



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

**Βελτίωση της διαχείρισης σφαλμάτων λογισμικού:
Μια προσέγγιση μηχανικής μάθησης για την
ιεράρχηση και ταξινόμηση των αναφορών
σφαλμάτων Jira**

Από

Ευφροσύνη Βίλλη

Υποβάλλεται

για την εκπλήρωση των προϋποθέσεων λήψης

Μεταπτυχιακού Διπλώματος

στην κατεύθυνση «Μεγάλα Δεδομένα και Αναλυτική»

του ΠΜΣ «Πληροφοριακά Συστήματα και Υπηρεσίες»

στο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Φεβρουάριος 2024

Επιβλέπων/Επιβλέπουσα: Ανδριάννα Πρέντζα

Ακαδημαϊκή Θέση: Καθηγήτρια

Πανεπιστήμιο Πειραιώς. Κάτοχος όλων των δικαιωμάτων

/ University of Piraeus. All rights reserved.

Συγγραφέας/Author ..Ευφροσύνη Βίλλη

ΣΕΛΙΔΑ ΕΓΚΥΡΟΤΗΤΑΣ

Όνοματεπώνυμο Φοιτητή/Φοιτήτριας: Ευφροσύνη Βίλλη

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας: Βελτίωση της διαχείρισης σφαλμάτων λογισμικού: Μια προσέγγιση μηχανικής μάθησης για την ιεράρχηση και ταξινόμηση των αναφορών σφαλμάτων Jira.

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών “Πληροφορικά Συστήματα & Υπηρεσίες” του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις 29/02/2024 από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή

Επιβλέπουσα (Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς):

Καθ. Ανδριάννα Πρέντζα

Μέλος Εξεταστικής Επιτροπής: Καθ. Δημοσθένης Κυριαζής

Μέλος Εξεταστικής Επιτροπής: Καθ. Μιχαήλ Φιλιππάκης

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

Η Ευφροσύνη Βίλλη, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Βελτίωση της διαχείρισης σφαλμάτων λογισμικού: Μια προσέγγιση μηχανικής μάθησης για την ιεράρχηση και ταξινόμηση των αναφορών σφαλμάτων Jira», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Επιπλέον δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται τις διατάξεις που προβλέπει η Ελληνική και Κοινωνική Νομοθεσία περί πνευματικής ιδιοκτησίας.

Η ΔΗΛΟΥΣΑ

Όνοματεπώνυμο: Ευφροσύνη Βίλλη

Αριθμός Μητρώου: 2103

Υπογραφή:



ΠΕΡΙΛΗΨΗ

Η αποτελεσματική διαχείριση των αποθετηρίων σφαλμάτων στα συστήματα λογισμικού αποτελεί σημαντική πρόκληση λόγω της συνεχούς και συντριπτικής ροής αναφορών νέων ζητημάτων. Η χειραγώγηση αυτών των αναφορών είναι τόσο επίπονη όσο και χρονοβόρα, με αποτέλεσμα τις συχνές καθυστερήσεις στην επίλυση σημαντικών ελαττωμάτων. Η παρούσα εργασία διερευνά τις περιπλοκές της διαχείρισης σφαλμάτων σε αποθήκες λογισμικού, τονίζοντας την ουσιαστική επιρροή της χειροκίνητης παρέμβασης στο χρόνο που απαιτείται για την επίλυση σφάλματος. Η προτεινόμενη τεχνική επιδιώκει να βελτιστοποιήσει τις διαδικασίες διαχείρισης σφαλμάτων και να επιταχύνει την επίλυση των βασικών ζητημάτων με την αυτοματοποίηση της διαδικασίας προτεραιότητας και ταξινόμησης, μέσω μιας λεπτομερούς ανάλυσης των αναφορών σφαλμάτων Jira και της εφαρμογής αλγορίθμων ταξινόμησης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αυτοματοποιημένος έλεγχος λογισμικού με χρήση τεχνικών Τεχνητής Νοημοσύνης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Διαχείριση σφαλμάτων, Ταξινόμηση σφαλμάτων, Εποπτευόμενη μηχανική μάθηση, Ταξινόμηση πολλαπλών κατηγοριών, Αυτοματισμός ροών εργασίας

ABSTRACT

Effectively managing bug repositories in software systems poses a significant challenge due to the constant and overwhelming flow of new issue reports. The manual handling of these reports is both laborious and time-consuming, resulting in frequent delays in resolving major defects. This thesis explores the intricacies of bug management in software repositories, highlighting the substantial influence of manual intervention on the time it takes to resolve bugs. The suggested technique seeks to optimize bug management processes and expedite the resolution of key issues by automating the prioritization and classification process, through a detailed analysis of Jira bug reports and the implementation of classification algorithms.

SUBJECT AREA: Automated Software Testing with the use of AI techniques

KEYWORDS: Bug Management, Bug Triage, Supervised Machine Learning, Multiclass Classification, Workflow Automation

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω την κα. Ανδριάννα Πρέντζα, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου το συγκεκριμένο θέμα, καθώς και για τη συνεργασία και την πολύτιμη συμβολή της με τα σχόλια και την καθοδήγηση για την ολοκλήρωση της παρούσας διατριβής.

Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου για την υποστήριξη που μου προσέφερε καθ' όλη τη διάρκεια αυτού του μεταπτυχιακού προγράμματος.

