ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

«Μελέτη και σύγκριση αλγορίθμων και τεχνικών κατηγοριοποίησης πολλαπλών ετικετών»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Αθηνάς Μαχαιρά

Επιβλέπων : Στέφανος Ουγιάρογλου
Επίκουρος Καθηγητής, ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Μάρτιος 2024

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Αθηνάς Μαχαιρά

Επιβλέπων : Στέφανος Ουγιάρογλου
Επίκουρος Καθηγητής ΔΙ.ΠΑ.Ε.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 2 Μαρτίου 2024.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Στέφανος Ουγιάρογλου	Περικλής Χατζημίσιος	Στάθης Κασδερίδης
Επίκουρος Καθηγητής ΔΙ.ΠΑ.Ε.	Καθηγητής ΔΙ.ΠΑ.Ε.	Επισκέπτης Καθηγητής ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Μάρτιος 2024

(Υπογραφή)

.....

Αθηνά Μαχαιρά

Μηχανικών Πληροφορικής Τ.Ε.

Περίληψη

Η παρούσα διπλωματική εργασία επικεντρώνεται στη μελέτη και σύγκριση αλγορίθμων και τεχνικών κατηγοριοποίησης πολλαπλών ετικετών, με στόχο την κατανόηση των μετρικών απόδοσής τους. Αναλύεται η έννοια της κατηγοριοποίησης πολλαπλών ετικετών, επισημαίνοντας τη σημασία της στην επεξεργασία και ανάλυση περίπλοκων συνόλων δεδομένων. Εξετάζονται διάφορες προσεγγίσεις, όπως οι τεχνικές μετασχηματισμού προβλήματος και ειδικοί αλγόριθμοι για την επίλυση τέτοιων προβλημάτων.

Επίσης, παρουσιάζεται η βιβλιοθήκη Scikit-multilearn ως ένα εργαλείο για την πρακτική υλοποίηση αυτών των τεχνικών. Η εργασία συμπεριλαμβάνει μια πειραματική μελέτη όπου συγκρίνονται διάφοροι αλγόριθμοι με βάση πραγματικά σύνολα δεδομένων, αξιολογώντας την απόδοσή τους σε διαφορετικές παραμέτρους. Τέλος, η εργασία καταλήγει σε συμπεράσματα σχετικά με την εφαρμοσιμότητα των διαφόρων τεχνικών και προτείνει μελλοντικές κατευθύνσεις για την έρευνα στον τομέα της κατηγοριοποίησης πολλαπλών ετικετών.

Λέξεις Κλειδιά:<< Κατηγοριοποίηση πολλαπλών ετικετών, Αλγόριθμοι κατηγοριοποίησης, Τεχνικές μετασχηματισμού προβλήματος, Επεξεργασία δεδομένων, Ανάλυση δεδομένων, Scikit-multilearn, Πειραματική μελέτη, Απόδοση αλγορίθμων, Μετρικές απόδοσης>>

Abstract

This thesis focuses on the study and comparison of algorithms and techniques for multi-label classification, aiming to understand their performance metrics. The concept of multi-label classification is analyzed, highlighting its importance in processing and analyzing complex data sets. Various approaches are examined, such as problem transformation techniques and specific algorithms for solving such problems. Additionally, the Scikit-multilearn library is presented as a tool for the practical implementation of these techniques. The work includes an experimental study where various algorithms are compared based on real data sets, evaluating their performance on different parameters. Finally, the thesis concludes with findings regarding the applicability of the various techniques and suggests future directions for research in the field of multi-label classification.

Keywords:<< Multi-label classification, Algorithms, Performance metrics, Data processing, Data analysis, Problem transformation techniques, Scikit-multilearn, Experimental study, Evaluation>>

Πίνακας περιεχομένων

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

1

Εισαγωγή

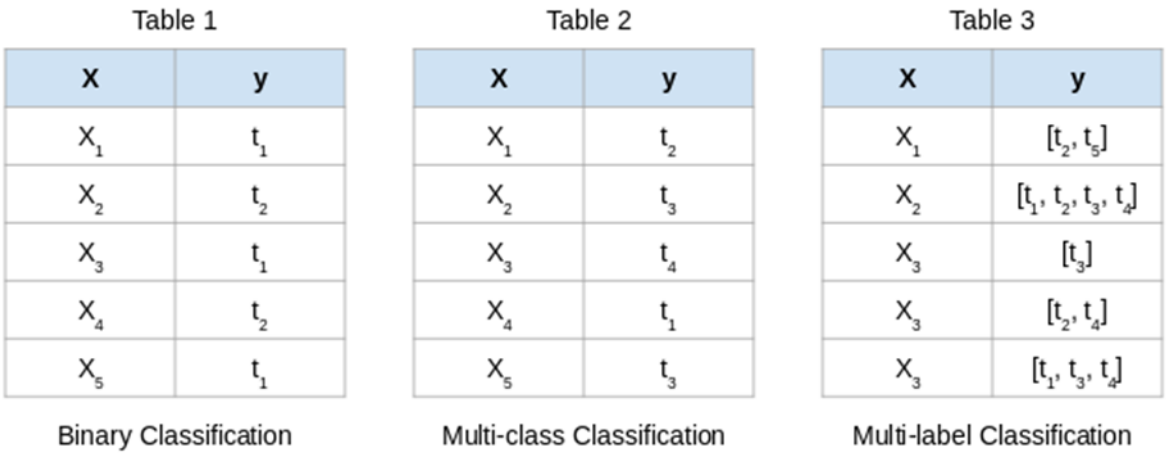
1.1 Κατηγοριοποίηση

Η εξέλιξη της τεχνολογίας και η αυξημένη διαθεσιμότητα των δεδομένων έχουν οδηγήσει σε σημαντικές προκλήσεις και ευκαιρίες στον τομέα της επεξεργασίας και ανάλυσης δεδομένων. Μια από τις πιο σημαντικές πτυχές στην επεξεργασία δεδομένων είναι η κατηγοριοποίηση, μια διαδικασία που στοχεύει στην αναγνώριση της κατηγορίας στην οποία ανήκει ένα δεδομένο ή ένα σύνολο δεδομένων. Η παραδοσιακή κατηγοριοποίηση, ωστόσο, αντιμετωπίζει προκλήσεις όταν έρχεται αντιμέτωπη με πολυπλοκότητες όπως τα προβλήματα πολλαπλών ετικετών.

Η κατηγοριοποίηση αποτελεί έναν από τους βασικούς πυλώνες της μηχανικής μάθησης και της επεξεργασίας δεδομένων. Αναφέρεται στη διαδικασία ταξινόμησης των δεδομένων σε διακριτές κατηγορίες ή κλάσεις, βάσει ενός συνόλου χαρακτηριστικών ή γνωρισμάτων. Ως εκ τούτου, βοηθά στην απλούστευση και στην κατανόηση μεγάλων συνόλων δεδομένων, καθιστώντας τα πιο διαχειρίσιμα και ερμηνεύσιμα. Στόχος της κατηγοριοποίησης είναι η ανάπτυξη μοντέλων πρόβλεψης που μπορούν να αναθέσουν νέα δεδομένα σε προκαθορισμένες κατηγορίες, βασιζόμενα στις πληροφορίες που έχουν αποκομιστεί κατά τη διάρκεια της εκπαίδευσης (training).

1.2 *Προβλήματα κατηγοριοποίησης*

Ανάλογα με τη φύση του προβλήματος και του συνόλου δεδομένων, η κατηγοριοποίηση μπορεί να διακριθεί σε διάφορους τύπους προβλημάτων, εκ των οποίων τα πιο συνηθισμένα είναι τα δυαδικά προβλήματα, τα προβλήματα πολλών κατηγοριών, και τα προβλήματα πολλαπλών ετικετών.



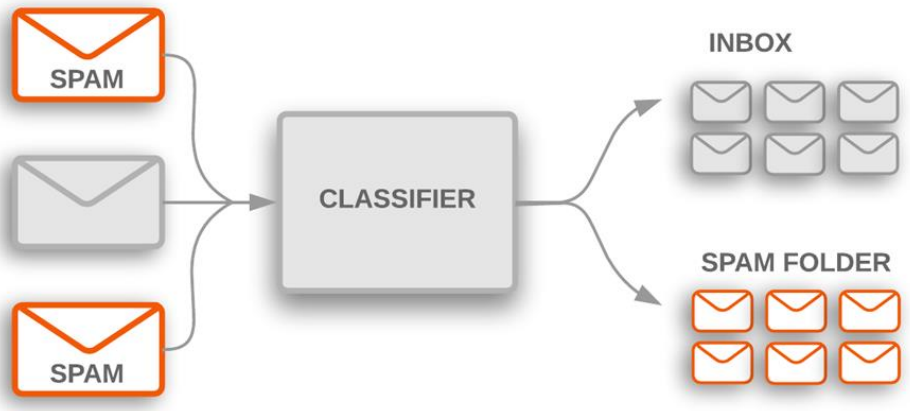
Σχήμα 1 Προβλήματα κατηγοριοποίησης Binary, Multi-class, Multi-label. [14]

Στο παραπάνω σχήμα όπου x τα δεδομένα εισόδου και y η κλάση που προβλέπεται από την ταξινόμηση. Στο Table 1 (binary classification) η κατηγοριοποίηση επιφέρει δύο κλάσεις t1 και t2 και αντιστοιχεί μία και μόνο σε κάθε δείγμα x. Μια μόνο κλάση αντιστοιχεί σε κάθε δείγμα x και στο Table 2 (multi-class classification) με περισσότερες διαθέσιμες επιλογές κλάσεων (t1, t2, t3, t4). Στο Table 3 (multi-label classification) είναι εμφανές ότι κάθε δείγμα x μπορεί να αντιστοιχεί σε παραπάνω από μία κλάσεις μεταξύ των t1, t2, t3, t4, t5.

1.2.1 *Δυαδικά προβλήματα*

Τα δυαδικά προβλήματα κατηγοριοποίησης αντιπροσωπεύουν την πιο βασική μορφή κατηγοριοποίησης, όπου το σύνολο δεδομένων μπορεί να διαχωριστεί σε δύο διακριτές κατηγορίες ή τάξεις, συχνά συμβολικά αναπαριστώμενες από τα δυο διαφορετικά αποτελέσματα που λαμβάνει ένας ταξινομητής: 0 ή 1, ναι ή όχι, θετικός ή αρνητικός.

Ένα βασικό παράδειγμα δυαδικής κατηγοριοποίησης είναι η ανίχνευση spam email. Σε αυτήν την περίπτωση, ο σκοπός είναι να καθοριστεί εάν ένα email είναι spam (κακόβουλο) ή όχι. Ανάλογα με τις λέξεις κλειδιά, το στυλ κειμένου και άλλες παραμέτρους που εξετάζονται στα emails, ο ταξινομητής θα λάβει μια απόφαση για την κατηγοριοποίησή τους.



Σχήμα 2 Δυαδική ταξινόμηση email σε spam [15]

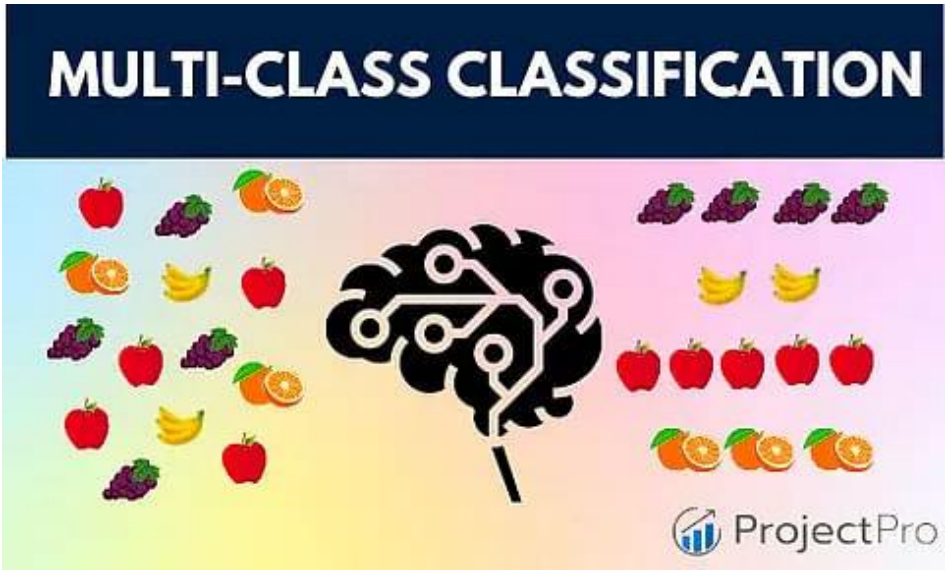
Άλλα παραδείγματα περιλαμβάνουν την ιατρική διάγνωση (παρουσία ή απουσία νόσου), την ανίχνευση πιστωτικής απάτης (νόμιμη ή μη συναλλαγή με πιστωτική κάρτα) και την ανίχνευση παραβίασης ψηφιακής ασφάλειας (κανονική ή ύποπτη ψηφιακή συμπεριφορά).

Τα δυαδικά προβλήματα κατηγοριοποίησης είναι ιδιαίτερα σημαντικά λόγω της ευρείας εφαρμογής τους σε πολλά πεδία και της σχετικά απλούστερης μεθοδολογίας ανάλυσης.

1.2.2 Προβλήματα πολλών κατηγοριών

Τα προβλήματα πολλών κατηγοριών, γνωστά και ως πολυκατηγορική κατηγοριοποίηση (multiclass classification), αφορούν την ταξινόμηση των δεδομένων σε μία από πολλές κατηγορίες. Αυτή η διαδικασία είναι περίπλοκη λόγω της ανάγκης να διαχειριστεί κανείς μεγάλο αριθμό διακριτών κατηγοριών και τις πιθανές αλληλεπιδράσεις μεταξύ τους.

Για παράδειγμα, ένα φρούτο μπορεί να καταταχθεί ως μήλο, πορτοκάλι, σταφύλι ή άλλη κατηγορία.



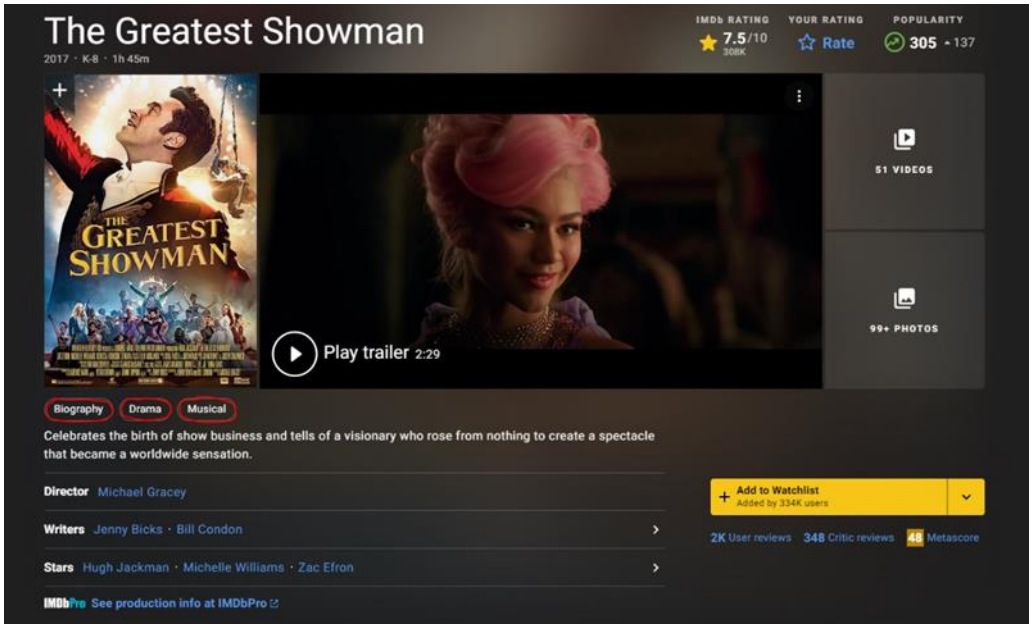
Σχήμα 3 Παράδειγμα ταξινόμησης φρούτων σε κατηγορίες [17]

Η πρόκληση εδώ περιλαμβάνει την κατανόηση και την εξαγωγή των χαρακτηριστικών που καθορίζουν την καταλληλότητα ενός δείγματος για κάθε μία από τις διαθέσιμες κατηγορίες, απαιτώντας εξελιγμένους αλγορίθμους και τεχνικές για την ορθότερη ταξινόμηση.

1.2.3 Προβλήματα πολλαπλών ετικετών

Τα προβλήματα πολλαπλών ετικετών εισάγουν μια ακόμη πιο πολύπλοκη διάσταση στην κατηγοριοποίηση. Σε αυτό το είδος προβλημάτων, κάθε παράδειγμα δεν ανατίθεται μόνο σε μία κατηγορία, αλλά μπορεί να σχετίζεται ταυτόχρονα με πολλαπλές κατηγορίες. Η πολυπλοκότητα εδώ προέρχεται από την ανάγκη να αναγνωριστούν και να προβλεφθούν οι σχέσεις μεταξύ των διαφόρων ετικετών και το πώς αυτές οι σχέσεις επηρεάζουν την ταξινόμηση ενός δεδομένου.

Ένα ενδεικτικό παράδειγμα που αναδεικνύει την πολυπλοκότητα αυτού του είδους της κατηγοριοποίησης είναι η ταξινόμηση της ταινίας "The Greatest Showman". Είναι μια ταινία που διαφεύγει την απλουστευτική κατηγοριοποίηση σε μία μόνο ετικέτα. Πρόκειται για ένα μουσικό δράμα που βασίζεται στη ζωή του P.T. Barnum, ιδρυτή του Barnum & Bailey Circus, και εξερευνά θέματα όπως η αγάπη, η φιλία, η αποδοχή και η επιδίωξη των ονείρων. Ως εκ τούτου, η ταινία μπορεί να καταταχθεί σε πολλαπλές κατηγορίες: μουσική, δράμα, βιογραφία, και ίσως ακόμη και ως οικογενειακή ταινία.



Σχήμα 4 Πολλαπλές ετικέτες σχετικά με το είδος της ταινίας "The Greatest Showman" [16]

Η πρόκληση στην κατηγοριοποίηση τέτοιων πολυδιάστατων ταινιών απαιτεί την ανάπτυξη αλγορίθμων και μεθοδολογιών που μπορούν να αναλύσουν και να ερμηνεύσουν το πλούσιο περιεχόμενο και τα θεματικά στοιχεία της ταινίας. Οι αλγόριθμοι πρέπει να είναι σε θέση να αντλούν χαρακτηριστικά από το σενάριο, τη σκηνοθεσία, τη μουσική, τις ερμηνείες, και άλλα στοιχεία, ώστε να αποφασίσουν πώς μια ταινία όπως το "The Greatest Showman" συνδυάζει τα διάφορα είδη. Επιπλέον, η ικανότητα να αναγνωρίζουν τις αλληλεπιδράσεις και τις σχέσεις μεταξύ των διαφόρων ετικετών μπορεί να προσφέρει μια πιο συνολική και διεισδυτική ανάλυση, αποκαλύπτοντας την πολυδιάστατη φύση του κινηματογραφικού έργου.

