεωρίων γράφων όπως και πρακτικά τις τεχνικές μηχανικής μάθησης που εφαρμόζονται σε δεδομένα γράφων.

2.4.1 Εφαρμογές θεωρίας γράφων

Διάφορα είδη σχέσεων και διεργασιών σε βιολογικά, κοινωνικά, πληροφοριακά και φυσικά συστήματα μπορούν να μοντελοποιηθούν χρησιμοποιώντας γράφους. [9] Οι γράφοι είναι ένα χρήσιμο εργαλείο για την αναπαράσταση πολλών ζητημάτων του πραγματικού κόσμου.

- Στη *βιολογία*, για παράδειγμα όταν μια κορυφή αντιπροσωπεύει μια τοποθεσία όπου υπάρχει ένα συγκεκριμένο είδος και οι ακμές αντικατοπτρίζουν τα πρότυπα μετανάστευσης ή την κίνηση μεταξύ των περιοχών, έτσι η θεωρία γράφων μπορεί να είναι χρήσιμη στις προσπάθειες διατήρησης ειδών.
- Στην *κοινωνιολογία*, για παράδειγμα η θεωρία γράφων εφαρμόζεται επίσης συχνά για τη διερεύνηση της μετάδοσης φημών ιδιαίτερα μέσω της χρήσης λογισμικού ανάλυσης κοινωνικών δικτύων. Υπάρχουν πολλές ποικιλίες γράφων που εμπίπτουν στην κατηγορία των κοινωνικών δικτύων, έτσι οι γράφοι φιλίας και γνωριμίας δείχνουν πόσο γνωστά είναι τα άτομα μεταξύ τους.

- Στην *επιστήμη των υπολογιστών* φυσιολογικές και οι μη φυσιολογικές συνδεδεμένες δομές είναι γράφοι που χρησιμοποιούνται για την απεικόνιση δικτύων επικοινωνίας, δομών δεδομένων, υπολογιστικών συσκευών, υπολογιστικής ροής και τα λοιπά.
- Στη *φυσική και τη χημεία* η θεωρία γράφων χρησιμοποιείται επίσης για την εξέταση μορίων. Για παράδειγμα συλλέγοντας στατιστικά στοιχεία για τα θεωρητικά χαρακτηριστικά γράφων που σχετίζονται με την ατομική τοπολογία, οι επιστήμονες μπορούν να διερευνήσουν ποσοτικά την τρισδιάστατη δομή πολύπλοκων προσομοιωμένων ατομικών δομών.
- Και τελικά στα *συστήματα προτάσεων*, δηλαδή εφαρμογές που κάνουν συστάσεις, όπως στην περίπτωση του πειράματος μας που θα μελετήσουμε αργότερα, ένα μοντέλο γράφου είναι μια εξαιρετική επιλογή. Μπορούν να αποθηκευτούν συσχετισμοί γράφων μεταξύ διαφορετικών τύπων πληροφοριών, όπως φίλοι, ιστορικό αγορών ή ενδιαφέροντα πελατών. Μια πολύ προσιτή βάση δεδομένων γράφων μπορεί να χρησιμοποιηθεί για να προτείνει πράγματα σε έναν χρήστη με βάση αυτά που έχουν επιλέξει άλλοι χρήστες που έχουν παρόμοια ενδιαφέροντα ή ιστορικά επιλογών.

2.4.2 Τεχνικές μηχανικής μάθησης σε δεδομένα γράφων

Τα μοντέλα μηχανικής μάθησης σε δεδομένα γράφων χρησιμοποιούνται εκτεταμένα λόγω των δωρεάν λύσεων ανοιχτού κώδικα που έχουν. Χρησιμοποιώντας μηχανική μάθηση με νευρωνικά δίκτυα γράφων μπορούμε να φτιάξουμε συστήματα συστάσεων. Τα νευρωνικά δίκτυα γράφων επικεντρώνονται σε τρόπους μάθησης που χρησιμοποιούν αυτές τις πληροφορίες για να βελτιώσουν την απόδοση για εργασίες όπως η πρόβλεψη αξιολόγησης. Χρησιμοποιώντας πληροφορίες γειτονίας τα νευρωνικά δίκτυα γράφων στοχεύουν να μάθουν αναπαραστάσεις οντοτήτων ή ενσωματώσεις κόμβων (node embeddings) πιο αποτελεσματικά. Άρα γενικότερα τα νευρωνικά δίκτυα γράφων που εκτελούν μηχανική μάθηση επιχειρούν:

- *Διάδοση μηνυμάτων* για διάδοση πληροφοριών
- *Συνάθροιση και ενημέρωση* που είναι η λειτουργία για την επεξεργασία των πληροφοριών που λαμβάνονται.
- *Επανάληψη* η οποία καθορίζει την έκταση του σήματος.

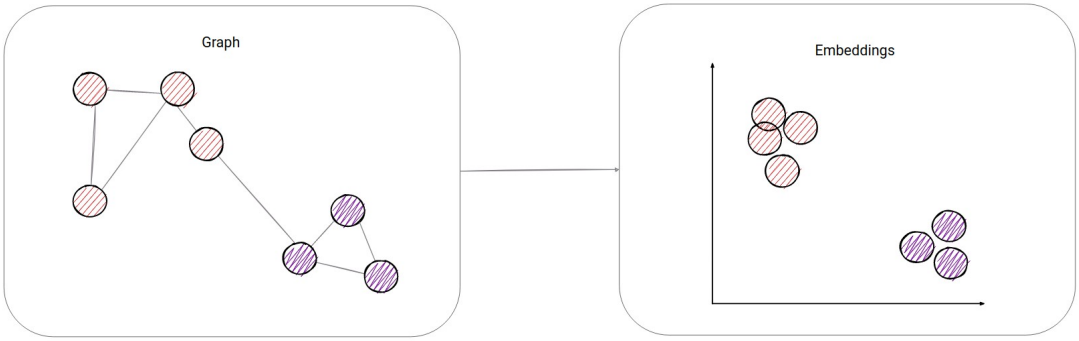
Πιο συγκεκριμένα στα συστήματα συστάσεων τύπου νευρωνικών δικτύων γράφων όπου προσπαθούμε να βρούμε μια επιθυμητή λίστα αντικειμένων για τους χρήστες έχουμε τρείς βασικούς τύπους ανάλογα τι προτιμά ειδικότερα ο χρήστης:

- Σχετικά με *προτιμήσεις του χρήστη*.
- Σχετικά με *χαρακτηριστικά αντικειμένου*.
- Ή σχετικά με *χρήστη-προηγούμενες αλληλεπιδράσεις αντικειμένων.*

Υπάρχουν διάφορα διαθέσιμα μοντέλα που βασίζονται σε νευρωνικά δίκτυα γράφων για την αντιμετώπιση του προβλήματος της σύστασης, ανάλογα με τον τρόπο με τον οποίο μοντελοποιούμε τα διαθέσιμα δεδομένα ως γράφο. Αυτά τα μοντέλα μπορούν να χωριστούν γενικά σε τρεις κατηγορίες.

- *Αλληλεπίδρασης χρήστη-στοιχείου*, τα δεδομένα μοντελοποιούνται ως διμερής γράφος όπως το παράδειγμα που θα δούμε σε πείραμα αργότερα.
- *Κοινωνικών δικτύων*, δύο είδη γράφων μπορούν να επιλεγούν για τη μοντελοποίηση των δεδομένων, είτε ένας ετερογενής γράφος με χρήστες και αντικείμενα που συνδέονται από άκρες χρήστη-χρήστη και στοιχείου χρήστη, ή ένας διμερής γράφος στοιχείου-χρήστη και ένας ξεχωριστός γράφος κοινωνικού δικτύου που αποτελείται από χρήστες και άκρες που εκφράζουν κάποια σχέση μεταξύ τους.
- *Γράφο γνώσης*, αυτή η ομάδα μοντέλων εστιάζει στη βελτίωση της αναπαράστασης του αντικειμένου, η οποία οδηγεί σε βελτιωμένες προτάσεις αντικειμένων με βάση τις προηγούμενες αλληλεπιδράσεις του χρήστη με παρόμοια αντικείμενα.

Μια ενσωμάτωση κόμβου (node embedding) η οποία είναι μια αναπαράσταση των κόμβων σε ένα γράφο μπορεί να μαθευτεί από ένα νευρωνικό δίκτυο γράφου. Η κατηγοριοποίηση κόμβων και η πρόβλεψη συνδέσμων είναι μόνο δύο από τις πολλές εφαρμογές για ενσωματώσεις κόμβων (node embeddings). Μπορούμε να σκεφτούμε μια ενσωμάτωση ως μια συνάρτηση που λαμβάνει μια είσοδο (όπως έναν κόμβο σε ένα γράφο) και βγάζει ένα διάνυσμα. Οι ενσωματώσεις κόμβων (node embeddings) είναι ένας τύπος ενσωμάτωσης που αντιστοιχίζει τους κόμβους ενός γράφου σε διανύσματα. [51]



Εικόνα 6: Οπτικοποίηση από κόμβους σε διανύσματα. [60]

Στην ‘Εικόνα 6’ βλέπουμε την οπτικοποίηση της ενσωμάτωσης κόμβου (node embedding) από κόμβους γράφου αριστερά σε διανύσματα γραφήματος δεξιά. Οι ενσωματώσεις κόμβων (node embeddings) προσφέρουν πολλά πλεονεκτήματα:

- *Παραδοσιακές τεχνικές βελτιστοποίησης νευρωνικών δικτύων* μπορούν να χρησιμοποιηθούν για την εκπαίδευση των ενσωματώσεων κόμβων (node embeddings).
- *Οποιοσδήποτε τύπος γράφου* μπορεί να χρησιμοποιηθεί για την εκπαίδευση των ενσωματώσεων κόμβων (node embeddings).

- *Κόμβοι μπορούν να αναπαρασταθούν ταυτόχρονα σε διάφορους γράφους* χρησιμοποιώντας ενσωματώσεις κόμβων (node embeddings).

Αναφορικά οι ενσωματώσεις κόμβων (node embeddings) μπορούν να κατηγοριοποιηθούν σε τρείς βασικούς τύπους:

- Οι *τεχνικές που βασίζονται στην παραγοντοποίηση* οι οποίες παραγοντοποιούν τον προκύπτοντα πίνακα για να πάρουν την ενσωμάτωση.
- Οι *τυχαίες τεχνικές περιπάτου* οι οποίες δειγματοποιούν γειτονιές δικτύου για κόμβους χρησιμοποιώντας μια μεθοδολογία περιπάτου.
- Οι *τεχνικές που βασίζονται σε Βαθιά Νευρωνικά Δίκτυα* χρησιμοποιούνται σε γράφους ως αποτέλεσμα της βαθιάς μάθησης όπως στο παράδειγμα μας που θα δούμε αργότερα.

2.5 Προβλήματα θεωρίας γράφων

Λόγω της πολυεπίπεδης προσέγγισης στη θεωρία των γράφων κατά τη χρήση γράφων θα αντιμετωπίσουμε κάποια χαρακτηριστικά προβλήματα όπως:

- *Απαρίθμιση γράφων*, περιγράφει μια κατηγορία προβλημάτων συνδυαστικής απαρίθμησης στα οποία πρέπει να μετρηθούν μη κατευθυνόμενοι ή κατευθυνόμενοι γράφοι ορισμένων τύπων, συνήθως ως συνάρτηση του αριθμού των κορυφών του γράφου. [10]
- Το πρόβλημα *ισομορφισμού υπογράφων*, είναι μια υπολογιστική εργασία στη θεωρητική επιστήμη των υπολογιστών όπου δύο γράφοι G και H παρέχονται ως είσοδοι και ο στόχος είναι να ανακαλύψουμε εάν ο G περιλαμβάνει υπογράφο που είναι ισόμορφος του H.
- Ο *χρωματισμός γράφου*, όπως αναφέρεται στη θεωρία γράφων είναι μια συγκεκριμένη περίπτωση επισήμανσης γράφων. Περιλαμβάνει την αντιστοίχιση ετικετών που μερικές φορές αναφέρονται ως χρώματα σε κόμβους γράφων, ενώ λαμβάνονται υπόψη συγκεκριμένοι περιορισμοί.
- *Υπαγωγή και ενοποίηση*, οι θεωρίες μοντελοποίησης περιορισμών ασχολούνται με οικογένειες κατευθυνόμενων γράφων που συνδέονται με μερική σειρά, οι γράφοι σε αυτές τις εφαρμογές είναι διατεταγμένοι σύμφωνα με την ιδιαιτερότητά τους, δηλαδή οι πιο περιορισμένοι γράφοι που είναι πιο συγκεκριμένοι εμπίπτουν στην κατηγορία των ευρύτερων γράφων που είναι γενικοί και μεγάλοι.
- *Προβλήματα κάλυψης*, είναι υπολογιστικά προβλήματα στην επιστήμη των υπολογιστών που διερευνούν εάν μια συγκεκριμένη συνδυαστική δομή καλύπτει μια άλλη ή πόσο μεγάλη πρέπει να είναι η δομή για να γίνει αυτό.
- *Προβλήματα αποσύνθεσης*, ορίζονται προβλήματα που αφορούν την κατάτμηση του συνόλου ακμών ενός γράφου με όσες κορυφές χρειάζεται να συνοδεύουν τις άκρες κάθε μισού του διαμερίσματος. Η αποσύνθεση ενός γράφου σε υπογράφους που είναι ισόμορφοι σε ένα σταθερό γράφο είναι συχνά πρόκληση.

ΚΕΦΑΛΑΙΟ 3: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

3.1 Εισαγωγή στη μηχανική μάθηση

Ο κλάδος μελέτης της μηχανικής μάθησης (machine learning ή ML) της τεχνητής νοημοσύνης (artificial intelligence ή AI) εστιάζει στη δημιουργία και ανάλυση στατιστικών αλγορίθμων που μπορούν να γενικεύουν καλά και να εκτελούν εργασίες χωρίς την ανάγκη ρητών οδηγιών. Τα παραγωγικά τεχνητά νευρωνικά δίκτυα (generative artificial neural networks) έχουν πρόσφατα επιδείξει κέρδη στην απόδοση τους σε σχέση με πολλές προηγούμενες μεθόδους. Οι προσεγγίσεις μηχανικής μάθησης έχουν εφαρμοστεί σε μεγάλα μοντέλα γλωσσών, αναγνώριση ομιλίας, φιλτράρισμα email, γεωργία και ιατρική, όπου είναι πολύ δαπανηρό να αναπτυχθούν αλγόριθμοι για την εκτέλεση των απαραίτητων εργασιών. [11]

Οι τεχνικές μαθηματικής βελτιστοποίησης ή μαθηματικού προγραμματισμού δίνουν το μαθηματικό υπόβαθρο της μηχανικής μάθησης. Η εξόρυξη δεδομένων είναι ένα σχετικό ή και παράλληλο πεδίο μελέτης, που εστιάζει στη διερευνητική ανάλυση δεδομένων (exploratory data analysis) μέσω της μη-εποπτευόμενης μάθησης [12] την οποία ειδικότερα θα δούμε αργότερα.

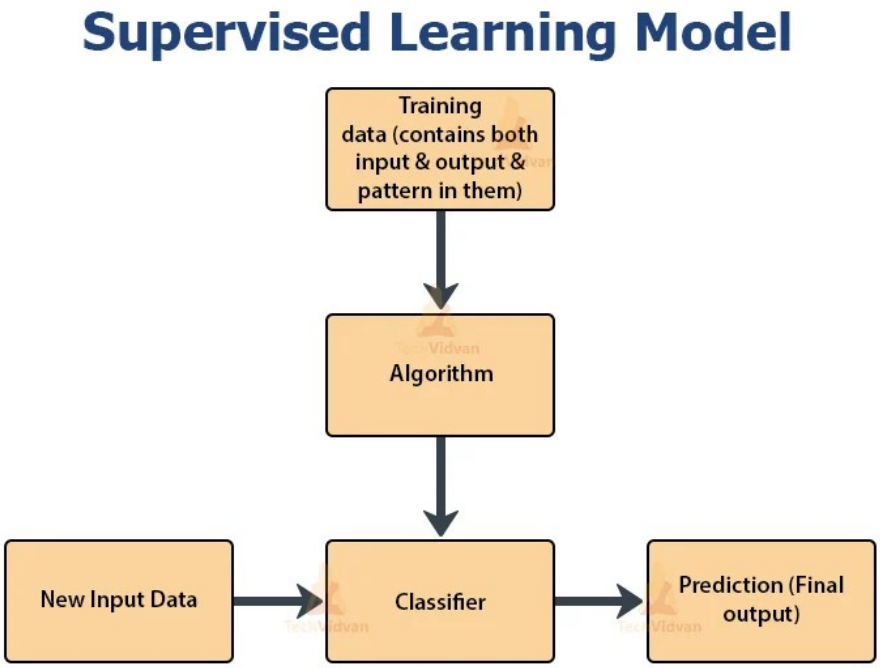
3.2 Μορφές μηχανικής μάθησης

Ανάλογα με τον τύπο του σήματος ή της ανάδρασης στον οποίο έχει πρόσβαση το σύστημα μάθησης, οι τεχνικές μηχανικής μάθησης κατηγοριοποιούνται συνήθως σε τέσσερις μεγάλες ομάδες ανάλογες με τα πρότυπα μάθησης.

- *Εποπτευόμενη μάθηση (supervised learning)*, ο στόχος του υπολογιστή είναι να μάθει έναν γενικό κανόνα που αντιστοιχίζει τις εισόδους σε εξόδους παρέχοντάς του παραδείγματα εισόδων και τα επιδιωκόμενα αποτελέσματά τους τα οποία παρέχονται από έναν ‘δάσκαλο’.
- *Μη-εποπτευόμενη μάθηση (unsupervised learning)*, ο αλγόριθμος μάθησης αφήνεται να καταλάβει τη δομή στην εισαγωγή μόνος του χωρίς ετικέτες (labels). Η μη-εποπτευόμενη μάθηση μπορεί να χρησιμοποιηθεί ως μέσο για την επίτευξη ενός σκοπού για παράδειγμα την εκμάθηση χαρακτηριστικών (features) ή ως στόχος από μόνος του για παράδειγμα την εύρεση κρυφών μοτίβων στα δεδομένα.
- *Ημιεποπτευόμενη μάθηση (semi-supervised learning)*, είναι μια μορφή μηχανικής μάθησης που βρίσκεται ανάμεσα στην εποπτευόμενη μάθηση και τη μη-εποπτευόμενη μάθηση, καθώς αν αναμιχθούν παραδείγματα εκπαίδευσης χωρίς ετικέτα με μερικά με ετικέτα βελτιώνεται η ακρίβεια της μάθησης.
- *Ενισχυτική μάθηση (reinforced learning)*, κατά την αλληλεπίδραση με ένα δυναμικό περιβάλλον, ένα πρόγραμμα υπολογιστή πρέπει να ολοκληρώσει μια συγκεκριμένη εργασία για παράδειγμα ανταγωνισμό σε ένα παιχνίδι, το πρόγραμμα λαμβάνει ανατροφοδότηση παρόμοια με κίνητρα καθώς κινείται στην περιοχή του προβλήματος και προσπαθεί να τα μεγιστοποιήσει.