ΠΕΡΙΕΧΟΜΕΝΑ	6
ΛΙΣΤΑ ΠΙΝΑΚΩΝ/ LIST OF TABLES	8
ΛΙΣΤΑ ΕΙΚΟΝΩΝ/ LIST OF IMAGES	9
1 ΕΙΣΑΓΩΓΗ	10
1.1 ΥΠΟΒΑΘΡΟ ΚΑΙ ΚΙΝΗΤΡΑ.....	10
1.2 ΕΡΕΥΝΗΤΙΚΟ ΠΡΟΒΛΗΜΑ ΚΑΙ ΣΤΟΧΟΙ	10
1.3 ΠΕΔΙΟΕΦΑΡΜΟΓΗΣΚΑΙ ΠΕΡΙΟΡΙΣΜΟΙ ΤΗΣ ΕΡΕΥΝΑΣ	11
2 ΔΟΚΙΜΕΣ ΛΟΓΙΣΜΙΚΟΥ	12
2.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	12
2.2 ΤΥΠΟΙ ΔΟΚΙΜΩΝ ΛΟΓΙΣΜΙΚΟΥ.....	12
2.3 ΠΡΟΚΛΗΣΕΙΣ ΣΤΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	12
2.4 ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ.....	12
2.5 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗΣ ΔΟΚΙΜΗΣ ΛΟΓΙΣΜΙΚΟΥ	13
2.6 ΕΡΓΑΛΕΙΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΩΝ ΔΟΚΙΜΩΝ	13
3 ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ ΚΑΙ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΣΤΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	15
3.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ ΚΑΙ ΤΗΝ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΣΤΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	15
3.2 ΕΦΑΡΜΟΓΕΣ ΤΗΣ NLP ΚΑΙ ΤΗΣ ΑΙ ΣΤΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	15
3.3 ΠΡΟΚΛΗΣΕΙΣ ΤΗΣ NLP ΚΑΙ ΤΗΣ ΑΙ ΣΤΗ ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	15
3.4 ΕΡΓΑΛΕΙΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΩΝ ΔΟΚΙΜΩΝ ΜΕ ΧΡΗΣΗ NLP/ΑΙ.....	16
3.5 ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ SPACY	19
3.6 ΤΑΞΙΝΟΜΗΣΗ ΑΝΑΦΟΡΩΝ ΣΦΑΛΜΑΤΩΝ JIRA	20
3.7 ΧΡΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ SPACY ΓΙΑ ΤΗΝ ΤΑΞΙΝΟΜΗΣΗ ΑΝΑΦΟΡΩΝ ΣΦΑΛΜΑΤΩΝ JIRA	20
4 ΜΕΘΟΔΟΛΟΓΙΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	22
4.1 ΕΙΣΑΓΩΓΗ.....	22
4.2 ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΒΛΗΜΑΤΩΝ ΣΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	22
4.3 ΘΕΜΑΤΑ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ	23

4.4	ΚΑΤΗΓΟΡΙΕΣ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΜΕ ΒΑΣΗ ΤΗ ΜΕΘΟΔΟ ΕΚΠΑΙΔΕΥΣΗΣ.....	23
5	ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	25
6	ΜΕΘΟΔΟΛΟΓΙΑ ΤΑΞΙΝΟΜΗΣΗΣ ΑΝΑΦΟΡΩΝ ΣΦΑΛΜΑΤΩΝ JIRA ΜΕ ΧΡΗΣΗ NLP ΚΑΙ ΑΙ	28
6.1	ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΤΑΞΙΝΟΜΗΣΗ ΑΝΑΦΟΡΩΝ ΣΦΑΛΜΑΤΩΝ JIRA.....	28
6.2	ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΕΠΟΠΤΕΥΟΜΕΝΗΣ ΜΑΘΗΣΗΣ ΚΑΙ ΤΗΣ ΤΑΞΙΝΟΜΗΣΗΣ ΠΟΛΛΑΠΛΩΝ ΤΑΞΕΩΝ	29
6.3	ΣΥΛΛΟΓΗ ΚΑΙ ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ.....	30
6.4	ΕΞΑΓΩΓΗ ΚΑΙ ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ	32
6.4.1	TF-IDF VECTORIZER.....	32
6.5	ΕΚΠΑΙΔΕΥΣΗ ΤΑΞΙΝΟΜΗΤΩΝ.....	34
6.5.1	Η ΜΕΘΟΔΟΣ TRAIN-TEST SPLIT	34
6.5.2	Η ΜΕΘΟΔΟΣ ΔΙΑΣΤΑΥΡΟΥΜΕΝΗΣ ΕΠΙΚΥΡΩΣΗΣ K-FOLD	35
6.6	ΤΑΞΙΝΟΜΗΤΕΣ	36
6.6.1	ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΥ GRIDSEARCHCV ΚΑΙ ΤΟΥ RANDOMIZEDSEARCHCV	36
6.6.2	ΤΑΞΙΝΟΜΗΤΗΣ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ (DECISION TREE CLASSIFIER)	37
6.6.3	ΤΑΞΙΝΟΜΗΤΗΣ RANDOM FOREST.....	37
6.6.4	ΠΟΛΥΩΝΥΜΙΚΗ ΛΟΓΙΣΤΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ (MULTINOMIAL LOGISTIC REGRESSION)	38
6.6.5	ΣΤΟΧΑΣΤΙΚΗ ΚΑΘΟΔΟΣ ΚΛΙΣΗΣ (STOCHASTIC GRADIENT DESCENT)	38
6.6.6	ΓΡΑΜΜΙΚΟΣ ΤΑΞΙΝΟΜΗΤΗΣ ΔΙΑΝΥΣΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ (LINEAR SUPPORT VECTOR CLASSIFIER).....	38
7	ΑΠΟΤΕΛΕΣΜΑΤΑ	40
7.1	ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΤΑΞΙΝΟΜΗΣΗΣ ΤΩΝ ΑΝΑΦΟΡΩΝ ΣΦΑΛΜΑΤΩΝ JIRA ΜΕ ΧΡΗΣΗ NLP	40
7.1.1	ΑΝΙΧΝΕΥΣΗ ΥΠΕΡΠΡΟΣΑΡΜΟΓΗΣ.....	40
7.2	ΠΕΡΙΛΗΨΗ ΜΕΘΟΔΟΥ ΕΚΠΑΙΔΕΥΣΗΣ ΤΩΝ ΤΑΞΙΝΟΜΗΤΩΝ	40
7.3	ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	41
8	ΣΥΖΗΤΗΣΗ	52
9	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ	53
10	ΑΝΑΦΟΡΕΣ	54