1.3 Κίνητρο

Η ανάπτυξη και η εξέλιξη της κατηγοριοποίησης πολλαπλών ετικετών στον τομέα της επιστήμης δεδομένων και της μηχανικής μάθησης υπαγορεύεται από την αυξανόμενη πολυπλοκότητα και ποικιλομορφία των διαθέσιμων δεδομένων. Στο σύγχρονο ψηφιακό περιβάλλον, όπου τα δεδομένα παράγονται, συλλέγονται και αναλύονται σε ασύλληπτους ρυθμούς, η ανάγκη για πιο εξελιγμένες τεχνικές κατηγοριοποίησης γίνεται επιτακτική. Η παραδοσιακή μονοδιάστατη κατηγοριοποίηση συχνά αδυνατεί να ανταποκριθεί στην πρόκληση της αντιστοίχισης των δεδομένων σε πολλαπλές, διακριτές κατηγορίες, αναγκάζοντας τους ερευνητές και τους επαγγελματίες να εξερευνήσουν νέους δρόμους και προσεγγίσεις. Η κατηγοριοποίηση πολλαπλών ετικετών, με την ικανότητα να διαχειρίζεται πολυδιάστατες ετικέτες για κάθε δείγμα δεδομένων, προσφέρει μια πολλά υποσχόμενη λύση

σε αυτή την πρόκληση, επιτρέποντας πιο λεπτομερείς και ακριβείς αναλύσεις. Το κίνητρο αυτής της έρευνας ενισχύεται από την ανάγκη να ξεπεραστούν οι προκλήσεις που ανακύπτουν στην κατηγοριοποίηση πολλαπλών ετικετών, οδηγώντας στην ανάπτυξη της επόμενης ενότητας όπου αναλύεται η συνεισφορά της τρέχουσας εργασίας στην επίλυση των εν λόγω προβλημάτων.

1.4 Συνεισφορά

Η προκειμένη διπλωματική εργασία φιλοδοξεί να εμβαθύνει στον τομέα της κατηγοριοποίησης πολλαπλών ετικετών, προσφέροντας μια ολοκληρωμένη και ενδελεχή αξιολόγηση των διάφορων υφιστάμενων αλγορίθμων και τεχνικών. Μέσα από αυτή την προσέγγιση, η εργασία αποσκοπεί να κατανοήσει πιο βαθιά την αποτελεσματικότητα, τα οφέλη και τις προκλήσεις που συνδέονται με κάθε μέθοδο. Αναγνωρίζοντας τα πλεονεκτήματα και τα μειονεκτήματα των διαφόρων προσεγγίσεων, η εργασία επιδιώκει να προτείνει βελτιώσεις και να διαμορφώσει νέες κατευθύνσεις για τη μελλοντική έρευνα στο πεδίο.

Μέσω της διενέργειας λεπτομερών πειραματικών αναλύσεων, αυτή η εργασία αποκτά πολύτιμες πρακτικές γνώσεις, αποκαλύπτοντας την πραγματική απόδοση των αλγορίθμων υπό διάφορες συνθήκες και σενάρια. Η αξιολόγηση αυτή προσδοκά να προσανατολίσει τους ερευνητές και τους επαγγελματίες σχετικά με την επιλογή της πιο κατάλληλης τεχνικής για συγκεκριμένες εφαρμογές, ενισχύοντας την ικανότητα να προσεγγίζουν πιο αποτελεσματικά τα πολύπλοκα προβλήματα κατηγοριοποίησης.

1.5 Οργάνωση της διπλωματικής

Η δομή της διπλωματικής εργασίας οργανώνεται ως εξής: Το Κεφάλαιο 2 αναλύει τις διάφορες τεχνικές και αλγορίθμους για την κατηγοριοποίηση πολλαπλών ετικετών, παρουσιάζοντας τις βασικές θεωρητικές αρχές και τις προκλήσεις τους. Το Κεφάλαιο 3 επικεντρώνεται στην παρουσίαση της βιβλιοθήκης Scikit-multilearn και στην ανάλυση των κυριότερων αλγορίθμων που παρέχει. Το Κεφάλαιο 4 περιλαμβάνει την πειραματική μελέτη, παρουσιάζοντας τα αποτελέσματα της σύγκρισης των αλγορίθμων και των τεχνικών. Τέλος, το Κεφάλαιο 5 παρουσιάζει τα συμπεράσματα της εργασίας και προτείνει μελλοντικές κατευθύνσεις για έρευνα.

2

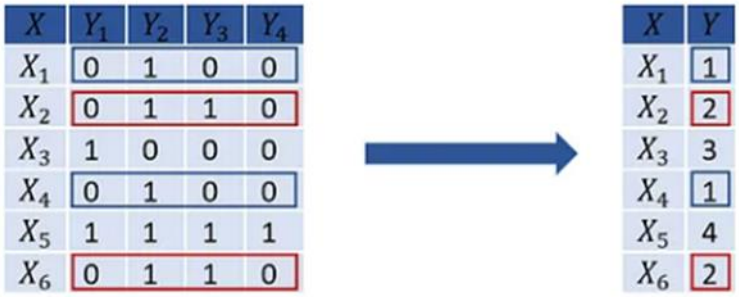
Αλγόριθμοι και Τεχνικές για προβλήματα πολλαπλών ετικετών

2.1 Τεχνικές μετασχηματισμού προβλήματος

Οι τεχνικές μετασχηματισμού προβλήματος αποτελούν θεμελιώδη προσέγγιση στην αντιμετώπιση προβλημάτων κατηγοριοποίησης πολλαπλών ετικετών στην επιστήμη δεδομένων και τη μηχανική μάθηση. Αυτές οι τεχνικές μετατρέπουν το αρχικό πολύπλοκο πρόβλημα σε ένα ή περισσότερα απλούστερα προβλήματα, τα οποία είναι πιο ευχερώς διαχειρίσιμα. Δύο από τις πιο διαδεδομένες τεχνικές είναι το Binary Relevance και το Label Powerset, οι οποίες προσφέρουν διαφορετικές προσεγγίσεις για την επίλυση του ίδιου προβλήματος.

2.1.1 Label Powerset

Η τεχνική Label Powerset προσεγγίζει το πρόβλημα μετατρέποντας το σε ένα πρόβλημα πολλαπλών κλάσεων κατηγοριοποίησης. Σε αυτή την περίπτωση, κάθε δυνατός συνδυασμός ετικετών που εμφανίζεται στο σύνολο δεδομένων θεωρείται ως μία ξεχωριστή "μεγάλη ετικέτα". Αυτό σημαίνει ότι ο ταξινομητής δεν πρέπει απλώς να αποφασίσει αν μια ετικέτα ανήκει ή όχι στο δείγμα, αλλά ποιος από τους πολλούς δυνατούς συνδυασμούς ετικετών αντιστοιχεί στο δείγμα. Αυτή η προσέγγιση λαμβάνει υπόψη τις σχέσεις μεταξύ των ετικετών, αλλά μπορεί να οδηγήσει σε μια εκρηκτική αύξηση του αριθμού των "μεγάλων ετικετών" που πρέπει να διαχειριστεί ο ταξινομητής, κάτι που καθιστά την προσέγγιση αυτή πιο περίπλοκη και απαιτητική σε υπολογιστικούς πόρους.

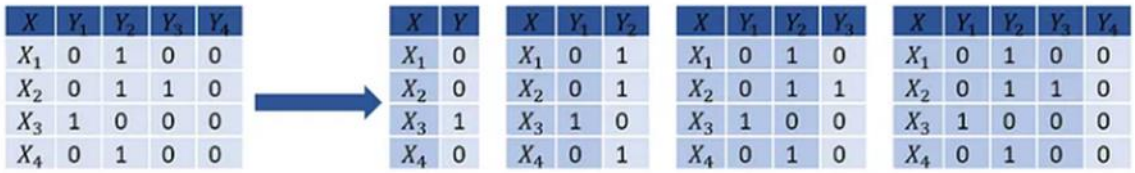


Σχήμα 5 Διαδικασία μετασχηματισμού δεδομένων πολλαπλών ετικετών με την τεχνική Label Powerset [24]

2.1.2 Classifier Chain

Η τεχνική Classifier Chain είναι μια μέθοδος για την πρόβλεψη πολλαπλών σχετιζόμενων ετικετών ή στόχων. Αυτή η τεχνική εφαρμόζεται συνήθως σε προβλήματα πολυετικετικής ταξινόμησης, όπου κάθε παράδειγμα μπορεί να ανήκει σε πολλές κατηγορίες ταυτόχρονα. Η βασική ιδέα πίσω από την τεχνική Classifier Chain είναι η δημιουργία μιας αλυσίδας από δυαδικούς ταξινομητές, όπου ο κάθε ταξινομητής εκπαιδεύεται να προβλέπει την παρουσία ή την απουσία μιας ετικέτας, λαμβάνοντας υπόψη όχι μόνο τα χαρακτηριστικά του παραδείγματος αλλά και τις προβλέψεις των προηγούμενων ταξινομητών στην αλυσίδα.

Η προσέγγιση αυτή επιτρέπει στους ταξινομητές να λαμβάνουν υπόψη τις σχέσεις μεταξύ των ετικετών, κάτι που μπορεί να βελτιώσει την ακρίβεια των προβλέψεων σε σύγκριση με τις ανεξάρτητες προσεγγίσεις, όπου κάθε ετικέτα προβλέπεται ανεξάρτητα από τις υπόλοιπες. Η σειρά των ετικετών στην αλυσίδα μπορεί να επηρεάσει την απόδοση του μοντέλου, και συνεπώς, η επιλογή μιας καλής σειράς αποτελεί σημαντικό μέρος της διαδικασίας εκπαίδευσης.

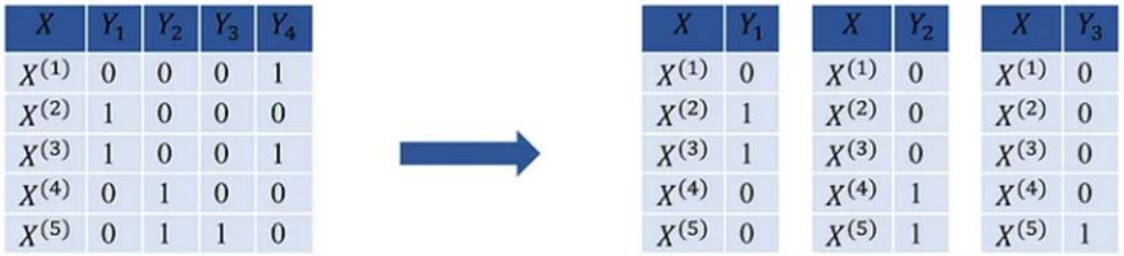


Σχήμα 6 Διαδικασία μετασχηματισμού δεδομένων πολλαπλών ετικετών με την τεχνική Classifier Chain [24]

2.1.3 Binary Relevance

Η τεχνική Binary Relevance προσεγγίζει το πρόβλημα κατηγοριοποίησης πολλαπλών ετικετών μέσω της δημιουργίας διακριτών δυαδικών ταξινομητών για κάθε ετικέτα. Αυτό σημαίνει ότι

για κάθε ετικέτα στο σύνολο δεδομένων, δημιουργείται ένας ανεξάρτητος ταξινομητής, ο οποίος αποφασίζει αν η ετικέτα αυτή ανήκει ή όχι στο δείγμα. Αυτή η προσέγγιση είναι σχετικά απλή και εύκολη στην υλοποίηση, αλλά μπορεί να παραβλέπει τις σχέσεις μεταξύ των ετικετών, καθώς κάθε ταξινομητής λειτουργεί ανεξάρτητα.



Σχήμα 7 Διαδικασία μετασχηματισμού δεδομένων πολλαπλών ετικετών με την τεχνική Binary Relevance [24]

2.1.3.1 k-Nearest Neighbors (kNN)

Ο αλγόριθμος k-Nearest Neighbors (kNN) αποτελεί μία από τις πιο απλές, και ταυτόχρονα ισχυρές, τεχνικές στην επιστήμη των δεδομένων. Βασίζεται στην αρχή ότι τα δείγματα με παρόμοια χαρακτηριστικά τείνουν να ανήκουν στην ίδια κατηγορία. Η κατηγοριοποίηση ενός νέου δείγματος πραγματοποιείται με βάση την κατηγοριοποίηση των k πλησιέστερων δειγμάτων του στο χώρο των χαρακτηριστικών.

2.1.3.2 Ενσωμάτωση της BR με τον kNN: BRkNNa και BRkNNb

Η ενσωμάτωση της τεχνικής Binary Relevance με τον αλγόριθμο kNN οδηγεί στις παραλλαγές BRkNNa και BRkNNb, προσφέροντας μια πιο δυναμική προσέγγιση για την κατηγοριοποίηση πολλαπλών ετικετών.

Στον BRkNNa [13], για κάθε ετικέτα, εκτελείται μια ανεξάρτητη διαδικασία kNN για να προσδιοριστεί αν αυτή πρέπει να αποδοθεί στο δείγμα ή όχι, βασιζόμενη αποκλειστικά στην παρουσία της ετικέτας στα k πλησιέστερα δείγματα. Αυτό παρέχει μια σαφή και άμεση μεθοδολογία, αλλά παραμένει περιορισμένο στην εξέταση των ετικετών ανεξάρτητα.

Στην περίπτωση του BRkNNb [13], η προσέγγιση είναι πιο εκλεπτυσμένη. Εκτός από την απλή καταμέτρηση των ετικετών στα k πλησιέστερα δείγματα, λαμβάνεται υπόψη και η απόσταση

αυτών των δειγμάτων από το δείγμα προς κατηγοριοποίηση. Αυτή η προσέγγιση αυξάνει την ακρίβεια της κατηγοριοποίησης λαμβάνοντας υπόψη την άμεση "γειτνίαση" των ετικετών σε σχέση με το δείγμα.

Καθώς οι BRkNNa και BRkNNb προσφέρουν διαφορετικές μεθοδολογίες για την επίλυση του ίδιου προβλήματος, η επιλογή μεταξύ τους εξαρτάται από τις συγκεκριμένες ανάγκες της εφαρμογής, τη φύση των δεδομένων και τους διαθέσιμους υπολογιστικούς πόρους.

2.2 Αλγόριθμοι για προβλήματα πολλαπλών ετικετών

Η πολυπλοκότητα που ενέχεται στην κατηγοριοποίηση πολλαπλών ετικετών απαιτεί την ανάπτυξη εξειδικευμένων μεθόδων που μπορούν να αναγνωρίζουν και να διαχειρίζονται τις αλληλεπιδράσεις και τις συσχετίσεις μεταξύ των ετικετών. Αυτό οδηγεί στην ανάγκη για αλγορίθμους που υπερβαίνουν τις παραδοσιακές μεθόδους και προσφέρουν πιο προηγμένες λύσεις.

2.2.1 MLkNN: Multi-Label k-Nearest Neighbors

Ο Multi-Label k-Nearest Neighbors (MLkNN) [2] είναι ένας αλγόριθμος βασισμένος στην τεχνική k-Nearest Neighbors (kNN), που προσαρμόζεται για να αντιμετωπίζει προβλήματα κατηγοριοποίησης πολλαπλών ετικετών. Αναπτύχθηκε για να λαμβάνει υπόψη την αλληλεπίδραση μεταξύ των ετικετών, προσπαθώντας να βελτιώσει την απόδοση σε σύνθετα σύνολα δεδομένων όπου κάθε δείγμα μπορεί να ανήκει σε πολλαπλές κατηγορίες ταυτόχρονα.

Η λειτουργία του MLkNN ξεκινά με τον προσδιορισμό των k πλησιέστερων γειτόνων για κάθε δείγμα εκπαίδευσης μέσω μιας διαδικασίας που μοιάζει με τον κλασσικό kNN. Στη συνέχεια, αντί να εφαρμόζει απλώς μια ψηφοφορία μεταξύ των ετικετών των πλησιέστερων γειτόνων, ο MLkNN χρησιμοποιεί μια στατιστική μέθοδο για να υπολογίσει την πιθανότητα εμφάνισης κάθε ετικέτας βάσει της παρουσίας ή της απουσίας της στους γείτονες. Αυτή η προσέγγιση επιτρέπει στον MLkNN να κάνει πιο ενημερωμένες προβλέψεις για τις ετικέτες που ανήκουν σε κάθε δείγμα, λαμβάνοντας υπόψη την πιθανότητα συν-εμφάνισης των ετικετών.

2.2.2 MLTSVM: Multi-Label Twin Support Vector Machine

Ο Multi-Label Twin Support Vector Machine (MLTSVM) [29] είναι μια επέκταση του τυπικού μοντέλου Support Vector Machine (SVM) για την αντιμετώπιση προβλημάτων κατηγοριοποίησης πολλαπλών ετικετών. Αυτός ο αλγόριθμος βασίζεται στην ιδέα του να κατασκευάζει ζεύγη από ταξινομητές SVM για κάθε ετικέτα, όπου ο ένας ταξινομητής εστιάζει στην ελαχιστοποίηση του σφάλματος για τα δείγματα που ανήκουν στην κατηγορία και ο άλλος στα δείγματα που δεν ανήκουν. Αυτό το προσεγγιστικό σχήμα επιτρέπει στον MLTSVM να είναι πιο ευέλικτος και αποτελεσματικός στην αντιμετώπιση της πολυπλοκότητας των πολλαπλών ετικετών, προσφέροντας μια ισορροπημένη προσέγγιση μεταξύ της ακρίβειας της κατηγοριοποίησης και της γενίκευσης.

2.2.3 MLARAM: Multi-Label Adaptive Resonance Associative Map

Ο Multi-Label Adaptive Resonance Associative Map (MLARAM) [29] είναι ένας αλγόριθμος που βασίζεται σε νευρωνικά δίκτυα και στη θεωρία των Adaptive Resonance Theory (ART) δικτύων, προσαρμοσμένος για την κατηγοριοποίηση πολλαπλών ετικετών. Ο MLARAM εφαρμόζει έναν μηχανισμό αντιστοίχισης βασισμένο στη συχνότητα ανταπόκρισης, ο οποίος δίνει τη δυνατότητα στο μοντέλο να ανακαλύπτει και να ταξινομεί ειδικά πρότυπα μέσα στο σύνολο δεδομένων. Αυτός ο μηχανισμός επιτρέπει στον αλγόριθμο να διατηρεί υψηλή αποδοτικότητα και ευελιξία, προσαρμοζόμενος σε νέα δεδομένα χωρίς να υποστεί απώλεια απόδοσης λόγω "κορεσμού" του μοντέλου, κάτι που είναι ιδιαίτερα σημαντικό σε πολυετικετικά σύνολα δεδομένων.