3.2.1 Εποπτευόμενη μάθηση

Οι αλγόριθμοι εποπτευόμενης μάθησης δημιουργούν ένα μαθηματικό μοντέλο ενός συνόλου δεδομένων που περιέχει τόσο τις εισόδους όσο και τις επιθυμητές εξόδους. [13] Μια συλλογή από παραδείγματα εκπαίδευσης συνθέτουν τα δεδομένα, τα οποία αναφέρονται ως δεδομένα εκπαίδευσης. Κάθε παράδειγμα εκπαίδευσης περιέχει μία ή περισσότερες εισόδους καθώς και το εποπτικό σήμα δηλαδή το επιθυμητό αποτέλεσμα. Τα δεδομένα εκπαίδευσης αντιπροσωπεύονται από έναν πίνακα στο μαθηματικό μοντέλο και κάθε δείγμα εκπαίδευσης αντιπροσωπεύεται από έναν πίνακα ή διάνυσμα που αναφέρεται επίσης ως διάνυσμα χαρακτηριστικών. Μέσω της επαναληπτικής βελτιστοποίησης μιας αντικειμενικής συνάρτησης οι εποπτευόμενοι αλγόριθμοι μάθησης μαθαίνουν μια συνάρτηση που μπορεί να χρησιμοποιηθεί για την πρόβλεψη της εξόδου που σχετίζεται με νέες εισόδους. [14] Για εισόδους που δεν συμπεριλήφθηκαν στο εκπαιδευτικό σύνολο δεδομένων ο αλγόριθμος μπορεί να εκτιμήσει με ακρίβεια την έξοδο με τη χρήση μιας βέλτιστης συνάρτησης. Ένας αλγόριθμος λέγεται ότι έχει μάθει να εκτελεί μια εργασία όταν σταδιακά αυξάνει την ακρίβεια των εξόδων ή των προβλέψεών του.



Εικόνα 7: Εποπτευόμενη μηχανική μάθηση. [61]

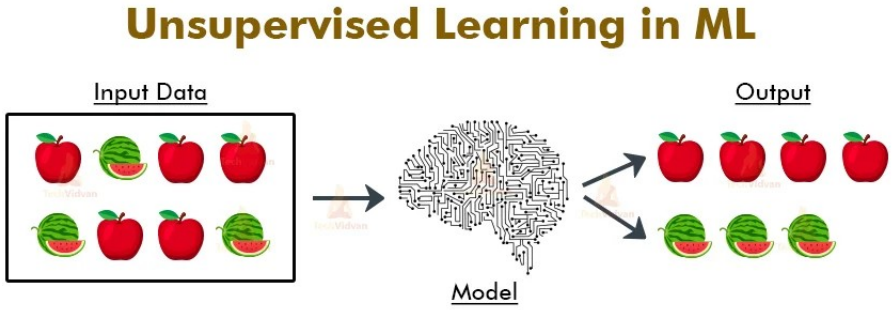
Στην ‘Εικόνα 7’ βλέπουμε το μοντέλο λειτουργίας της εποπτευόμενης μηχανικής μάθησης, η είσοδος των δεδομένων εκπαίδευσης και επιθυμητών αποτελεσμάτων περνά από τον αλγόριθμο που επιχειρεί βελτιστοποίηση, ώστε όταν εισαχθούν τα νέα δεδομένα να βγούν πετυχημένες προβλέψεις.

Μαθηματικά η εποπτευόμενη μάθηση μπορεί να αναπαρασταθεί ως μια γραμμική συνάρτηση για παράδειγμα y=f(x), ενώ αναφορικά μερικοί δημοφιλείς τύποι αλγορίθμων εποπτευόμενης μάθησης είναι:

- *Διαδραστικής μάθησης*, είναι μια ειδική περίπτωση μηχανικής μάθησης στην οποία ένας αλγόριθμος μάθησης μπορεί να ρωτήσει αλληλεπιδραστικά έναν χρήστη (ή κάποια άλλη πηγή πληροφοριών) για να επισημάνει νέα σημεία δεδομένων με τις επιθυμητές εξόδους. [15]
- *Ταξινόμησης*, η ταξινόμηση είναι μια εποπτευόμενη μέθοδος μηχανικής εκμάθησης όπου το μοντέλο προσπαθεί να προβλέψει τη σωστή ετικέτα ενός δεδομένου δεδομένων εισόδου. [16]
- *Παλινδρόμησης*, είναι μια τεχνική για τη διερεύνηση της σχέσης μεταξύ ανεξάρτητων μεταβλητών ή χαρακτηριστικών και μιας εξαρτημένης μεταβλητής ή αποτελέσματος. [17]

3.2.2 Μη-εποπτευόμενη μάθηση

Όταν δίνεται ένα σύνολο δεδομένων που αποτελείται αποκλειστικά από εισόδους, οι αλγόριθμοι μη-εποπτευόμενης μάθησης μπορούν να αναγνωρίσουν τη δομή των δεδομένων συσταδοποιώντας (clustering) σημεία δεδομένων. Κατά συνέπεια δεδομένα δοκιμής που δεν έχουν επισημανθεί, κατηγοριοποιηθεί ή ταξινομηθεί χρησιμοποιούνται για την εκπαίδευση των αλγορίθμων. Οι αλγόριθμοι μάθησης χωρίς επίβλεψη βρίσκουν μοτίβα στα δεδομένα και λαμβάνουν αποφάσεις με βάση το εάν αυτά τα μοτίβα υπάρχουν ή όχι σε κάθε νέο κομμάτι δεδομένων. Μια βασική εφαρμογή της μάθησης χωρίς επίβλεψη είναι στο πεδίο της εκτίμησης πυκνότητας (density estimation) στη στατιστική όπως η εύρεση της συνάρτησης πιθανότητας πυκνότητας (probability density function). [18] Η διαδικασία αντιστοίχισης ενός συνόλου δεδομένων σε υποσύνολα ή συστάδες έτσι ώστε οι παρατηρήσεις από διαφορετικές συστάδες να είναι ανόμοιες και οι παρατηρήσεις από την ίδια συστάδα να είναι παρόμοιες με βάση ένα ή περισσότερα προκαθορισμένα κριτήρια, είναι γνωστή ως ανάλυση συστάδων.



Εικόνα 8: Μη-εποπτευόμενη μηχανική μάθηση. [62]

Στην ‘Εικόνα 8’ βλέπουμε το μοντέλο λειτουργίας της μη-εποπτευόμενης μηχανικής μάθησης, με την είσοδο ανάμεικτων δεδομένων ο αλγόριθμος επιχειρεί να ταξινομήσει τις εξόδους πετυχημένα ξεχωρίζοντας τα χαρακτηριστικά βάση ομοιότητας.

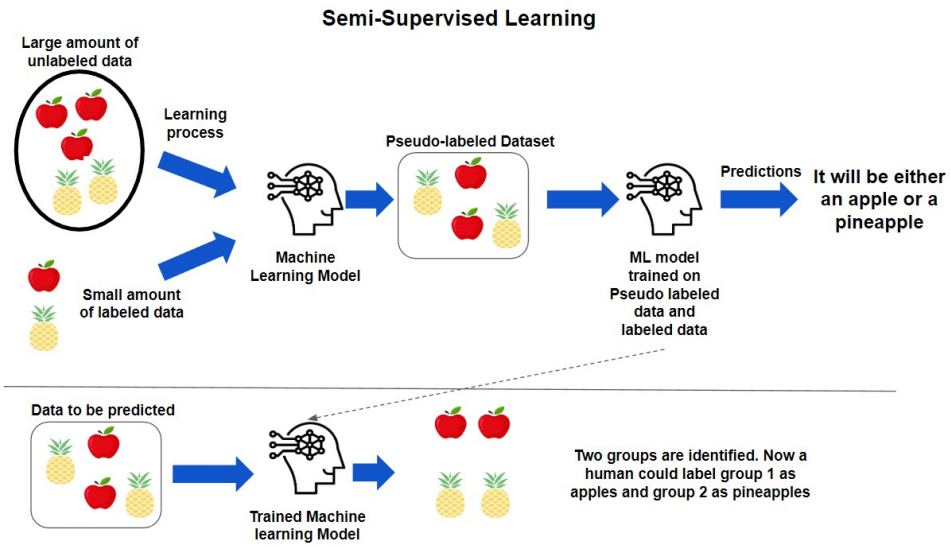
Στην μη-εποπτευόμενη μάθηση δεν υπάρχει ετικετοποίηση, το μοντέλο μαθαίνει μέσω της εκπαίδευσης από τα δεδομένα με τον αλγόριθμο που θα επιλεχθεί. Διαφορετικές τεχνικές ομαδοποίησης κάνουν διαφορετικές υποθέσεις σχετικά με τη δομή των

δεδομένων, που συχνά ορίζονται από κάποια μετρική ομοιότητας και αξιολογούνται. Μερικοί δημοφιλείς αλγόριθμοι μη εποπτευόμενης μάθησης αναφορικά είναι:

- *Κ-μέσων*, επιχειρεί να χωρίσει n παρατηρήσεις σε k συμπλέγματα στα οποία κάθε παρατήρηση ανήκει στο σύμπλεγμα με τον πλησιέστερο μέσο όρο που αναφέρεται και ως κέντρο συστάδων.
- *Ιεραρχική συσταδοποίηση*, ο αλγόριθμος δημιουργεί συστάδες μετρώντας τις ανομοιότητες μεταξύ των δεδομένων.

3.2.3 Ημι-εποπτευόμενη μάθηση

Η αδύναμη εποπτεία γνωστή και ως ημι-εποπτευόμενη μάθηση έχει κερδίσει εξέχουσα θέση στον τομέα της μηχανικής μάθησης καθώς τα μεγάλα γλωσσικά μοντέλα έχουν γίνει πιο περίπλοκα και απαιτούν σημαντικό όγκο δεδομένων για την εκπαίδευση. Χαρακτηρίζεται από το συνδυασμό ενός μεγάλου όγκου δεδομένων χωρίς ετικέτα που χρησιμοποιούνται κυρίως ως παράδειγμα μη εποπτευόμενης μάθησης, με περιορισμένο αριθμό δεδομένων με ανθρώπινη επισήμανση που χρησιμοποιούνται αποκλειστικά σε πιο δαπανηρό και χρονοβόρο παράδειγμα εποπτευόμενης μάθησης.



Εικόνα 9: Ημι-εποπτευόμενη μηχανική μάθηση. [63]

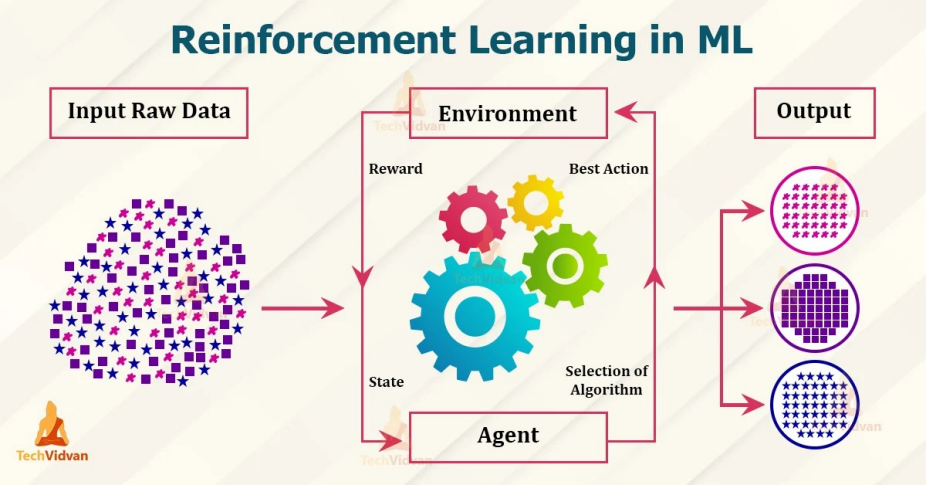
Στην 'Εικόνα 9' βλέπουμε το μοντέλο λειτουργίας της ημι-εποπτευόμενης μηχανικής μάθησης, όπου συνδιάζονται ένα μεγαλύτερο κομμάτι δεδομένων με τον τρόπο της μη-εποπτεόμενης μάθησης που είδαμε προηγουμένως με ένα μικρότερο κομμάτι δεδομένων της εποπτευόμενης μάθησης που είδαμε προηγουμένως επιχειρώντας τελικά μέσω του συνδιασμού αυτού καλύτερες προβλέψεις.

Στην ημι-εποπτευόμενη μάθηση οι ετικέτες εκπαίδευσης είναι θορυβώδεις, περιορισμένες ή ανακριβείς. Ωστόσο, αυτές οι ετικέτες είναι συχνά φθηνότερες ή ευκολότερες στην απόκτηση με αποτέλεσμα μεγαλύτερα πιο αποτελεσματικά σετ

εκπαίδευσης. [19]

3.2.4 Ενισχυτική μάθηση

Το πεδίο της μηχανικής μάθησης που ονομάζεται ενισχυτική μάθηση μελετά τον τρόπο με τον οποίο οι πράκτορες λογισμικού πρέπει να συμπεριφέρονται σε μια δεδομένη κατάσταση για να μεγιστοποιήσουν μια έννοια γνωστή ως αθροιστική ανταμοιβή. Λόγω της ευρείας φύσης της η ενισχυτική μάθηση βρίσκει εφαρμογή σε πολλά διαφορετικά πεδία. Στην ενισχυτική μάθηση το περιβάλλον τυπικά αναπαρίσταται ως διαδικασία απόφασης Markov (Markov decision process ή MDP). Το MDP παρέχει ένα μαθηματικό πλαίσιο για τη μοντελοποίηση της λήψης αποφάσεων σε καταστάσεις όπου τα αποτελέσματα είναι εν μέρει τυχαία και εν μέρει υπό τον έλεγχο ενός υπεύθυνου λήψης αποφάσεων. Πολλοί αλγόριθμοι ενισχυτικής μάθησης χρησιμοποιούν τεχνικές δυναμικού προγραμματισμού. [20]



Εικόνα 10: Ενισχυτική μηχανική μάθηση. [64]

Στην ‘Εικόνα 10’ βλέπουμε το μοντέλο λειτουργίας της ενισχυτικής μηχανικής μάθησης, όπου μετά την εισαγωγή των δεδομένω ο αλγόριθμος μες το περιβάλλον μέσω του πράκτορα προσπαθεί να μεγιστοποιήσει την ανταμοιβή του ώστε να βελτιώσει τις προβλέψεις του.

Η επιβράβευση του αλγόριθμου με ανατροφοδότηση του επιτρέπει να μάθει από τα λάθη του και να παράγει καλύτερα αποτελέσματα στο μέλλον, λόγω του σημαντικού πλεονεκτήματος που προσφέρει σε τεχνολογίες όπως η τεχνητή νοημοσύνη αυτό το είδος μάθησης ερευνάται ευρέως σε όλο τον κόσμο. Μερικοί δημοφιλείς τύποι ενισχυτικής μάθησης αναφορικά είναι:

- *Θετική ενισχυτική μάθηση*, είναι η προσθήκη ενός θετικού αποτελέσματος για την ενίσχυση της συμπεριφοράς.
- *Αρνητική ενισχυτική μάθηση*, είναι η αφαίρεση ενός αρνητικού αποτελέσματος για την ενίσχυση μιας συμπεριφοράς.

3.3 Μοντέλα μηχανικής μάθησης

Η δημιουργία ενός μοντέλου που εκπαιδεύεται με δεδομένα εκπαίδευσης και στη συνέχεια έχει τη δυνατότητα να αναλύει περισσότερα δεδομένα προκειμένου να παρέχει προβλέψεις είναι ένας τρόπος με τον οποίο εκτελείται η μηχανική εκμάθηση. Τα συστήματα μηχανικής μάθησης έχουν μελετηθεί και χρησιμοποιηθεί με πολλούς τύπους μοντέλων.

Τα ζητήματα ταξινόμησης (classification) και παλινδρόμησης (regression) είναι οι δύο κύριες κατηγορίες προβλημάτων μηχανικής μάθησης και εκεί μπορούμε να κατηγοριοποιήσουμε και τα μοντέλα. [21]

- Τα *μοντέλα ταξινόμησης* πρόβλεπουν κατηγορικές διακριτες τιμές όπως μπλε ή κόκκινο, αληθές ή ψευδές και λοιπά.
- Τα *μοντέλα παλινδρόμησης* προβλέπουν αριθμητικές ή συνεχείς τιμές όπως αξίες αντικειμένων, τάσεις αγοράς και λοιπά.

Εικόνα 11: Μοντέλο παλινδρόμησης και ταξινόμησης. [65]

Στην ‘Εικόνα 11’ βλέπουμε τη διαφορετική λειτουργία μεταξύ των μοντέλων παλινδρόμησης και ταξινόμησης, στην παλινδρόμηση επιχειρείται να βρεθεί κατά πόσο τα χαρακτηριστικά ταιριάζουν με ένα αποτέλεσμα, ενώ στην ταξινόμηση επιχειρείται να ταξινομηθούν τα χαρακτηριστικά σε προκαθορισμένες ομάδες.

3.3.1 Μοντέλα ταξινόμησης

Η διαδικασία αναγνώρισης, κατανόησης και τοποθέτησης εννοιών και αντικειμένων σε προκαθορισμένες ομάδες, μερικές φορές γνωστές ως υποπληθυσμοί, είναι γνωστή ως ταξινόμηση. Αυτά τα προκατηγοριοποιημένα σύνολα δεδομένων εκπαίδευσης επιτρέπουν στα προγράμματα μηχανικής μάθησης να ταξινομούν μελλοντικές πληροφορίες σε κατάλληλες και σχετικές κατηγορίες χρησιμοποιώντας μια ποικιλία αλγορίθμων. Οι μέθοδοι ταξινόμησης μηχανικής μάθησης χρησιμοποιούν δεδομένα εκπαίδευσης εισόδου για να εκτιμήσουν την πιθανότητα τα επόμενα δεδομένα να ταιριάζουν σε μια από τις προκαθορισμένες κατηγορίες. Όπως χρησιμοποιείται από

τους κορυφαίους παρόχους υπηρεσιών email του σήμερα, μία από τις πιο δημοφιλείς χρήσεις της ταξινόμησης είναι το φιλτράρισμα των email σε κατηγορίες ‘spam’ ή ‘non-spam’. [22] Τα πιο δημοφιλή μοντέλα ταξινόμισης είναι:

- *Λογιστική παλινδρόμιση*, μια εποπτευόμενη προσέγγιση μηχανικής μάθησης που ονομάζεται λογιστική παλινδρόμηση χρησιμοποιείται κυρίως για προβλήματα ταξινόμησης στα οποία ο στόχος είναι να εκτιμηθεί η πιθανότητα ότι ένα δεδομένο αντικείμενο θα ανήκει σε μια συγκεκριμένη κλάση. Η λογιστική παλινδρόμηση είναι ο όρος για τους αλγόριθμους ταξινόμησης που την χρησιμοποιούν. Χρησιμοποιείται για την πρόβλεψη της κατηγορικής εξαρτημένης μεταβλητής χρησιμοποιώντας ένα δεδομένο σύνολο ανεξάρτητων μεταβλητών. [23]
- *Θεώρημα Μπέυζ*, μια ομάδα αλγορίθμων ταξινόμησης που βασίζονται στο θεώρημα του Μπέυζ είναι γνωστοί ως ταξινομητές Μπέυζ. Στην πραγματικότητα είναι μια οικογένεια αλγορίθμων και όχι μια ενιαία μέθοδος και βασίζονται όλοι στην ίδια αρχή, δηλαδή κάθε ζεύγος χαρακτηριστικών που ταξινομείται είναι ανεξάρτητο μεταξύ του. Η θεμελιώδης υπόθεση του Μπέυζ είναι ότι κάθε χαρακτηριστικό κάνει μια ανεξάρτητη και ίση συμβολή στο αποτέλεσμα. [24]
- *Δένδρο απόφασης*, κατασκευάζει μια δομή δέντρου που μοιάζει με ένα διάγραμμα ροής, με κάθε εσωτερικό κόμβο να σημαίνει μια δοκιμή σε ένα χαρακτηριστικό, κάθε κλάδος να ορίζει ένα αποτέλεσμα δοκιμής και κάθε κόμβο φύλλου (τερματικός κόμβος) να περιέχει ένα όνομα κλάσης. Ένα δέντρο αποφάσεων είναι μια δομή δέντρου που μοιάζει με διάγραμμα ροής όπου κάθε εσωτερικός κόμβος υποδηλώνει το χαρακτηριστικό, οι κλάδοι υποδηλώνουν τους κανόνες και οι κόμβοι φύλλων δηλώνουν το αποτέλεσμα του αλγορίθμου. [25]
- *Τυχαία δένδρα απόφασης*, είναι μια τεχνική εκμάθησης συνόλου που δημιουργεί πολλά δέντρα αποφάσεων κατά τη διάρκεια της φάσης εκπαίδευσης για εργασίες όπως ταξινόμηση. Για εργασίες ταξινόμησης η έξοδος του τυχαίου δάσους είναι η κλάση που επιλέγεται από τα περισσότερα δέντρα. [26]
- *Κ-κοντινότεροι γείτονες (K-nearest neighbors)*, ένα στοιχείο κατανέμεται στην κατηγορία που είναι πιο συχνή μεταξύ των k πλησιέστερων γειτόνων του (k είναι ένας θετικός αριθμός, συνήθως μικρός) με βάση την πολλαπλή ψήφο των γειτόνων του. Το αντικείμενο απλώς τοποθετείται στην κλάση αυτού του πλησιέστερου γείτονα εάν k=1. Το αποτέλεσμα είναι η εγγραφή σε μια τάξη. Δεδομένου ότι αυτός ο αλγόριθμος βασίζεται στην απόσταση για ταξινόμηση, εάν τα χαρακτηριστικά αντιπροσωπεύουν διαφορετικές φυσικές μονάδες ή βρίσκονται σε πολύ διαφορετικές κλίμακες, τότε η κανονικοποίηση των δεδομένων εκπαίδευσης μπορεί να βελτιώσει δραματικά την ακρίβειά τους. [27]
- *Μηχανές διανυσμάτων στήριξης (support vector machines ή SVM)*, ο πρωταρχικός στόχος του αλγόριθμου SVM είναι να εντοπίσει το καλύτερο υπερεπίπεδο σε έναν χώρο N-διάστατων που μπορεί να χρησιμοποιηθεί για τη διαίρεση των σημείων δεδομένων σε διάφορες κατηγορίες χώρου χαρακτηριστικών. Το υπερεπίπεδο προσπαθεί να διατηρήσει τη μεγαλύτερη δυνατή προσωρινή μνήμη μεταξύ των πλησιέστερων σημείων διαφόρων κατηγοριών. Ο αριθμός των χαρακτηριστικών καθορίζει τη διάσταση του υπερεπιπέδου. Ο SVM αντιστοιχίζει παραδείγματα εκπαίδευσης σε σημεία στο χώρο, ώστε να μεγιστοποιήσει το πλάτος του χάσματος μεταξύ των δύο

κατηγοριών. Στη συνέχεια, τα νέα παραδείγματα χαρτογραφούνται στον ίδιο χώρο και προβλέπεται ότι ανήκουν σε μια κατηγορία με βάση την πλευρά του κενού που πέφτουν. [28]

3.3.2 Μοντέλα παλινδρόμησης

Μια μέθοδος για την εξέταση της σχέσης μεταξύ ανεξάρτητων μεταβλητών ή χαρακτηριστικών και μιας εξαρτημένης μεταβλητής ή αποτελέσματος είναι η παλινδρόμηση μηχανικής μάθησης. Είναι μια τεχνική για την προγνωστική μοντελοποίηση μηχανικής μάθησης, η οποία χρησιμοποιεί έναν αλγόριθμο για την πρόβλεψη συνεχών αποτελεσμάτων. Μία από τις πιο δημοφιλείς χρήσεις των μοντέλων μηχανικής μάθησης, ιδιαίτερα στην εποπτευόμενη μηχανική μάθηση, είναι η επίλυση προβλημάτων παλινδρόμησης. Η σχέση μεταξύ ανεξάρτητων παραγόντων και μιας εξαρτημένης μεταβλητής ή ενός αποτελέσματος διδάσκεται στους αλγόριθμους μέσω της εκπαίδευσης. Το μοντέλο μπορεί στη συνέχεια να χρησιμοποιηθεί για να συμπληρώσει ένα κενό δεδομένων ή να προβλέψει το αποτέλεσμα φρέσκων, αόρατων δεδομένων εισόδου. Η ανάλυση παλινδρόμησης είναι αναπόσπαστο μέρος οποιουδήποτε μοντέλου πρόβλεψης, έτσι είναι μια κοινή μέθοδος που βρίσκεται στα προγνωστικά αναλυτικά στοιχεία που υποστηρίζονται από μηχανική μάθηση. Εκτός από την ταξινόμηση, η παλινδρόμηση είναι μια κοινή χρήση για μοντέλα εποπτευόμενης μηχανικής εκμάθησης. [29]

Τα πιο γνωστά μοντέλα παλινδρόμισης είναι:

- *Γραμμική παλινδρόμηση*, χρησιμοποιώντας ως βάση την ανεξάρτητη μεταβλητή εισόδου, προβλέπει τις συνεχείς μεταβλητές εξόδου. Η γραμμική παλινδρόμηση ήταν ο πρώτος τύπος ανάλυσης παλινδρόμησης που μελετήθηκε αυστηρά και χρησιμοποιήθηκε εκτενώς σε πρακτικές εφαρμογές. [30]
- *Παλινδρόμηση κορυφής*, είναι μια μέθοδος εκτίμησης των συντελεστών των μοντέλων πολλαπλής παλινδρόμησης σε σενάρια όπου οι ανεξάρτητες μεταβλητές έχουν υψηλή συσχέτιση. [31] Ως μέθοδος κανονικοποιεί μη-οριοθετημένα προβλήματα δηλαδή προβλήματα που δεν έχουν ξεκάθαρη ή μοναδική λύση.
- *Δένδρο απόφασης*, είναι η διαδικασία με την οποία το αναμενόμενο αποτέλεσμα θεωρείται πραγματικός αριθμός. Γενικότερα, η έννοια του δέντρου παλινδρόμησης μπορεί να επεκταθεί σε οποιοδήποτε είδος αντικειμένου εξοπλισμένο με ζευγαρωμένες ανομοιότητες όπως κατηγορικές ακολουθίες. [32]
- *Τυχαίο δάσος*, η κύρια ιδέα εδώ είναι να χρησιμοποιήθουν περισσότερα από ένα δέντρα αποφάσεων για να καθορίσουμε το τελικό αποτέλεσμα αντί να εξαρτόμαστε μόνο από ένα. Στην περίπτωση ενός προβλήματος παλινδρόμησης, η τελική έξοδος είναι ο μέσος όρος όλων των εξόδων. [33]
- *Κ-κοντινότερος γείτονας*, η τιμή ιδιότητας του αντικειμένου είναι η έξοδος. Ο μέσος όρος των τιμών των k πλησιέστερων γειτόνων είναι αυτή η τιμή. Η έξοδος ορίζεται απλώς στην τιμή του απλού πλησιέστερου γείτονα εάν k=1. Αν και η μέθοδος είναι αρκετά ελκυστική, γίνεται γρήγορα μη-πρακτική όταν η διάσταση αυξάνεται, δηλαδή όταν υπάρχουν πολλές ανεξάρτητες μεταβλητές. [34]

- *Παλινδρόμηση νευρωνικού δικτύου (neural network regression)*, επειδή η γραμμική παλινδρόμηση μπορεί να μάθει μόνο τη γραμμική σχέση μεταξύ των χαρακτηριστικών και του στόχου και δεν μπορεί να μάθει τη σύνθετη μη-γραμμική σχέση, τα τεχνητά νευρωνικά δίκτυα προτιμώνται για παλινδρόμηση από τη γραμμική παλινδρόμηση. Χρειαζόμαστε πρόσθετες στρατηγικές για να ανακαλύψουμε την περίπλοκη μη-γραμμική σχέση μεταξύ του στόχου και των χαρακτηριστικών. Τα τεχνητά νευρωνικά δίκτυα είναι μία από αυτές τις μεθόδους. Τα τεχνητά νευρωνικά δίκτυα έχουν τη δυνατότητα να μάθουν τη σύνθετη σχέση μεταξύ των χαρακτηριστικών και του στόχου λόγω της παρουσίας Συνάρτησης Ενεργοποίησης (Activation Function) σε κάθε επίπεδο. [35]

3.4 Εφαρμογές μηχανικής μάθησης σχετικές με δεδομένα γράφων

Μπορούμε να εξετάσουμε γενικά τα πεδία εφαρμογής της μηχανικής μάθησης όπως επίσης και ειδικότερα πως η μηχανική μάθηση αξιοποιεί τα δεδομένα γράφων.

3.4.1 Εφαρμογές μηχανικής μάθησης

Υπάρχουν πολλά πεδία εφαρμογής της μηχανικής μάθησης, όπως γεωργία, ανατομία, αστρονομία, αυτοματοποιημένες λήψεις αποφάσεων, τραπεζικά συστήματα, συμπεριφερολογία, βιοιατρική, χημειοπληροφορική, πολιτική επιστήμη, κλιματική επιστήμη, υπολογιστικά δίκτυα, ανίχνευση απάτης πιστωτικών καρτών, ποιότητα δεδομένων, κατηγοριοποίηση dna, οικονομία, ανάλυση οικονομικής αγοράς, αναγνώριση γραφής, υγεία, ανάκτηση πληροφοριών, ασφάλεια, ανίχνευση απάτης ίντερνετ, γλωσσολογία, μηχανική μάθηση, μηχανή μετάφρασης, μάρκετινγκ, ιατρική διάγνωση, διαδικτυακή διαφήμιση, συστήματα συστάσεων, ρομποτική, μηχανές αναζήτησης, ανάλυση συναισθήματος, αναγνώριση ομιλίας, τηλεποικινωνίες, αυτοματοποιημένη απόδειξη θεωρημάτων, χρονική σειρά πρόβλεψης, τομογραφίκή ανακατασκευή, ανάλυση συμπεριφοράς χρήστη και πολλά άλλα ακόμα.

Το 2006, ο πάροχος υπηρεσιών πολυμέσων Netflix πραγματοποίησε τον πρώτο διαγωνισμό ‘Βραβείο Netflix’ για να βρει ένα πρόγραμμα για την καλύτερη πρόβλεψη των προτιμήσεων των χρηστών και τη βελτίωση της ακρίβειας του υπάρχοντος αλγόριθμου προτάσεων ταινιών Cinematch κατά τουλάχιστον 10%. Μια κοινή ομάδα που αποτελείται από ερευνητές από την AT&T Labs-Research σε συνεργασία με τις ομάδες Big Chaos και Pragmatic Theory κατασκεύασε ένα μοντέλο συνόλου για να κερδίσει το Μεγάλο Βραβείο το 2009 για 1 εκατομμύριο δολάρια. [36] Λίγο μετά την απονομή του βραβείου, το Netflix συνειδητοποίησε ότι οι βαθμολογίες των θεατών δεν ήταν οι καλύτεροι δείκτες των προτύπων προβολής τους (‘όλα είναι μια σύσταση’) και άλλαξαν τη μηχανή συστάσεων ανάλογα. [37]

3.4.2 Μηχανική μάθηση βασισμένη σε γράφους

Όπως είδαμε σε προηγούμενο κεφάλαιο ένας γράφος αποτελείται από κόμβους που συνδέονται με τις όποιες σχέσεις, ενώ τους μοντελοποιούμε χρησιμοποιώντας κόμβους που συνδέονται με ακμές και έτσι αναπαριστούνται δεδομένα και οι ιδιότητες μεταξύ τους. Η μηχανική μάθηση σχετική με τα δεδομένα γράφων αναφέρεται ως μηχανική

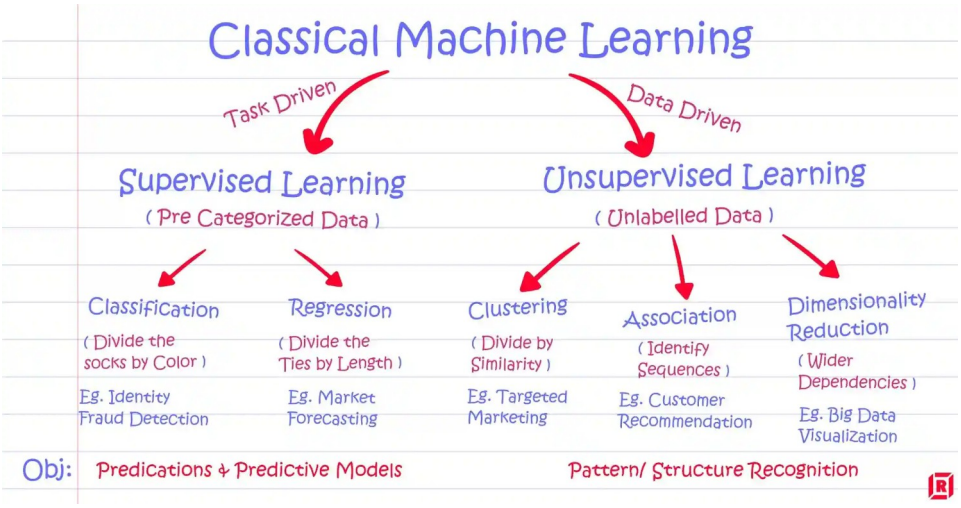
μάθηση γράφων (graph machine learning ή GML), υπάρχουν πολυάριθμες εφαρμογές για την GML όπως στην αλυσίδα εφοδιασμού, τον εντοπισμό απάτης, τις συστάσεις και άλλους τομείς. Οι διάφορες εργασίες μηχανικής μάθησης που μπορεί να ολοκληρώσει η μηχανική μάθηση γράφων μπορούν να χωριστούν βασικά στην εποπτευόμενη μηχανική μάθηση γράφων και στη μη-εποπτευόμενη μηχανική μάθηση γράφων. [52]

Στην εποπτευόμενη μηχανική μάθηση γράφων όπως στο πείραμα μας σε επόμενο κεφάλαιο μπορούμε να κάνουμε:

- *Προβλέψεις διακριτών ή συνεχών ιδιοτήτων ενός κόμβου*.
- *Πρόβλεψη σύνδεσης* όπου γίνεται μια καλή εικασία για το αν πρέπει να υπάρχει μια σχέση.
- Πρόβλεψη μιας διακριτής ή συνεχούς ιδιότητας ενός γράφου ή υπογράφου η οποία είναι γνωστή ως *πρόβλεψη ιδιοτήτων γράφου*.

Στην μη-εποπτευόμενη μηχανική μάθηση γράφων μπορούμε να κάνουμε:

- *Εκμάθηση αναπαράστασης*, για μείωση διαστάσεων χωρίς θυσία κρίσιμων σημάτων.
- Χρησιμοποιώντας μια προσέγγιση ομαδοποίησης που ονομάζεται *ανίχνευση κοινότητας* μπορεί κανείς να βρει ομάδες κόμβων σε ένα γράφο που σχετίζονται σε μεγάλο βαθμό μεταξύ τους.
- Η *ομοιότητα* είναι η διαδικασία αναγνώρισης και αξιολόγησης παρόμοιων ζευγών κόμβων μέσα σε ένα δίκτυο.
- *Εύρεση μονοπατιού και κεντρικότητα*, η εύρεση μονοπατιού εντοπίζει τα λιγότερο ‘ακριβά’ μονοπάτια σε ένα γράφο, ενώ η κεντρικότητα εντοπίζει σημαντικούς κόμβους.



Εικόνα 12: Οι στόχοι ορίζουν την επιλογή τύπου μηχανικής μάθησης γράφου. [66]

Στην ‘Εικόνα 12’ βλέπουμε ένα παράδειγμα πως ανάλογα τους στόχους του

συστήματος (για παράδειγμα πρόβλεψη ή αναγνώριση δομών) επιλέγουμε τον τύπο της μηχανικής μάθησης γράφου.

3.5 Περιορισμοί μηχανικής μάθησης

Παρόλο που η μηχανική μάθηση έχει φέρει επανάσταση σε πολλούς τομείς, τα προγράμματά της συχνά δεν ανταποκρίνονται στις προσδοκίες. Οι λόγοι για αυτό είναι πολλοί: έλλειψη (κατάλληλων) δεδομένων, έλλειψη πρόσβασης στα δεδομένα, μεροληψία δεδομένων, προβλήματα απορρήτου, κακώς επιλεγμένες εργασίες και αλγόριθμοι, λάθος εργαλεία και άτομα, έλλειψη πόρων και προβλήματα αξιολόγησης. [38] Οι προσπάθειες χρήσης μηχανικής μάθησης στην υγειονομική περίθαλψη με το σύστημα IBM Watson απέτυχαν ακόμη και μετά από χρόνια και δισεκατομμύρια δολάρια που επενδύθηκαν. [39] Για αυτούς τους περιορισμούς μπορεί να ευθύνονται:

- *Αλγοριθμική προκατάληψη*, χαρακτηρίζει επαναλαμβανόμενα, συστηματικά λάθη σε ένα σύστημα υπολογιστή που οδηγούν σε ‘άδικα αποτελέσματα’, όπως ‘προνόμιο’ μιας κατηγορίας έναντι μιας άλλης με τρόπους που αποκλίνουν από τον επιδιωκόμενο σκοπό του αλγορίθμου.
- *Επεξηγήσιμη τεχνητή νοημοσύνη (explainable AI, XAI)*, περιγράφει τον τρόπο δημιουργίας ενός συστήματος τεχνητής νοημοσύνης που επιτρέπει στους ανθρώπους να διατηρούν πνευματικό έλεγχο πάνω σε αυτό ή πώς να το κάνουν. Το XAI αντιμετωπίζει την τάση του ‘μαύρου κουτιού’ της μηχανικής μάθησης, όπου ακόμη και οι σχεδιαστές της AI δεν μπορούν να εξηγήσουν γιατί κατέληξε σε μια συγκεκριμένη απόφαση. [40]
- *Υπερπροσαρμογή (overfitting)*, στη μαθηματική μοντελοποίηση, η υπερπροσαρμογή είναι ‘η παραγωγή μιας ανάλυσης που αντιστοιχεί πολύ στενά ή ακριβώς σε ένα συγκεκριμένο σύνολο δεδομένων, και ως εκ τούτου μπορεί να μην ταιριάζει σε πρόσθετα δεδομένα ή να προβλέψει μελλοντικές παρατηρήσεις με αξιοπιστία’. [41]

ΚΕΦΑΛΑΙΟ 4: ΜΕΘΟΔΟΛΟΓΙΑ ΠΕΙΡΑΜΑΤΙΚΗΣ ΜΕΛΕΤΗΣ

4.1 Εισαγωγή πειράματος

Στο πλαίσιο αυτής της διπλωματικής, αφού είδαμε στα προηγούμενα δυο κεφάλαια τη θεωρία των δεδομένων γράφων όπως και τη θεωρία της μηχανικής μάθησης, σε αυτό το κεφάλαιο περνάμε σε ένα πρακτικό κομμάτι πειράματος, αυτό το καταφέρνουμε εξετάζοντας ένα σύνολο δεδομένων, το οποίο φέρνουμε στην κατάλληλη μορφή γράφων και θα εκπαιδεύσουμε με δύο διαφορετικά σενάρια ένα σύστημα μηχανικής μάθησης, ώστε τελικά να δούμε τα αποτελέσματα.