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 1: ΕΡΓΑΛΕΙΑ ΓΙΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΟ ΕΛΕΓΧΟ ΛΟΓΙΣΜΙΚΟΥ	14
ΠΙΝΑΚΑΣ 2: ΕΡΓΑΛΕΙΑ ΓΙΑ ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΟ ΕΛΕΓΧΟ ΛΟΓΙΣΜΙΚΟΥ ΜΕ ΧΡΗΣΗ NLP/AI	18
ΠΙΝΑΚΑΣ 3: ΠΑΡΑΜΕΤΡΟΙ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΣΤΟΝ ΚΑΘΕ ΤΑΞΙΝΟΜΗΤΗ.....	41
ΠΙΝΑΚΑΣ 4: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΑΞΙΝΟΜΗΤΩΝ ΓΙΑ MAX_FEATURES = 100	42
ΠΙΝΑΚΑΣ 5: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΑΞΙΝΟΜΗΤΩΝ ΓΙΑ MAX_FEATURES = 200	44
ΠΙΝΑΚΑΣ 6: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΑΞΙΝΟΜΗΤΩΝ ΓΙΑ MAX_FEATURES = 500	46
ΠΙΝΑΚΑΣ 7: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΑΞΙΝΟΜΗΤΩΝ ΓΙΑ MAX_FEATURES = 800	48
ΠΙΝΑΚΑΣ 8: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΑΞΙΝΟΜΗΤΩΝ ΓΙΑ MAX_FEATURES = 2000	50

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1: Χαρακτηριστικά της βιβλιοθήκης SpaCy 21

1. Εισαγωγή

1.1 Υπόβαθρο και κίνητρα

Η ταξινόμηση αναφορών σφαλμάτων είναι ένα πρόβλημα που αντιμετωπίζεται συχνά στη βιομηχανία λογισμικού. Οι αναφορές σφαλμάτων είναι συνήθως κείμενα που περιγράφουν ένα πρόβλημα που εντοπίστηκε στο λογισμικό και περιλαμβάνουν πληροφορίες όπως τον τύπο του σφάλματος, το περιβάλλον όπου εμφανίστηκε, την έκδοση του λογισμικού κ.λπ. Συνήθως, η διαχείριση αναφορών σφαλμάτων γίνεται με μηχανικό τρόπο, με αποτέλεσμα την αναπαραγωγή του ίδιου προβλήματος σε πολλές αναφορές και την καθυστέρηση της αντιμετώπισης πραγματικών προβλημάτων. Συμπερασματικά, η διαχείριση αυτών των αναφορών είναι σημαντική για τη βελτίωση της ποιότητας του λογισμικού και την εξοικονόμηση χρόνου και πόρων στη διαδικασία αντιμετώπισης των προβλημάτων. Ως εκ τούτου, έχουν αναπτυχθεί αυτοματοποιημένα συστήματα εντοπισμού σφαλμάτων για την εξομάλυνση αυτής της διαδικασίας. Ωστόσο, τα αυτοματοποιημένα συστήματα εντοπισμού σφαλμάτων μπορούν να δημιουργήσουν μεγάλο αριθμό αναφορών σφαλμάτων, η διαχείριση των οποίων μπορεί να είναι δύσκολη. Η αυτόματη ταξινόμηση των αναφορών σφαλμάτων σε κατηγορίες μπορεί να βοηθήσει στην αυτοματοποίηση αυτής της διαδικασίας, κατηγοριοποιώντας τις αναφορές σφαλμάτων σε διαφορετικές κατηγορίες, όπως το λειτουργικό σύστημα, η βαθμίδα σοβαρότητας του σφάλματος, και το επίπεδο προτεραιότητας για τη διόρθωση του προβλήματος. Οι τεχνικές επεξεργασίας φυσικής γλώσσας (NLP) και τεχνητής νοημοσύνης (AI) μπορούν να χρησιμοποιηθούν για την ταξινόμηση των αναφορών σφαλμάτων, κάτι το οποίο μπορεί να βοηθήσει στην αποτελεσματικότερη διαχείρισή τους.

1.2 Ερευνητικό πρόβλημα και στόχοι

Στόχος της παρούσας έρευνας είναι να διερευνήσει τη χρήση τεχνικών NLP και AI για την ταξινόμηση των αναφορών σφαλμάτων που δημιουργούνται στο Jira ως προς την σοβαρότητά τους. Οι συγκεκριμένοι ερευνητικοί στόχοι είναι οι εξής:

- Επισκόπηση της βιβλιογραφίας σχετικά με τη δοκιμή λογισμικού, την αυτοματοποιημένη δοκιμή λογισμικού και τις τεχνικές NLP/AI στη δοκιμή λογισμικού.
- Ανάπτυξη μοντέλων ταξινόμησης με χρήση τεχνικών NLP/AI για την ταξινόμηση αναφορών σφαλμάτων στο Jira.
- Αξιολόγηση της απόδοσης των μοντέλων ταξινόμησης.

Στην εργασία θα χρησιμοποιηθούν δεδομένα αναφορών σφαλμάτων από ανοιχτές πηγές λογισμικού και θα υλοποιηθεί ένα πρωτότυπο σύστημα ταξινόμησης αναφορών σφαλμάτων σε γλώσσα Python. Το συγκεκριμένο ερευνητικό πρόβλημα μπορεί να αποδοθεί στην κατηγορία προβλημάτων ταξινόμησης, με στόχο την ταξινόμηση σφαλμάτων στις κατηγορίες:

Σημαντικό (Major), Ελάσσων (Minor), Κρίσιμο (Critical), Ανασταλτικό (Blocker), Ασήμαντο (Trivial).

Οι αλγόριθμοι που εφαρμόζονται στη μοντελοποίηση της ταξινόμησης, έχουν συγκριθεί και παρουσιάζεται ο πιο αποτελεσματικός.

1.3 Πεδίο εφαρμογής και περιορισμοί της έρευνας

Η παρούσα έρευνα επικεντρώνεται στη χρήση τεχνικών NLP και AI για την ταξινόμηση αναφορών σφαλμάτων που παράγονται στο Jira. Θα χρησιμοποιηθεί ένα δείγμα αναφορών σφαλμάτων που παράγονται στο Jira για την ανάπτυξη και την αξιολόγηση του μοντέλου ταξινόμησης, ενώ δεν θα καλυφθούν άλλες πτυχές των δοκιμών λογισμικού, όπως οι δοκιμές μονάδας (Unit Testing), οι δοκιμές ολοκλήρωσης (Integration Testing) και οι δοκιμές συστήματος (System Testing).