3

Η Βιβλιοθήκη Scikit-multilearn

3.1 Η βιβλιοθήκη

Στο πλαίσιο της επιστημονικής έρευνας και της ανάπτυξης εφαρμογών που αφορούν την κατηγοριοποίηση πολλαπλών ετικετών, η ανάγκη για εργαλεία και βιβλιοθήκες που παρέχουν αποτελεσματικές λύσεις είναι περισσότερο επιτακτική από ποτέ. Η βιβλιοθήκη Scikit-multilearn αποτελεί μία τέτοια πρωτοβουλία, προσφέροντας μια πλατφόρμα ανοιχτού κώδικα στην Python για την κατηγοριοποίηση πολλαπλών ετικετών. Στο παρόν κεφάλαιο, θα εξετάσουμε τις βασικές λειτουργίες της Scikit-multilearn, καθώς και ορισμένους από τους πιο δημοφιλείς αλγορίθμους που παρέχει.

Η Scikit-multilearn [18] είναι μια σύγχρονη, ευέλικτη βιβλιοθήκη, βασισμένη στην προϋπάρχουσα βιβλιοθήκη Scikit-learn, που επιτρέπει την εύκολη υλοποίηση και εξέταση διάφορων αλγορίθμων κατηγοριοποίησης πολλαπλών ετικετών σε Python. Παρέχει πρόσβαση σε ένα ευρύ φάσμα μεθόδων, από βασικές τεχνικές μετασχηματισμού προβλήματος μέχρι πιο προηγμένες τεχνικές προσαρμοσμένες για πολυετικετική κατηγοριοποίηση.

Παρακάτω θα επεξηγηθούν οι κυριότερες τεχνικές και αλγόριθμοι που περιλαμβάνει η βιβλιοθήκη οι BRkNNa, BRkNNb, MLkNN, MLARAM και MLTSVM.

3.2 BRkNNa

Ο BRkNNaClassifier είναι μια τεχνική που χρησιμοποιείται στη βιβλιοθήκη Scikit-multilearn για την επίλυση προβλημάτων ταξινόμησης με πολλαπλές ετικέτες (multi-label classification) μέσω ενός προσαρμοσμένου αλγορίθμου k-Nearest Neighbors (kNN). Η θεωρία πίσω από τον BRkNNaClassifier βασίζεται στην υπόθεση ότι μπορούμε να διαχωρίσουμε ένα πολύπλοκο πρόβλημα ταξινόμησης με πολλαπλές ετικέτες σε πολλά απλούστερα δυαδικά προβλήματα

ταξινόμησης, ένα για κάθε ετικέτα. Κάθε δυαδικό πρόβλημα ταξινόμησης λύνεται ανεξάρτητα με τη χρήση του k-Nearest Neighbors, προσδιορίζοντας αν ένα δείγμα ανήκει σε μια συγκεκριμένη κλάση (ετικέτα) ή όχι, βάσει της "γειτνίασής" του με άλλα δείγματα.

Ορίσματα του BRkNNaClassifier

Ο BRkNNaClassifier διαθέτει το εξής βασικό όρισμα:

k: Ο αριθμός των πλησιέστερων γειτόνων που θα ληφθούν υπόψη για την ταξινόμηση. Είναι το κύριο όρισμα που επηρεάζει την απόδοση του ταξινομητή.

Πιθανές Περιπτώσεις Χρήσης

Ο BRkNNaClassifier είναι κατάλληλος για μια πληθώρα εφαρμογών στις οποίες υπάρχουν πολλαπλές ετικέτες ανά δείγμα, όπως:

Ταξινόμηση κειμένου, όπου ένα έγγραφο μπορεί να ανήκει σε πολλές κατηγορίες ταυτόχρονα.

Ιατρική διάγνωση όπου ένας ασθενής μπορεί να έχει συγχρόνως πολλαπλές παθήσεις.

Αναγνώριση σκηνών σε εικόνες, όπου μια εικόνα μπορεί να περιέχει πολλά αντικείμενα ή θέματα.

Παράδειγμα Εφαρμογής

Στο παράδειγμα που ακολουθεί η συνάρτηση make_multilabel_classification από την βιβλιοθήκη sklearn.datasets της Python δημιουργεί ένα σύνολο πολυετικετικών δεδομένων με 100 δείγματα, κάθε ένα από τα οποία έχει 20 χαρακτηριστικά και ανήκει κατά μέσο όρο σε 2 από τις 5 διαθέσιμες κλάσεις. Στη συνέχεια, η συνάρτηση train_test_split της βιβλιοθήκης sklearn.model_selection, χρησιμοποιείται για να διαχωρίσει δεδομένα σε σύνολα εκπαίδευσης (train) και δοκιμής (test), δεσμεύοντας το 20% των δειγμάτων ως σύνολο δοκιμής, ενώ το υπόλοιπο 80% θα αποτελέσει το σύνολο εκπαίδευσης. Ο ταξινομητής BRkNNaClassifier ορίζεται με k=3 γεγονός που σημαίνει ότι θα κοιτάξει τους τρεις πλησιέστερους γείτονες ενός δείγματος για να αποφασίσει την ταξινόμησή του. Κατόπιν, ακολουθεί η εκπαίδευση και η πρόβλεψη κάνοντας χρήση των συναρτήσεων fit και predict του ταξινομητή αντίστοιχα, και τέλος η συνάρτηση accuracy_score της βιβλιοθήκης sklearn.metrics επιστρέφει την τιμή της ακρίβειας (accuracy).

```
from skmultilearn.adapt import BRkNNaClassifier
from sklearn.datasets import make_multilabel_classification
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import accuracy_score

# Δημιουργία συνόλου δεδομένων για multi-label ταξινόμηση

X, y = make_multilabel_classification(n_samples=100, n_features=20, n_classes=5, n_labels=2)

# Διαχωρισμός σε σύνολο εκπαίδευσης και δοκιμής

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Δημιουργία του BRkNNaClassifier με k=3

classifier = BRkNNaClassifier(k=3)

# Εκπαίδευση του ταξινομητή

classifier.fit(X_train, y_train)

# Πρόβλεψη των ετικετών στο σύνολο δοκιμής

predictions = classifier.predict(X_test)

# Υπολογισμός της ακρίβειας της πρόβλεψης

accuracy = accuracy_score(y_test, predictions)

print(f'Accuracy: {accuracy}')
```

3.3 BRkNNb

Ο BRkNNbClassifier είναι μια παραλλαγή του BRkNNaClassifier που υλοποιείται στη βιβλιοθήκη Scikit-multilearn για την επίλυση προβλημάτων πολλαπλής ταξινόμησης ετικετών (multi-label classification). Η τεχνική αυτή επικεντρώνεται στη χρήση του αλγορίθμου k-Nearest Neighbors (kNN) σε συνδυασμό με την προσέγγιση Binary Relevance (BR), όμως με έναν διαφορετικό τρόπο από τον BRkNNaClassifier. Στην προσέγγιση του BRkNNbClassifier, η διαφορά από τον BRkNNaClassifier είναι στον τρόπο προσδιορισμού της κλάσης ενός δείγματος. Αντί να ψάχνει για τους k πλησιέστερους γείτονες σε όλο το σύνολο δεδομένων και στη συνέχεια να αποφασίζει με βάση τις ετικέτες τους, ο BRkNNbClassifier ψάχνει για τους k πλησιέστερους γείτονες μέσα στο σύνολο των δειγμάτων για κάθε δυνατή ετικέτα ξεχωριστά. Αυτό σημαίνει ότι ο αλγόριθμος υπολογίζει την πιθανότητα συμμετοχής ενός δείγματος σε κάθε κλάση βάσει της πλησιέστερης γειτονιάς του εντός της εκάστοτε κλάσης.

Ορίσματα του BRkNNbClassifier

Ο BRkNNbClassifier διαθέτει το εξής όρισμα:

k: Ο αριθμός των πλησιέστερων γειτόνων που θα ληφθούν υπόψη για τον προσδιορισμό της κλάσης ενός δείγματος.

Πιθανές Περιπτώσεις Χρήσης

Ο BRkNNbClassifier είναι ιδιαίτερα χρήσιμος σε αντίστοιχα σενάρια πολλαπλής ταξινόμησης ετικετών με τον BRkNNaClassifier και επιπλέον θα ήταν ιδανικός σε σενάρια όπως:

Ταξινόμηση μουσικής, ένα τραγούδι μπορεί να ανήκει σε πολλαπλά μουσικά είδη και ακούσματα ή να έχει πολλαπλά χαρακτηριστικά που το καθιστούν κατάλληλο για διάφορες διαθέσεις ή σκηνές. Ο BRkNNbClassifier μπορεί να χρησιμοποιηθεί για να αναλύσει τα χαρακτηριστικά του ήχου και να κατηγοριοποιήσει τα τραγούδια στις αντίστοιχες ετικέτες.

Στην ανάλυση κοινωνικών δικτύων, ένα δημοσίευμα μπορεί να περιέχει πολλαπλά θέματα ή να απευθύνεται σε πολλαπλές ομάδες. Ο BRkNNbClassifier μπορεί να εκπαιδευτεί για να αναγνωρίζει διάφορα θέματα ή τάσεις από τα δεδομένα κειμένου των δημοσιεύσεων, βοηθώντας στην καλύτερη κατανόηση των ενδιαφερόντων των χρηστών.

Παράδειγμα Εφαρμογής

Στο παράδειγμα που ακολουθεί η συνάρτηση make_multilabel_classification από την βιβλιοθήκη sklearn.datasets της Python δημιουργεί ένα σύνολο πολυετικετικών δεδομένων με 100 δείγματα, κάθε ένα από τα οποία έχει 20 χαρακτηριστικά και ανήκει κατά μέσο όρο σε 2 από τις 3 διαθέσιμες κλάσεις. Στη συνέχεια, η συνάρτηση train_test_split της βιβλιοθήκης sklearn.model_selection, χρησιμοποιείται για να διαχωρίσει δεδομένα σε σύνολα εκπαίδευσης (train) και δοκιμής (test), δεσμεύοντας το 30% των δειγμάτων ως σύνολο δοκιμής, ενώ το υπόλοιπο 70% θα αποτελέσει το σύνολο εκπαίδευσης. Ο ταξινομητής BRkNNbClassifier ορίζεται χωρίς την παράμετρο k γεγονός που σημαίνει ότι η τιμή του k θα είναι η default τιμή που είναι το 10 και έτσι ο ταξινομητής θα κοιτάξει τους δέκα πλησιέστερους γείτονες ενός δείγματος για να αποφασίσει την ταξινόμησή του. Κατόπιν, ακολουθεί η εκπαίδευση και η πρόβλεψη κάνοντας χρήση των συναρτήσεων fit και predict του ταξινομητή αντίστοιχα, και τέλος η συνάρτηση hamming_loss της βιβλιοθήκης sklearn.metrics επιστρέφει την τιμή της απώλειας Hamming (hamming loss).

```
from skmultilearn.adapt import BRkNNbClassifier
from sklearn.datasets import make_multilabel_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import hamming_loss
# Δημιουργία συνόλου δεδομένων για multi-label ταξινόμηση
```

```
X, y = make_multilabel_classification(n_samples=100, n_features=20, n_classes=3, n_labels=2)
# Διαχωρισμός σε σύνολο εκπαίδευσης και δοκιμής
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
# Δημιουργία του BRkNNbClassifier
classifier = BRkNNbClassifier()
# Εκπαίδευση του ταξινομητή
classifier.fit(X_train, y_train)
# Πρόβλεψη των ετικετών στο σύνολο δοκιμής
predictions = classifier.predict(X_test)
# Υπολογισμός της απώλειας Hamming της πρόβλεψης
hamming_loss_value = hamming_loss(y_test, predictions)
print(f'Hamming Loss: {hamming_loss_value}')
```

3.4 MLkNN

Ο MLkNN (Multi-Label k-Nearest Neighbors) είναι ένας αλγόριθμος μηχανικής μάθησης που αποτελεί προσαρμογή του κλασικού k-Nearest Neighbors (kNN) για την επίλυση προβλημάτων πολλαπλής ταξινόμησης ετικετών (multi-label classification). Στον πυρήνα του, ο MLkNN ακολουθεί τη βασική ιδέα του kNN: η ταξινόμηση ενός δείγματος βασίζεται στις πιο κοντινές γειτονιές του στο χώρο των χαρακτηριστικών. Ωστόσο, ο MLkNN τροποποιεί αυτή τη λογική για να χειριστεί πολλαπλές ετικέτες, χρησιμοποιώντας μια ειδική τεχνική βασισμένη στην πιθανότητα για να προβλέψει την παρουσία ή απουσία κάθε ετικέτας. Για κάθε ετικέτα, ο αλγόριθμος υπολογίζει την πιθανότητα παρουσίας ή απουσίας της ετικέτας βασιζόμενος στον αριθμό των γειτόνων που έχουν αυτή την ετικέτα. Αυτό γίνεται μέσω ενός συστήματος ψηφοφορίας που λαμβάνει υπόψη τις ετικέτες των k πλησιέστερων γειτόνων του δείγματος.

Ορίσματα του MLkNN

Ο MLkNN στη βιβλιοθήκη Scikit-multilearn υποστηρίζει διάφορα ορίσματα, τα σημαντικότερα εκ των οποίων είναι:

k: Ο αριθμός των πλησιέστερων γειτόνων που θα χρησιμοποιηθούν για την πρόβλεψη. Είναι η βασική παράμετρος που καθορίζει το πώς ο αλγόριθμος θα βρει την κοντινότερη γειτονιά ενός δείγματος.

s: Παράμετρος εξομάλυνσης για την αποφυγή διαίρεσης με μηδέν κατά τον υπολογισμό των πιθανοτήτων. Βελτιώνει την αξιοπιστία των προβλέψεων όταν ο αριθμός των γειτόνων με συγκεκριμένη ετικέτα είναι πολύ μικρός.

ignore_first_neighbours: Πόσοι από τους πρώτους γείτονες να αγνοηθούν κατά τον υπολογισμό των πιθανοτήτων. Η ρύθμιση αυτή μπορεί να είναι χρήσιμη σε περιπτώσεις όπου οι πλησιέστεροι γείτονες δεν είναι αντιπροσωπευτικοί ή είναι θορυβώδεις.

Πιθανές Περιπτώσεις Χρήσης

Ο MLkNN θα ήταν ιδανικός για εφαρμογή σε περιπτώσεις όπως:

Ανίχνευση Θεμάτων σε Συλλογές Εγγράφων: Αυτόματη κατηγοριοποίηση εγγράφων ή ειδήσεων σε πολλαπλές κατηγορίες θεμάτων, βοηθώντας στην οργάνωση και την εύρεση πληροφοριών.

Πρόβλεψη Περιβαλλοντικών Κινδύνων: Χρήση σε περιβαλλοντικές επιστήμες για την κατηγοριοποίηση περιοχών με βάση πολλαπλούς περιβαλλοντικούς κινδύνους ή παράγοντες, όπως ρύπανση, εκπομπές, και απώλεια βιοποικιλότητας.

Παράδειγμα Εφαρμογής

Στο παράδειγμα που ακολουθεί η συνάρτηση make_multilabel_classification από την βιβλιοθήκη sklearn.datasets της Python δημιουργεί ένα σύνολο πολυετικετικών δεδομένων με 100 δείγματα, κάθε ένα από τα οποία έχει 20 χαρακτηριστικά και ανήκει κατά μέσο όρο σε 2 από τις 3 διαθέσιμες κλάσεις. Στη συνέχεια, η συνάρτηση train_test_split της βιβλιοθήκης sklearn.model_selection, χρησιμοποιείται για να διαχωρίσει δεδομένα σε σύνολα εκπαίδευσης (train) και δοκιμής (test), δεσμεύοντας το 30% των δειγμάτων ως σύνολο δοκιμής, ενώ το υπόλοιπο 70% θα αποτελέσει το σύνολο εκπαίδευσης. Ο ταξινομητής MLkNN ορίζεται με k=3 γεγονός που σημαίνει ότι θα κοιτάξει τους τρεις πλησιέστερους γείτονες ενός δείγματος για να αποφασίσει την ταξινόμησή του. Ειδικότερα ορίζεται και το s = 1 που σημαίνει ότι εφαρμόζεται μια βασική μορφή εξομάλυνσης. Αυτό σημαίνει ότι προστίθεται μία μονάδα (1) στον αριθμό των περιπτώσεων όπου ένας γείτονας έχει μια συγκεκριμένη ετικέτα και μία μονάδα (1) στον αριθμό των περιπτώσεων όπου ένας γείτονας δεν έχει την ετικέτα. Αυτό βοηθά στην αποφυγή του προβλήματος που θα προέκυπτε εάν καμία από τις περιπτώσεις δεν εμφανιζόταν στους πλησιέστερους γείτονες, οδηγώντας σε μια πιθανότητα που θα ήταν αδύνατο να υπολογιστεί λογικά λόγω διαίρεσης με το μηδέν. Επιπροσθέτως, με την παράμετρο ignore_first_neighbours

να έχει την τιμή 0 εξασφαλίζετε ότι δεν πρόκειται να αγνοηθούν κάποιοι από τους πρώτους εγγύτερους γείτονες. Κατόπιν, ακολουθεί η εκπαίδευση και η πρόβλεψη κάνοντας χρήση των συναρτήσεων fit και predict του ταξινομητή αντίστοιχα. Τέλος, τυπώνονται οι προβλέψεις.

```
from skmultilearn.adapt import MLkNN
from sklearn.datasets import make_multilabel_classification
from sklearn.model_selection import train_test_split
# Δημιουργία συνόλου δεδομένων για multi-label ταξινόμηση
X, y = make_multilabel_classification(n_samples=100, n_features=20, n_classes=3, n_labels=2)
# Διαχωρισμός σε σύνολο εκπαίδευσης και δοκιμής
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
# Δημιουργία του MLkNN με όλες τις διαθέσιμες παραμέτρους
classifier = MLkNN(k=3, s=1.0, ignore_first_neighbours=0)
# Εκπαίδευση του ταξινομητή
classifier.fit(X_train, y_train)
# Πρόβλεψη των ετικετών στο σύνολο δοκιμής
predictions = classifier.predict(X_test)
# Εμφάνιση των προβλέψεων
print(predictions.toarray())
```

3.5 *MLARAM*

Ο MLARAM (Multi-Label Averaged Rank Association Map) είναι μια τεχνική ταξινόμησης για προβλήματα με πολλαπλές ετικέτες, που υλοποιείται στη βιβλιοθήκη Scikit-multilearn. Αυτή η τεχνική είναι μια επέκταση του Averaged Rank Association Map (ARAM), που είναι ένας νευρωνικός αλγόριθμος βασισμένος σε μνήμη, ο οποίος εκπαιδεύει ένα σύνολο ταξινομητών, έναν για κάθε ετικέτα. Ο MLARAM λειτουργεί με τη δημιουργία ενός συνόλου μικρών, τοπικών μοντέλων (νευρωνικών δικτύων) που αποθηκεύονται σε μια δομή μνήμης. Κάθε ένα από αυτά τα μοντέλα εκπαιδεύεται σε ένα μικρό υποσύνολο των δεδομένων, και η ταξινόμηση γίνεται με βάση την πλειοψηφία των ψήφων ή την πιθανότητα που προκύπτει από τη συνένωση των προβλέψεων όλων των μοντέλων.