4.2 Στόχοι πειράματος

Στόχοι του πειράματος μας γενικότερα είναι να πάρουμε ένα σύνολο δεδομένων, να το μετατρέψουμε σε γράφο και να το χρησιμοποιήσει ένα μοντέλο τύπου γράφου νευρωνικού δικτύου που κάνει προβλέψεις για να δούμε τα αποτελέσματα που βγαίνουν. Ειδικότερα θα πάρουμε ένα σύνολο δεδομένων σχετικό με ταινίες, θα το εξετάσουμε, μετά θα το φέρουμε σε κατάλληλη μορφή γράφων και τελικά θα δημιουργήσουμε ένα σύστημα συστάσεων που χρησιμοποιεί γράφο νευρωνικού δικτύου για να εκπαιδευτεί και να κάνει προβλέψεις ταινιών για σύσταση σε χρήστες βασισμένο σε προυπάρχουσες βαθμολογίες που έχουν δώσει και μετά θα κρίνουμε πως λειτουργεί το σύστημα μας με τις κατάλληλες μετρικές. Έπειτα θα δούμε αν αλλάζοντας κριτήρια εκπαίδευσης πως αντιδρά το σύστημα μας ώστε να βγάλουμε συμπεράσματα. Όλα αυτά θα εξηγηθούν ειδικότερα στις επόμενες παραγράφους.

Σε κάθε βήμα του πειράματος μας, από την αρχή της επιλογής του συνόλου δεδομένων, την αποτύπωση σε γράφο, τη δημιουργία του μοντέλου μας, έως το τέλος με την αξιολόγηση των αποτελεσμάτων μας θα εξηγούμε πρώτα τι κάνουμε και γιατί το κάνουμε ενώ στη συνέχεια θα εξηγούμε αναλυτικότερα πως λειτουργεί κάθε κομμάτι.

4.3 Προγραμματιστικά εργαλεία και κώδικας εφαρμογής

Σε αυτό το σημείο μπορούμε να δούμε γενικά τα προγραμματιστικά εργαλεία που χρησιμοποιήσαμε όπως επίσης και ειδικότερα τον κώδικα της εφαρμογής όπως τον αναπτύξαμε σχετικά με τις λειτουργίες του.

4.3.1 Προγραμματιστικά εργαλεία

Για το πείραμα μας θα χρησιμοποιήσουμε τα κατάλληλα προγραμματιστικά εργαλεία, τα οποία αναλυτικότερα είναι.

- Η γλώσσα προγραμματισμού *python* έκδοσης 3.9. Η python έχει κερδίσει τη θέση της ως μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού μεταξύ των επαγγελματιών του ML χάρη στην ευανάγνωστη σύνταξη, τις εκτεταμένες βιβλιοθήκες και τη συμβατότητα μεταξύ πλατφορμών. [42]

- Την γλώσσα την εγκατεστήσαμε μέσω της πλατφόρμας *anaconda* έκδοσης 2022.10. [43]
- Η πλατφόρμα μας προσφέρει τον *spyder IDE* για γράψιμο κώδικα.

Η python ειδικότερα έρχεται με αρκετές προεγκατεστημένες βιβλιοθήκες που μας προσφέρουν λειτουργίες για τον κώδικα μας, ενώ όσες δεν είναι προεγκατεστημένες και τις χρειαζόμαστε μπορούμε να τις εγκαταστήσουμε οι ίδιοι μέσω του anaconda Prompt της πλατφόρμας anaconda με την κατάλληλη εντολή μορφής 'conda', γενικότερα οι βιβλιοθήκες που θα μας χρειαστούν αφορούν επεξεργασία δεδομένων, οπτικοποίηση τους και μηχανική μάθηση. Για τις βιβλιοθήκες μας.

- *NumPy* προσθέτει δυνατότητα για χρήση για μεγάλων πολυδιάστατων πινάκων και μια ευρεία γκάμα μαθηματικών συναρτήσεων.
- *Pandas* προσθέτει δομές δεδομένων και λειτουργίες για το χειρισμό αριθμητικών πινάκων και χρονοσειρών.
- *Matplotlib* προσθέτει την ικανότητα σχεδίασης για τη γλώσσα και προσθέτει λειτουργίες στην numpy.
- *Seaborn* βασίζεται στο matplotlib και προσθέτει την ικανότητα σχεδίασης πολύπλοκων γραφημάτων.
- *Random* προσθέτει την ικανότητα να επιλέγει τυχαίους αριθμούς και τυχαίες επιλογές.
- *Torch* προσθέτει πολυδιάστατους πίνακες όπως το numpy με ισχυρή επιτάχυνση κάρτας γραφικών χρήσιμη για νευρωνικά δίκτυα.
- *String* προσθέτει ορισμένες σταθερές και κλάσεις για χειρισμό δεδομένων τύπου συμβολοσειρών.
- *Re* προσθέτει ικανότητα αντιστοίχισης τυπικών εκφράσεων.
- *Tqdm* προσθέτει τη μπάρα ολοκλήρωσης που μετρά την πρόοδο των βρόχων.
- *Torch.nn.functional* προσθέτει συναρτήσεις συνέλιξης, χρήσιμη για νευρωνικά δίκτυα.
- *Torch_geometric.transforms*, προσθέτει μετασχηματισμό αντικειμένων, χρήσιμη για νευρωνικά δίκτυα.

4.3.2 Κώδικας εφαρμογής

Όπως είδαμε προηγουμένως έχουμε βρεί την γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε, έχουμε στήσει την πλατφόρμα για να γράψουμε κώδικα και έχουμε εγκαταστήσει τις βιβλιοθήκες που θα χρειαστούμε. Κατεβάζουμε επίσης το σύνολο δεδομένων που θα χρησιμοποιήσουμε το οποίο θα αναλύσουμε ειδικά αργότερα. Οπότε τελικά μπορούμε να συνεχίσουμε δημιουργώντας την εφαρμογή. [54] Αναλύουμε τις λειτουργίες της εφαρμογής με τη σειρά που αυτές εμφανίζονται στον κώδικα ο οποίος περιέχει τα ανάλογα σχόλια για εύκολη κατανόηση.

- *'#1 Importing Libraries'*, αυτή η ενότητα εισάγει διάφορες βιβλιοθήκες και πακέτα python που είναι απαραίτητα για την επεξεργασία δεδομένων, την οπτικοποίηση και τη μηχανική μάθηση.
- *'#2 Loading Data'*, ο κώδικας διαβάζει δεδομένα ταινιών και βαθμολογιών από αρχεία .csv χρησιμοποιώντας βιβλιοθήκη pandas και τα αποθηκεύει σε dataframes.
- *'#3 Data Exploration'*, αυτή η ενότητα εξερευνεί τα φορτωμένα δεδομένα.
- *'#4 Data Preprocessing'*, τα είδη ταινιών χωρίζονται και μετατρέπονται σε μεταβλητές δείκτη. Αυτές οι μεταβλητές δείκτη υποδεικνύουν εάν μια ταινία ανήκει σε ένα συγκεκριμένο είδος (π.χ. 'Δράση', 'Περιπέτεια', 'Δράμα'). Αυτές οι πληροφορίες αποθηκεύονται στη μεταβλητή movie_feat, η οποία είναι ένας τανυστής.
- *'#5 User and Movie ID Mapping'*, αυτή η ενότητα αντιστοιχίζει μοναδικά αναγνωριστικά χρηστών και ταινιών σε διαδοχικές τιμές. Δημιουργεί αντιστοιχίσεις για δεδομένα χρήστη και ταινίας, οι οποίες χρησιμοποιούνται για την κατασκευή ακμών που συνδέουν τους χρήστες με ταινίες.
- *'#6 Creating User-Movie Edges'*, τα άκρα που συνδέουν τους χρήστες με ταινίες δημιουργούνται με βάση τα αντιστοιχισμένα αναγνωριστικά χρήστη και ταινιών. Οι δείκτες ακμών που προκύπτουν αποθηκεύονται στο edge_index_user_to_movie.
- *'#7 Creating a Heterogeneous Graph'*, κατασκευάζεται ένας ετερογενής γράφος για να αναπαραστήσει τα δεδομένα. Περιλαμβάνει κόμβους χρήστη και ταινίας, τις δυνατότητες τους και τις αλληλεπιδράσεις χρήστη-ταινίας. Αυτό το τμήμα διασφαλίζει επίσης ότι υπάρχουν μη-κατευθυνόμενες άκρες για τη μετάδοση μηνυμάτων.
- *'#8 Data Splitting'*, το σύνολο δεδομένων χωρίζεται σε σύνολα εκπαίδευσης, επικύρωσης και δοκιμών. Ορισμένες ακμές δεσμεύονται για σκοπούς επικύρωσης και δοκιμής χρησιμοποιώντας μια στρατηγική διαχωρισμού τυχαίων συνδέσμων.
- *'#9 Creating a Data Loader'*, δημιουργείται ένας φορτωτής δεδομένων για την αποτελεσματική δειγματοληψία δεδομένων για εκπαίδευση. Χρησιμοποιεί το LinkNeighborLoader από την pytorch geometric για δειγματοληψία και φόρτωση δεδομένων..
- *'#10 Defining the GNN Model'*, αυτό το μέρος ορίζει ένα μοντέλο νευρωνικού δικτύου γράφου (GNN). Αποτελείται από δύο επίπεδα GraphSAGE (SAGEConv) για τη μετάδοση μηνυμάτων.
- *'#11 Defining a Classifier'*, ενας ταξινομητής ορίζεται για να κάνει προβλέψεις σε επίπεδο ακμής υπολογίζοντας τα εσωτερικά γινόμενα μεταξύ των ενσωματώσεων κόμβου (node embeddings) προέλευσης και προορισμού.
- *'#12 Creating the Main Model'*, δημιουργείται το κύριο μοντέλο σύστασης. Συνδυάζει ενσωματώσεις χρηστών και ταινιών, εφαρμόζει το GNN και κάνει προβλέψεις χρησιμοποιώντας τον ταξινομητή.

- *'#13 Training the Model'*, το μοντέλο σύστασης εκπαιδεύεται χρησιμοποιώντας τα δεδομένα εκπαίδευσης. Υπολογίζει την απώλεια και ενημερώνει τις παραμέτρους του μοντέλου χρησιμοποιώντας οπίσθιο πολλαπλασιασμό για να βρεί λάθη ανάποδα από τις εξόδους στις εισόδους. Οι τιμές 'AUC' (Area Under the ROC Curve) συλλέγονται για κάθε εποχή για να αξιολογηθεί η απόδοση του μοντέλου.
- *'#14 Plotting AUC and Loss'*, στο τέλος της εκπαίδευσης, ο κώδικας δημιουργεί δύο γραφήματα, ένα για τις τιμές 'AUC' σε κάθε epoch και ένα άλλο για τις τιμές 'Loss' σε κάθε epoch. Αυτά τα γραφήματα παρέχουν πληροφορίες για την πρόοδο εκπαίδευσης του μοντέλου.

Εμείς μπορούμε να αλλάζουμε τα epochs εντός του κώδικα και να βλέπουμε τα τελικά αποτελέσματα στην κονσόλα. Η τελική έκδοση του κώδικα της εφαρμογής για αυτή τη διπλωματική εργασία τρέχει σε python 3.9 και βρίσκεται στο GitHub: https://github.com/sav206436/FinalCodeDiplomatiki2143.git [53]

4.4 Κατανόηση του συνόλου δεδομένων

Αρχικά, θέλουμε να κατανοήσουμε τη δομή, τα περιεχόμενα του συνόλου δεδομένων μας και τις διάφορες σχέσεις μεταξύ των αντικειμένων μας.

4.4.1 Περιγραφή συνόλου δεδομένων MovieLens 100K

Το σύνολο δεδομένων που θα χρησιμοποιήσουμε στο πείραμα μας είναι το MovieLens 100K, είναι της GroupLens και περιέχει 100.000 Βαθμολογίες ταινιών (ratings) που έδωσαν 1.000 χρήστες (users) για 1.700 ταινίες (movies) και κυκλοφόρησε 4/1998. [44] Το επιλέξαμε γιατί ένα σύνολο δεδομένων αντιπροσωπεύει αλληλεπιδράσεις δηλαδή βαθμολογίες των χρηστών σε ταινίες ώστε μετά να τις χρησιμοποιήσουμε για να κάνουμε προβλέψεις με το μοντέλο που θα φτιάξουμε και συγκεκριμένα αυτό το σύνολο δεδομένων είναι ένα από τα πιο διαδεδομένα datasets για πειράματα στον τομέα της μηχανικής μάθησης.

Όταν το κατεβάσουμε στον υπολογιστή μας έχουμε τέσσερα αρχεία τύπου '.csv' και ένα αρχείο 'README', διαβάζοντας το 'README' μαθαίνουμε για κάθε ένα από τα υπόλοιπα αρχεία μας ότι:

- Το *links.csv* περιέχει στην επικεφαλίδα τις στήλες 'movieId', 'imdbId', 'tmdbId' και κάθε γραμμή μετά χρησιμεύει στο να συσχετίσουμε τα δεδομένα μας με άλλες γνωστές βάσεις δεδομένων αντίστοιχων πληροφοριών όπως το IMDB και το TheMovieDB.
- Το *movies.csv*, περιέχει στην επικεφαλίδα τις στήλες 'movieId', 'title', 'genres' και κάθε γραμμή μετά αντιπροσωπεύει μια ταινία.
- Το *ratings.csv*, περιέχει στην επικεφαλίδα τις στήλες 'userId', 'movieId', 'rating' και 'timestamp' δηλαδή δευτερόλεπτα από τα μεσάνυχτα της 1ης Ιανουαρίου 1970 Συντονισμένη Παγκόσμια Ώρα (coordinated universal time, UTC) μέχρι τη στιγμή της βαθμολόγησης και κάθε γραμμή μετά έχει τη βαθμολογία ενός χρήστη για μια ταινία.

- Το *tags.csv*, περιέχει στην επικεφαλίδα τις στήλες ‘userId’, ‘movieId’, ‘tag’, ‘timestamp’ και κάθε γραμμή μετά έχει το σχόλιο ενός χρήστη για μια ταινία.

Εμείς θα ασχοληθούμε ειδικότερα με το ‘movies.csv’ και το ‘ratings.csv’, πρώτα τα φορτώνουμε στον κώδικα μας και μετά τα μετατρέπουμε σε δύο dataframes, το ‘movies_df’ και το ‘ratings_df’. Το dataframe είναι δομή δεδομένων σε πίνακα που παρέχεται από τη βιβλιοθήκη Pandas και βολεύει για αποθήκευση και επεξεργασία δομημένων δεδομένων δηλαδή αρκετά τακτοποιημένων δεδομένων και εύκολα στη μεταγλώτιση κατά τη μηχανική μάθηση όπως τα δικά μας. Η στήλη movieId λειτουργεί ως ευρετήριο (index) για το dataframe. Αυτό σημαίνει ότι κάθε σειρά στο dataframe θα προσδιορίζεται από την κάθε μοναδική τιμή movieId και έτσι μπορούμε εύκολα να έχουμε πρόσβαση και να χειριζόμαστε τα δεδομένα ταινιών για συγχώνευση, φιλτράρισμα ή εύρεση συγκεκριμένων ταινιών στα επόμενα βήματα. Άρα.

- Δημιουργούμε το *movies_df* διαβάζοντας τα δεδομένα της ταινίας από το αρχείο ‘movies.csv’. Η παράμετρος ‘index_col='movieId'’ ορίζει τη στήλη 'movieId' ως ευρετήριο του dataframe.
- Δημιουργούμε το *ratings_df* διαβάζοντας τα δεδομένα αξιολογήσεων χρηστών από το αρχείο ‘ratings.csv’.

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

Εικόνα 13: Οι πρώτες σειρές του ‘movies_df’.

Στην ‘Εικόνα 13’ βλέπουμε τις πρώτες σειρές του ‘movies_df’, με τις στήλες movieid, τον τίτλο της κάθε ταινίας και τα είδη που ανήκει.

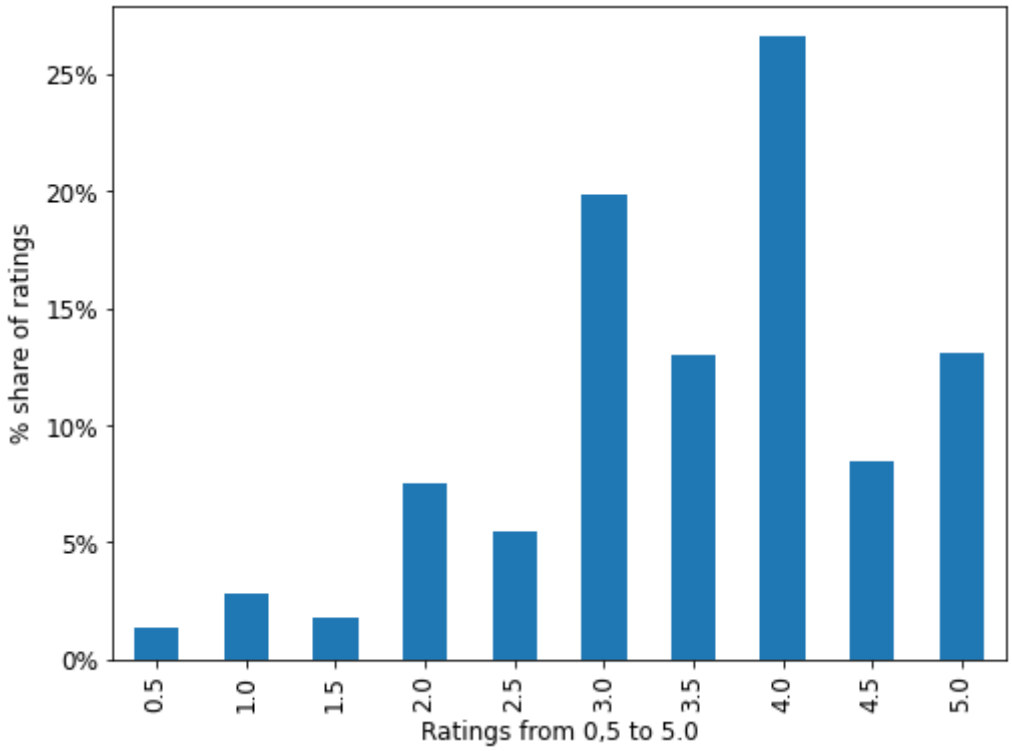
4.4.2 Διερευνητική ανάλυση δεδομένων

Θα κάνουμε μια διερευνητική ανάλυση του συνόλου δεδομένων, αυτό γιατί θέλουμε να εξετάσουμε κάποιες βασικές πληροφορίες ώστε να βγάλουμε κάποια χρήσιμα συμπεράσματα στον τομέα της ανάλυσης δεδομένων.

Έτσι όπως βλέπουμε παρακάτω στην ‘Εικόνα 12’ φτιάχνουμε ένα διάγραμμα X, Y που θα αναπαριστά στο Y την ποσοστιαία κατανομή όλων των βαθμολογιών (% share of ratings) για όλες τις βαθμολογίες που έδωσαν οι χρήστες στο X (Ratings from 0.5 to 5.0). [45] Άρα:

- Χρησιμοποιούμε το *ratings_df* για να φτιάξουμε το διάγραμμα και να οπτικοποιήσουμε την κατανομή των βαθμολογιών που δίνονται από τους χρήστες.

- Η συνάρτηση *value_counts()* υπολογίζει τη συχνότητα κάθε μοναδικής τιμής αξιολόγησης.
- Η *sort_index()* ταξινομεί τις μοναδικές τιμές αξιολόγησης.
- Τα *plt.xlabel()*, *plt.ylabel()*, *plt.xticks()* και *plt.yticks()* ορίζουν τις ετικέτες και τους άξονες.
- Η *fig.savefig()* αποθηκεύει την γραφική παράσταση ως αρχείο εικόνας



Εικόνα 14: Ποσοστιαία κατανομή βαθμολογιών.