2. Δοκιμές λογισμικού

2.1 Εισαγωγή στη δοκιμή λογισμικού

Η δοκιμή λογισμικού είναι η διαδικασία επαλήθευσης και επικύρωσης του λογισμικού για να διασφαλιστεί ότι ανταποκρίνεται στις απαιτήσεις και τις προδιαγραφές. Η δοκιμή λογισμικού είναι σημαντική επειδή βοηθά στον εντοπισμό ατελειών και σφαλμάτων στο λογισμικό, τα οποία μπορεί να έχουν σοβαρές συνέπειες για τους χρήστες και τους πελάτες [1].

2.2 Τύποι δοκιμών λογισμικού

Υπάρχουν διάφοροι τύποι δοκιμών λογισμικού [1], μεταξύ των οποίων:

- Δοκιμές μονάδας (Unit Testing): Η δοκιμή μονάδας είναι η διαδικασία δοκιμής μεμονωμένων στοιχείων ή ενοτήτων του λογισμικού.
- Δοκιμές ολοκλήρωσης (Integration Testing): Η δοκιμή ολοκλήρωσης είναι η διαδικασία δοκιμής της αλληλεπίδρασης μεταξύ διαφορετικών ενοτήτων ή συστατικών στοιχείων του λογισμικού.
- Δοκιμή συστήματος (System Testing): Η δοκιμή συστήματος είναι η διαδικασία δοκιμής ολόκληρου του συστήματος λογισμικού για να διασφαλιστεί ότι πληροί τις απαιτήσεις και τις προδιαγραφές.
- Δοκιμή αποδοχής (Acceptance Testing) : Η δοκιμή αποδοχής είναι η διαδικασία δοκιμής του λογισμικού από τη σκοπιά του χρήστη για να διασφαλιστεί ότι ανταποκρίνεται στις απαιτήσεις και τις προσδοκίες του.

2.3 Προκλήσεις στη δοκιμή λογισμικού

Η δοκιμή λογισμικού μπορεί να αποτελέσει πρόκληση λόγω των ακόλουθων παραγόντων:

- Πολυπλοκότητα: Το σύγχρονο λογισμικό είναι συχνά πολύπλοκο, γεγονός που καθιστά δύσκολη τη δοκιμή του.
- Περιορισμοί χρόνου: Η δοκιμή μπορεί να είναι χρονοβόρα και μπορεί να υπάρχουν χρονικοί περιορισμοί για την ολοκλήρωση της δοκιμής πριν από την κυκλοφορία του λογισμικού.
- Κόστος: Η δοκιμή μπορεί να είναι δαπανηρή και μπορεί να υπάρχουν περιορισμοί στον προϋπολογισμό που πρέπει να ληφθούν υπόψη.
- Μεταβαλλόμενες απαιτήσεις: Οι απαιτήσεις μπορεί να αλλάξουν κατά τη διάρκεια του κύκλου ζωής της ανάπτυξης λογισμικού, γεγονός που μπορεί να επηρεάσει τη δοκιμή.

2.4 Αυτοματοποιημένη δοκιμή λογισμικού

Η αυτοματοποιημένη δοκιμή λογισμικού είναι η χρήση εργαλείων λογισμικού για την αυτόματη εκτέλεση εργασιών ελέγχων. Συμβάλλει στην εξοικονόμηση χρόνου και στη μείωση του κινδύνου σφαλμάτων. Τα αυτοματοποιημένα εργαλεία δοκιμών μπορούν να εκτελέσουν

επαναλαμβανόμενες εργασίες, όπως οι δοκιμές παλινδρόμησης (Regression Testing), πιο αποτελεσματικά από τις χειροκίνητες δοκιμές.

2.5 Πλεονεκτήματα της αυτοματοποιημένης δοκιμής λογισμικού

Υπάρχουν διάφορα πλεονεκτήματα στη χρήση αυτοματοποιημένων δοκιμών λογισμικού [2], μεταξύ των οποίων:

- Ταχύτερη δοκιμή: Η αυτοματοποιημένη δοκιμή μπορεί να είναι ταχύτερη από τη χειροκίνητη δοκιμή.
- Συνέπεια: Η αυτοματοποιημένη δοκιμή μπορεί να διασφαλίσει ότι οι δοκιμές εκτελούνται με συνέπεια.
- Επαναχρησιμοποίηση: Τα αυτοματοποιημένα σενάρια δοκιμών μπορούν να επαναχρησιμοποιηθούν για μελλοντικές δοκιμές.
- Κάλυψη: Η αυτοματοποιημένη δοκιμή μπορεί να καλύψει μεγαλύτερο αριθμό περιπτώσεων δοκιμής από τη χειροκίνητη δοκιμή.

2.6 Εργαλεία αυτοματοποιημένων δοκιμών

Τα εργαλεία αυτοματοποιημένων δοκιμών είναι εφαρμογές λογισμικού που έχουν σχεδιαστεί για την αυτοματοποίηση της διαδικασίας δοκιμών εφαρμογών λογισμικού. Βοηθούν στον εντοπισμό ατελειών, σφαλμάτων ή σφαλμάτων στον κώδικα λογισμικού με την εκτέλεση αυτοματοποιημένων δοκιμών. Παρακάτω παρατίθεται μια επεξήγηση ορισμένων κοινών εργαλείων αυτοματοποιημένων δοκιμών μαζί με έναν πίνακα σύγκρισης [3][4][5]:

- Selenium: Το Selenium είναι ένα δημοφιλές εργαλείο ανοικτού κώδικα που χρησιμοποιείται για την αυτοματοποίηση προγραμμάτων περιήγησης στο διαδίκτυο. Υποστηρίζει πολλαπλές γλώσσες προγραμματισμού, όπως Java, C#, Python κ.λπ., καθιστώντας το αρκετά ευέλικτο. Το Selenium WebDriver παρέχει ένα ισχυρό API για την αλληλεπίδραση με στοιχεία ιστού και την εκτέλεση ενεργειών όπως κλικ, πληκτρολόγηση κ.λπ. Υποστηρίζει δοκιμές σε πολλαπλά προγράμματα περιήγησης και απαιτεί δεξιότητες προγραμματισμού για τη συγγραφή σεναρίων δοκιμών.
- JUnit: Το JUnit είναι ένα ευρέως χρησιμοποιούμενο πλαίσιο δοκιμών μονάδας για Java. Παρέχει σημειώσεις για τον ορισμό μεθόδων δοκιμής και ισχυρισμούς για την επαλήθευση των αναμενόμενων αποτελεσμάτων. Ενσωματώνεται καλά με εργαλεία κατασκευής όπως το Maven και το Ant και επικεντρώνεται σε δοκιμές μονάδας, κυρίως για εφαρμογές Java.
- TestNG: Το TestNG είναι ένα άλλο πλαίσιο δοκιμών για Java, εμπνευσμένο από το JUnit, αλλά με πρόσθετα χαρακτηριστικά. Υποστηρίζει διάφορους τύπους δοκιμών, όπως unit, functional, integration κ.λπ. Προσφέρει χαρακτηριστικά όπως παραμετροποίηση, ομαδοποίηση δοκιμών, παράλληλη εκτέλεση και άλλα. Μπορεί να ενσωματωθεί με άλλα εργαλεία και πλαίσια όπως το Selenium για δοκιμές ιστού.
- Cucumber: Το Cucumber είναι ένα εργαλείο ανάπτυξης με γνώμονα τη συμπεριφορά (BDD / Behavior-driven development) που επιτρέπει τη συγγραφή δοκιμών σε απλό κείμενο. Οι δοκιμές γράφονται σε σύνταξη Gherkin, καθιστώντας τις ευανάγνωστες και κατανοητές από μη τεχνικούς ενδιαφερόμενους. Υποστηρίζει πολλές γλώσσες

προγραμματισμού, όπως Java, Ruby, JavaScript κ.λπ. Ενθαρρύνει τη συνεργασία μεταξύ προγραμματιστών, δοκιμαστών και επιχειρηματικών αναλυτών.

Πίνακας 1: Εργαλεία για αυτοματοποιημένο έλεγχο λογισμικού

Εργαλείο	Σκοπός	Χαρακτηριστικά	Περιπτώσεις χρήσης
Selenium	Αυτοματοποίηση προγράμματος περιήγησης ιστού	Ανοικτός κώδικας Υποστηρίζει πολλαπλές γλώσσες προγραμματισμού Υποστήριξη δοκιμών Cross-browser	Αυτοματοποιημένη δοκιμή εφαρμογών ιστού σε διαφορετικά προγράμματα περιήγησης Δοκιμές παλινδρόμησης Λειτουργικές δοκιμές Δοκιμές End-to-end
JUnit	Δοκιμές μονάδας για Java	Πλαίσιο δοκιμών μονάδας για Java Παρέχει σημειώσεις για μεθόδους δοκιμής και ισχυρισμούς Ενσωματώνεται καλά με εργαλεία κατασκευής όπως το Maven και το Ant	Δοκιμές μονάδας κλάσεων και μεθόδων Java Ανάπτυξη με γνώμονα τη δοκιμή Pipeline συνεχούς ολοκλήρωσης και παράδοσης (CI/CD)
TestNG	Πλαίσιο δοκιμών για Java	Υποστηρίζει διάφορους τύπους δοκιμών Προσφέρει χαρακτηριστικά όπως παραμετροποίηση, ομαδοποίηση δοκιμών, παράλληλη εκτέλεση Ενσωματώνεται με άλλα εργαλεία και πλαίσια όπως το Selenium για δοκιμές ιστού	Δοκιμές εφαρμογών Java σε διάφορα επίπεδα Παραμετρικές δοκιμές Ομαδοποίηση και παράλληλη εκτέλεση δοκιμών Πολύπλοκα σενάρια δοκιμών που απαιτούν προηγμένα χαρακτηριστικά
Cucumber	Εργαλείο ανάπτυξης με γνώμονα τη συμπεριφορά	Εφαρμόζει τις αρχές BDD Οι δοκιμές γράφονται σε απλό κείμενο χρησιμοποιώντας σύνταξη Gherkin Ενθαρρύνει τη συνεργασία μεταξύ των ενδιαφερομένων Υποστηρίζει πολλαπλές γλώσσες προγραμματισμού	Ανάπτυξη με γνώμονα τη συμπεριφορά (BDD) Συνεργασία μεταξύ προγραμματιστών, δοκιμαστών και επιχειρηματικών αναλυτών Έλεγχος αποδοχής Αυτοματοποιημένη δοκιμή ιστοριών χρήστη ή χαρακτηριστικών Δημιουργία ζωντανής τεκμηρίωσης

3. Επεξεργασία Φυσικής Γλώσσας και Τεχνητή Νοημοσύνη στη δοκιμή λογισμικού

3.1 Εισαγωγή στην Επεξεργασία Φυσικής Γλώσσας και την Τεχνητή Νοημοσύνη στη δοκιμή λογισμικού

Η Επεξεργασία Φυσικής Γλώσσας (NLP) είναι ένα διεπιστημονικό πεδίο μελέτης που επικεντρώνεται στην αλληλεπίδραση μεταξύ της ανθρώπινης γλώσσας και των υπολογιστών. Η NLP έχει πολυάριθμες εφαρμογές σε διάφορους τομείς, και μία από τις σημαντικότερες περιπτώσεις χρήσης της είναι η ταξινόμηση κειμένου. Οι τεχνικές επεξεργασίας φυσικής γλώσσας μπορούν να χρησιμοποιηθούν για την ανάλυση κειμένου φυσικής γλώσσας σε αναφορές σφαλμάτων και σε περιπτώσεις ελέγχου λογισμικού, ενώ η τεχνητή νοημοσύνη μπορεί να χρησιμοποιηθεί για την ανάπτυξη μοντέλων που μπορούν να μάθουν από ιστορικά δεδομένα και να κάνουν προβλέψεις για το λογισμικό που δοκιμάζεται. Συνεπώς, μπορούν να συμβάλλουν στη δοκιμή λογισμικού για την αυτοματοποίηση της διαδικασίας δοκιμής, αλλά και για τη βελτίωση της ακρίβειας και της αποτελεσματικότητάς της [6]. Σε αυτό το κεφάλαιο, θα διερευνήσουμε τη χρήση της βιβλιοθήκης NLP SpaCy για την ταξινόμηση αναφορών σφαλμάτων Jira με χρήση NLP/AI.