Ορίσματα του MLARAM

vigilance: Αυτό το όρισμα καθορίζει το επίπεδο της αυστηρότητας για την αποδοχή ενός δείγματος από έναν υπάρχοντα ταξινομητή (νευρώνα) μέσα στο μοντέλο. Ένας υψηλός δείκτης vigilance σημαίνει ότι μόνο τα δείγματα που ταιριάζουν πολύ κοντά στα χαρακτηριστικά ενός νευρώνα θα ενσωματωθούν σε αυτόν, ενώ ένας χαμηλότερος δείκτης επιτρέπει μεγαλύτερη ευελιξία.

threshold: Το όρισμα threshold στον MLARAM καθορίζει ένα κατώφλι για την απόφαση αν ένα δείγμα πρέπει να ενσωματωθεί σε έναν υπάρχοντα νευρώνα ή αν πρέπει να δημιουργηθεί ένας νέος νευρώνας για αυτό. Το threshold είναι ένας σημαντικός παράγοντας για τη ρύθμιση της γενικότητας ή της ειδικότητας του μοντέλου, αλλάζοντας το κατώφλι για την αποδοχή νέων δειγμάτων.

Πιθανές Περιπτώσεις Χρήσης

Ο MLARAM είναι ένας ευέλικτος αλγόριθμος που μπορεί να εφαρμοστεί σε ποικίλες περιπτώσεις όπου η πολλαπλή ταξινόμηση ετικετών είναι απαραίτητη. Αυτό σημαίνει πως μπορεί να βρει εφαρμογή σε διάφορους τομείς, όπως:

Συστάσεις Περιεχομένου: Σε συστήματα συστάσεων όπου ένα προϊόν ή περιεχόμενο μπορεί να αντιστοιχιστεί σε πολλαπλές κατηγορίες ενδιαφερόντων ταυτόχρονα, βελτιώνοντας την προσωποποίηση της εμπειρίας του χρήστη.

Εκπαιδευτικές Εφαρμογές: Στην προσαρμοστική εκπαίδευση, όπου ο MLARAM μπορεί να βοηθήσει στην κατηγοριοποίηση εκπαιδευτικού υλικού ή δραστηριοτήτων βάσει πολλαπλών δεξιοτήτων ή εκπαιδευτικών στόχων που πρέπει να επιτευχθούν.

Παράδειγμα Εφαρμογής

Στο παράδειγμα που ακολουθεί η συνάρτηση make_multilabel_classification από την βιβλιοθήκη sklearn.datasets της Python δημιουργεί ένα σύνολο πολυετικετικών δεδομένων με 100 δείγματα, κάθε ένα από τα οποία έχει 20 χαρακτηριστικά και ανήκει κατά μέσο όρο σε 2 από τις 5 διαθέσιμες κλάσεις. Στη συνέχεια, η συνάρτηση train_test_split της βιβλιοθήκης sklearn.model_selection, χρησιμοποιείται για να διαχωρίσει δεδομένα σε σύνολα εκπαίδευσης (train) και δοκιμής (test), δεσμεύοντας το 30% των δειγμάτων ως σύνολο δοκιμής, ενώ το υπόλοιπο 70% θα αποτελέσει το σύνολο εκπαίδευσης. Ο ταξινομητής MLARAM ορίζεται με vigilance=0.9 γεγονός που υποδηλώνει υψηλό επίπεδο αυστηρότητας στην αποδοχή νέων δειγμάτων από τους υπάρχοντες νευρώνες, δηλαδή, ένα δείγμα πρέπει να έχει πολύ κοντινή αντιστοιχία με τα χαρακτηριστικά ενός νευρώνα για να ενσωματωθεί. Ειδικότερα ορίζεται και το threshold = 0.05 που ορίζει το κατώφλι για την απόφαση αν ένα δείγμα θα αντιστοιχιστεί σε

έναν υπάρχοντα νευρώνα ή αν θα προκαλέσει τη δημιουργία ενός νέου. Αυτό βοηθά στη διατήρηση της ισορροπίας μεταξύ της γενίκευσης και της ειδίκευσης του μοντέλου. Κατόπιν, ακολουθεί η εκπαίδευση και η πρόβλεψη κάνοντας χρήση των συναρτήσεων fit και predict του ταξινομητή αντίστοιχα. Τέλος, τυπώνονται οι προβλέψεις.

```
from skmultilearn.adapt import MLARAM
from sklearn.datasets import make_multilabel_classification
from sklearn.model_selection import train_test_split
# Δημιουργία συνόλου δεδομένων για multi-label ταξινόμηση
X, y = make_multilabel_classification(n_samples=100, n_features=20, n_classes=5, n_labels=2)
# Διαχωρισμός σε σύνολο εκπαίδευσης και δοκιμής
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
# Δημιουργία του MLARAM με όλες τις διαθέσιμες παραμέτρους
classifier = MLARAM(vigilance=0.9, threshold=0.05)
# Εκπαίδευση του ταξινομητή
classifier.fit(X_train, y_train)
# Πρόβλεψη των ετικετών στο σύνολο δοκιμής
predictions = classifier.predict(X_test)
# Εμφάνιση των προβλέψεων
print(predictions)
```

3.6 MLTSVM

Ο MLTSVM (Multi-Label Twin Support Vector Machine) είναι μια προσέγγιση στη μάθηση με επίβλεψη που επεκτείνει την ιδέα των Twin Support Vector Machines (TWSVM) σε προβλήματα πολλαπλών ετικετών. Αυτός ο αλγόριθμος αναπτύχθηκε για να αντιμετωπίσει την πρόκληση της ταυτόχρονης πρόβλεψης πολλαπλών ετικετών για κάθε δείγμα, χρησιμοποιώντας ένα μοντέλο που δημιουργεί δύο υπερεπίπεδα για κάθε ετικέτα με σκοπό τη μεγιστοποίηση της απόστασης από τα πλησιέστερα δείγματα της αντίθετης κλάσης.

Ορίσματα του MLTSVM

c_k: Παράμετρος που ελέγχει τον συμβιβασμό μεταξύ της επίτευξης μικρής απόστασης από το υπερεπίπεδο και της ελαχιστοποίησης των λαθών ταξινόμησης για κάθε κλάση. Κάθε c_k αντιστοιχεί σε μια διαφορετική ετικέτα στο πρόβλημα πολλαπλών ετικετών.

sor_omega: Παράμετρος που χρησιμοποιείται στην επίλυση των βελτιστοποιητικών προβλημάτων που προκύπτουν στον MLTSVM, συγκεκριμένα στην επίλυση των γραμμικών εξισώσεων μέσω της μεθόδου Successive Over-Relaxation (SOR). Αυτή η παράμετρος ελέγχει τον ρυθμό σύγκλισης της μεθόδου.

threshold: Κατώφλι για την τερματισμό της επαναληπτικής διαδικασίας. Αυτή η τιμή καθορίζει πότε θα θεωρηθεί ότι η διαδικασία έχει συγκλίνει, βάσει της διαφοράς των διαδοχικών εκτιμήσεων.

lambda_param: Παράμετρος που ελέγχει την επίδραση του όρου κανονικοποίησης στη συνάρτηση στόχου του MLTSVM. Αυτός ο όρος βοηθά στην αποφυγή του overfitting με την προσθήκη μιας ποινής για μεγάλες τιμές των βαρών του μοντέλου.

max_iteration: Ο μέγιστος αριθμός επαναλήψεων για την επίλυση του βελτιστοποιητικού προβλήματος. Αυτή η παράμετρος καθορίζει το μέγιστο αριθμό επαναλήψεων πριν τον αυτόματο τερματισμό της διαδικασίας, ανεξάρτητα από το αν έχει επιτευχθεί σύγκλιση.

Πιθανές Περιπτώσεις Χρήσης

Ο MLTSVM μπορεί να βρει εφαρμογή σε διάφορους τομείς, όπως:

Αναγνώριση Ανθρώπινης Δραστηριότητας: Σε εφαρμογές παρακολούθησης ή συστημάτων ευφυούς σπιτιού, η αναγνώριση διαφορετικών τύπων ανθρώπινης δραστηριότητας από αισθητήρια δεδομένα (π.χ., αισθητήρες κίνησης) μπορεί να επωφεληθεί από την χρήση MLTSVM για την ταυτόχρονη ταξινόμηση πολλαπλών δραστηριοτήτων.

Εκτίμηση Ρίσκου Πολλαπλών Χρηματοοικονομικών Προϊόντων: Στη χρηματοοικονομική ανάλυση, η εκτίμηση του ρίσκου για πολλαπλά χρηματοοικονομικά προϊόντα μπορεί να επωφεληθεί από την χρήση MLTSVM για την ταυτόχρονη πρόβλεψη του ρίσκου σε διάφορες αγορές ή κατηγορίες προϊόντων.

Παράδειγμα Εφαρμογής

Στο παράδειγμα που ακολουθεί θα χρησιμοποιηθεί το υπάρχον dataset ‘emotions’ της Python, το οποίο φορτώνεται με χρήση της συνάρτησης load_dataset και με την ένδειξη ‘undivided’ ώστε αρχικά να μην γίνει διαχωρισμός σε δεδομένα εκπαίδευσης και δοκιμής. Στη συνέχεια, η συνάρτηση train_test_split της βιβλιοθήκης sklearn.model_selection, χρησιμοποιείται για να διαχωρίσει τα δεδομένα σε σύνολα εκπαίδευσης (train) και δοκιμής (test), δεσμεύοντας το 10% των δειγμάτων ως σύνολο δοκιμής, ενώ το υπόλοιπο 90% θα αποτελέσει το σύνολο

εκπαίδευσης. Ο ταξινομητής MLTSVM ορίζεται με c_k=3 ρυθμίζοντας έτσι την τιμή του παράγοντα κανονικοποίησης για την τιμωρία των περιθωρίων στην ταξινόμηση. Στις Μηχανές Διανυσματικής Υποστήριξης (SVM), αυτή η παράμετρος ελέγχει το trade-off μεταξύ της επίτευξης ενός μικρότερου περιθωρίου διαχωρισμού και της εξασφάλισης ότι τα δείγματα θα ταξινομηθούν σωστά. Μια υψηλότερη τιμή του c_k δίνει μεγαλύτερη έμφαση στη σωστή ταξινόμηση όλων των δειγμάτων, ακόμη και εις βάρος ενός μικρότερου περιθωρίου. Όσον αφορά την παράμετρο sor_omega αναφέρεται στο ρυθμό χαλάρωσης της μεθόδου επαναληπτικής υπερχαλάρωσης (Successive Over-Relaxation, SOR) που χρησιμοποιείται για την επίλυση των βελτιστοποιητικών προβλημάτων στο MLTSVM. Η τιμή 1.0 αντιστοιχεί στην κλασική μέθοδο Gauss-Seidel χωρίς υπερχαλάρωση. Τιμές μεγαλύτερες από 1.0 ενδέχεται να επιταχύνουν τη σύγκλιση, αλλά επίσης αυξάνουν τον κίνδυνο αστάθειας. Η παράμετρος threshold=0.06 ορίζει το κατώφλι για την τερματισμό της επαναληπτικής διαδικασίας στην επίλυση του βελτιστοποιητικού προβλήματος. Όταν η διαφορά στην αξιολόγηση της συνάρτησης στόχου μεταξύ δύο διαδοχικών επαναλήψεων είναι μικρότερη από αυτό το κατώφλι, η διαδικασία επανάληψης σταματά. Αυτό βοηθά στον περιορισμό του χρόνου εκτέλεσης και στην αποφυγή άσκοπων υπολογισμών. Επιπροσθέτως, η παράμετρος lambda_param έχοντας την τιμή 1 ελέγχει την επίδραση του όρου κανονικοποίησης στην τιμωρία της πολυπλοκότητας του μοντέλου στην απώλεια. Μια υψηλότερη τιμή για το lambda_param αυξάνει την επιρροή της κανονικοποίησης, ενθαρρύνοντας πιο απλά μοντέλα που μπορεί να γενικεύουν καλύτερα σε αδιερεύνητα δεδομένα. Το max_iteration=400 ορίζει τον μέγιστο αριθμό επαναλήψεων για την επαναληπτική διαδικασία. Ο περιορισμός αυτός εξασφαλίζει ότι η διαδικασία θα τερματιστεί ακόμα και αν δεν επιτευχθεί η επιθυμητή σύγκλιση, προλαμβάνοντας την άπειρη εκτέλεση σε περίπτωση που η διαδικασία δεν καταφέρει να συγκλίνει. Κατόπιν, ακολουθεί η εκπαίδευση και η πρόβλεψη κάνοντας χρήση των συναρτήσεων fit και predict του ταξινομητή αντίστοιχα. Τέλος, τυπώνονται οι προβλέψεις.

```
from skmultilearn.adapt import MLTSVM
from sklearn.model_selection import train_test_split
# Φόρτωση του dataset - χωρίς διαχωρισμό σε train/set
X, y, _, _ = load_dataset('emotions', 'undivided')
# Διαχωρισμός σε σύνολο εκπαίδευσης και δοκιμής
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
# Δημιουργία του MLTSVM με όλες τις διαθέσιμες παραμέτρους
classifier = MLTSVM(c_k=3, sor_omega=1.0, threshold=0.06, lambda_param=1.0, max_iteration=400)
```

```
# Εκπαίδευση του ταξινομητή
classifier.fit(X_train, y_train)
# Πρόβλεψη των ετικετών στο σύνολο δοκιμής
predictions = classifier.predict(X_test)
# Εμφάνιση των προβλέψεων
print(predictions)
```

4

Πειραματική μελέτη

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, η πειραματική μελέτη αποτελεί ένα κρίσιμο κεφάλαιο που στοχεύει στην εκτίμηση και τη σύγκριση της αποτελεσματικότητας διαφόρων αλγορίθμων κατηγοριοποίησης πολλαπλών ετικετών, όπως αυτοί υλοποιούνται στη βιβλιοθήκη Scikit-multilearn. Μέσω αυτού του κεφαλαίου, θα επιχειρηθεί μια συστηματική αξιολόγηση της απόδοσης αυτών των αλγορίθμων, χρησιμοποιώντας διαφορετικά σύνολα δεδομένων και μετρικές απόδοσης.

Η επιλογή των αλγορίθμων περιλαμβάνει τόσο τις βασικές όσο και τις προηγμένες μεθόδους κατηγοριοποίησης πολλαπλών ετικετών που διατίθενται μέσω της Scikit-multilearn, προκειμένου να καλυφθεί ένα ευρύ φάσμα προσεγγίσεων. Από τον BRkNN έως τον MLkNN και τους πιο εξειδικευμένους αλγορίθμους όπως ο MLARAM και ο MLTSVM, κάθε αλγόριθμος θα υποβληθεί σε δοκιμασία υπό διάφορες παραμέτρους και συνθήκες για την αξιολόγηση της ευρεσιτεχνίας και της αποδοτικότητάς του.

Για την πειραματική διαδικασία, θα χρησιμοποιηθούν διάφορα σύνολα δεδομένων του Mulan, το οποίο αποτελεί μια συλλογή από σύνολα δεδομένων πολλαπλών ετικετών που είναι ειδικά σχεδιασμένα για την αξιολόγηση τέτοιων αλγορίθμων. Η επιλογή του Mulan ως της βάσης για την πειραματική μελέτη δικαιολογείται από την ποικιλομορφία και την πληρότητα των δεδομένων που περιλαμβάνει, προσφέροντας έναν πλούσιο χώρο δοκιμών για την εκτίμηση της απόδοσης.

Οι μετρικές που θα χρησιμοποιηθούν για την αξιολόγηση των αλγορίθμων περιλαμβάνουν την Ακρίβεια (Accuracy), την Ορθότητα (Precision), την Ανάκληση (Recall) καθώς και πιο εξειδικευμένες μετρικές που είναι συνήθεις στην κατηγοριοποίηση πολλαπλών ετικετών, όπως η Απώλεια Hamming (Hamming Loss). Αυτές οι μετρικές θα παρέχουν μια σφαιρική εικόνα της αποτελεσματικότητας κάθε αλγορίθμου σε διαφορετικές πτυχές της κατηγοριοποίησης πολλαπλών ετικετών.

Στόχος είναι η πειραματική μελέτη να αποκαλύψει τις δυνατότητες και τα όρια κάθε μεθόδου, να εντοπίσει τις προκλήσεις που εγείρονται κατά την εφαρμογή τους σε πραγματικά σύνολα δεδομένων και να προσφέρει καθοδήγηση σχετικά με την επιλογή του κατάλληλου αλγορίθμου ανάλογα με τις ανάγκες της εφαρμογής. Αυτή η συστηματική προσέγγιση επιδιώκει να συμβάλει στη βαθύτερη κατανόηση της κατηγοριοποίησης πολλαπλών ετικετών και να διευρύνει τις δυνατότητες της μηχανικής μάθησης σε αυτό το πεδίο.

4.1 Σύνολα δεδομένων

Τα Mulan datasets [19] αποτελούν μια σημαντική πηγή δεδομένων που αφορά στο πεδίο της κατηγοριοποίησης πολλαπλών ετικετών. Το όνομα "Mulan" προέρχεται από την έννοια της "Multi-Label learning", μια μέθοδο μηχανικής μάθησης όπου κάθε δείγμα μπορεί να ανήκει ταυτόχρονα σε πολλαπλές ετικέτες-κατηγορίες. Αυτό διαφέρει από την παραδοσιακή μονοκατηγορική κατηγοριοποίηση, όπου κάθε δείγμα ανήκει μόνο σε μία κατηγορία.

Τα datasets του Mulan περιλαμβάνουν διάφορους τύπους δεδομένων, όπως κείμενα, εικόνες και γενετικές πληροφορίες, καθιστώντας τα χρήσιμα για μια πληθώρα εφαρμογών σε διαφορετικούς τομείς.

Στην πειραματική μελέτη που ακολουθεί χρησιμοποιήθηκαν τα παρακάτω datasets:

Bibtex: Αφορά δεδομένα από εγγραφές βιβλιογραφίας στο BibTeX format. Οι ετικέτες αντιστοιχούν σε λέξεις-κλειδιά που περιγράφουν το περιεχόμενο των δημοσιεύσεων. Χρησιμοποιείται για εργασίες ταξινόμησης και συστάσεων βιβλιογραφικών αναφορών.

Birds: Αυτό το dataset περιλαμβάνει ηχογραφήσεις τραγουδιών πουλιών. Οι ετικέτες αντιστοιχούν στα είδη των πουλιών που εκπέμπουν τα τραγούδια. Χρησιμοποιείται σε εργασίες αναγνώρισης ήχου και ταξινόμησης.