Στην 'Εικόνα 14' συμπεραίνουμε πως η πλειοψηφία των χρηστών βαθμολογεί με 3.0 έως 4.0 αστέρια, ακολουθούν οι βαθμολογίες απο 4.5 εως 5.0 αστέρια, ενώ πολλοί λιγότεροι χρήστες βαθμολογούν απο 0.5 έως 2.0 αστέρια.

4.5 Ορισμός σχέσεων κόμβων-ακμών και προεπεξεργασία δεδομένων

Στην επόμενη φάση ορίζουμε τις σχέσεις μεταξύ των αντικειμένων μας ώστε μετά να προεπεξεργαστούμε τα δεδομένα μας για να είναι σε κατανοητή μορφή από το μοντέλο μας αργότερα.

4.5.1 Ορισμός κόμβων και ακμών

Πριν προχωρήσουμε πρέπει να ορίσουμε ποιοί θα είναι οι κόμβοι και οι ακμές μας καθώς και τις σχέσεις μεταξύ τους.

Στην περίπτωση μας κόμβοι είναι οι χρήστες και οι ταινίες ενώ ακμές είναι οι βαθμολογίες. Δηλαδή οι χρήστες βαθμολογούν ταινίες και οι ταινίες βαθμολογούνται απο χρήστες. Άρα:

- Εχουμε *κόμβους χρηστών*, όπου κάθε μοναδικό 'userId' αντιπροσωπεύει ένα κόμβο χρήστη.
- Έχουμε *κόμβους ταινιών*, όπου κάθε μοναδικό 'movieId' αντιπροσωπεύει ένα κόμβο ταινίας.
- Έχουμε *ακμές χρηστών-ταινιών*, που αντιπροσωπεύουν τις αντιδράσεις, δηλαδή οι χρήστες που βαθμολόγησαν τις ταινίες. Το 'edge_index_user_to_movie' συνδέει τους κόμβους χρήστη με τους κόμβους ταινιών με βάση τις δραστηριότητες αξιολόγησης του χρήστη.
- *Ο τύπος των ακμών είναι τύπου αλληλεπίδρασης αξιολόγησης*, δηλαδή η βαθμολογία είναι η αλληλεπίδραση που αντιπροσωπεύει τη σχέση μεταξύ χρηστών και ταινιών, επίσης επιτρέπεται η μετάδοση μηνυμάτων και προς τις δύο κατευθύνσεις. Αυτές οι ακμές θα χρησιμοποιηθούν αργότερα για εκπαίδευση (training), επικύρωση (validation) και δοκιμή (testing) του συστήματος συστάσεων.

4.5.2 Προεπεξεργασία δεδομένων με μεταβλητές δείκτη

Σε αυτό το σημείο πρέπει να κάνουμε προεπεξεργασία των δεδομένων μας, γιατί με την τωρινή τους μορφή τα δεδομένα του συνόλου δεδομένων μας δεν μπορούν να χρησιμοποιηθούν ακόμα από το μοντέλο που θέλουμε να φτιάξουμε αργότερα.

Κάνουμε μετατρομή των δεδομένων του είδους της ταινίας σε μεταβλητές δείκτη (indicator-dummy-variables), αυτό το κάνουμε γιατί θέλουμε τα μεικτά δεδομένα του είδους να μετατραπούν σε μια μορφή που μπορεί εύκολα να χρησιμοποιηθούν ως είσοδοι για ένα μοντέλο σύστασης αργότερα. [46] Όπως είναι τώρα τα είδη των ταινιών μας είναι μεικτός τύπος δεδομένων που κατηγοριοποιούν όπως πχ 'Action', 'Adventure', 'Drama', 'Horror' και λοιπά ενώ οι αλγόριθμοι μηχανικής μάθησης λειτουργούν με αριθμητικά δεδομένα. Οι μεταβλητές δείκτη είναι μεταβλητές με μια δυαδική τιμή εκ των 0 ή 1 και για να κατηγοριοποιήσει κάποιο αποτέλεσμα όπως για παράδειγμα μια ταινία αν ανήκει στο είδος 'Action' θα λάβει την τιμή 1, ενώ αν δεν ανήκει στις υπόλοιπες θα λάβει την τιμή 0. Για αυτό λοιπόν χρησιμοποιούμε τη μέθοδο str.get_dummies('|') της βιβλιοθήκης Pandas. Άρα:

- Το *movies_df['genres']* ανακτά τη στήλη 'genres' από το dataframe 'movies_df', το οποίο περιέχει τα πολλαπλά είδη ταινιών σε μορφή συμβολοσειρών χωρισμένα με το σύμβολο '|'.
- Το *.str.get_dummies('|')* εφαρμόζεται στη στήλη 'genres', η οποία χωρίζει αυτές τις συμβολοσειρές με βάση το σύμβολο '|' και δημιουργεί μεταβλητές δείκτη (εικονικές μεταβλητές) για κάθε μοναδικό είδος που βρίσκεται στο σύνολο δεδομένων. Αυτό έχει ως αποτέλεσμα ένα dataframe 'genres' όπου κάθε στήλη αντιπροσωπεύει ένα διαφορετικό είδος και η παρουσία ενός είδους για μια συγκεκριμένη ταινία υποδεικνύεται από μια δυαδική τιμή 1 ή 0.

- Το *movie_feat=torch.from_numpy(genres.values).to(torch.float)*, μετά τη δημιουργία εικονικών μεταβλητών, το dataframe 'genres' που προκύπτει μετατρέπεται στον τανυστή PyTorch 'movie_feat' χρησιμοποιώντας torch.from_numpy(). Αυτός ο τανυστής αντιπροσωπεύει τα χαρακτηριστικά του είδους των ταινιών, όπου κάθε σειρά αντιστοιχεί σε μια ταινία και κάθε στήλη αντιπροσωπεύει ένα διαφορετικό είδος.

movieId	Action	Adventure	Drama	Horror
1	0	1	0	0
2	0	1	0	0
3	0	0	0	0
4	0	0	1	0
5	0	0	0	0

Εικόνα 15: Οι πρώτες σειρές του 'genres'.

Στην 'Εικόνα 15' βλέπουμε τις πρώτες σειρές του 'genres' με τις στήλες 'movieId' και τα πρώτα τέσσερα είδη ταινιών, με '0' αν δεν ανήκει η ταινία σε αυτό το είδος και '1' αν ανήκει.

4.6 Χαρτογράφηση και δημιουργία κόμβων χρηστών-ταινιών

Μετά θέλουμε να αποτυπώσουμε κάθε μοναδική ύπαρξη ως κόμβο του γράφου μας βάση των σχέσεων που ορίσαμε νωρίτερα.

4.6.1 Χαρτογράφηση

Πρώτα κάνουμε τη χαρτογράφηση (mapping), αυτό είναι είναι η αντιστοίχιση τιμών ή πεδίων απο μια βάση δεδομένων σε μια άλλη και μας διευκολύνει στο χειρισμό δεδομένων. Εμείς κάνουμε ξεχωριστά χαρτογράφηση των 'userId's και 'movieId's σε διαδοχικές τιμές, αυτό είναι μια κοινή τεχνική στα συστήματα συστάσεων που βασίζονται σε δεδομένα γράφων. Απλοποιεί την αναπαράσταση των κόμβων χρηστών και ταινιών στο γράφο επιτρέποντας αργότερα αποτελεσματικότερη επεξεργασία δεδομένων. Η αντιστοίχιση διασφαλίζει ότι τα μοναδικά αναγνωριστικά χρήστη και ταινιών αντιπροσωπεύονται από ακέραιες τιμές που ξεκινούν από το 0 και αυξάνονται διαδοχικά. Μετά βοηθάνε στην κατασκευή του γράφου, αφού τα αντιστοιχισμένα αναγνωριστικά χρήστη και ταινιών χρησιμοποιούνται για τη δημιουργία ακμών που συνδέουν τους χρήστες με ταινίες στον ετερογενή γράφο. Άρα.

- Τα *unique_user_id* λαμβάνονται εξάγοντας τα μοναδικά αναγνωριστικά από τις αντίστοιχες στήλες στο 'ratings_df'.
- Τα *unique_movie_id* λαμβάνεται εξάγοντας τα μοναδικά αναγνωριστικά από τις αντίστοιχες στήλες στο 'ratings_df'.
- Το *pd.RangeIndex(len(unique_user_id))* δημιουργεί μια σειρά διαδοχικών ακεραίων αριθμών που ξεκινούν από το 0 έως το μήκος των μοναδικών αναγνωριστικών χρηστών-ταινιών. Αυτό το εύρος χρησιμεύει ως αντιστοιχισμένα

αναγνωριστικά για χρήστες και ταινίες.

```
Mapping of userIds to consecutive values:
--------------------------------------------------
   userId  mappedID
0       1         0
1       2         1
2       3         2
3       4         3
4       5         4
```

Εικόνα 16: Χαρτογράφηση σε διαδοχικές τιμές των 'userId's'.

Στην 'Εικόνα 16' βλέπουμε τις πρώτες σειρές της χαρτογράφησης σε διαδοχικές τιμές των 'userId's'.

4.6.2 Δημιουργία ακμών χρηστών-ταινιών

Αφού έχουμε κάνει τη χαρτογράφηση μπορούμε να περάσουμε στη δημιουργία των ακμών χρηστών-ταινιών μέσω συγχωνεύσεων, αυτές χρησιμοποιούνται για την αντιστοίχιση μοναδικών αναγνωριστικών χρηστών και ταινιών σε διαδοχικές τιμές και για τη δημιουργία των ακμών που συνδέουν τους χρήστες με τις ταινίες. Άρα:

- Οι *λειτουργίες συγχώνευσης pd.merge* χρησιμοποιούνται για τη λήψη αναγνωριστικών χρηστών και ταινιών που αντιστοιχίζονται σε διαδοχικές τιμές.
- Οι *ratings_user_id* και *ratings_movie_id* προέρχονται από τη συγχώνευση των αρχικών δεδομένων αξιολογήσεων με τα αντίστοιχα πλαίσια δεδομένων 'unique_user_id' και 'unique_movie_id' στα προυπάρχοντα 'userId's' και 'movieId's', χαρτογραφόντας τα σε διαδοχικές τιμές.

```
Merges create user-movie edges as ratings_user_id:
--------------------------------------------------
   userId  mappedID
0       1         0
1       1         0
2       1         0
3       1         0
4       1         0
```

Εικόνα 17: Δημιουργία ακμών μεταξύ χρηστών-ταινιών.

Στην 'Εικόνα 17' βλέπουμε τις πρώτες σειρές της δημιουργίας ακμών μεταξύ χρηστών-ταινιών μετά από τις κατάλληλες συγχωνεύσεις, οι αντιστοιχίσεις αυτές επιτρέπουν τη δημιουργία ακμών μεταξύ χρηστών και ταινιών με βάση αυτά τα αντιστοιχισμένα αναγνωριστικά.

4.7 Καταχώρηση δεικτών ακμών

Πλέον μπορούμε να φτιάξουμε τους δείκτες ακμών (edge indices) που αντιπροσωπεύουν τις συνδέσεις ή τις σχέσεις μεταξύ κόμβων σε ένα γράφο. Συγκεκριμένα στην PyTorch Geometric, οι δείκτες ακμών αναφέρονται στην αναπαράσταση ακμών μεταξύ κόμβων σε ένα γράφο.

Γι' αυτό αποθηκεύουμε τις ακμές σαν δείκτες ακμών, αυτό το κάνουμε γιατί έτσι αποθηκεύουμε και προετοιμάζουμε τα δεδομένα για τη δημιουργία του γράφου του συστήματος συστάσεων. Θα φτιάξουμε τους δείκτες με πίνακα 'COO format'. Στον πίνακα 'COO format' της βιβλιοθήκης numpy έχουμε τριπλή παράταξη με γραμμή, στήλη και δεδομένα, δηλαδή το δεδομένο[i] είναι η τιμή στη θέση (σειρά[i], στήλη[i]) και επιτρέπει διπλες καταχωρήσεις. tensors είναι πολυδιάστατοι πίνακες n-διαστάσεων με ομοιόμορφο τύπο δεδομένων. Άρα:

- Το *edge_index_user_to_movie* δημιουργείται με τη στοίβαξη των αντιστοιχισμένων 'userId' και 'movieId' ως σειρές tensor, σχηματίζοντας τους τελικούς δείκτες ακμών από χρήστες σε ταινίες. Αυτοί οι δείκτες ακμών αντιπροσωπεύουν συνδέσεις μεταξύ των χρηστών και των ταινιών που έχουν βαθμολογήσει, όπου κάθε ευρετήριο στήλης αντιστοιχεί σε μια ταινία και οι τιμές στους δείκτες σειρών υποδεικνύουν συνδέσεις χρήστη-ταινίας.

```
Final edge_index pointing from users to movies:
-----------------------------------------------
tensor([[   0,    0,    0,  ...,  609,  609,  609],
        [   0,    1,    2,  ..., 3121, 1392, 2873]])
```

Εικόνα 18: Τελική Καταχώρηση δεικτών ακμών.

Στην 'Εικόνα 18' βλέπουμε πως συνδιάσαμε το 'ratings_user_id' και το 'ratings_movie_id' για να δημιουργήσουμε τον tensor 'edge_index_user_to_movie' με σχήμα (2, 100836) όπου η πρώτη σειρά είναι δείκτες χρηστών και η δεύτερη σειρά δείκτες ταινιών.

4.8 Κατασκευή ετερογενούς μη-κατευθυνόμενου γράφου

Τελικά μπορούμε να κατασκευάσουμε τον γράφο μας που θα αναπαριστά τα δεδομένα μας. Ο γράφος είναι σημαντικός για την δομή δεδομένων που θα τροφοδοτηθούν αργότερα στο μοντέλο σημαντικής μάθησης και μάλιστα ο πιο διαδεδομένος τύπος στον τομέα των συστάσεων είναι οι ετερογενείς γράφοι όπως θα κάνουμε εμείς. Αυτός ο τύπος γράφων μπορούν και αποθηκεύουν διαφορετικούς τύπους οντοτήτων με διαφορετικούς τύπους σχέσεων μεταξύ τους, στην περίπτωση μας δύο τύπους δηλαδή χρήστες και ταινίες. Μετά χρησιμοποιούνται σαν είσοδοι για μοντέλα νευρωνικών δικτύων. Σε ένα μη-κατευθυνόμενο γράφο όλες οι ακμές μπορούν να στείλουν μηνύματα και στις δύο πλευρές, δηλαδή σε κάθε ακμή δίνεται η ικανότητα ανεξάρτητου προσανατολισμού προς κάθε άκρη.

Εμείς θα κατασκευάσουμε έναν ετερογενή γράφο με κόμβους τους χρήστες και τις ταινίες και μη-κατευθυνόμενες ακμές που αντιπροσωπεύουν τις αντιδράσεις χρηστών-

ταινιών οι οποίες είναι οι βαθμολογίες, για παράδειγμα στην περίπτωση μας οι χρήστες βαθμολογούν τις ταινίες και οι ταινίες βαθμολογούνται από τους χρήστες. Άρα:

- Χρησιμοποιούμε το *HeteroData()* της PyTorch Geometric για την αναπαράσταση της ετερογενούς δομής δεδομένων γράφου.
- Οι κόμβοι για χρήστες και ταινίες αναγνωρίζονται και εκχωρούνται αναγνωριστικά κόμβων ως *'nodeId's'*.
- Διασφαλίζουμε μη-κατευθυνόμενες ακμές για τη μετάδοση μηνυμάτων χρησιμοποιώντας τον μετασχηματισμό *T.ToUndirected()* από την PyTorch Geometric.

4.9 Ορισμός προβλήματος μηχανικής μάθησης

Θέλουμε τα δεδομένα του Movielens100K που μόλις μετατρέψαμε σε ετερογενή μη-κατευθυνόμενο γράφο, να τα χρησιμοποιήσει το σύστημα μηχανικής μάθησης που θα φτιάξουμε αργότερα ώστε να προβλέπει και αλλάζοντας τα κριτίρια εκπαίδευσης να δούμε πως επηρεάζεται η απόδοση.

4.10 Προετοιμασία δεδομένων μηχανικής μάθησης

Στη συνέχει πρέπει πρώτα να προετοιμάσουμε τα δεδομένα μας ώστε να χρησιμοποιηθούν αργότερα κατά τη μηχανική μάθηση.

4.10.1 Διαχωρισμός δεδομένων

Έχοντας προηγουμένως κατασκευάσει τον γράφο μας μπορούμε να περάσουμε στο επόμενο βήμα που είναι η προετοιμασία των δεδομένων με διαχωρισμό δηλαδή splitting, αυτό πρέπει να γίνει γιατί προετοιμάζει τα δεδομένα μας με τον διαχωρισμό σε δεδομένα εκπαίδευσης, επικύρωσης και δοκιμής. Ο διαχωρισμός δεδομένων είναι απαραίτητος όταν τα μοντέλα βασίζονται σε δεδομένα καθώς διασφαλίζει τη δημιουργία μοντέλων μηχανικής μάθησης. Συνήθως, συναντάμε διαχωρισμό δύο ή τρίων splits του κύριου συνόλου δεδομένων, εάν έχουμε δύο διαχωρισμούς το ένα είναι για εκπαίδευση και το δεύτερο για δοκιμή, ενώ εάν έχουμε τρείς διαχωρισμούς αυτό σημαίνει πως το ένα θα χρησιμοποιηθεί για εκπαίδευση, το άλλο για δοκιμή και το τελευταίο για επικύρωση. Άρα.

- Χρησιμοποιούμε τον μετασχηματισμό T.RandomLinkSplit() από το PyTorch Geometric για να εκτελέσουμε τον διαχωρισμό δεδομένων. Τυχαία χωρίζουμε τις άκρες σε σύνολα εκπαίδευσης, επικύρωσης και δοκιμής με βάση καθορισμένα ποσοστά *(num_val=0.1, num_test=0.1)* του συνολικού αριθμού ακμών στο σύνολο δεδομένων.
- Η παράμετρος *disjoint_train_ratio=0,3* υποδηλώνει την αναλογία των ασυνεχών άκρων που χρησιμοποιούνται για την εκπαίδευση. Σε αυτή την περίπτωση, το 30% των ακμών εκπαίδευσης είναι ασυνεχείς για επίβλεψη.

Στο πείραμα μας έχουμε τρείς διαχωρισμούς, σε γενικές γραμμές συνηθίζεται το ποσοστό να είναι 80% εκπαίδευση, 10% επικύρωση και 10% δοκιμών και αυτό θα ακολουθήσουμε. Επίσης στο πείραμα μας έχουμε εποπτευόμενη μάθηση, η οποία ορίζεται από τη χρήση δεδομένων με ετικέτα όπως το δικό μας για εκπαίδευση αλγορίθμων που ταξινομούν τα δεδομένα ή προβλέπουν αποτελέσματα με ακρίβεια. Στην περίπτωση μας για το σετ εκπαίδευσης 30% θα χρησιμοποιηθούν για επίβλεψη δηλαδή να κάνουν ετικετοποίηση και 70% για να περνάνε μηνύματα δηλαδή να μαθαίνουν τη δομή του γράφου. Για εκτίμηση στο πείραμα μας ορίζουμε πως για κάθε μια θετική ακμή θα έχουμε δύο αρνητικές. Και τελικά βάση βάρους τα αρνητικά δείγματα τα αφήνουμε εκτός. Άρα:

- Ο μετασχηματισμός trtansform δημιουργεί σταθερές αρνητικές για αξιολόγηση με καθορισμένο ποσοστό *neg_sampling_ratio=2,0*. Τα αρνητικά άκρα κατά τη διάρκεια της εκπαίδευσης μπορούν να δημιουργηθούν επιτόπου για να βελτιωθεί η ευρωστία του μοντέλου και η ακρίβεια πρόβλεψης

4.10.2 Φορτωτής δεδομένων

Συνεχίζουμε με τον φορτωτή δεδομένων (data loader), ο φορτωτής δεδομένων ευθύνεται για την αποτελεσματική δειγματοληψία δεδομένων για εκπαίδευση και επικύρωση. Ειδικότερα η δουλειά ενός φορτωτή δεδομένων είναι να παίρνει μικρές παρτίδες από ένα σύνολο δεδομένων που ονομάζονται μίνι-παρτίδες (mini-batches), δίνοντάς μας την ευκολία να επιλέξουμε από διαφορετικούς τρόπους δειγματοληψίας. Ένας πολύ κοινός τρόπος δειγματολειψίας είναι η ομοιόμορφη δειγματοληψία μετά από ανακάτεμα των δεδομένων σε κάθε epoch. Ως epoch εννοούμε κάθε φορά που το σύνολο δεδομένων ολοκληρώνει ένα πέρασμα από τον αλγόριθμο. Ο φορτωτής δεδομένων είναι ιδιαίτερα χρήσιμος για να χειρίζεται μεγάλα σύνολα δεδομένων όπως το δικό μας που μπορεί να μη χωράνε στη μνήμη.