3.2 Εφαρμογές της NLP και της AI στη δοκιμή λογισμικού

Οι τεχνικές NLP και AI μπορούν να χρησιμοποιηθούν σε διάφορους τομείς της δοκιμής λογισμικού [7] [8], όπως:

- **Εντοπισμός σφαλμάτων:** Η NLP και η AI μπορούν να χρησιμοποιηθούν για την ταξινόμηση και την ιεράρχηση των αναφορών σφαλμάτων με βάση τη σοβαρότητά τους και τον αντίκτυπό τους στο λογισμικό.
- **Δημιουργία σεναρίων/περιπτώσεων δοκιμής:** Οι τεχνικές NLP και AI μπορούν να χρησιμοποιηθούν για τη δημιουργία περιπτώσεων δοκιμής με βάση τις απαιτήσεις και τις προδιαγραφές.
- **Ανάλυση αποτελεσμάτων δοκιμών:** Η NLP και η AI μπορούν να χρησιμοποιηθούν για την ανάλυση των αποτελεσμάτων δοκιμών για τον εντοπισμό προτύπων και τάσεων.

3.3 Προκλήσεις της NLP και της AI στη δοκιμή λογισμικού

Οι τεχνικές NLP και AI μπορούν να αντιμετωπίσουν διάφορες προκλήσεις στη δοκιμή λογισμικού, όπως:

- **Ποιότητα δεδομένων:** Η ακρίβεια και η ποιότητα των δεδομένων μπορεί να επηρεάσει την απόδοση των μοντέλων NLP και AI.
- **Πολυπλοκότητα των μοντέλων:** Η ανάπτυξη και η εφαρμογή σύνθετων μοντέλων NLP και AI μπορεί να είναι χρονοβόρα και να έχει αρκετές προκλήσεις.
- **Ερμηνευσιμότητα:** Τα μοντέλα NLP και AI μπορεί να είναι δύσκολο να ερμηνευθούν, γεγονός που μπορεί να δυσχεράνει την κατανόηση του τρόπου με τον οποίο καταλήγουν στα συμπεράσματά τους.

3.4 Εργαλεία αυτοματοποιημένων δοκιμών με χρήση NLP/AI

Τα εργαλεία που παρατίθενται παρακάτω αξιοποιούν τη δύναμη της τεχνητής νοημοσύνης και των τεχνικών επεξεργασίας φυσικής γλώσσας για την αυτοματοποίηση διαφόρων πτυχών των δοκιμών λογισμικού, όπως η δημιουργία, η εκτέλεση και η ανάλυση περιπτώσεων δοκιμών [7] [9]. Ωστόσο, είναι σημαντικό να σημειωθεί ότι δεν αντικαθιστούν τη χειροκίνητη δοκιμή και θα πρέπει να χρησιμοποιούνται σε συνδυασμό με άλλες μεθοδολογίες δοκιμών για μια ολοκληρωμένη προσέγγιση δοκιμών.

- **Testim:** Η Testim είναι μια πλατφόρμα δοκιμών αυτοματοποίησης βασισμένη σε τεχνητή νοημοσύνη που αξιοποιεί αλγορίθμους μηχανικής μάθησης για τη δημιουργία και τη συντήρηση δοκιμών. Προσφέρει αυτοθεραπευόμενες δοκιμές που μπορούν να προσαρμόζονται στις αλλαγές της υπό δοκιμή εφαρμογής και παρέχει δυνατότητες οπτικών δοκιμών, επιτρέποντας στους ελεγκτές να επικυρώνουν τις οπτικές πτυχές της εφαρμογής.
- **Applitools:** Το Applitools είναι ένα εργαλείο οπτικών δοκιμών που χρησιμοποιεί τεχνητή νοημοσύνη για την οπτική επικύρωση εφαρμογών ιστού και κινητών τηλεφώνων. Εντοπίζει αυτόματα τις οπτικές διαφορές μεταξύ των βασικών και των πραγματικών καταστάσεων του UI, επιτρέποντας τον γρήγορο και ακριβή εντοπισμό σφαλμάτων και ενσωματώνεται με υπάρχοντα πλαίσια αυτοματοποίησης δοκιμών και αγωγούς CI/CD.
- **ReTest:** Το ReTest είναι ένα εργαλείο αυτοματοποίησης δοκιμών με τεχνητή νοημοσύνη που επικεντρώνεται στη δοκιμή εφαρμογών ιστού από άκρο σε άκρο. Δημιουργεί και συντηρεί αυτόματα σενάρια δοκιμών χρησιμοποιώντας αλγορίθμους AI, μειώνοντας την προσπάθεια που απαιτείται για τη δημιουργία και τη συντήρηση δοκιμών. Η μηχανή AI του ReTest αναλύει τη συμπεριφορά της εφαρμογής για τον εντοπισμό σεναρίων δοκιμής και τη βελτιστοποίηση της κάλυψης των δοκιμών.
- **Mabl:** Το Mabl είναι μια αυτόνομη πλατφόρμα δοκιμών που συνδυάζει την τεχνητή νοημοσύνη με την αυτοματοποίηση δοκιμών χαμηλού κώδικα για τον εξορθολογισμό της διαδικασίας δοκιμών. Χρησιμοποιεί μηχανική μάθηση για τη δημιουργία και τη συντήρηση δοκιμών, τον εντοπισμό παλινδρομήσεων και την ιεράρχηση των προσπαθειών δοκιμών, ενώ ενσωματώνεται με δημοφιλή εργαλεία CI/CD και συστήματα παρακολούθησης προβλημάτων για απρόσκοπτη εκτέλεση δοκιμών και αναφορά αποτελεσμάτων.
- **Functionize:** Το Functionize είναι μια πλατφόρμα δοκιμών που βασίζεται στην τεχνητή νοημοσύνη και προσφέρει αυτόνομες δυνατότητες δοκιμής για εφαρμογές ιστού. Χρησιμοποιεί επεξεργασία φυσικής γλώσσας (NLP) για να ερμηνεύσει τις απαιτήσεις δοκιμής γραμμένες σε απλή αγγλική γλώσσα και να δημιουργήσει αντίστοιχες περιπτώσεις δοκιμών. Οι προσαρμοστικοί αλγόριθμοι του Functionize επιτρέπουν τις αυτοθεραπευτικές δοκιμές και την έξυπνη συντήρηση των δοκιμών.
- **Test.ai:** Το Test.ai είναι μια πλατφόρμα αυτοματισμού δοκιμών που βασίζεται στην AI και επικεντρώνεται στις δοκιμές εφαρμογών για κινητές συσκευές. Εκμεταλλεύεται την υπολογιστική όραση και τη μηχανική μάθηση για να εκτελέσει λειτουργικές δοκιμές, εξερευνητικές δοκιμασίες και εξετάσεις αναδρομής σε κινητές εφαρμογές. Το Test.ai αναγνωρίζει αυτόματα στοιχεία UI και δημιουργεί σενάρια δοκιμών χωρίς να βασίζεται σε παραδοσιακούς εντοπιστές.