Delicious: Περιέχει δεδομένα από την υπηρεσία social bookmarking Delicious. Τα δεδομένα περιλαμβάνουν ετικέτες που χρηστές έχουν αποδώσει σε ιστοσελίδες, και χρησιμοποιείται για ταξινόμηση και συστάσεις ιστοσελίδων.

Emotions: Αυτό το dataset σχετίζεται με τη μουσική και τις συναισθηματικές αντιδράσεις που προκαλεί. Συγκεκριμένα, περιέχει δεδομένα για τραγούδια και τις συναισθηματικές ετικέτες που τα χαρακτηρίζουν, όπως "χαρούμενο", "θλιμμένο", "ήρεμο", κλπ.

Enron: Το dataset αυτό περιέχει συλλογή email που ανταλλάχθηκαν στην εταιρεία Enron. Χρησιμοποιείται συχνά για εργασίες ανίχνευσης θεμάτων και ταξινόμησης κειμένου, καθώς τα emails έχουν ετικέτες που αντιστοιχούν σε διάφορα θέματα.

Genbase: Αυτό το dataset περιλαμβάνει γονιδιωματικά δεδομένα και οι ετικέτες αντιστοιχούν σε λειτουργικές κατηγορίες γονιδίων. Χρησιμοποιείται για την πρόβλεψη της λειτουργίας γονιδίων.

Mediamill: Περιλαμβάνει δεδομένα από βίντεο και οι ετικέτες περιγράφουν το περιεχόμενο των βίντεο (π.χ. δράση, τοπίο). Χρησιμοποιείται για εργασίες αναγνώρισης περιεχομένου βίντεο και ταξινόμησης.

Medical: Αυτό το dataset περιλαμβάνει δεδομένα από ιατρικές περιγραφές και οι ετικέτες αντιστοιχούν σε διαγνώσεις ή συμπτώματα. Χρησιμοποιείται για ταξινόμηση και πρόβλεψη ιατρικών καταστάσεων.

Scene: Περιέχει εικόνες από διάφορες σκηνές (π.χ. παραλίες, δάση, πόλεις) και οι ετικέτες αντιστοιχούν σε αντικείμενα ή χαρακτηριστικά της σκηνής. Χρησιμοποιείται για εργασίες ταξινόμησης εικόνας και αναγνώρισης σκηνών.

Yeast: Αφορά δεδομένα από μελέτες γενετικής της μαγιάς. Οι ετικέτες αντιστοιχούν σε λειτουργικές κατηγορίες των πρωτεϊνών της μαγιάς. Χρησιμοποιείται για την πρόβλεψη λειτουργικών κατηγοριών πρωτεϊνών.

Πίνακας 1 Γνωρίσματα συνόλων δεδομένων Mulan

name	domain	instances	nominal	numeric	labels	cardinality	density	distinct
bibtex	text	7395	1836	0	159	2.402	0.015	2856
birds	audio	645	2	258	19	1.014	0.053	133
delicious	text (web)	16105	500	0	983	19.020	0.019	15806
emotions	music	593	0	72	6	1.869	0.311	27
enron	text	1702	1001	0	53	3.378	0.064	753
genbase	biology	662	1186	0	27	1.252	0.046	32
mediamill	video	43907	0	120	101	4.376	0.043	6555
medical	text	978	1449	0	45	1.245	0.028	94
scene	image	2407	0	294	6	1.074	0.179	15
yeast	biology	2417	0	103	14	4.237	0.303	198

Στον παραπάνω πίνακα αναγράφονται αριθμητικές πληροφορίες που αφορούν τα datasets που θα χρησιμοποιηθούν στη συνέχεια στην πειραματική μελέτη. Παρακάτω ακολουθεί η ερμηνεία των εννοιών που αναφέρονται στον πίνακα.

Domain: Αναφέρεται στον πεδίο ή την κατηγορία των δεδομένων (π.χ., μουσική, εικόνες, κείμενο).

Instances: Ο αριθμός των παραδειγμάτων ή των δειγμάτων στο σύνολο δεδομένων.

Nominal: Ο αριθμός των χαρακτηριστικών που έχουν κατηγορικές ή ονομαστικές τιμές.

Numeric: Ο αριθμός των χαρακτηριστικών με αριθμητικές τιμές.

Labels: Ο συνολικός αριθμός των διαφορετικών ετικετών που μπορεί να έχει κάθε δείγμα.

Cardinality: Η μέση τιμή του αριθμού των ετικετών ανά δείγμα στο σύνολο δεδομένων.

Density: Η πυκνότητα των ετικετών, που υπολογίζεται ως η καρδιναλιότητα (cardinality) διαιρούμενη με τον συνολικό αριθμό των διαθέσιμων ετικετών.

Distinct: Ο αριθμός των διακριτών ή μοναδικών συνόλων ετικετών που εμφανίζονται στο σύνολο δεδομένων.

4.2 Εγκαθίδρυση πειραμάτων

Στον σύγχρονο κόσμο της τεχνολογίας, η δύναμη και η ταχύτητα των υπολογιστών έχουν ανοίξει νέους δρόμους στην επιστημονική έρευνα και ανάλυση. Με τη χρήση ενός υψηλών επιδόσεων υπολογιστή, είμαστε σε θέση να εγκαθιδρύσουμε μια πειραματική μελέτη που στοχεύει στην ανάπτυξη και δοκιμή προηγμένων αλγορίθμων μηχανικής μάθησης. Ο επεξεργαστής υψηλών επιδόσεων επιτρέπει την εκτέλεση περίπλοκων υπολογιστικών εργασιών με βέλτιστη ταχύτητα, ενώ η μεγάλη ποσότητα μνήμης RAM και ο υψηλής ταχύτητας σκληρός δίσκος SSD εξασφαλίζουν ότι η ανάλυση μεγάλων δεδομένων γίνεται αποδοτικά.

Συγκεκριμένα η διενέργεια της πειραματικής μελέτης έγινε σε υπολογιστή με τα ακόλουθα χαρακτηριστικά:

Επεξεργαστής (CPU): AMD Ryzen 5 1600 Six-Core Processor 3.20 GHz

Μνήμη RAM: 16.0 GB

Σκληρός Δίσκος (SSD): Samsung SSD 970 EVO Plus 1TB

Κάρτα Γραφικών (GPU): NVIDIA GeForce GTX 1060 6GB

Λειτουργικό Σύστημα: Windows 10 Pro

Στόχος της μελέτης είναι η κατανόηση, πως οι νέοι αλγόριθμοι μπορούν να αντιμετωπίσουν τις προκλήσεις που παρουσιάζουν τα μεγάλα δεδομένα, όπως η ανάγκη για ταχύτητα, η ακρίβεια και η ικανότητα να διαχειρίζονται πολύπλοκες, μη-γραμμικές σχέσεις μεταξύ των δεδομένων.

4.3 Πειραματικές μετρήσεις

Η πειραματική μελέτη που υλοποιήθηκε εφαρμόστηκε για κάθε αλγόριθμο και τεχνική προσέγγισης της βιβλιοθήκης Scikit-multilearn και για κάθε επιλεγμένο σύνολο δεδομένων που αναλύθηκε το Κεφάλαιο 4.1 και αφορά στην λήψη μετρήσεων των Training CPU Time, Testing CPU Time, Hamming Loss, Accuracy, Precision, Recall.

4.3.1 Training CPU Time

Το Training CPU Time εξ ’ορισμού αναφέρεται στον χρόνο που απαιτείται από μια μονάδα κεντρικής επεξεργασίας (CPU) για να εκπαιδεύσει ένα μοντέλο μηχανικής μάθησης ή τεχνητής νοημοσύνης. Η εκπαίδευση ενός μοντέλου περιλαμβάνει την ανάλυση και την επεξεργασία μεγάλων συνόλων δεδομένων για να μάθει και να προσαρμόζεται σε συγκεκριμένα πρότυπα ή τάσεις. Ο χρόνος που απαιτείται για την εκπαίδευση ενός μοντέλου εξαρτάται από πολλούς παράγοντες, όπως η πολυπλοκότητα του μοντέλου, ο όγκος και η ποιότητα των δεδομένων εκπαίδευσης, καθώς και η υποδομή υλικού και λογισμικού που χρησιμοποιείται.

Παράγοντες που επηρεάζουν το Training CPU Time

Πολυπλοκότητα του Μοντέλου: Μοντέλα με περισσότερες παραμέτρους και πιο σύνθετες αρχιτεκτονικές απαιτούν περισσότερο χρόνο εκπαίδευσης.

Όγκος Δεδομένων: Μεγαλύτερα σύνολα δεδομένων εκπαίδευσης αυξάνουν τον απαιτούμενο χρόνο, καθώς το μοντέλο πρέπει να διαπεράσει και να μάθει από περισσότερες περιπτώσεις.

Υποδομή Υλικού: Οι επιδόσεις της CPU (π.χ., ταχύτητα ρολογιού, αριθμός πυρήνων) και η διαθεσιμότητα άλλων πόρων (π.χ., GPU, TPU) επηρεάζουν την ταχύτητα εκπαίδευσης.

Βελτιστοποίηση Κώδικα και Αλγορίθμων: Η αποδοτικότητα του κώδικα και η επιλογή αλγορίθμων μπορούν να μειώσουν τον απαιτούμενο χρόνο εκπαίδευσης.

Μείωση του Training CPU Time

Για να μειωθεί ο χρόνος εκπαίδευσης, οι ερευνητές και οι μηχανικοί μπορούν να εφαρμόσουν διάφορες τεχνικές, όπως:

Χρήση GPU ή TPU: Η μεταφορά των υπολογισμών εκπαίδευσης σε GPU ή TPU μπορεί να προσφέρει σημαντικές βελτιώσεις στην ταχύτητα.

Βελτιστοποίηση Αλγορίθμων: Επιλογή αποδοτικότερων αλγορίθμων και τεχνικών όπως το batch training μπορούν να μειώσουν τον χρόνο εκπαίδευσης.

Παραλληλισμός και Διανεμημένη Επεξεργασία: Η διανεμημένη επεξεργασία και ο παραλληλισμός μπορούν να επιταχύνουν την εκπαίδευση με την ταυτόχρονη εκτέλεση υπολογισμών.

4.3.2 Testing CPU Time

Ο όρος Testing CPU Time αναφέρεται στο χρονικό διάστημα που απαιτεί μια κεντρική μονάδα επεξεργασίας (CPU) ώστε να διενεργήσει τη δοκιμαστική φάση ενός μοντέλου μηχανικής μάθησης ή τεχνητής νοημοσύνης. Αυτή η διαδικασία περιλαμβάνει την αξιολόγηση του μοντέλου χρησιμοποιώντας ένα σύνολο δεδομένων δοκιμής που δεν έχει χρησιμοποιηθεί κατά την εκπαίδευση, για να καθοριστεί πόσο καλά το μοντέλο μπορεί να προβλέψει ή να κατανοήσει τα δεδομένα. Αυτός ο χρόνος επηρεάζεται από παρόμοιους παράγοντες με αυτούς που επηρεάζουν τον χρόνο εκπαίδευσης, αλλά συνήθως είναι σημαντικά μικρότερος, καθώς το μοντέλο δεν χρειάζεται να αναπροσαρμόζεται.

Παράγοντες που Επηρεάζουν τον Testing CPU Time

Μέγεθος του Σετ Δεδομένων Δοκιμής: Ένα μεγαλύτερο σετ δεδομένων δοκιμής απαιτεί περισσότερο χρόνο για να διενεργηθεί η αξιολόγηση.

Πολυπλοκότητα του Μοντέλου: Μοντέλα με περισσότερες παραμέτρους και πιο σύνθετες δομές μπορεί να χρειάζονται περισσότερο χρόνο για να κάνουν προβλέψεις.

Υποδομή Υλικού: Η απόδοση του CPU, καθώς και η διαθεσιμότητα και η χρήση άλλων πόρων υλικού (π.χ., GPU), μπορεί να επηρεάσει την ταχύτητα των δοκιμών.

Μείωση του Testing CPU Time

Επιλογή Αποδοτικών Μοντέλων: Χρήση λιγότερο πολύπλοκων μοντέλων όταν είναι δυνατόν, χωρίς να θυσιάζεται η ακρίβεια.

Προ-επεξεργασία Δεδομένων: Εφαρμογή τεχνικών προ-επεξεργασίας για να μειωθεί ο όγκος των δεδομένων δοκιμής χωρίς να χάνεται σημαντική πληροφορία.

Χρήση Παραλληλισμού και Διανεμημένης Επεξεργασίας: Αξιοποίηση του παραλληλισμού και της διανεμημένης επεξεργασίας για να μειωθεί ο συνολικός χρόνος αξιολόγησης.

4.3.3 Hamming Loss

Η μετρική Hamming Loss (απώλεια Hamming) [6] αποτελεί μία σημαντική μέτρηση στον τομέα της μηχανικής μάθησης, ιδιαίτερα όταν αντιμετωπίζουμε προβλήματα πολλαπλών

ετικετών (multi-label classification). Η μετρική αυτή μετρά το ποσοστό των λανθασμένων προβλέψεων ,δηλαδή, τη διαφορά μεταξύ του πραγματικού και του προβλεπόμενου συνόλου ετικετών για ένα δεδομένο σύνολο δειγμάτων. Σε αντίθεση με άλλες μετρικές που επικεντρώνονται στην ακρίβεια των σωστά ταξινομημένων παραδειγμάτων, η Hamming Loss αναδεικνύει τη σημασία της αποφυγής λαθών προβλέψεων.

Η Hamming Loss ορίζεται ως το μέσο ποσοστό των λανθασμένων ετικετών (δηλαδή, των ψευδώς θετικών και ψευδώς αρνητικών) προς το συνολικό αριθμό των ετικετών, ανά δείγμα στο σύνολο δεδομένων. Μαθηματικά, μπορεί να εκφραστεί ως:

$$\frac{1}{|N| \cdot |L|} \sum_{i=1}^{|N|} \sum_{j=1}^{|L|} \mathbf{xor}(y_{i,j}, z_{i,j})$$

Σχήμα 8 Μαθηματικός ορισμός Hamming Loss [20]

όπου:

- yi,j είναι η πραγματική τιμή της j-οστής ετικέτας για το i-οστό δείγμα,
- zi,j είναι η προβλεπόμενη τιμή της j-οστής ετικέτας για το i-οστό δείγμα,
- N είναι ο αριθμός των δειγμάτων στο σετ δεδομένων, και
- L είναι το μέγεθος της ετικέτας (δηλαδή, ο αριθμός των δυνατών ετικετών).

Η xor συνάρτηση υπολογίζει 1 αν η πραγματική και η προβλεπόμενη τιμή διαφέρουν, διαφορετικά 0, για κάθε ετικέτα και κάθε δείγμα.

Η Hamming Loss υπολογίζει έναν αριθμός μεταξύ 0 και 1, όπου 0 σημαίνει τέλεια ακρίβεια και 1 σημαίνει πλήρη ανακρίβεια. Με άλλα λόγια, η Hamming Loss αντικατοπτρίζει το ποσοστό των λαθών στις προβλέψεις του μοντέλου ως προς την συνολική συλλογή των ετικετών σε όλα τα δείγματα, και άρα όσο πιο μικρή η τιμή της τόσο πιο λίγες οι λανθασμένες προβλέψεις.

Παράδειγμα υπολογισμού Hamming Loss

Ας υποθέσουμε ότι έχουμε ένα πρόβλημα πολλαπλής κατηγοριοποίησης ετικετών, όπου θέλουμε να κατηγοριοποιήσουμε έγγραφα σε πολλαπλές κατηγορίες-ετικέτες, και κάθε έγγραφο μπορεί να ανήκει σε μία ή περισσότερες κατηγορίες. Ακολουθεί ένα παράδειγμα του πώς μπορούμε να υπολογίσουμε την απώλεια Hamming για ένα τέτοιο πρόβλημα:

Υποθέτουμε ότι έχουμε ένα σύνολο εγγράφων και τις πραγματικές ετικέτες για κάθε έγγραφο, καθώς και τις προβλεπόμενες ετικέτες από τον κατηγοριοποιητή.

Ας θεωρήσουμε πέντε κατηγορίες (A, B, C, D, E) για τα έγγραφα.

Πραγματικές ετικέτες:

Έγγραφο 1: A, B, D

Έγγραφο 2: B, C, E

Έγγραφο 3: A, C, D

Έγγραφο 4: B, D, E

Προβλεπόμενες ετικέτες:

Έγγραφο 1: A, C, D

Έγγραφο 2: B, E

Έγγραφο 3: A, C

Έγγραφο 4: B, D, E

Τώρα, ας υπολογίσουμε την απώλεια Hamming για κάθε έγγραφο:

Για το Έγγραφο 1: Πραγματικές ετικέτες: A, B, D Προβλεπόμενες ετικέτες: A, C, D Λάθος ετικέτες: B (1 λάθος ετικέτα) Απώλεια Hamming για το Έγγραφο 1: 1/3 = 0.333

Για το Έγγραφο 2: Πραγματικές ετικέτες: B, C, E Προβλεπόμενες ετικέτες: B, E Λάθος ετικέτες: C (1 λάθος ετικέτα) Απώλεια Hamming για το Έγγραφο 2: 1/3 = 0.333

Για το Έγγραφο 3: Πραγματικές ετικέτες: A, C, D Προβλεπόμενες ετικέτες: A, C Λάθος ετικέτες: D (1 λάθος ετικέτα) Απώλεια Hamming για το Έγγραφο 3: 1/3 = 0.333

Για το Έγγραφο 4: Πραγματικές ετικέτες: B, D, E Προβλεπόμενες ετικέτες: B, D, E Λάθος ετικέτες: Καμία Απώλεια Hamming για το Έγγραφο 4: 0/3 = 0

Τώρα, ας υπολογίσουμε τη συνολική απώλεια Hamming για το σύνολο των δεδομένων:

Συνολική απώλεια Hamming = (0.333 + 0.333 + 0.333 + 0) / 4 ≈ 0.25

Έτσι, η απώλεια Hamming για αυτό το πρόβλημα πολλαπλής κατηγοριοποίησης με πέντε κατηγορίες είναι περίπου 0.25.

4.3.4 Accuracy

Η μετρική Accuracy (Ακρίβεια) [21] αποτελεί μία από τις πιο θεμελιώδεις και ευρέως χρησιμοποιούμενες μετρικές απόδοσης στον τομέα της μηχανικής μάθησης και της τεχνητής νοημοσύνης. Αναφέρεται στο ποσοστό των σωστών προβλέψεων που έκανε ένα μοντέλο σε σχέση με το σύνολο των παραδειγμάτων που δοκιμάστηκαν.

Η Ακρίβεια ορίζεται μαθηματικά ως το πηλίκο του αριθμού των σωστών προβλέψεων (σωστά θετικές και σωστά αρνητικές) προς το συνολικό αριθμό των προβλέψεων:

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Σχήμα 9 Τύπος υπολογισμού Accuracy [21]

Η Accuracy μπορεί να πάρει τιμές από 0 έως 1 με το 1 είναι να είναι η βέλτιστη τιμής της ακρίβειας και το 0 η χείριστη.