Εμείς χρησιμοποιούμε τον LinkNeighborLoader της PyTorch Geometric, συμπληρώνουμε παραμέτρους όπως ο αριθμός των γειτόνων που θα δειγματιστούν σε κάθε hop (20 στο πρώτο hop και 10 στο δεύτερο hop), την αναλογία αρνητικής δειγματοληψίας (2 προς 1) και το μέγεθος του παρτίδας (128).

- Ο φορτωτής μπορεί να κάνει δειγματοληψία γειτόνων για ένα δεδομένο σύνολο κόμβων. Εδώ, λαμβάνει δείγματα γειτόνων με βάση τον καθορισμένο αριθμό *num_neighbors=[20, 10]* για το πρώτο και το δεύτερο hop.
- Εκτελεί αρνητική δειγματοληψία κατά τη διάρκεια της εκπαίδευσης με *neg_sampling_ratio=2,0*
- Ο φορτωτής ασχολείται με ετικέτες ακμών *(edge_label_index and edge_label)* που είναι απαραίτητες για την εποπτευόμενη μάθηση.
- Συγκεντρώνει τα δεδομένα του δείγματος, βοηθώντας στην παράλληλη επεξεργασία και βελτιστοποιώντας την εκπαίδευση με mini-batch gradient descent *(batch_size=128)*
- Ανακατεύει τα δεδομένα εκπαίδευσης *(shuffle=True)*, εξασφαλίζοντας τυχαιότητα στα δείγματα εκπαίδευσης για καλύτερη γενίκευση.

4.11 Επιλογή μοντέλου γράφου νευρωνικού δικτύου

Στη συνέχεια μπορούμε να επιλέξουμε το μοντέλο μας τύπου γράφου νευρωνικού δικτύου, το οποίο έχει αρκετές ξεχωριστές λειτουργίες.

4.11.1 Μοντέλο γράφου νευρωνικού δικτύου

Τώρα μπορούμε να προχωρήσουμε στο μοντέλο νευρωνικού δικτύου που θα χρησιμοποιήσουμε, είναι το SAGEConv της βιβλιοθήκης PyTorch, χρησιμοποιείται για τη μετάδοση μηνυμάτων και την εκμάθηση αναπαραστάσεων κόμβων σε ένα γράφο [47]. Συγκεκριμένα το μοντέλο μας χρησιμοποιεί δύο στοίβες 'GraphSAGE convolutional layers' (SAGEConv) για τη μετάδοση μηνυμάτων. Το μοντέλο GNN παίρνει χαρακτηριστικά κόμβων και δεικτών ακμών ως δεδομένα εισόδου και εκτελεί μετάδοση μηνυμάτων για να μάθει τις αναπαραστάσεις των κόμβων. Η επιλογή δύο στοιβών μπορεί να διαφέρει ανάλογα με το πρόβλημα και το σύνολο δεδομένων αλλά είναι η πιο κοινή επιλογή για εφαρμογές μηχανικής μάθησης σε δεδομένα γράφων. Το μοντέλο λειτουργεί με τον εξής τύπο: Δεδομένου ενός γράφου G=(V,E) με κόμβους V, άκρες E, χαρακτηριστικά κόμβου Xv για v∈V και μια συνάρτηση τύπου 'AGGREGATE' που συγκεντρώνει πληροφορίες γείτονα.

Χρησιμοποιώντας μοντέλο GNN με 2 στοίβες 'SAGEConv' επιτρέπουμε στο μοντέλο να καταγράψει σύνθετες αλληλεπιδράσεις μεταξύ χρηστών και ταινιών σε έναν ετερογενή μη-κατευθυνόμενο γράφο ώστε μετά να φτιάξουμε ένα σύστημα συστάσεων. Οι δύο στοίβες πραγματοποιούν μετάδοση μηνυμάτων και επιτρέπουν ενσωμάτωση πληροφοριών από συνδεδεμένους κόμβους ώστε να δημιουργηθούν χρήσιμες αναπαραστάσεις και για τους χρήστες και για τις ταινίες, αυτές οι αναπαραστάσεις μετά θα χρησιμοποιηθούν για σύσταση.

4.11.2 Ταξινομητής

Στη συνέχεια ορίζουμε τον ταξινομητή (classifier) μας, στα συστήματα συστάσεων, ο στόχος είναι να προβλέψουμε πώς ένας χρήστης θα βαθμολογήσει ή θα αλληλεπιδράσει με μια συγκεκριμένη ταινία. Για να το επιτύχουμε αυτό, πρέπει να υπολογίσουμε προβλέψεις για κάθε ζεύγος χρήστη-ταινίας και ο ταξινομητής το κάνει αποτελεσματικά λαμβάνοντας αλληλεπιδράσεις χρήστη και ταινίας και υπολογίζοντας το εσωτερικό γινόμενο (dot product). Οι προβλέψεις συγκρίνονται με πραγματικές αξιολογήσεις ή αλληλεπιδράσεις για τον υπολογισμό της απώλειας τύπου 'loss' που θα δούμε αργότερα και την ενημέρωση του μοντέλου κατά τη διάρκεια της εκπαίδευσης. 'loss' είναι η ποινή για μια κακή πρόβλεψη, δηλαδή είναι ένας αριθμός που δείχνει πόσο κακή ήταν η πρόβλεψη του μοντέλου σε ένα μόνο παράδειγμα. Εάν η πρόβλεψη του μοντέλου είναι τέλεια η απώλεια είναι μηδενική διαφορετικά η απώλεια είναι μεγαλύτερη. Στην κλάση 'classifier':

- Η μέθοδος προώθησης περιλαμβάνει τρία ορίσματα, *x_user*, *x_movie* και *edge_label_index*.
- Οι *x_user* και *x_movie* αντιπροσωπεύουν τις ενσωματώσεις των χρηστών και των ταινιών αντίστοιχα, που εξάγονται από το νευρωνικό δίκτυο του γράφου.

- *edge_label_index* περιέχει τους δείκτες των ακμών στο γράφημα για τις οποίες πρέπει να γίνουν προβλέψεις.
- Στη μέθοδο προώθησης, πρώτα ανακτά τις ενσωματώσεις κόμβων (node embeddings) για κόμβους πηγής 'x_user' και προορισμού 'x_movie' χρησιμοποιώντας το *edge_label_index*.
- Στη συνέχεια υπολογίζει την αναπαράσταση σε επίπεδο ακμής εκτελώντας πολλαπλασιασμό βάσει στοιχείων *(*)* μεταξύ των ενσωματώσεων κόμβων (node embeddings) προέλευσης και προορισμού.
- Τέλος, υπολογίζει την πρόβλεψη ανά ακμή εποπτείας αθροίζοντας αυτούς τους πολλαπλασιασμούς βάσει στοιχείων κατά μήκος μιας συγκεκριμένης διάστασης *(dim=-1 αντιπροσωπεύει το άθροισμα κατά μήκος της τελευταίας διάστασης)*.

Ο ταξινομητής ορίζεται για να κάνει προβλέψεις ακμών υπολογίζοντας τα εσωτερικά γινόμενα μεταξύ των ενσωματώσεων κόμβου (node embeddings) πηγής και προορισμού.

4.11.3 Κύριο μοντέλο

Τώρα μπορούμε να δημιουργήσουμε το κύριο μοντέλο (main model), είναι το κύριο μοντέλο σύστασης το οποίο συνδιάζει ενσωματώσεις χρήστη και ταινίας, εφαρμόζει το GNN με δύο στοίβες για μετάδοση μηνυμάτων και κάνει προβλέψεις χρησιμοποιώντας τον ταξινομητή. Η κλάση 'Model' χρησιμεύει ως το κύριο σύστημα συστάσεων που χρησιμοποιεί μια αρχιτεκτονική GNN για την εκτέλεση προτάσεων που βασίζονται σε αλληλεπιδράσεις χρήστη-ταινίας. Άρα.

- Η μέθοδος *__init__* προετοιμάζει την αρχιτεκτονική και τα στοιχεία του μοντέλου.
- Η μέθοδος *forward* λαμβάνει το αντικείμενο εισόδου τύπου HeteroData που περιέχει πληροφορίες χρήστη και ταινίας, Εξάγει τις ενσωματώσεις χρηστών, τις ενσωματώσεις ταινιών και τις αλληλεπιδράσεις τους μέσω του GNN και του ταξινομητή.
- Περνά τις ενσωματώσεις χρήστη και ταινιών μέσω των *επιπέδων GNN* και υπολογίζει τις προβλέψεις σε επίπεδο ακμής χρησιμοποιώντας τον καθορισμένο ταξινομητή.
- Κατά τη διάρκεια της εκπαίδευσης, στιγμιότυπα αυτής της κλάσης 'Model' δημιουργούνται και εκπαιδεύονται με χρήση φορτωτών δεδομένων (*train_loader* και *val_loader*). Ο βελτιστοποιητής (optimizer) εφαρμόζεται για την ενημέρωση των παραμέτρων του μοντέλου με βάση την απώλεια που υπολογίζεται κατά τη διάρκεια κάθε epoch.
- Κατά τη παρεμβολή ή την επικύρωση, η μέθοδος προώθησης καλείται να κάνει προβλέψεις με βάση το εκπαιδευμένο μοντέλο χρησιμοποιώντας το σύνολο δεδομένων επικύρωσης *(val_loader)*.

Σε αυτό το σημείο επίσης μπορούμε να ρυθμίσουμε το μοντέλο μας ώστε να χρησιμοποιήσει τεχνολογία CUDA αν έχουμε την ανάλογη κάρτα γραφικών για πιο γρήγορη επεξεργασία ή επεξεργαστή που έχουμε όλοι στους υπολογιστές μας. Επίσης δημιουργούμε τις άδειες λίστες στην Python οι οποίες θα αποθηκεύουν τα epochs, τα

‘Loss’ και τα ‘AUC’ που θα εξηγήσουμε τον ρόλο των δυο τελευταίων σε επόμενο κεφάλαιο.

4.12 Εκπαίδευση του μοντέλου

Το μοντέλο σύστασης εκπαιδεύεται χρησιμοποιώντας τα δεδομένα εκπαίδευσης. Υπολογίζει την απώλεια και ενημερώνει τις παραμέτρους του μοντέλου. Οι τιμές ‘AUC’ συλλέγονται για κάθε epoch για να αξιολογηθεί η απόδοση του μοντέλου.

Για την εκπαίδευση του Model μας δημιουργούμε την ανάλογη εκπαιδευτική λούπα για το μοντέλο συστάσεων, εκεί ορίζουμε τον αριθμό epochs που θέλουμε για να εκπαιδευτεί το μοντέλο μας χρησιμοποιώντας τα δεδομένα εκπαίδευσης. Εκπαιδεύουμε το μοντέλο σύστασης χρησιμοποιώντας τα δεδομένα εκπαίδευσης, κάνουμε αξιολόγηση στην απόδοση στα δεδομένα επικύρωσης και παρακολουθούμε ‘Loss’ και ‘AUC’ για κάθε epoch. Η ‘AUC’ χρησιμοποιείται συνήθως για την αξιολόγηση της ικανότητας του μοντέλου να ταξινομεί αποτελεσματικά αντικείμενα (ταινίες) για σύσταση. Η Loss ποσοτικοποιεί την ποιότητα των προβλέψεων του μοντέλου. Άρα:

- *Επαναληπτική εκπαίδευση*, ένας βρόχος διατρέχει έναν καθορισμένο αριθμό epochs (Στο πείραμα μας αργότερα 5 ή 50 epochs).
- *Εκπαίδευση με παρτίδες*, τα δεδομένα εκπαίδευσης τροφοδοτούνται στο μοντέλο κατά παρτίδες (train_loader) χρησιμοποιώντας το LinkNeighborLoader από την PyTorch Geometric. Μαζεύει δείγματα γειτονικών κόμβων και ακμών για κάθε παρτίδα.
- *Αντίθετη διάδοση και βελτιστοποίηση*, σε κάθε επανάληψη, οι παράμετροι του μοντέλου βελτιστοποιούνται χρησιμοποιώντας αντίθετη διάδοση και τον βελτιστοποιητή τύπου Adam (optimizer.step())
- *Υπολογισμός απώλειας*, η απώλεια τύπου ‘Loss’ υπολογίζεται μεταξύ προβλέψεων μοντέλων και ετικετών αληθείας βάσης (F.binary_cross_entropy_with_logits).
- *Αξιολόγηση απόδοσης*, η απόδοση του μοντέλου αξιολογείται στο σύνολο επικύρωσης μετά από κάθε εποχή για τον υπολογισμό της μέτρησης της περιοχής κάτω από την καμπύλη (AUC).

4.13 Παρακολούθηση και απόδοση ανάλυσης του μοντέλου

Για να αξιολογήσουμε το μοντέλο μας χρησιμοποιούμε δύο μετρικές, η πρώτη είναι η δυαδική απώλεια διασταυρούμενης εντροπίας με logits (binary cross entropy loss with logits, ‘Loss’) και η δεύτερη η περιοχή κάτω από την καμπύλη ROC (area under the ROC curve, ‘AUC’), με αυτές πλεόν αντί απλώς να βλέπουμε όπως πριν πως ο κώδικας μας λειτουργεί μπορούμε να ποσοτικοποιήσουμε και να κρίνουμε την ποιότητα των αποτελεσμάτων κατά τη διάρκεια της μηχανικής μάθησης.

4.13.1 Δυαδική απώλεια διασταυρούμενης εντροπίας με logits

Η πρώτη μετρική είναι η δυαδική απώλεια διασταυρούμενης εντροπίας με logits (Loss), είναι ένα μέτρο της ομοιότητας ή της απόκλισης μεταξύ των προβλεπόμενων τιμών του μοντέλου και των πραγματικών τιμών. Χρησιμοποιείται στην εκπαίδευση μοντέλων μηχανικής μάθησης, συμπεριλαμβανομένων των νευρωνικών δικτύων, για να

καθοδηγήσει τις ενημερώσεις παραμέτρων του μοντέλου. Ο σκοπός της χρήσης της στην εκπαίδευση είναι να καθοδηγήσει το μοντέλο ώστε να βελτιώσει τις προβλέψεις του με την πάροδο του χρόνου. Με την επαναληπτική ενημέρωση των παραμέτρων του μοντέλου για την ελαχιστοποίηση της απώλειας το μοντέλο γίνεται καλύτερο στο να κάνει ακριβείς προβλέψεις. Στην περίπτωση μας η ελαχιστοποίηση των τιμών της είναι σημαντική για την εκπαίδευση του συστήματος συστάσης. Ο στόχος μας είναι να προβλέπουμε με ακρίβεια τον τρόπο που οι χρήστες θα βαθμολογήσουν τις ταινίες και δυαδική απώλεια διασταυρούμενης εντροπίας με logits χρησιμεύει ως μέτρο του πόσο καλά τα καταφέρνει το μοντέλο μας.

- *Pred*, αντιπροσωπεύει τις προβλεπόμενες βαθμολογίες/λογαριασμοί που δημιουργούνται από το μοντέλο για κάθε ακμή.
- *ground_truth*, περιέχει τις πραγματικές ετικέτες για κάθε ακμή. Για εργασίες δυαδικής κατηγοριοποίησης (binary classification), όπως η κατηγοριοποίηση ακμών, περιέχει τις αληθινές ετικέτες που υποδεικνύουν την παρουσία (θετική κλάση) ή την απουσία (αρνητική κλάση) μιας ακμής μεταξύ των κόμβων.
- Η συνάρτηση αυτή χρησιμοποιείται εδώ για τον υπολογισμό της απώλειας μεταξύ των προβλεπόμενων logits (pred) και των ετικετών βασικής αλήθειας (ground_truth). Η συνάρτηση εφαρμόζει εσωτερικά τη συνάρτηση ενεργοποίησης σιγμοειδούς στα logits (sigmoid activation function) και στη συνέχεια *υπολογίζει τη δυαδική απώλεια διασταυρούμενης εντροπίας μεταξύ αυτών των προβλέψεων που έχουν μετασχηματιστεί σε σιγμοειδή και των ετικετών αλήθειας*.

Ειδικότερα, αυτή η 'Loss' συνδυάζει μια σιγμοειδή στοίβα και το BCEloss σε μία μόνο κλάση. Αυτή η έκδοση είναι πιο σταθερή αριθμητικά από τη χρήση ενός απλού σιγμοειδούς που ακολουθείται από ένα BCEloss καθώς, συνδυάζοντας τις πράξεις σε ένα επίπεδο, εκμεταλλευόμαστε το τέχνασμα 'log-sum-exp' για αριθμητική σταθερότητα [48]

Η απώλεια (loss) τύπου binary cross entropy loss with logits υπολογίζεται από τον τύπο:

- $Binary\,Cross-Entropy\,Loss = -(1/N) * \Sigma[\, y_i * \log(\hat{y}_i) + (1-y_i) * \log(1-\hat{y}_i)]$

Όπου:

- N είναι *ο αριθμός των παραδειγμάτων* στο σύνολο δεδομένων.
- y_i είναι *η αληθινή ετικέτα-χαρακτηριστικό* για το i-οστό παράδειγμα (0 ή 1).
- $\hat{y}_i$ είναι *η προβλεπόμενη πιθανότητα για τη θετική κατηγοριοποίηση* και λαμβάνεται με την εφαρμογή μιας σιγμοειδούς συνάρτησης στα logits, όπου logits είναι οι ακατέργαστες-μη κανονικοποιημένες έξοδοι του μοντέλου.

Και λειτουργεί με τον εξής τρόπο:

- *Το μοντέλο παράγει ακατέργαστες βαθμολογίες (logit)* για κάθε κατηγορία.
- Οι ακατέργαστες βαθμολογίες (logits) *μετατρέπονται σε πιθανότητες χρησιμοποιώντας μια σιγμοειδή συνάρτηση*, κατηγοριοποιεί με 0 ή 1.

- *Η binary cross entropy loss συγκρίνει τις προβλεπόμενες πιθανότητες με τις αληθινές ετικέτες* τιμωρώντας το μοντέλο περισσότερο για αποκλίνουσες προβλέψεις.
- *Για βελτιστοποίηση οι παράμετροι του μοντέλου προσαρμόζονται για να ελαχιστοποιηθεί αυτή η απώλεια*, βελτιώνοντας την προγνωστική του απόδοση κατά την εκπαίδευση.

4.13.2 Περιοχή κάτω από την καμπύλη ROC

Η δεύτερη μετρική είναι η περιοχή κάτω από την καμπύλη ROC (AUC), είναι μια μέτρηση που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου διαδυκής ταξινόμησης [49]. Το ROC σημαίνει 'receiver operating characteristic' και το AUC αντιπροσωπεύει την περιοχή κάτω από την καμπύλη ROC. Άρα.