-
- **Diffblue:** Το Diffblue είναι ένα εργαλείο παραγωγής δοκιμών που βασίζεται στην τεχνητή νοημοσύνη, ειδικά σχεδιασμένο για τη δοκιμή μονάδων εφαρμογών Java. Δημιουργεί αυτόματα δοκιμές μονάδων για υπάρχουσες βάσεις κωδικού Java, βοηθώντας τους προγραμματιστές να βελτιώσουν την ποιότητα του κώδικα και να διατηρήσουν την κάλυψη δοκιμών. Η μηχανή AI της Diffblue αναλύει τη σημασιολογία του κώδικα για να δημιουργήσει ουσιαστικές και αποτελεσματικές περιπτώσεις δοκιμών.
 - **TestCraft:** Το TestCraft είναι μια πλατφόρμα αυτοματισμού δοκιμών με τεχνητή νοημοσύνη για εφαρμογές ιστού. Προσφέρει δυνατότητες αυτοματοποίησης δοκιμών χωρίς κώδικα, επιτρέποντας στους testers να δημιουργούν και να διατηρούν δοκιμές χρησιμοποιώντας μια οπτική διεπαφή. Η μηχανή τεχνητής νοημοσύνης του TestCraft βοηθά στη δημιουργία, συντήρηση και εκτέλεση δοκιμών, επιτρέποντας αποτελεσματικές και αξιόπιστες δοκιμές.
 - **Appvance IQ:** Το Appvance IQ είναι μια πλατφόρμα αυτοματισμού δοκιμών με τεχνητή νοημοσύνη που καλύπτει δοκιμές λειτουργίας, απόδοσης και ασφάλειας. Χρησιμοποιεί αλγόριθμους τεχνητής νοημοσύνης για να δημιουργήσει σενάρια δοκιμών, να εκτελέσει δοκιμές και να αναλύσει τα αποτελέσματα δοκιμής σε διάφορους τύπους εφαρμογών. Ο κινητήρας AI της Appvance IQ προσαρμόζεται στις αλλαγές στην εφαρμογή υπό δοκιμή, εξασφαλίζοντας την ανθεκτικότητα και την ακρίβεια των δοκιμών.
 - **Eggplant:** Το Eggplant είναι μια πλατφόρμα αυτοματισμού δοκιμών που ενσωματώνει τεχνολογία AI και αναγνώρισης εικόνας για δοκιμές GUI. Προσομοιώνει τις αλληλεπιδράσεις των χρηστών με τις εφαρμογές, επιτρέποντας στους testers να δημιουργήσουν ισχυρά και ρεαλιστικά σενάρια δοκιμών. Οι δυνατότητες AI της Eggplant επιτρέπουν την έξυπνη εκτέλεση και ανάλυση δοκιμών σε διάφορες πλατφόρμες και συσκευές.
 - **Tricentis Tosca:** Η Tricentis Tosca είναι μια πλατφόρμα δοκιμών που συνδυάζει τον έλεγχο αυτοματισμού, την τεχνητή νοημοσύνη και την ανάλυση. Προσφέρει σχεδιασμό περιπτώσεων δοκιμών που βασίζεται στην τεχνητή νοημοσύνη, αυτοματοποίηση δοκιμής με βάση τον κίνδυνο και προβλέψιμη ανάλυση για τη βελτιστοποίηση της κάλυψης των δοκιμασιών. Η Tricentis Tosca ενσωματώνει τις αλυσίδες εργαλείων DevOps για να επιτρέψει την αυτοματοποίηση των δοκιμών από άκρο σε άκρο και να επιταχύνει τους κύκλους απελευθέρωσης.
 - **Postbot (Postman AI-powered assistant):** Το Postbot είναι ένα AI-powered βοηθός που βοηθά τους προγραμματιστές και τους testers στις δοκιμές API. Παρέχει συστάσεις για τα endpoints των API, τις τιμές παραμέτρων και τους ισχυρισμούς δοκιμών με βάση την ιστορική χρήση και τα πρότυπα. Το Postbot ενισχύει την παραγωγικότητα και την ακρίβεια των δοκιμών API αξιοποιώντας προτάσεις και γνώσεις που βασίζονται στην τεχνητή νοημοσύνη.
 - **Rainforest QA:** Το Rainforest QA είναι μια πλατφόρμα δοκιμών ενισχυμένη με δυνατότητες AI. Συνδυάζει ανθρώπινους testers και αλγόριθμους AI για την εκτέλεση περιπτώσεων δοκιμών σε διάφορες πλατφόρμες και συσκευές. Η μηχανή AI της Rainforest QA αναλύει τα αποτελέσματα των δοκιμών για τον εντοπισμό των μοτίβων, τον καθορισμό προτεραιοτήτων για τα σφάλματα και τη βελτίωση της κάλυψης δοκιμής.
 - **Perfecto:** Η Perfecto είναι μια πλατφόρμα αυτοματισμού δοκιμών που προσφέρει δυνατότητες τεστ που βασίζονται στην τεχνητή νοημοσύνη για εφαρμογές ιστού και

κινητής τηλεφωνίας. Παρέχει έξυπνη αυτοματοποίηση δοκιμών, δοκιμές αυτοθεραπείας και προγνωστική ανάλυση για τη βελτιστοποίηση της αποτελεσματικότητας δοκιμής, ενώ υποστηρίζει την ενσωμάτωση με τα δημοφιλή εργαλεία CI/CD και τα πλαίσια δοκιμών για την απρόσκοπτη εκτέλεση και διαχείριση δοκιμής.