Έχει κρίσιμη σημασία στην αξιολόγηση και τη σύγκριση μοντέλων μηχανικής μάθησης, καθώς παρέχει έναν απλό και άμεσο τρόπο για την εκτίμηση της απόδοσής τους. Είναι ιδιαίτερα χρήσιμη σε περιπτώσεις όπου οι κλάσεις είναι ισορροπημένες και τα κόστη των ψευδώς θετικών και ψευδώς αρνητικών προβλέψεων είναι παρόμοια.

Παράδειγμα υπολογισμού Accuracy

Στην πολυετικετική κατηγοριοποίηση, η ακρίβεια (accuracy) δεν χρησιμοποιείται συνήθως ως η κύρια μετρική απόδοσης, επειδή δεν αποτυπώνει την πραγματική απόδοση του ταξινομητή όταν εμπλέκονται πολλαπλές ετικέτες. Ωστόσο, η ακρίβεια μπορεί ακόμη να υπολογιστεί για κάθε μεμονωμένη ετικέτα ή για το σύνολο των ετικετών.

Ας δούμε πώς μπορεί να υπολογιστεί η ακρίβεια για κάθε ετικέτα με ένα παράδειγμα:

Υποθέτουμε ότι έχουμε ένα πρόβλημα πολλαπλής κατηγοριοποίησης όπου κατηγοριοποιούμε έγγραφα σε τρεις κατηγορίες: A, B και C. Ας θεωρήσουμε ένα μικρό σύνολο δεδομένων με τέσσερα έγγραφα.

Πραγματικές ετικέτες:

Έγγραφο 1: A, B

Έγγραφο 2: B, C

Έγγραφο 3: A

Έγγραφο 4: C

Προβλεπόμενες ετικέτες:

Έγγραφο 1: A, B

Έγγραφο 2: B

Έγγραφο 3: A, C

Έγγραφο 4: B, C

Για να υπολογίσουμε την ακρίβεια για κάθε ετικέτα:

Για την Ετικέτα A: Θετικά Σωστά (TP_A): 2 (Έγγραφο 1, Έγγραφο 3) Θετικά Λάθος (FP_A): 0 Αρνητικά Λάθος (FN_A): 0 Ακρίβεια_A = TP_A / (TP_A + FP_A + FN_A) = 2 / 2 = 1.0

Για την Ετικέτα B: Θετικά Σωστά (TP_B): 1 (Έγγραφο 1) Θετικά Λάθος (FP_B): 0 Αρνητικά Λάθος (FN_B): 1 (Έγγραφο 2) Ακρίβεια_B = TP_B / (TP_B + FP_B + FN_B) = 1 / 2 = 0.5

Για την Ετικέτα C: Θετικά Σωστά (TP_C): 1 (Έγγραφο 2) Θετικά Λάθος (FP_C): 1 (Έγγραφο 3) Αρνητικά Λάθος (FN_C): 0 Ακρίβεια_C = TP_C / (TP_C + FP_C + FN_C) = 1 / 2 = 0.5

Για να υπολογίσουμε τη συνολική ακρίβεια, υπολογίζουμε την μέση ακρίβεια σε όλες τις ετικέτες:

Συνολική Ακρίβεια = (Ακρίβεια_A + Ακρίβεια_B + Ακρίβεια_C) / 3 = (1.0 + 0.5 + 0.5) / 3 ≈ 0.667

Έτσι, η συνολική ακρίβεια για αυτό το πρόβλημα πολλαπλής κατηγοριοποίησης είναι περίπου 0.667. Ωστόσο, είναι σημαντικό να σημειωθεί ότι η ακρίβεια μπορεί να μην προσφέρει μια πλήρη εικόνα της απόδοσης του μοντέλου, ειδικά σε σενάρια πολλαπλής κατηγοριοποίησης, όπου κάποιες ετικέτες μπορεί να είναι πιο συχνές από άλλες ή όπου η σημασία της σωστής πρόβλεψης ορισμένων ετικετών μπορεί να διαφέρει. Σε τέτοιες περιπτώσεις, ένα μοντέλο μπορεί να εμφανίζει υψηλή Accuracy προβλέποντας πάντα την πιο συχνή κλάση, αγνοώντας τις λιγότερο συχνές και πιθανώς πιο σημαντικές.

4.3.5 Precision

Η μετρική Precision (Ορθότητα) [7] είναι ένας κρίσιμος δείκτης αποτελεσματικότητας στον τομέα της μηχανικής μάθησης, ιδιαίτερα στην ταξινόμηση και την αναγνώριση προτύπων. Αφορά το λόγο μεταξύ των σωστά ταξινομημένων προβλέψεων προς τις συνολικές θετικές προβλέψεις. Με άλλα λόγια, μετράει πόσο "ακριβείς" είναι οι προβλέψεις θετικής κλάσης του μοντέλου.

Η Precision ορίζεται μαθηματικά ως το πηλίκο των True Positives (TP) προς το άθροισμα των True Positives και των False Positives (FP):

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

Σχήμα 10 Τύπος υπολογισμού Precision [22]

Η ορθότητα είναι ιδιαίτερα σημαντική σε εφαρμογές όπου το κόστος των ψευδώς θετικών προβλέψεων είναι υψηλό. Για παράδειγμα, σε ένα σύστημα ανίχνευσης απάτης, ένα ψευδώς θετικό αποτέλεσμα μπορεί να σημαίνει την άδικη αναστολή ενός λογαριασμού χρήστη, ενώ στην ιατρική διάγνωση, ένα ψευδώς θετικό μπορεί να οδηγήσει σε άσκοπες, επακόλουθες ιατρικές εξετάσεις.

Η ορθότητα δεν πρέπει να εξετάζεται μόνη της. Ένα μοντέλο μπορεί να έχει υψηλή ορθότητα αλλά να είναι πολύ επιλεκτικό στις προβλέψεις του, αγνοώντας πολλές θετικές περιπτώσεις (True Positives) και έτσι να έχει χαμηλή ανάκληση (Recall). Για αυτό, συχνά συνδυάζεται με άλλες μετρικές, όπως το Recall, για να προσφέρει μια πιο ολοκληρωμένη εικόνα της απόδοσης του μοντέλου.

Παράδειγμα υπολογισμού Precision

Η Ορθότητα (Precision) στην κατηγοριοποίηση πολλαπλών ετικετών μετρά το ποσοστό των σωστά προβλεπόμενων θετικών παρατηρήσεων προς το σύνολο των προβλεπόμενων θετικών για κάθε ετικέτα. Είναι μια σημαντική μετρική, ιδιαίτερα όταν το κόστος ενός ψευδώς θετικού είναι υψηλό. Ακολουθεί ένα παράδειγμα για να εξηγήσουμε πώς υπολογίζεται η ακρίβεια σε ένα περιβάλλον πολυετικετικής κατηγοριοποίησης:

Ας υποθέσουμε ότι έχουμε ένα πρόβλημα κατηγοριοποίησης πολλαπλών ετικετών όπου κατηγοριοποιούμε αναρτήσεις ιστολογίων σε τρεις κατηγορίες: Τεχνολογία (T), Ψυχαγωγία (E) και Αθλητισμός (S). Ας θεωρήσουμε ένα μικρό σύνολο δεδομένων με τρεις αναρτήσεις ιστολογίων.

Πραγματικές Ετικέτες:

Ανάρτηση 1: T, E

Ανάρτηση 2: E, S

Ανάρτηση 3: T, S

Προβλεπόμενες Ετικέτες:

Ανάρτηση 1: T

Ανάρτηση 2: E, S

Ανάρτηση 3: T, E

Τώρα, ας υπολογίσουμε την ακρίβεια για κάθε ετικέτα.

Για την Ετικέτα T (Τεχνολογία):

Θετικά Σωστά (TP_T): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε σωστά την Τ. Σε αυτή την περίπτωση, TP_T = 2 (Ανάρτηση 1 και 3 προσδιορίζονται σωστά ως Τ).

Θετικά Λάθος (FP_T): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε λανθασμένα την Τ όταν δεν είναι η πραγματική ετικέτα. Σε αυτή την περίπτωση, FP_T = 0 (Δεν υπάρχουν λανθασμένες προβλέψεις για Τ).

Precision_T = TP_T / (TP_T + FP_T) = 2 / (2 + 0) = 1.0

Για την Ετικέτα Ε (Ψυχαγωγία):

Θετικά Σωστά (TP_E): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε σωστά την Ε. Σε αυτή την περίπτωση, TP_E = 1 (Ανάρτηση 2 προσδιορίζεται σωστά ως Ε).

Θετικά Λάθος (FP_E): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε λανθασμένα την Ε όταν δεν είναι η πραγματική ετικέτα. Σε αυτή την περίπτωση, FP_E = 1 (Ανάρτηση 3 προβλέπεται λανθασμένα ως Ε).

Precision_E = TP_E / (TP_E + FP_E) = 1 / (1 + 1) = 0.5

Για την Ετικέτα S (Αθλητισμός):

Θετικά Σωστά (TP_S): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε σωστά την S. Σε αυτή την περίπτωση, TP_S = 1 (Ανάρτηση 2 προσδιορίζεται σωστά ως S).

Θετικά Λάθος (FP_S): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε λανθασμένα την S όταν δεν είναι η πραγματική ετικέτα. Σε αυτή την περίπτωση, FP_S = 0 (Δεν υπάρχουν λανθασμένες προβλέψεις για S).

Precision_S = TP_S / (TP_S + FP_S) = 1 / (1 + 0) = 1.0

Για να πάρουμε μια συνολική έννοια της ορθότητας (Precision) σε όλες τις ετικέτες, μπορούμε να υπολογίσουμε τη μέση ορθότητα για κάθε ετικέτα:

Συνολικό Precision = (Precision_T + Precision_E + Precision_S) / 3 = (1.0 + 0.5 + 1.0) / 3 ≈ 0.83

Επομένως, η συνολική ορθότητα για αυτό το πρόβλημα πολλαπλής κατηγοριοποίησης είναι περίπου 0.83. Αυτό το παράδειγμα δείχνει πώς η ορθότητα μπορεί να προσφέρει ενδείξεις για την ικανότητα του ταξινομητή να προβλέπει σωστά θετικές ετικέτες ενώ αποφεύγει τα ψευδώς θετικά.

4.3.6 Recall

Η μετρική Recall [7] , επίσης γνωστή ως ευαισθησία ή ανάκληση, είναι ένας βασικός δείκτης απόδοσης στον τομέα της μηχανικής μάθησης, ιδιαίτερα σε εφαρμογές ταξινόμησης και

αναγνώρισης προτύπων. Η Recall μετρά το ποσοστό των πραγματικών θετικών περιπτώσεων που έχουν προβλεφθεί σωστά από το σύνολο των πραγματικών θετικών στα δεδομένα. Αυτό σημαίνει ότι αξιολογεί πόσο καλά το μοντέλο μπορεί να αναγνωρίσει τις θετικές περιπτώσεις, ανεξάρτητα από το πόσες θετικές προβλέψεις έκανε συνολικά.

Η Recall ορίζεται μαθηματικά ως το πηλίκο των True Positives (TP) προς το άθροισμα των True Positives και των False Negatives (FN):

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

Σχήμα 11 Τύπος υπολογισμού Recall [22]

Η σημασία της Recall είναι κρίσιμη σε εφαρμογές όπου οι συνέπειες της χαμηλής ανίχνευσης των θετικών περιπτώσεων είναι σοβαρές. Για παράδειγμα, στην ιατρική διάγνωση, ένα υψηλό Recall είναι ζωτικής σημασίας για την ανίχνευση όλων των πιθανών περιπτώσεων μιας ασθένειας, ενώ σε συστήματα ασφαλείας, είναι σημαντικό για την ανίχνευση όλων των απειλών.

Ένας περιορισμός της Recall είναι ότι δεν λαμβάνει υπόψη τις ψευδώς θετικές προβλέψεις (False Positives). Ένα μοντέλο μπορεί να έχει υψηλή Recall προβλέποντας την πλειονότητα των περιπτώσεων ως θετικές, αλλά αυτό μπορεί να οδηγήσει σε πολλά λάθη. Γι' αυτό συχνά συνδυάζεται με την Precision για μια πιο ισορροπημένη εκτίμηση της απόδοσης του μοντέλου.

Παράδειγμα υπολογισμού Recall

Ας δούμε ένα παράδειγμα για να εξηγήσουμε πώς υπολογίζεται η ανάκληση (Recall) σε ένα περιβάλλον κατηγοριοποίησης πολλαπλών ετικετών:

Ας υποθέσουμε ότι εργαζόμαστε πάνω σε ένα πρόβλημα κατηγοριοποίησης πολλαπλών ετικετών όπου στόχος μας είναι να κατηγοριοποιήσουμε εικόνες με βάση τα αντικείμενα που περιέχουν. Οι δυνατές κατηγορίες είναι: Ζώο (A), Κτίριο (B) και Αυτοκίνητο (C). Ας θεωρήσουμε ένα σύνολο δεδομένων με τέσσερις εικόνες.

Πραγματικές Ετικέτες:

Εικόνα 1: A, C

Εικόνα 2: B, C

Εικόνα 3: A, B

Εικόνα 4: C

Εικόνα 4: C

Προβλεπόμενες Ετικέτες:

Εικόνα 1: A

Εικόνα 2: B, C

Εικόνα 3: B

Εικόνα 4: A, C

Υπολογισμός Ανάκλησης για Κάθε Ετικέτα:

Για την Ετικέτα A (Ζώο):

Θετικά Σωστά (TP_A): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε σωστά Α. Σε αυτή την περίπτωση, TP_A = 1 (Η εικόνα 1 προσδιορίζεται σωστά ως περιέχουσα Ζώο).

Αρνητικά Λάθος (FN_A): Ο αριθμός των φορών που ο ταξινομητής αποτυγχάνει να προβλέψει Α όταν πράγματι είναι παρόν. Σε αυτή την περίπτωση, FN_A = 1 (Η εικόνα 3 περιέχει Ζώο, αλλά δεν προβλέφθηκε).

Ανάκληση_A = TP_A / (TP_A + FN_A) = 1 / (1 + 1) = 0.5

Για την Ετικέτα B (Κτίριο):

Θετικά Σωστά (TP_B): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε σωστά Β. Σε αυτή την περίπτωση, TP_B = 2 (Οι εικόνες 2 και 3 προσδιορίζονται σωστά ως περιέχουσες Κτίριο).

Αρνητικά Λάθος (FN_B): Ο αριθμός των φορών που ο ταξινομητής αποτυγχάνει να προβλέψει Β όταν πράγματι είναι παρόν. Σε αυτή την περίπτωση, FN_B = 0 (Όλα τα Κτίρια προβλέπονται σωστά).

Ανάκληση_B = TP_B / (TP_B + FN_B) = 2 / (2 + 0) = 1.0

Για την Ετικέτα C (Αυτοκίνητο):

Θετικά Σωστά (TP_C): Ο αριθμός των φορών που ο ταξινομητής προέβλεψε σωστά C. Σε αυτή την περίπτωση, TP_C = 2 (Οι εικόνες 2 και 4 προσδιορίζονται σωστά ως περιέχουσες Αυτοκίνητο).

Αρνητικά Λάθος (FN_C): Ο αριθμός των φορών που ο ταξινομητής αποτυγχάνει να προβλέψει C όταν πράγματι είναι παρόν. Σε αυτή την περίπτωση, FN_C = 1 (Η εικόνα 1 περιέχει Αυτοκίνητο, αλλά δεν προβλέφθηκε).

Ανάκληση_C = TP_C / (TP_C + FN_C) = 2 / (2 + 1) ≈ 0.67

Για να αποκτήσουμε μια συνολική εικόνα της τιμής της ανάκλησης σε όλες τις ετικέτες, μπορούμε να υπολογίσουμε τον μέσο όρο των σκορ ανάκλησης για κάθε ετικέτα:

Συνολική Ανάκληση = (Ανάκληση_A + Ανάκληση_B + Ανάκληση_C) / 3 = (0.5 + 1.0 + 0.67) / 3 ≈ 0.72

Επομένως, η συνολική ανάκληση για αυτό το πρόβλημα κατηγοριοποίησης πολλαπλών ετικετών είναι περίπου 0.72. Αυτό το παράδειγμα δείχνει πώς η ανάκληση είναι μια κρίσιμη μετρική στην αξιολόγηση της ικανότητας ενός ταξινομητή να αναγνωρίζει όλες τις σχετικές περιπτώσεις κάθε κατηγορίας, ιδιαίτερα σε περιβάλλοντα κατηγοριοποίησης πολλαπλών ετικετών όπου κάθε περίπτωση μπορεί να ανήκει σε πολλαπλές ετικέτες.