- Κατά την εκπαίδευση το μοντέλο εκπαιδεύεται χρησιμοποιώντας μια δυαδική απώλεια διασταυρούμενης εντροπίας *(F.binary_cross_entropy_with_logits)* για τη βελτιστοποίηση των παραμέτρων με βάση το σφάλμα πρόβλεψης μεταξύ της εξόδου του μοντέλου και των ετικετών αλήθειας βάσης (αλληλεπιδράσεις χρήστη-ταινίας).
- Μετά από κάθε περίοδο εκπαίδευσης, η τιμή 'AUC' υπολογίζεται στο σύνολο δεδομένων επικύρωσης. Οι προβλέψεις από το μοντέλο συγκρίνονται με τις ετικέτες βασικής αλήθειας (αλληλεπιδράσεις μεταξύ χρηστών και ταινιών) χρησιμοποιώντας το *roc_auc_score*.
- Οι τιμές 'AUC' για κάθε epoch αποθηκεύονται στη λίστα *auc_values* .

Στην περίπτωση μας για τη μέτρηση της ποιότητας των προτάσεων που παρέχονται από το μοντέλο προτάσεων που βασίζεται στο GNN εκπαιδεύει το μοντέλο και υπολογίζει τις τιμές 'AUC' κατά τη διάρκεια της εκπαίδευσης για να κατανοήσει πόσο καλά αποδίδει το μοντέλο όσον αφορά τη διάκριση μεταξύ των ταινιών που αρέσουν στους χρήστες και εκείνων που δεν τους αρέσουν. Στη συνέχεια οπτικοποιούμε αυτές τις τιμές 'AUC' ανα epoch για να παρακολουθούμε την πρόοδο εκπαίδευσης του μοντέλου και να βοηθήσει στις καλύτερες επιλογές.

ΚΕΦΑΛΑΙΟ 5: ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΕΙΡΑΜΑΤΙΚΗΣ ΜΕΛΕΤΗΣ

5.1 Ορισμός παραμέτρων και μετρικών

Πρώτα θα δοκιμάσουμε πως λειτουργεί το μοντέλο μας για 5 epochs και θα δούμε τις τιμές Δυαδικής Απώλειας Διασταυρούμενης Εντροπίας με Logits (Loss) και περιοχής κάτω από την καμπύλη ROC (AUC). Μετά θα δοκιμάσουμε αντίστοιχα για 50 epochs. Τελικά θα δούμε κατα πόσο επηρεάζεται ο χρόνος μάθησης και η ποιότητα των προτάσεων μέσω αυτών των αλλαγών όπως θα αποτυπωθούν από τις μετρικές μας.

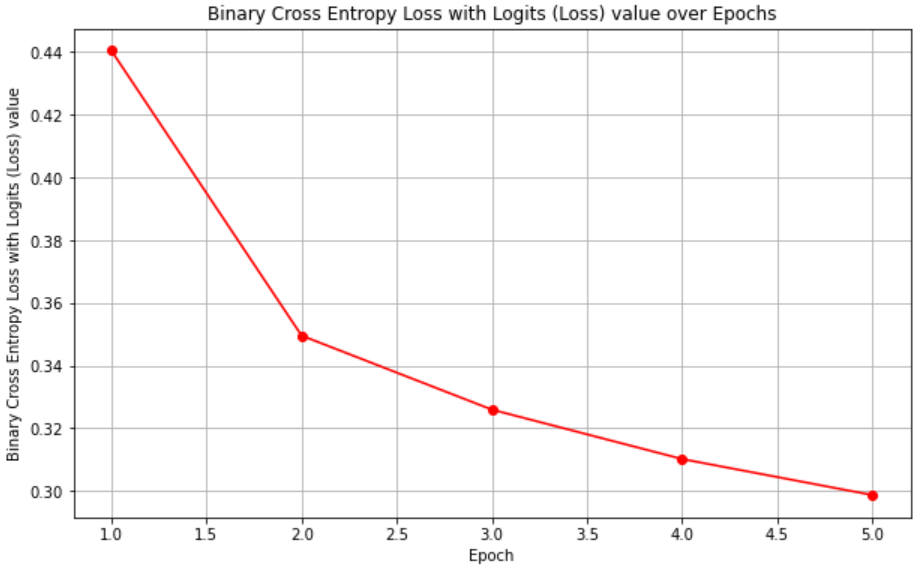
5.2 Πρώτο πείραμα με 5 epochs

Ξεκινάμε να δοκιμάσουμε το μοντέλο μας για 5 epochs με επεξεργαστή.

```
Device: 'cpu'
Training
100%|██████████| 190/190 [00:16<00:00, 11.21it/s]
Epoch: 001, Loss: 0.4405
AUC calculation
100%|██████████| 79/79 [00:06<00:00, 12.03it/s]
AUC: 0.9012
Training
100%|██████████| 190/190 [00:16<00:00, 11.36it/s]
Epoch: 002, Loss: 0.3496
AUC calculation
100%|██████████| 79/79 [00:06<00:00, 11.97it/s]
AUC: 0.9169
Training
100%|██████████| 190/190 [00:16<00:00, 11.32it/s]
Epoch: 003, Loss: 0.3260
AUC calculation
100%|██████████| 79/79 [00:06<00:00, 12.14it/s]
AUC: 0.9190
Training
100%|██████████| 190/190 [00:16<00:00, 11.29it/s]
Epoch: 004, Loss: 0.3102
AUC calculation
100%|██████████| 79/79 [00:06<00:00, 12.39it/s]
AUC: 0.9236
Training
100%|██████████| 190/190 [00:16<00:00, 11.28it/s]
Epoch: 005, Loss: 0.2987
AUC calculation
100%|██████████| 79/79 [00:06<00:00, 12.00it/s]
AUC: 0.9284
```

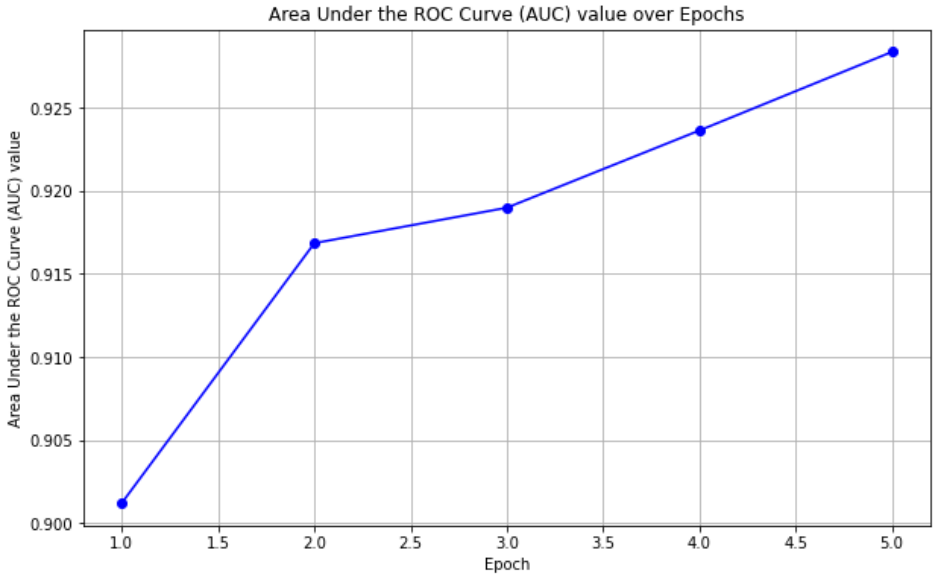
Εικόνα 19: Πως εμφανίζονται 5 epochs στην κονσόλα.

Κατά τη διάρκεια της εκπαίδευσης έως το τέλος στην κονσόλα μας η διαδικασία εμφανίζεται όπως βλέπουμε στην ‘Εικόνα 19’, βλέπουμε στο πρωτο epoch πως η ‘Loss’ ξεκίνησε με τιμή 0.4405, επίσης η ‘AUC’ ξεκίνησε με τιμή 0.9012. Ενώ στο τελευταίο epoch η ‘Loss’ κατέληξε με τιμή 0.2987 και η ‘AUC’ με τιμή 0.9284. Άρα στην πρώτη φάση του πειράματος μας σαν πρώτη εικόνα το μοντέλο μας για 5 epochs σίγουρα αποδίδει καλύτερα καθώς όσο αυξάνονται τα epochs θέλουμε τη ‘Loss’ να μειώνεται και την ‘AUC’ να αυξάνεται.



Εικόνα 20: 'Loss' για 5 epochs.

Στην 'Εικόνα 20' βλέπουμε το γράφημα για την 'Loss' για κάθε epoch με 5 epochs, στον άξονα Χ έχουμε κάθε epoch και στον Ψ την τιμή 'Loss', οι απώλειες μειώνονται άρα το μοντέλο μας προβλέπει όλο και καλύτερα. Συμπεραίνουμε πως το μοντέλο μας εκπαιδεύεται καλά γιατί ελαχιστοποιούνται οι απώλειες.



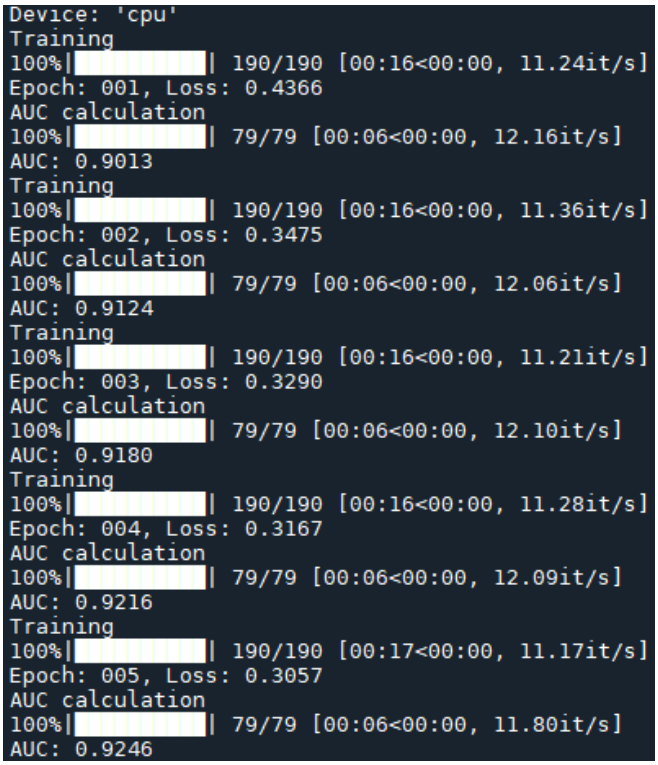
Εικόνα 21: 'AUC' για 5 epochs.

Στην 'Εικόνα 21' βλέπουμε το γράφημα για την 'AUC' για κάθε epoch με 5 epochs, στον άξονα Χ έχουμε κάθε epoch και στον Ψ την τιμή 'AUC', η τιμή της περιοχής αυξάνεται

άρα το μοντέλο μας κάνει όλο και πιο πετυχημένες προβλέψεις. Συμπεραίνουμε πως το μοντέλο μας εκπαιδεύεται καλά γιατί βελτιώνονται οι προβλέψεις.

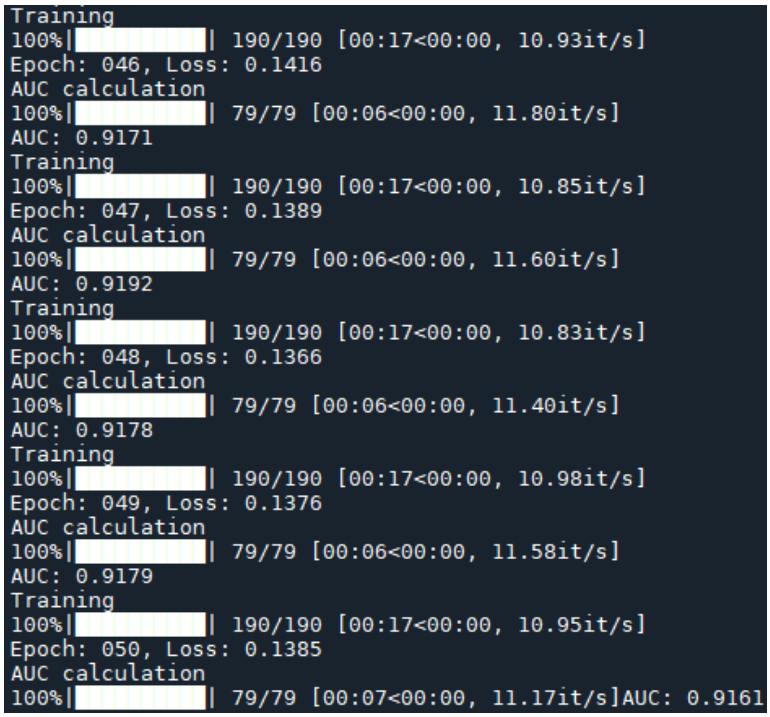
5.3 Δεύτερο πείραμα με 50 epochs

Στη συνέχεια δοκιμάζουμε να δεκαπλασιάσουμε τα epochs δηλαδή να τα κάνουμε 50 επίσης με επεξεργαστή, κατά τη διάρκεια της εκπαίδευσης λόγω του μεγάλου όγκου στην κονσόλα εδώ θα αποτυπώσουμε τα 5 πρώτα και 5 τελευταία epochs των 50 epoch που δοκιμάζουμε, ενώ εμείς εννοείται τα παρακολουθούσαμε όλα.



Εικόνα 22: Πως εμφανίζονται τα 5 πρώτα epochs των 50 στην κονσόλα.

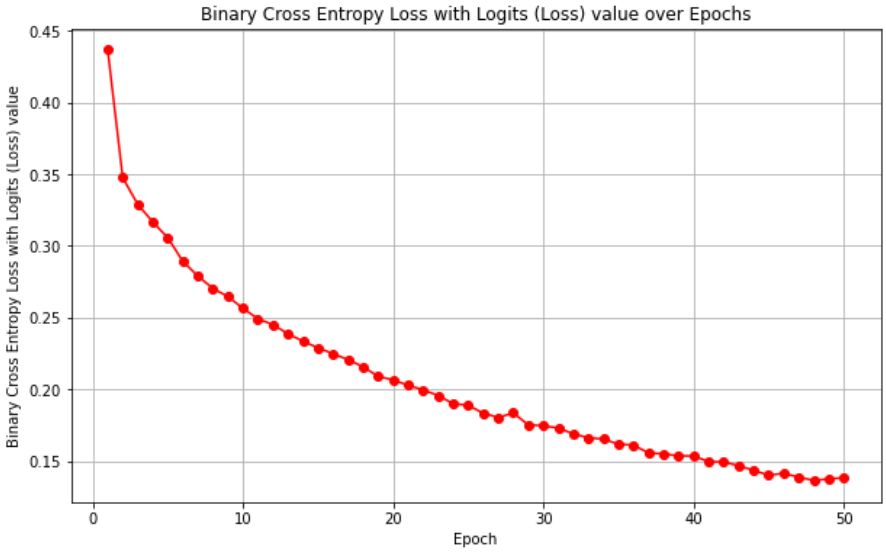
Στην κονσόλα μας τα πρώτα 5 epochs των 50 epochs εμφανίζονται όπως βλέπουμε στην ‘Εικόνα 22’.



Εικόνα 23: Πως εμφανίζονται τα 5 τελευταία epochs των 50 στην κονσόλα.

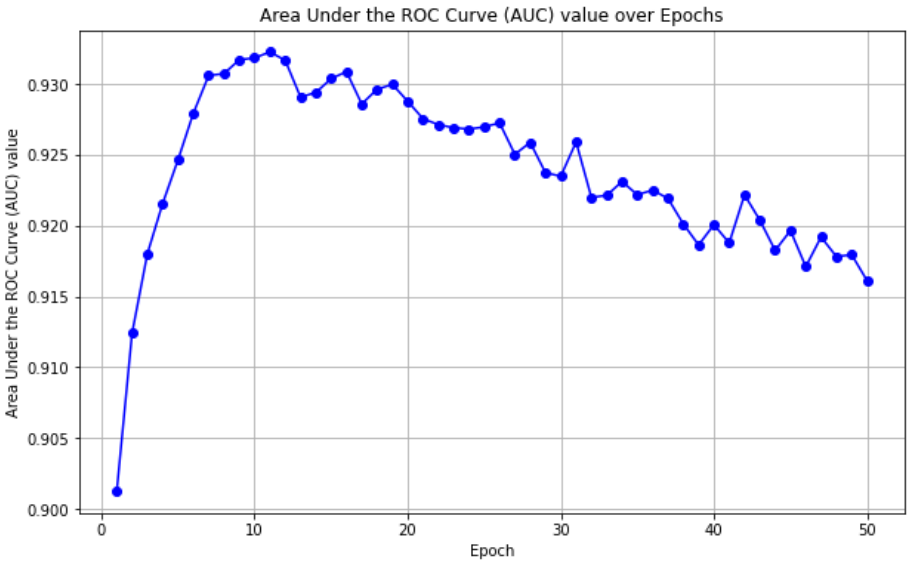
Και τα τελευταία 5 epochs των 50 epochs στην κονσόλα μας εμφανίζονται όπως βλέπουμε στην ‘Εικόνα 23’.

Στην ‘Εικόνα 22’ βλέπουμε στο πρώτο epoch πως η ‘Loss’ ξεκίνησε με 0.4366, επίσης η ‘AUC’ ξεκίνησε με 0.9013. Ενώ στην ‘Εικόνα 23’ βλέπουμε πως στο τελευταίο epoch η ‘Loss’ κατέληξε με τιμή 0.1385 και η ‘AUC’ με τιμή 0.9161. Άρα στη δεύτερη φάση του πειράματος μας σαν πρώτη εικόνα το μοντέλο μας για 50 epochs σίγουρα αποδίδει καλύτερα καθώς όσο αυξάνονται τα epochs θέλουμε τη ‘Loss’ να μειώνεται και την ‘AUC’ να αυξάνεται.



Εικόνα 24: 'Loss' για 50 epochs.

Στην 'Εικόνα 24' βλέπουμε το γράφημα για την 'Loss' για κάθε epoch με 50 epochs, στον άξονα Χ έχουμε κάθε epoch και στον Ψ την τιμή 'Loss', οι απώλειες μειώνονται άρα το μοντέλο μας προβλέπει όλο και καλύτερα. Συμπεραίνουμε πως το μοντέλο μας εκπαιδεύεται καλά γιατί ελαχιστοποιούνται οι απώλειες.



Εικόνα 25: 'AUC' για 50 epochs.

Στην 'Εικόνα 25' βλέπουμε το γράφημα για την 'AUC' για κάθε epoch με 50 epochs, στον άξονα Χ έχουμε κάθε epoch και στον Ψ την τιμή 'AUC', η τιμή 'AUC' αυξάνεται εως το ενδέκατο epoch και είναι καλό γιατί σημαίνει πως στο μοντέλο βελτιώνονται οι προβλέψεις, όμως μετά το ενδέκατο epoch αρχίζει να μειώνεται ως το τέλος αν και στο τέλος παραμένει καλύτερη από την τιμή εκκίνησης. Τελικά βλέπουμε μεν πως η τελική

‘AUC’ είναι μεγαλύτερη από την αρχική άρα θα μπορούσαμε να πούμε πως τελικά βελτιώθηκαν οι προβλέψεις, όμως επίσης παρατηρούμε πως συγκεκριμένα οι προβλέψεις κορυφώθηκαν στο ενδέκατο epoch ενώ μετά ώς το πεντηκοστό epoch παρουσίασαν τάση χειροτέρευσης αν και ακόμα παρέμεναν καλύτερες από το ξεκίνημα. Συμπεραίνουμε πως είτε το σύστημα μας αποδίδει τέλεια ως τα έντεκα epochs και δε χρειάζονται άλλα epochs είτε συντρέχει κάποιο πρόβλημα ώστε να χειροτερεύσουν μετά οι προβλέψεις όπως πχ αναφορικά είδαμε προηγουμένως αλγοριθμική προκατάληψη, εξηγήσιμη AI ή υπερπροσαρμογή.