4.3.7 Κώδικας (Python) πειραματικής μελέτης

Ο παρακάτω κώδικας είναι ενδεικτικός και υλοποιήθηκε για την πειραματική μελέτη που έγινε και αφορά στον αλγόριθμο BRkNNaClassifier και στο Mulan dataset “emotions”. Τροποποιώντας τον παρακάτω κώδικα αντικαθιστώντας το όνομα του dataset και του αλγορίθμου υλοποιήθηκε η μελέτη και για τα υπόλοιπα επιλεγμένα σύνολα δεδομένων του Mulan (Κεφάλαιο 4.1) και τους αλγορίθμους κατηγοριοποίησης (Κεφάλαιο 3). Αρχικά το dataset ‘emotions’ της Python, φορτώνεται με χρήση της συνάρτησης load_dataset και με την ένδειξη ‘undivided’ ώστε αρχικά να μην γίνει διαχωρισμός σε δεδομένα εκπαίδευσης και δοκιμής. Στη συνέχεια, η συνάρτηση train_test_split της βιβλιοθήκης sklearn.model_selection, χρησιμοποιείται για να διαχωρίσει τα δεδομένα σε σύνολα εκπαίδευσης (train) και δοκιμής (test), δεσμεύοντας το 20% των δειγμάτων ως σύνολο δοκιμής, ενώ το υπόλοιπο 80% θα αποτελέσει το σύνολο εκπαίδευσης. Ο ταξινομητής BRkNNaClassifier ορίζεται χωρίς παραμέτρους (default k=10. Κατόπιν, ξεκινάει η καταμέτρηση του χρόνου εκπαίδευσης (Training CPU Time), αμέσως ακολουθεί η εκπαίδευση του μοντέλου με τη συνάρτηση fit και έπειτα ολοκληρώνεται η καταμέτρηση του χρόνου εκπαίδευσης και τυπώνεται ο καταγεγραμμένος χρόνος. Ακολουθεί η έναρξη της καταμέτρησης του χρόνου δοκιμής (Testing CPU Time), καλείται αμέσως μετά η συνάρτηση predict του ταξινομητή και στη συνέχεια τερματίζεται και τυπώνεται η χρονική καταμέτρηση του Testing CPU Time. Τέλος, υπολογίζονται και τυπώνονται οι προβλέψεις για τις μετρικές Hamming Loss, Accuracy, Precision, Recall.

```
from skmultilearn.adapt import BRkNNaClassifier
from skmultilearn.dataset import load_dataset
from sklearn.metrics import hamming_loss, accuracy_score, precision_score, recall_score
```

```
from sklearn.model_selection import train_test_split
import time
# Φόρτωση του dataset - χωρίς διαχωρισμό σε train/set
X, y, _, _ = load_dataset('emotions', 'undivided')
# Διαχωρισμός του σετ δεδομένων σε εκπαίδευση και δοκιμή
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
# Αρχικοποίηση του BRkNNaClassifier
classifier = BRkNNaClassifier()
# Αρχική χρονική σήμανση για την εκπαίδευση
start_train = time.time()
# Εκπαίδευση του μοντέλου
classifier.fit(X_train, y_train)
# Τερματική χρονική σήμανση για την εκπαίδευση
end_train = time.time()
# Χρόνος εκπαίδευσης (CPU time)
train_time = end_train - start_train
print("Training CPU Time:", train_time)
# Αρχική χρονική σήμανση για την πρόβλεψη του test set
start_test = time.time()
# Προβλέψεις με το εκπαιδευμένο μοντέλο
y_pred = classifier.predict(X_test)
# Τερματική χρονική σήμανση για την εκπαίδευση
end_test = time.time()
# Χρόνος εκπαίδευσης (CPU time)
test_time = end_test - start_test
print("Testing CPU Time:", test_time)
# Υπολογισμός των μετρικών
hamming_loss_value = hamming_loss(y_test, y_pred)
accuracy_value = accuracy_score(y_test, y_pred)
precision_value = precision_score(y_test, y_pred, average='micro')
recall_value = recall_score(y_test, y_pred, average='micro')
# Εμφάνιση των αποτελεσμάτων
```

```
print("Hamming Loss:", hamming_loss_value)
print("Accuracy:", accuracy_value)
print("Precision:", precision_value)
print("Recall:", recall_value)
```

Πίνακας 2 Πειραματικές μετρήσεις

Dataset	Metrics	BRkNNa	BRkNNb	MLkNN	MLARAM	MLTSVM
bibtex	*Training CPU Time*	0.009835958	0.010002613	32.17414284	3.762850046	2324.319849
	Testing CPU Time	0.866207123	1.265366793	4.887376785	98.12754154	0.061014175
	Hamming loss	0.014769953	0.029521331	0.014942499	0.021725541	0.035935835
	Accuracy	0.030178326	0.000686342	0.068587106	0.181879648	0
	Precision	0.785276074	0.018727705	0.565377532	0.293391617	0.239601576
	Recall	0.070914127	0.01749514	0.170271769	0.325813821	0.60687552
birds	*Training CPU Time*	0.001999855	0.001991034	0.500113726	0.052024364	4.724068403
	Testing CPU Time	0.039019108	0.092019081	0.112034082	0.071003914	0.002000332
	Hamming loss	0.053855569	0.560179519	0.055895553	0.1374949	0.329661363
	Accuracy	0.480620155	0.015503876	0.488372093	0.007751938	0
	Precision	0.666666667	0.01486698	0.375	0.138181818	0.119369369
	Recall	0.015037594	0.142857143	0.045112782	0.275362319	0.803030303
delicious	*Training CPU Time*	0.015003204	0.013992071	385.4578483	8.830998659	4254.692265
	Testing CPU Time	3.306737661	5.806315184	53.10002065	328.8166749	1.113556533
	Hamming loss	0.018287387	0.037725993	0.018669758	0.020182279	0.020162283
	Accuracy	0.003771395	0	0.002636204	0.011487116	0
	Precision	0.633640553	0.019600872	0.550746775	0.426455289	0.615883422
	Recall	0.110032163	0.019673556	0.151049667	0.141269919	0.915153454
emotions	*Training CPU Time*	0.009992123	0.004010916	0.227051497	0.039008379	0.48110795
	Testing CPU Time	0.037007809	0.387086153	0.085018873	0.015003681	0.002000093
	Hamming loss	0.260928962	0.343395099	0.265027322	0.341736695	0.490437158
	Accuracy	0.204918033	0	0.180327869	0.092436975	1
	Precision	0.614285714	0.443682664	0.596026491	0.449339207	0.381944444
	Recall	0.385650224	0.438104449	0.403587444	0.461538462	0.986547085
enron	*Training CPU Time*	0.004008532	0.004000425	2.551587105	2.869650602	111.1401711
	Testing CPU Time	0.098011971	0.172038317	0.455093384	3.472650051	0.002991199
	Hamming loss	0.062079809	0.107422825	0.055513142	0.072594478	0.137985253
	Accuracy	0.051724138	0	0.066473988	0.073313783	0

	Precision	0.618257261	0.015834348	0.640127389	0.260997067	0.292826221
	Recall	0.123960067	0.010906047	0.336683417	0.07745866	0.785892116
genbase	*Training CPU Time*	0.011002064	0.011002541	1.078253269	0.062013865	184.5755072
	Testing CPU Time	0.208061695	0.298067093	0.362085342	0.061014414	0.015003204
	Hamming loss	0.042974272	0.088210348	0.045518801	0.067947647	0.158608991
	Accuracy	0.20610687	0	0.213740458	0.210526316	1
	Precision	0.72972973	0.006896552	0.583333333	0.210526316	0.230769231
	Recall	0.159763314	0.005917160	0.165680473	0.167664671	0.994082840
mediamill	*Training CPU Time*	0.063015461	0.064023972	577.4717543	73.16805124	23455.54546
	Testing CPU Time	115.7376857	124.233608	134.3885059	211.1497626	1.531736784
	Hamming loss	0.029954385	0.085217062	0.030235112	0.033920643	0.033134734
	Accuracy	0.135048964	0	0.133796402	0.065247096	0
	Precision	0.765446572	0.053911231	0.732940218	0.721048599	0.735343441
	Recall	0.454402516	0.057347670	0.486594380	0.345211581	0.826564657
medical	*Training CPU Time*	0.001000404	0.002000809	1.253283262	0.104023218	31.36909986
	Testing CPU Time	0.038999081	0.080018282	0.227061749	0.301068544	0.003000736
	Hamming loss	0.016585366	0.048563686	0.013550136	0.021995465	0.073604336
	Accuracy	0.4	0.024390244	0.497560976	0.443877551	1
	Precision	0.838461538	0.04	0.808510638	0.587121212	0.258810573
	Recall	0.452282158	0.037344398	0.630705394	0.645833333	0.975103734
scene	*Training CPU Time*	0.011012793	0.008001804	3.952062845	3.213727951	42.57958961
	Testing CPU Time	0.820186615	0.851192951	1.06008029	3.641824961	0.008001566
	Hamming loss	0.094744122	0.284578147	0.088520055	0.104426003	0.487551867
	Accuracy	0.599585062	0.170124481	0.624481328	0.605809129	1
	Precision	0.824	0.185185185	0.793258427	0.684124386	0.267326733
	Recall	0.597678917	0.174081238	0.6827853	0.79316888	0.992263056
yeast	*Training CPU Time*	0.005999565	0.004010916	2.163490057	0.655147791	23.3276639
	Testing CPU Time	0.380087376	0.387086153	0.490125179	0.798181057	0.004001141
	Hamming loss	0.20481203	0.343395099	0.209022556	0.222255018	0.651991614
	Accuracy	0.176842105	0	0.178947368	0.227272727	0
	Precision	0.73337766	0.443682664	0.705167173	0.631736527	0.321055096
	Recall	0.534399225	0.438104449	0.561743341	0.622418879	0.998059195

Από τις παραπάνω πειραματικές μετρήσεις εύλογα προκύπτουν τα εξής συμπεράσματα για τους αλγορίθμους αναφορικά με τις μετρικές.

BRkNNa

Training CPU Time: Έχει τον χαμηλότερο μέσο χρόνο εκπαίδευσης, κάτι που τον καθιστά ιδιαίτερα αποδοτικό για ταχεία εκπαίδευση σε σύνολα δεδομένων.

Testing CPU Time: Έχει μεγαλύτερο χρόνο δοκιμής από ότι εκπαίδευσης, που ωστόσο κυμαίνεται σε χαμηλές τιμές.

Hamming Loss: Παρουσιάζει μια από τις χαμηλότερες τιμές, υποδηλώνοντας καλή απόδοση στην πρόβλεψη ετικετών.

Accuracy, Precision, Recall: Έχει ισορροπημένες τιμές σε αυτές τις μετρικές, δείχνοντας ένα καλό επίπεδο αποδοτικότητας και αξιοπιστίας στις προβλέψεις του. Επιπλέον, παρουσιάζει το μεγαλύτερο Precision σε σχέση τους υπόλοιπους αλγορίθμους.

BRkNNb

Training CPU Time: Έχει χαμηλό χρόνο εκπαίδευσης, αλλά λίγο υψηλότερο σε σύγκριση με τον BRkNNa.

Testing CPU Time: Έχει μεγαλύτερο χρόνο δοκιμής από ότι εκπαίδευσης, που ωστόσο κυμαίνεται σε χαμηλές τιμές, επίσης λίγο υψηλότερες από τις αντίστοιχες του BRkNNa.

Hamming Loss: Η απόδοσή του είναι σημαντικά χειρότερη σε σύγκριση με τον BRkNNa, υποδηλώνοντας πιθανά περισσότερα λάθη στην πρόβλεψη ετικετών.

Accuracy, Precision, Recall: Οι τιμές του σε αυτές τις μετρικές είναι χαμηλότερες, υποδηλώνοντας πως ο αλγόριθμος αυτός ίσως δεν είναι η καλύτερη επιλογή για εφαρμογές που απαιτούν υψηλή ακρίβεια και ανάκληση.

MLkNN

Training CPU Time: Παρουσιάζει σημαντικά υψηλότερο χρόνο εκπαίδευσης σε σύγκριση με τους BRkNN αλγορίθμους, υποδηλώνοντας μεγαλύτερη υπολογιστική απαίτηση.

Testing CPU Time: Αντίστοιχα, έχει και μεγαλύτερο χρόνο δοκιμής σε σύγκριση με τους BRkNN αλγορίθμους.

Hamming Loss: Έχει παρόμοια απόδοση με τον BRkNNa, δείχνοντας καλή ικανότητα στην πρόβλεψη ετικετών.

Accuracy, Precision, Recall: Οι τιμές του σε αυτές τις μετρικές είναι σχετικά καλές, αρκετά κοντά στις αντίστοιχες του BRkNNa.

MLARAM

Training CPU Time: Έχει μέτριο χρόνο εκπαίδευσης, χαμηλότερο από τον MLkNN αλλά υψηλότερο από τους BRkNN αλγορίθμους.

Testing CPU Time: Έχει το μεγαλύτερο χρόνο δοκιμής από όλους τους αλγορίθμους που υποδηλώνει πιθανώς μεγαλύτερη πολυπλοκότητα στην αξιολόγηση των μοντέλων.

Hamming Loss: Δεν ξεχωρίζει ιδιαίτερα σε αυτή τη μετρική, έχοντας μια μέση απόδοση.

Accuracy, Precision, Recall: Οι τιμές του δείχνουν μέτρια απόδοση, κάνοντάς τον κατάλληλο για εφαρμογές που δεν απαιτούν την υψηλότερη δυνατή ακρίβεια ή ανάκληση.

MLTSVM

Training CPU Time: Παρουσιάζει τον υψηλότερο χρόνο εκπαίδευσης, κάτι που τον καθιστά τον πιο υπολογιστικά απαιτητικό αλγόριθμο.

Testing CPU Time: Έχει ωστόσο και το μικρότερο χρόνο δοκιμής από όλους τους αλγορίθμους.

Hamming Loss: Έχει την χειρότερη απόδοση σε σύγκριση με τους άλλους αλγορίθμους, υποδηλώνοντας περισσότερα λάθη στην πρόβλεψη.

Accuracy, Precision, Recall: Παρά την χαμηλή ακρίβεια και ορθότητα, έχει την υψηλότερη ανάκληση, υποδηλώνοντας ότι είναι πολύ καλός στο να εντοπίζει θετικές περιπτώσεις αλλά με κάποιο κόστος στο να προβλέπει εσφαλμένα θετικά.

Συμπερασματικά, κάθε αλγόριθμος παρουσιάζει έναν διαφορετικό συμβιβασμό μεταξύ χρόνου εκπαίδευσης, απόδοσης στις διάφορες μετρικές αξιολόγησης και υπολογιστικής απαιτητικότητας. Η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από τις συγκεκριμένες απαιτήσεις της εφαρμογής και την ισορροπία μεταξύ απόδοσης και υπολογιστικού κόστους.

4.4 Στατιστικοί έλεγχοι

Το Friedman test [28] είναι μια μη παραμετρική στατιστική δοκιμασία που χρησιμοποιείται για τη σύγκριση περισσότερων των δύο (k>=2) συνδεδεμένων δειγμάτων. Ιδιαίτερα χρήσιμο σε πειράματα όπου οι μετρήσεις επαναλαμβάνονται στις ίδιες μονάδες, το Friedman test βοηθά στην ανίχνευση διαφορών στις μετρήσεις μεταξύ των ομάδων. Για την υλοποίηση του Friedman τεστ χρησιμοποιήθηκε το εργαλείο PSPP.

Το PSPP είναι ένα στατιστικό λογισμικό που χρησιμοποιείται για την ανάλυση δεδομένων. Αποτελεί μια ελεύθερη και ανοιχτού κώδικα εναλλακτική στο SPSS (Statistical Package for

the Social Sciences), ένα δημοφιλές εμπορικό πρόγραμμα στατιστικής ανάλυσης. Σχεδιασμένο για εύκολη χρήση, το PSPP προσφέρει μια πληθώρα στατιστικών δυνατοτήτων, καθιστώντας το κατάλληλο για ερευνητές, εκπαιδευτικούς, φοιτητές, και κάθε άλλον που ασχολείται με την ανάλυση δεδομένων.

Πίνακας 3 Στατιστική συγκριτική ανάλυση Friedman (ΜΟ Κατάταξης μετρικών)

Αλγόριθμος	*Training CPU Time*	*Testing CPU Time*	*Hamming Loss*	*Accuracy*	*Precision*	*Recall*
BRkNNa	1.60	2.20	1.30	3.20	4.70	2.00
BRkNNb	1.40	3.40	4.30	1.50	1.20	1.40
MLkNN	3.90	4.20	1.70	3.90	3.90	3.10
MLARAM	3.10	4.20	3.20	3.60	2.60	3.50
MLTSVM	5.00	1.00	4.50	2.80	2.60	5.00

Από τα παραπάνω αποτελέσματα του Friedman τεστ προκύπτουν τα εξής συμπεράσματα:

- Ο BRkNNa παρουσιάζει το μικρότερο Training CPU Time γεγονός που τον καθιστά τον πιο γρήγορο αλγόριθμο αναφορικά με το χρόνο εκπαίδευσης. Επιπλέον έχει και το χαμηλότερο Hamming Loss προβλέποντας τις λιγότερες λάθος ετικέτες, αλλά και το μεγαλύτερο Precision. Όλα τα παραπάνω καθιστούν τον BRkNNa έναν αλγόριθμο ικανοποιητικό τόσο σε προβλέψεις όσο και σε υπολογιστικούς πόρους.
- Από την άλλη, ο MLTSVM είναι ο πιο αργός όσον αφορά το χρόνο εκπαίδευσης και ο πιο γρήγορος αναφορικά με το χρόνο δοκιμής. Παρέχει το χειρότερο Hamming Loss και ταυτόχρονα το καλύτερο Recall με τα Accuracy και Precision να έχουν μια μέση τιμή.
- Ο BRkNNb είναι αρκετά κοντά στην απόδοση του BRkNNa και διαφοροποιείται με μεγαλύτερες τιμές στα Testing CPU Time, Hamming Loss και με αρκετά μικρότερες τιμές στα Accuracy, Precision, Recall.
- Τέλος, ο MLkNN είναι αρκετά κοντά στις επιδόσεις με τον MLARAM με τη διαφορά ότι έχει μικρότερο και άρα καλύτερο Hamming Loss.

Πέρα από το Friedman test, υλοποιήθηκε η στατιστική συγκριτική ανάλυση και με την εφαρμογή του Wilcoxon test.

Στον τομέα της επιστημονικής έρευνας, είναι συχνά αναγκαίο να συγκρίνουμε δύο συνδεδεμένα δείγματα ή μετρήσεις για να κατανοήσουμε αν υπάρχει σημαντική διαφορά μεταξύ τους. Το Wilcoxon Signed Rank Test [28] αποτελεί μια στατιστική δοκιμασία που

εφαρμόζεται για τον σκοπό αυτό, ειδικά όταν οι διανομές των δειγμάτων δεν ακολουθούν την κανονική κατανομή. Είναι μια μη παραμετρική δοκιμασία που χρησιμοποιείται για να συγκρίνει τα μέσα δύο συνδεδεμένων δειγμάτων, παρέχοντας μια αποτελεσματική μέθοδο για την ανάλυση της στατιστικής σημασίας των διαφορών τους.

Η διαδικασία του Wilcoxon Signed Rank Test ακολουθεί μερικά βασικά βήματα:

Διαφορές Ζευγών: Για κάθε ζεύγος παρατηρήσεων, υπολογίζεται η διαφορά τους.

Κατάταξη Διαφορών: Οι απόλυτες τιμές των διαφορών κατατάσσονται από τη μικρότερη στη μεγαλύτερη.

Υπολογισμός Στατιστικής Τιμής: Συνυπολογίζοντας το πρόσημο κάθε διαφοράς, υπολογίζεται ένα συνολικό άθροισμα για τις κατατάξεις των θετικών και των αρνητικών διαφορών.

Το αποτέλεσμα του τεστ εκφράζεται μέσω μιας στατιστικής τιμής (συνήθως συμβολίζεται ως W). Η σημασία της διαφοράς μεταξύ των δύο δειγμάτων κρίνεται με βάση τη σύγκριση της τιμής W με έναν πίνακα κρίσιμων τιμών για το Wilcoxon test, ανάλογα με το επίπεδο σημαντικότητας (α) που έχει οριστεί (συνήθως 0.05 ή 5%). Εάν η τιμή W είναι μικρότερη από την κρίσιμη τιμή, τότε η διαφορά θεωρείται στατιστικά σημαντική.

Στην πειραματική μελέτη των αλγορίθμων ή των τεχνικών επεξεργασίας δεδομένων, το Wilcoxon Signed Rank Test χρησιμοποιείται για να συμπληρώσει την ανάλυση, επιβεβαιώνοντας στατιστικά την έγκυρη αξιολόγηση των αποτελεσμάτων. Για παράδειγμα, στην αξιολόγηση διαφορετικών αλγορίθμων ως προς την ακρίβεια ή τον χρόνο εκτέλεσης, το τεστ αυτό παρέχει έναν αξιόπιστο τρόπο για την εκτίμηση της σημασίας των παρατηρούμενων διαφορών, βοηθώντας στην κατανόηση της πραγματικής αποδοτικότητας και απόδοσης των μεθόδων που εξετάζονται.

Πίνακας 4 Στατιστικό τεστ Wilcoxon βάσει της τιμής Training CPU Time

Αλγόριθμοι ανά ζεύγη	*Αποτελέσματα*	*N*	*ΜΟ Κατάταξης*	*Άθροισμα των Κατατάξεων*	*Σημαντικότητα*
BRkNNb - BRkNNa	Ήττες	6	6.50	39.00	.241
	Νίκες	4	4.00	16.00	
	Ισοπαλίες	0			
	Σύνολο	10			
	Ήττες	0	NaN	.00	.005

MLkNN - BRkNNa	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM- BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM- BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM- MLkNN	Ήττες	9	5.89	53.00	.009
	Νίκες	1	2.00	2.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLkNN	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			

	Σύνολο	10			
MLTSVM - MLARAM	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			

Πίνακας 5 Στατιστικό τεστ Wilcoxon βάσει της τιμής Testing CPU Time

Αλγόριθμοι ανά ζεύγη	*Αποτελέσματα*	*N*	*ΜΟ Κατάταξης*	*Άθροισμα των Κατατάξεων*	*Σημαντικότητα*
BRkNNb - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM - BRkNNa	Ήττες	2	2.00	4.00	.017
	Νίκες	8	6.38	51.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - BRkNNa	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNb	Ήττες	1	7.00	7.00	.037
	Νίκες	9	5.33	48.00	
	Ισοπαλίες	0			
	Σύνολο	10			

MLARAM-BRkNNb	Ήττες	3	2.67	8.00	.047
	Νίκες	7	6.71	47.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM-BRkNNb	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM-MLkNN	Ήττες	3	2.33	7.00	.037
	Νίκες	7	6.86	48.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLkNN	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLARAM	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			

Πίνακας 6 Στατιστικό τεστ Wilcoxon βάσει της τιμής Hamming Loss

Αλγόριθμοι ανά ζεύγη	*Αποτελέσματα*	*N*	*ΜΟ Κατάταξης*	*Άθροισμα των Κατατάξεων*	*Σημαντικότητα*
BRkNNb - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
	Ήττες	3	8.33	25.00	.799

MLkNN - BRkNNa	Νίκες	7	4.29	30.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNb	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM- BRkNNb	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM- BRkNNb	Ήττες	3	5.33	16.00	.241
	Νίκες	7	5.57	39.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM- MLkNN	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLkNN	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			

	Σύνολο	10			
MLTSVM - MLARAM	Ήττες	2	1.50	3.00	.013
	Νίκες	8	6.50	52.00	
	Ισοπαλίες	0			
	Σύνολο	10			

Πίνακας 7 Στατιστικό τεστ Wilcoxon βάσει της τιμής Accuracy

Αλγόριθμοι ανά ζεύγη	*Αποτελέσματα*	*N*	*ΜΟ Κατάταξης*	*Άθροισμα των Κατατάξεων*	*Σημαντικότητα*
BRkNNb - BRkNNa	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNa	Ήττες	3	3.33	10.00	.074
	Νίκες	7	6.43	45.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM - BRkNNa	Ήττες	3	8.33	25.00	.799
	Νίκες	7	4.29	30.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - BRkNNa	Ήττες	6	3.67	22.00	0.575
	Νίκες	4	8.25	33.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			

MLARAM-BRkNNb	Ήττες	1	1.00	1.00	.007
	Νίκες	9	6.00	54.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM-BRkNNb	Ήττες	2	1.50	3.00	.115
	Νίκες	4	4.50	18.00	
	Ισοπαλίες	4			
	Σύνολο	10			
MLARAM-MLkNN	Ήττες	6	6.00	36.00	.386
	Νίκες	4	4.75	19.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLkNN	Ήττες	6	3.67	22.00	0.575
	Νίκες	4	8.25	33.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLARAM	Ήττες	6	3.50	21.00	0.508
	Νίκες	4	8.50	34.00	
	Ισοπαλίες	0			
	Σύνολο	10			

Πίνακας 8 Στατιστικό τεστ Wilcoxon βάσει της τιμής Precision

Αλγόριθμοι ανά ζεύγη	*Αποτελέσματα*	*N*	*ΜΟ Κατάταξης*	*Άθροισμα των Κατατάξεων*	*Σημαντικότητα*
BRkNNb - BRkNNa	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
	Ήττες	9	5.89	53.00	.009

MLkNN - BRkNNa	Νίκες	1	2.00	2.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM - BRkNNa	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - BRkNNa	Ήττες	8	6.50	52.00	0.13
	Νίκες	2	1.50	3.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM- BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM- BRkNNb	Ήττες	2	2.50	5.00	.022
	Νίκες	8	6.25	50.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM- MLkNN	Ήττες	10	5.50	55.00	.005
	Νίκες	0	NaN	.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLkNN	Ήττες	8	6.50	52.00	0.13
	Νίκες	2	1.50	3.00	
	Ισοπαλίες	0			

	Σύνολο	10			
MLTSVM - MLARAM	Ήττες	6	6.33	38.00	0.285
	Νίκες	4	4.25	17.00	
	Ισοπαλίες	0			
	Σύνολο	10			

Πίνακας 9 Στατιστικό τεστ Wilcoxon βάσει της τιμής Recall

Αλγόριθμοι ανά ζεύγη	*Αποτελέσματα*	*N*	*ΜΟ Κατάταξης*	*Άθροισμα των Κατατάξεων*	*Σημαντικότητα*
BRkNNb - BRkNNa	Ήττες	8	6.00	48.00	.037
	Νίκες	2	3.50	7.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM - BRkNNa	Ήττες	2	4.50	9.00	.059
	Νίκες	8	5.75	46.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - BRkNNa	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLkNN - BRkNNb	Ήττες	2	1.50	3.00	0.13
	Νίκες	8	6.50	52.00	
	Ισοπαλίες	0			
	Σύνολο	10			

MLARAM-BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM-BRkNNb	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLARAM-MLkNN	Ήττες	3	6.33	19.00	.386
	Νίκες	7	5.14	36.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLkNN	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			
MLTSVM - MLARAM	Ήττες	0	NaN	.00	.005
	Νίκες	10	5.50	55.00	
	Ισοπαλίες	0			
	Σύνολο	10			

Από τα παραπάνω στατιστικά Wilcoxon tests που υλοποιήθηκαν για κάθε σύνολο πειραματικών τιμών με βάση κάθε μετρική ξεχωριστά προκύπτουν χρήσιμα συμπεράσματα σχετικά με την πληροφορία σε πόσα δείγματα τιμών υστερεί (Ήττες) , σε πόσα υπερτερεί (Νίκες) και σε πόσα έρχεται σε ισοπαλία κάθε ζεύγος αλγορίθμων. Τέλος, λαμβάνεται η πληροφορία της σημαντικότητας που υποδηλώνει αν εντοπίζεται σημαντική συγκριτική διαφορά μεταξύ των συγκρινόμενων αλγορίθμων.

5

Συμπεράσματα και μελλοντικές κατευθύνσεις

5.1 Συμπεράσματα

Η διπλωματική εργασία παρουσίασε μια εκτενή μελέτη και σύγκριση διάφορων αλγορίθμων και τεχνικών κατηγοριοποίησης πολλαπλών ετικετών, επιδιώκοντας να κατανοήσει τις δυνατότητες και τους περιορισμούς τους. Μέσα από την πειραματική μελέτη, η εργασία αποκάλυψε σημαντικές παρατηρήσεις σχετικά με την αποδοτικότητα, την ακρίβεια, και την εφαρμοσιμότητα των αναλυθέντων μεθόδων.

Ένα κρίσιμο συμπέρασμα από την έρευνα είναι ότι οι διάφορες τεχνικές και αλγόριθμοι προσφέρουν μια ευρεία γκάμα αποτελεσματικότητας, εξαρτώμενης από τη φύση και την πολυπλοκότητα των δεδομένων. Η επιλογή της κατάλληλης μεθόδου απαιτεί μια λεπτομερή κατανόηση τόσο των δεδομένων όσο και των στόχων της ανάλυσης. Επιπλέον, η ανάπτυξη και η βελτίωση των υπαρχόντων αλγορίθμων μέσω της έρευνας και της πειραματικής εφαρμογής αποτελεί μια συνεχή διαδικασία, καθώς ο τομέας της μηχανικής μάθησης εξελίσσεται.

Αυτή η διαδικασία αναδεικνύει τη σημασία της διαρκούς αξιολόγησης και της προσαρμογής των αλγορίθμων στις συνεχώς αλλαγμένες συνθήκες και στις νέες ανακαλύψεις που προκύπτουν από την επιστημονική έρευνα. Καθώς προχωρούμε προς την επίτευξη πιο εξελιγμένων και αποδοτικών τεχνικών, η συνεισφορά της τρέχουσας εργασίας στον τομέα της κατηγοριοποίησης πολλαπλών ετικετών διαδραματίζει έναν ουσιαστικό ρόλο στην κατεύθυνση αυτή, προσδίδοντας στην κοινότητα μια βάση για περαιτέρω ανάπτυξη και βελτίωση.

5.2 Μελλοντικές κατευθύνσεις

Η έρευνα στο πεδίο της κατηγοριοποίησης πολλαπλών ετικετών ανοίγει έναν πλούσιο ορίζοντα για μελλοντικές εξερευνήσεις και βελτιώσεις. Με βάση τα ευρήματα της παρούσας εργασίας, προτείνονται οι εξής κατευθύνσεις για μελλοντική έρευνα:

Βελτίωση Αλγορίθμων: Η ανάπτυξη και ο εμπλουτισμός των υφιστάμενων αλγορίθμων με νέες τεχνικές και προσεγγίσεις, όπως το deep learning και τα νευρωνικά δίκτυα, μπορούν να προσφέρουν βαθύτερη κατανόηση και καλύτερη διαχείριση των προβλημάτων πολλαπλών ετικετών.

Πώς μπορεί να Επιτευχθεί η Βελτίωση των Αλγορίθμων:

- **Αξιολόγηση και Επιλογή Χαρακτηριστικών (Feature Selection and Engineering):** Η βελτιστοποίηση της επιλογής και της κατασκευής των χαρακτηριστικών μπορεί να βελτιώσει σημαντικά την απόδοση των αλγορίθμων, επιτρέποντας την ακριβέστερη αναγνώριση των σχέσεων μεταξύ των δεδομένων και των προβλεπόμενων ετικετών.

- **Εφαρμογή Σύνθετων Μοντέλων:** Η εξερεύνηση και εφαρμογή πιο σύνθετων μοντέλων μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα και τα μοντέλα βαθιάς μάθησης, μπορεί να προσφέρει βελτιωμένες δυνατότητες πρόβλεψης, καθώς αυτά τα μοντέλα είναι ικανά να αναγνωρίζουν πιο πολύπλοκες μη γραμμικές σχέσεις στα δεδομένα.

- **Βελτιστοποίηση Υπερπαραμέτρων (Hyperparameter Tuning):** Η διαδικασία της ρύθμισης και της βελτιστοποίησης των υπερπαραμέτρων των αλγορίθμων μπορεί να οδηγήσει σε σημαντικές βελτιώσεις της απόδοσης, καθώς η σωστή ρύθμιση επηρεάζει άμεσα την ικανότητα του μοντέλου να μαθαίνει και να προβλέπει.

- **Διαχείριση Ανισορροπίας στα Δεδομένα (Handling Data Imbalance):** Σε περιπτώσεις όπου τα δεδομένα εκπαίδευσης είναι ανισορροπημένα, η εφαρμογή τεχνικών όπως το oversampling της λιγότερο εκπροσωπημένης κλάσης ή το undersampling της περισσότερο εκπροσωπημένης μπορεί να βελτιώσει την απόδοση του μοντέλου.

- **Ενσωμάτωση Διαφορετικών Προσεγγίσεων (Ensemble Methods):** Η χρήση συνδυαστικών μεθόδων, όπως τα ensemble models που συνδυάζουν τις προβλέψεις από πολλαπλά μοντέλα, μπορεί να αυξήσει την ακρίβεια και την αξιοπιστία των προβλέψεων.

Εξερεύνηση Νέων Συνόλων Δεδομένων: Η εφαρμογή και η δοκιμή των αλγορίθμων σε νέα και διαφορετικά σύνολα δεδομένων μπορεί να αποκαλύψει πρόσθετες προκλήσεις και ευκαιρίες για βελτίωση, ενισχύοντας τη γενίκευση και την εφαρμοσιμότητα των τεχνικών.

Διεπιστημονικές Προσεγγίσεις: Η ενσωμάτωση γνώσεων και τεχνικών από άλλες επιστήμες, όπως η ψυχολογία, η κοινωνιολογία, και η γλωσσολογία, μπορεί να προσφέρει νέες προοπτικές και να βελτιώσει την κατανόηση των πολυπλοκοτήτων που εμπεριέχονται στην κατηγοριοποίηση πολλαπλών ετικετών.

- **Ψυχολογία και Κοινωνιολογία**: Η ενσωμάτωση γνώσεων από την ψυχολογία και την κοινωνιολογία μπορεί να βοηθήσει στην κατανόηση της ανθρώπινης συμπεριφοράς και των κοινωνικών διαστάσεων που επηρεάζουν την ετικέτες στα δεδομένα. Για παράδειγμα, στην ανάλυση κειμένου ή στην επεξεργασία φυσικής γλώσσας, η κατανόηση των ψυχολογικών και κοινωνικών παραγόντων μπορεί να βοηθήσει στην ερμηνεία του συναισθηματικού φορτίου και της πολιτικής ή κοινωνικής σημασίας των κειμένων.
- **Γλωσσολογία:** Η εφαρμογή γλωσσολογικών θεωριών και εργαλείων μπορεί να βελτιώσει την ανάλυση κειμένου και την επεξεργασία φυσικής γλώσσας, βοηθώντας στην αναγνώριση και κατανόηση των γλωσσικών δομών, των σημασιολογικών διαστάσεων και της χρήσης της γλώσσας σε διαφορετικά πλαίσια. Αυτό μπορεί να είναι ιδιαίτερα χρήσιμο στην κατηγοριοποίηση πολλαπλών ετικετών όταν αντιμετωπίζουμε κείμενα με πολυσημία ή αμφισημία.
- **Τεχνητή Νοημοσύνη και Κοινωνικές Επιστήμες**: Η συνδυασμένη χρήση μεθόδων από την τεχνητή νοημοσύνη και τις κοινωνικές επιστήμες μπορεί να προσφέρει προηγμένες λύσεις για την ανάλυση και την κατανόηση των σύνθετων κοινωνικών δικτύων, της διάδοσης πληροφοριών και της ανθρώπινης συμπεριφοράς στο διαδίκτυο, προσφέροντας έτσι πιο εμπεριστατωμένες και πολυδιάστατες προοπτικές στην κατηγοριοποίηση δεδομένων.

Η ενσωμάτωση αυτών των μελλοντικών κατευθύνσεων στην έρευνα και την ανάπτυξη των συστημάτων κατηγοριοποίησης πολλαπλών ετικετών θα επιτρέψει την περαιτέρω εξέλιξη του πεδίου, προσφέροντας πιο προηγμένες και ευέλικτες λύσεις που μπορούν να ανταποκριθούν στις συνεχώς αυξανόμενες ανάγκες της επιστήμης δεδομένων και της μηχανικής μάθησης.

Βιβλιογραφία

[1] T. Hastie, R. Tibshirani, και J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media, 2009.

[2] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds., Springer, 2010, pp. 667-685.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[4] F. Chollet, Deep Learning with Python. Manning Publications, 2017.

[5] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. O'Reilly Media, 2019.

[6] R.E. Schapire, Y. Singer: Boostexter: a boosting-based system for text categorization. Machine Learning 39 pp 135–168, 2000

[7] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008.

[8] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Packt Publishing, 2019.

[9] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An Introduction to Statistical Learning: with Applications in R," Springer, 2013.

[10] J.D. Kelleher, B.M. Namee, and A. D'Arcy, "Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies," MIT Press, 2015.

[11] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer, 2009.

[12] F. Cady, "The Data Science Handbook," Wiley, 2017.

[13] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, “An empirical study of lazy multilabel classification algorithms,” in Lecture Notes in Computer Science, 2008, pp. 401–406. doi: 10.1007/978-3-540-87881-0_40.

[14] P. Joshi, “Predicting Movie Genres using NLP – An Awesome Introduction to Multi-Label Classification,” Analytics Vidhya, Jul. 19, 2022. https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/

[15] Prabhakaran, “Email Spam Classification - prabhakaran98 - Medium,” Medium, Apr. 18, 2022. [Online]. Available: https://medium.com/@yesprabhakaran98/email-spam-classification-92b661d3b700

[16] “The Greatest Showman (2017) | Biography, drama, musical,” IMDb, Dec. 20, 2017. https://www.imdb.com/title/tt1485796/?ref_=nv_sr_srsg_0_tt_8_nm_0_q_the%2520greate

[17] “How to Solve a Multi Class Classification Problem with Python?,” ProjectPro, Oct. 30, 2023. https://www.projectpro.io/article/multi-class-classification-python-example/547

[18] “scikit-multilearn.” http://scikit.ml/api/skmultilearn.html

[19] “Documentation.” https://mulan.sourceforge.net/datasets.html

[20] Csvankhede, “Performance matrix in Machine Learning,” CSVANKHEDE, Sep. 04, 2019. https://csvankhede.wordpress.com/2019/07/01/performance-matrix-in-machine-learning/

[21] “Accuracy | CloudFactory Computer Vision Wiki,” CloudFactory Computer Vision Wiki. https://wiki.cloudfactory.com/docs/mp-wiki/metrics/accuracy

[22] R. Kundu, “F1 Score in Machine Learning: Intro & Calculation,” V7, Apr. 20, 2023. https://www.v7labs.com/blog/f1-score-guide

[23] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Sydney, Australia, May 2004, pp. 22-30.

[24] W. Madhusanka, “Multi-label Classification - Wimukthi Madhusanka - Medium,” Medium, Jan. 06, 2022. [Online]. Available: https://wimukthimadhusanka85.medium.com/multi-label-classification-e259896182da

[25] S. Zhang and Z. Zhou, "A Review on Multi-Label Learning Algorithms," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1819-1837, Aug. 2014.

[26] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," Pattern Recognition, vol. 40, no. 7, pp. 2038-2048, Jul. 2007.

[27] Y. Zhang and Z. Zhou, "Multi-label dimensionality reduction via dependency maximization," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 2, pp. 310-323, Feb. 2013.

[28] Ougiaroglou, S., Mastromanolis, T., Evangelidis, G., Margaris, D. Fast Training Set Size Reduction Using Simple Space Partitioning Algorithms. Information 2022, 13, 572. https://doi.org/10.3390/info13120572

[29] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, “Comprehensive comparative study of multi-label classification methods,” Expert Systems With Applications, vol. 203, p. 117215, Oct. 2022, doi: 10.1016/j.eswa.2022.117215.