5.4 Ερμηνεία αποτελεσμάτων

Είδαμε πως και για 5 και για 50 epochs η ‘Loss’ μειωνόταν άρα βελτιωνόταν γιατί ελαχιστοποιούσε τις απώλειες με παρόμοιο ρυθμό. Όμως στην ‘AUC’ είδαμε μια διαφορά μεταξύ 5 και 50 epochs. Ενώ και στις 2 περιπτώσεις τελικά μπορούμε να πούμε πως βελτιώθηκαν συγκριτικά μεταξύ αρχής και τέλους οι προβλέψεις, αν δούμε στην πρώτη περίπτωση των 5 epochs συνέχεια βελτιωνόταν, ενώ στη δεύτερη περίπτωση των 50 epochs κορυφώθηκε στα 11 epochs και μετά άρχισε να χειροτερεύει, για αυτό μπορεί να ευθύνονται όπως είδαμε νωρίτερα η αλγοριθμική προκατάληψη, η επεξηγήσιμη τεχνητή νοημοσύνη ή η υπερπροσαρμογή. Βέβαια μπορούμε και στα 5 και στα 50 epochs να την χαρακτηρίσουμε ‘εξαιρετική’ καθώς παραμένει πάνω από 0.9. [50]

Επιλέξαμε 5 epochs που θεωρούνται λίγα αλλά γρήγορα και 50 που είναι περισσότερα αλλά αυξήθηκε αναλογικά και η ώρα εκπαίδευσης. Καθώς και στις δύο περιπτώσεις του πειράματος είχαμε ‘εξαιρετικά’ αποτελέσματα θα επιλέγαμε ιδανικά την πρώτη περίπτωση καθώς κερδίζουμε χρόνο ενώ τα αποτελέσματα παραμένουν ‘εξαιρετικά’.

ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή τη διπλωματική εξερευνήσαμε τη θεωρία των γράφων στην οποία είδαμε τους τύπους τους, τον τρόπο που αποτυπώνονται, τα πεδία των εφαρμογών τους γενικά και τις τεχνικές μηχανικής μάθησης σε δεδομένα γράφων ειδικά, όπως και τα προβλήματα κατά την εφαρμογή τους. Είδαμε πόσο πολύ χρησιμεύει σε δεδομένα προτίμησης όπως συστήματα συστάσεων ταινιών. Επίσης αντίστοιχα είδαμε και την μηχανική μάθηση εξετάζοντας τις μορφές της, τα μοντέλα της, τα πεδία εφαρμογών της γενικά και την εφαρμογή της σε δεδομένα γράφων ειδικότερα, όπως και τα προβλήματα κατά την εφαρμογή της. Αντίστοιχα καταλάβαμε γιατί τα νευρωνικά δίκτυα γράφων είναι από τα καλύτερα για εκπαίδευση πάνω σε δεδομένα ταινιών. Τελικά για να τα καταλάβουμε έμπρακτα κάναμε ένα πείραμα με το σύνολο δεδομένων MovieLens, πρώτα κάναμε μια διερευνητική ανάλυση βλέποντας την ποσοσταία κατανομή των βαθμολογιών των χρηστών σε ταινίες και διαπιστώσαμε πως λίγοι βαθμολογούν χαμηλά. Έπειτα το μετατρέψουμε σε ετερογενή μη-κατευθυνόμενο γράφο που βοηθά πάρα πολύ σε συστήματα συστάσεων επειδή επιτρέπει το πέρασμα πληροφορίας και προς τις δύο πλευρές. Τελικά τον χρησιμοποιήσαμε σε σύστημα μηχανικής μάθησης τύπου νευρωνικού δικτύου γράφου για να κάνει προβλέψεις με 5 και 50 epochs και συγκρίναμε τα αποτελέσματα της κάθε περίπτωσης με ‘Loss’ και ‘AUC’. Παρατηρήσαμε πως έχει ‘εξαιρετικά’ αποτελέσματα και στις δύο περιπτώσεις. Βέβαια μπορούμε να πούμε πως για το σύνολο δεδομένων μας μόλις στο 11ο epoch έδινε ήδη ‘εξαιρετικές’ προβλέψεις και δε χρειαζόταν να σπαταλήσουμε περισσότερο χρόνο.

Ένα σύστημα συστάσεων ταινιών που χρησιμοποιεί νευρωνικό δίκτυο γράφου αποδίδει ήδη καλά, γιατί όμως όχι και καλύτερα; Μελλοντικά με τη βελτίωση των τεχνολογιών μπορεί να δούμε εξειδικευμένους αλγορίθμους οι οποίοι βγάζουν ακόμα πιο πετυχημένα αποτελέσματα βάση των αναγκών των χρηστών, ακόμα και σε συντομότερο χρονικό διάστημα, ή ακόμα και με μικρότερα δείγματα.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός όρος
Graph	Γράφος
Vertice	Κορυφή
Node	Κόμβος
Edge	Ακμή
Loop	Βρόχος
Node embedding	Ενσωμάτωση κόμβου
Machine learning	Μηχανική μάθηση
Artificial intelligence	Τεχνητή νοημοσύνη
Neural network	Νευρωνικό δίκτυο
Exploratory data analysis	Διερευνητική ανάλυση δεδομένων
Supervised learning	Εποπτευόμενη μάθηση
Unsupervised learning	Μη-εποπτευόμενη μάθηση
Label	Ετικέτα
Feature	Χαρακτηριστικό
Semi-supervised learning	Ημι-εποπτευόμενη μάθηση
Reinforced learning	Ενισχυτική μάθηση
Clustering	Συσταδοποίηση
Density estimation	Εκτίμηση πυκνότητας
Probability density function	Συνάρτηση πιθανότητας πυκνότητας
Markov decision process	Διαδικασία Απόφασης Μάρκοβ
Classification	Ταξινόμηση
Regression	Παλινδρόμηση
K-nearest neighbors	Κ-κοντινότεροι γείτονες
Support vector machines	Μηχανές διανυσμάτων στήριξης
Neural network regression	Παλινδρόμηση νευρωνικού δικτύου
Explainable AI	Επεξηγήσιμη τεχνητή νοημοσύνη
Overfitting	Υπερπροσάρμογη
Training	Εκπαίδευση
Validation	Επικύρωση
Testing	Δοκιμή
Indicator-dummy-variable	Μεταβλητή δείκτη
Heterogeneous graph	Ετερογενής γράφος
Mapping	Χαρτογράφηση
Edge indice	Δείκτης ακμής
Data loader	Φορτωτής δεδομένων
Mini-batch	Μίνι-παρτίδα
Classifier	Ταξινομητής
Dot product	Εσωτερικό γινόμενο
Binary cross entropy loss with logits	Δυαδική απώλεια διασταυρούμενης εντροπίας με logits
Area under the ROC curve	Περιοχή κάτω από την καμπύλη ROC

ΣΥΝΤΜΗΣΕΙΣ–ΑΡΚΤΙΚΟΛΕΞΑ–ΑΚΡΩΝΥΜΙΑ

Συντμήσεις	Πλήρης ανάπτυξη ονομασιών
G	Graph
V	Vertice
E	Edge
φ:E	Συνάρτηση πρόσπτωσης
ML	Machine Learning
AI	Artificial Intelligence
MDP	Markov Decision Process
SVM	Support Vector Machine
XAI	Explainable AI
SAGEConv	GraphSAGE convolutional layers
Loss	Binary Cross Entropy Loss with Logits
AUC	Area Under the ROC Curve

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

1. Bitnine.net, 2023. https://bitnine.net/blog-graph-database/graph-based-recommendation-system (accessed Jan. 10, 2024).
2. “Machine Learning in Recommendation Systems: an Overview,” www.itransition.com. https://www.itransition.com/machine-learning/recommendation-systems (accessed Jan. 10, 2024).
3. Lists, Decisions and Graphs. S. Gill Williamson.
4. Lists, Decisions and Graphs. S. Gill Williamson.
5. Lists, Decisions and Graphs. S. Gill Williamson.
6. Lists, Decisions and Graphs. S. Gill Williamson.
7. J. V. Kepner and J. R. Gilbert, Graph algorithms in the language of linear algebra. Philadelphia: Society For Industrial And Applied Mathematics, 2011.
8. G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, “Algorithms for drawing graphs: an annotated bibliography,” Computational Geometry, vol. 4, no. 5, pp. 235–282, Oct. 1994, doi: https://doi.org/10.1016/0925-7721(94)00014-x (accessed Jan. 10, 2024).
9. P. Shah et al., “Characterizing the role of the structural connectome in seizure dynamics,” Brain, vol. 142, no. 7, pp. 1955–1972, May 2019, doi: https://doi.org/10.1093/brain/awz125 (accessed Jan. 10, 2024).
10. F. Harary and E. M. Palmer, Graphical Enumeration. Elsevier, 2014.
11. “IEEE Xplore Full-Text PDF”:, ieeexplore.ieee.org. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9244647 (accessed Jan. 10, 2024).
12. C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.
13. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. New Jersey: Pearson, 2010.
14. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of machine learning. Cambridge, Massachusetts: The Mit Press, 2018.
15. B. Settles, “Active Learning Literature Survey,” 2009. Available: https://research.cs.wisc.edu/techreports/2009/TR1648.pdf (accessed Jan. 10, 2024).
16. Z. Keita, “Classification in Machine Learning: A Guide for Beginners,” www.datacamp.com, Sep. 2022. https://www.datacamp.com/blog/classification-machine-learning (accessed Jan. 10, 2024).
17. D. Castillo, “Machine Learning Regression Explained,” Seldon, Oct. 29, 2021. https://www.seldon.io/machine-learning-regression-explained (accessed Jan. 10, 2024).
18. A. B. Tucker, Computer Science Handbook. CRC Press, 2004.
19. “Weak Supervision: The New Programming Paradigm for Machine Learning · Stanford DAWN,” Stanford.edu, Jul. 15, 2017. https://dawn.cs.stanford.edu/2017/07/16/weak-supervision/ (accessed Jan. 10, 2024).
20. M. Wiering and Martijn Otterlo, Reinforcement Learning : State-of-the-Art. Berlin: Springer, 2012.
21. “Machine Learning Models: What They Are and How to Build Them,” Coursera. https://www.coursera.org/articles/machine-learning-models (accessed Jan. 10, 2024).

22. "Classification in Machine Learning | The Best Classification Models," Simplilearn.com. https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning (accessed Jan. 10, 2024).

23. "Understanding Logistic Regression," GeeksforGeeks, May 09, 2017. https://www.geeksforgeeks.org/understanding-logistic-regression/ (accessed Jan. 10, 2024).

24. N. Kumar, "Naive Bayes Classifiers - GeeksforGeeks," GeeksforGeeks, Jan. 14, 2019. https://www.geeksforgeeks.org/naive-bayes-classifiers/ (accessed Jan. 10, 2024).

25. GeeksForGeeks, "Decision Tree - GeeksforGeeks," GeeksforGeeks, Oct. 16, 2017. https://www.geeksforgeeks.org/decision-tree/ (accessed Jan. 10, 2024).

26. "Wayback Machine," Archive.org, 2011. https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf (accessed Jan. 10, 2024).

27. T. Hastie, J. Friedman, and R. Tibshirani, The Elements of Statistical Learning : Data Mining, Inference, and Prediction. New York, Ny Springer New York Imprint: Springer, 2001.

28. Wikipedia Contributors, "Support-vector machine," Wikipedia, Apr. 20, 2019. https://en.wikipedia.org/wiki/Support_vector_machine (accessed Jan. 10, 2024).

29. D. Castillo, "Machine Learning Regression Explained," Seldon, Oct. 29, 2021. https://www.seldon.io/machine-learning-regression-explained (accessed Jan. 10, 2024).

30. X. Yan and X. Su, Linear Regression Analysis: Theory And Computing. Singapore, Sg World Scientific Publishing Company, 2009.

31. D. E. Hilt and D. W. Seegrist, Ridge, a computer program for calculating ridge regression estimates. 1977.

32. M. Studer, G. Ritschard, A. Gabadinho, and N. S. Müller, "Discrepancy Analysis of State Sequences," Sociological Methods & Research, vol. 40, no. 3, pp. 471–510, Aug. 2011, doi: https://doi.org/10.1177/0049124111415372. (accessed Jan. 10, 2024).

33. A. Dutta, "Random Forest Regression in Python - GeeksforGeeks," GeeksforGeeks, Jun. 14, 2019. https://www.geeksforgeeks.org/random-forest-regression-in-python/ (accessed Jan. 10, 2024).

34. A. Teixeira-Pinto, 2 K-nearest Neighbours Regression | Machine Learning for Biostatistics. Available: https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html (accessed Jan. 10, 2024).

35. "Regression Analysis Using Artificial Neural Networks," Analytics Vidhya, Aug. 16, 2021. https://www.analyticsvidhya.com/blog/2021/08/a-walk-through-of-regression-analysis-using-artificial-neural-networks-in-tensorflow/ (accessed Jan. 10, 2024).

36. "BellKor Home Page," web.archive.org, Nov. 10, 2015. https://web.archive.org/web/20151110062742/http://www2.research.att.com/~volinsky/netflix/ (accessed Jan. 10, 2024).

37. "The Netflix Tech Blog: Netflix Recommendations: Beyond the 5 stars (Part 1)," web.archive.org, May 31, 2016. https://web.archive.org/web/20160531002916/http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html (accessed Jan. 10, 2024).

38. "9 Reasons why your machine learning project will fail," KDnuggets. https://www.kdnuggets.com/2018/07/why-machine-learning-project-fail.html (accessed Jan. 10, 2024).

39. C. Ross and I. Swetlitz, "IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show," STAT, Jul. 25, 2018.

https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/ (accessed Jan. 10, 2024).

40. D. Castelvecchi, “Can we open the black box of AI?,” Nature, vol. 538, no. 7623, pp. 20–23, Oct. 2016, doi: https://doi.org/10.1038/538020a (accessed Jan. 10, 2024).

41. Wikipedia Contributors, “Overfitting,” Wikipedia, Feb. 23, 2019. https://en.wikipedia.org/wiki/Overfitting (accessed Jan. 10, 2024).

42. “6 Reasons Why Is Python Used for Machine Learning,” New Horizons. https://www.newhorizons.com/resources/blog/why-is-python-used-for-machine-learning (accessed Jan. 10, 2024).

43. “Free Download,” Anaconda. https://www.anaconda.com/download (accessed Jan. 10, 2024).

44. “MovieLens 100K Dataset,” GroupLens, Sep. 23, 2015. https://grouplens.org/datasets/movielens/100k/ (accessed Jan. 10, 2024).

45. T. Seidakhmetov, “Graph Neural Network based Movie Recommender System,” Stanford CS224W GraphML Tutorials, Feb. 09, 2022. https://medium.com/stanford-cs224w/graph-neural-network-based-movie-recommender-system-5876b9686df3 (accessed Jan. 10, 2024).

46. T. by Abby, “Dummy Variables in Machine Learning.,” Medium, Sep. 30, 2021. https://techynotes.medium.com/dummy-variables-in-machine-learning-b3991367bd59 (accessed Jan. 10, 2024).

47. “torch_geometric.nn.conv.SAGEConv — pytorch_geometric documentation,” pytorch-geometric.readthedocs.io. https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.SAGEConv.html#torch_geometric.nn.conv.SAGEConv (accessed Jan. 10, 2024).

48. “BCEWithLogitsLoss — PyTorch 1.6.0 documentation,” pytorch.org. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html#torch.nn.BCEWithLogitsLoss (accessed Jan. 10, 2024).

49. “torcheval.metrics.BinaryAUROC — TorchEval main documentation,” pytorch.org. https://pytorch.org/torcheval/stable/generated/torcheval.metrics.BinaryAUROC.html (accessed Jan. 10, 2024).

50. D. W. Hosmer, S. Lemeshow, and I. Netlibrary, Applied logistic regression. New York: Wiley, 2000.

51. O. Hussein, “Graph Neural Networks Series | Part 3 | Node embedding,” The Modern Scientist, May 16, 2023. https://medium.com/the-modern-scientist/graph-neural-networks-series-part-3-node-embedding-36613cc967d5 (accessed Jan. 31, 2024).

52. Z. Blumenfeld, “Graph Machine Learning: An Overview,” Medium, Apr. 04, 2023. https://towardsdatascience.com/graph-machine-learning-an-overview-c996e53fab90 (accessed Jan. 31, 2024).

53. sav206436, “sav206436/FinalCodeDiplomatiki2143,” GitHub, Jan. 31, 2024. https://github.com/sav206436/FinalCodeDiplomatiki2143.git (accessed Jan. 31, 2024).

54. “Movielens - Movie Recommendation using GNN - PyG,” kaggle.com. https://www.kaggle.com/code/akkefa/movielens-movie-recommendation-using-gnn-pyg (accessed Feb. 3, 2024).

55. HI2Rec: Exploring knowledge in heterogeneous information for movie ... Available at: https://www.researchgate.net/publication/331515560_HI2Rec_Exploring_Knowledge_in_Heterogeneous_Information_for_Movie_Recommendation (accessed Feb. 2, 2024).

56. Wikipedia Contributors, “Graph theory,” Wikipedia, Apr. 29, 2019. https://en.wikipedia.org/wiki/Graph_theory (accessed Feb. 02, 2024).

57. Wikipedia Contributors, “Graph theory,” Wikipedia, Apr. 29, 2019. https://en.wikipedia.org/wiki/Graph_theory (accessed Feb. 02, 2024).

58. “When to use a Pie chart vs a Bar graph?,” piechartmaker.co. https://piechartmaker.co/when-to-use-pie-chart-vs-bar-graph (accessed Feb. 02, 2024).

59. “Home,” notegraph.com. https://notegraph.com/desktop-app/ (accessed Feb. 02, 2024).

60. “Introduction to Node Embedding,” memgraph.com. https://memgraph.com/blog/introduction-to-node-embedding (accessed Feb. 2, 2024).

61. “Supervised Learning Algorithm in Machine Learning,” TechVidvan, Jul. 11, 2020. https://techvidvan.com/tutorials/supervised-learning/ (accessed Feb. 2, 2024).

62. “Unsupervised Learning - Machine Learning Algorithms,” TechVidvan, Jul. 11, 2020. https://techvidvan.com/tutorials/unsupervised-learning/ (accessed Feb. 2, 2024).

63. C. Atten, “The Ultimate Beginner Guide of Semi-Supervised Learning,” Medium, Sep. 25, 2023. https://medium.datadriveninvestor.com/the-ultimate-beginner-guide-of-semi-supervised-learning-3bd11cb19835 (accessed Feb. 2, 2024).

64. “Reinforcement Learning Algorithms and Applications,” TechVidvan, Aug. 01, 2020. https://techvidvan.com/tutorials/reinforcement-learning/ (accessed Feb. 2, 2024).

65. J. Terra, “Regression vs. Classification in Machine Learning for Beginners | Simplilearn,” Simplilearn.com, Apr. 28, 2022. https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article (accessed Feb. 02, 2024).

66. R. Io, “Supervised vs Unsupervised Learning: Key Differences,” Medium, Oct. 03, 2019. https://medium.com/@recrosoft.io/supervised-vs-unsupervised-learning-key-differences-cdd46206cdcb (accessed Feb. 02, 2024).