Πίνακας 2: Εργαλεία για αυτοματοποιημένο έλεγχο λογισμικού με χρήση NLP/AI

Εργαλείο	Σκοπός	Χαρακτηριστικά	Περιπτώσεις χρήσης
Testim	Αυτοματοποιημένες δοκιμές UI	Δημιουργία δοκιμών με τη βοήθεια του AI, self-healing δοκιμές	Δοκιμές εφαρμογών ιστού
Applitools	Οπτικός έλεγχος	Οπτικός έλεγχος AI για την ανίχνευση θεμάτων διάταξης του UI	Εξασφάλιση συνέπειας UI
ReTest	Δημιουργία δοκιμών	NLP για μετατροπή προδιαγραφών σε περιπτώσεις δοκιμών	Δοκιμές με βάση τις προδιαγραφές
Mabl	Αυτόνομη δοκιμή	Εκτέλεση δοκιμών με γνώμονα το AI, συντήρηση δοκιμών	Συνεχείς δοκιμές, δοκιμές παλινδρόμησης
Functionize	AI-powered δοκιμή	Δοκιμές self-healing, δημιουργία δοκιμών μέσω AI	Δοκιμές εφαρμογών ιστού και κινητών
Test.ai	Εξερεύνηση με AI	AI εξερεύνηση της εφαρμογής, εύρεση σφαλμάτων	Δοκιμές εφαρμογών κινητού
Diffblue	Αυτοματοποιημένες δοκιμές κώδικα	AI για τη δημιουργία δοκιμών μονάδων από τον κώδικα	Δοκιμές μονάδας, κάλυψη κώδικα
TestCraft	Αυτοματοποιημένες δοκιμές χωρίς κώδικα	Δημιουργία δοκιμών με AI, δεν απαιτείται προγραμματισμός	Δοκιμές εφαρμογών ιστού και κινητών
Appvance IQ	Έλεγχος δοκιμών με	AI για τη	Συνεχείς δοκιμές,

	AI	δημιουργία σεναρίων δοκιμών, self-healing δοκιμές	δοκιμές παλινδρόμησης
Eggplant	Ευφυείς δοκιμές	Δημιουργία περιπτώσεων δοκιμών με βάση το AI, προγνωστική αναλυτική	Δοκιμές GUI και εφαρμογών κινητού
Tricentis Tosca	Πλατφόρμα αυτοματοποίησης δοκιμών	Σχεδιασμός περιπτώσεων δοκιμών με AI, risk-based δοκιμές	End-to-end έλεγχος, διαχείριση δοκιμών
Postbot (Postman AI-powered assistant)	Έλεγχος API	Δημιουργία και επικύρωση δοκιμών API με AI	Έλεγχος και επικύρωση API
Rainforest QA	Crowdsourced έλεγχος	Βελτιστοποίηση δοκιμών με AI, εκτέλεση δοκιμών σε πραγματικό χρόνο	Crowdsourced και αυτοματοποιημένος έλεγχος
Perfecto	Συνεχείς δοκιμές	Εκτέλεση δοκιμών με AI, αναλυτική δοκιμών	Συνεχείς δοκιμές, Δοκιμές εφαρμογών κινητού

3.5 Επισκόπηση της SpaCy

Το SpaCy είναι μια βιβλιοθήκη NLP ανοικτού κώδικα που παρέχει εργαλεία για επεξεργασία κειμένου, αναγνώριση οντοτήτων, επισημείωση μέρους του λόγου και ανάλυση εξαρτήσεων, μεταξύ άλλων λειτουργιών. Το SpaCy είναι κατασκευασμένο με Cython, η οποία είναι μια μεταγλωττισμένη γλώσσα που είναι ταχύτερη από την καθαρή Python. Αυτό καθιστά το SpaCy κατάλληλο για τη γρήγορη και αποτελεσματική επεξεργασία μεγάλου όγκου δεδομένων κειμένου, κάτι που μπορεί να είναι σημαντικό σε εργασίες ταξινόμησης σφαλμάτων όπου πρέπει να αναλυθεί μεγάλος αριθμός αναφορών σφαλμάτων. Συμπερασματικά, εφόσον έχει σχεδιαστεί για να είναι γρήγορο, αποδοτικό και κλιμακούμενο, καθίσταται ως μια δημοφιλή επιλογή για εργασίες NLP, ιδίως σε κλάδους όπως η χρηματοδότηση, η υγειονομική περίθαλψη και τα μέσα ενημέρωσης [10].

3.6 Ταξινόμηση αναφορών σφαλμάτων Jira

Το Jira είναι ένα δημοφιλές εργαλείο παρακολούθησης προβλημάτων και διαχείρισης έργων που χρησιμοποιείται από ομάδες ανάπτυξης λογισμικού. Επιτρέπει στους χρήστες να δημιουργούν ζητήματα, όπως αναφορές σφαλμάτων, αιτήσεις χαρακτηριστικών και εργασίες, μεταξύ άλλων. Οι αναφορές σφαλμάτων Jira είναι περιγραφές κειμένου για ελαττώματα λογισμικού, που περιλαμβάνουν λεπτομέρειες όπως τα βήματα για την αναπαραγωγή του προβλήματος, την αναμενόμενη συμπεριφορά καθώς και την πραγματική συμπεριφορά. Στην προκειμένη περίπτωση, η ταξινόμηση των αναφορών σφαλμάτων Jira είναι το έργο της αυτόματης κατηγοριοποίησης των αναφορών σφαλμάτων με βάση το περιεχόμενό τους. Το έργο αυτό είναι απαραίτητο για τις ομάδες ανάπτυξης λογισμικού, καθώς μπορεί να τις βοηθήσει να δώσουν προτεραιότητα σε περισσότερο επείγοντα ζητήματα καθώς και να καταναείμουν αποτελεσματικά τους πόρους τους. Οι παραδοσιακές μέθοδοι για την ταξινόμηση αναφορών σφαλμάτων περιλαμβάνουν χειροκίνητη επισήμανση και προσεγγίσεις που βασίζονται σε κανόνες, οι οποίες είναι χρονοβόρες και επιρρεπείς σε σφάλματα. Οι προσεγγίσεις που βασίζονται σε NLP και AI μπορούν να αυτοματοποιήσουν αυτό το έργο, μειώνοντας τον χρόνο και την προσπάθεια που απαιτούνται για την ταξινόμηση των αναφορών σφαλμάτων [11].

3.7 Χρήση της βιβλιοθήκης SpaCy για την ταξινόμηση αναφορών σφαλμάτων Jira

Οι αναφορές σφαλμάτων υποβάλλονται σε προεπεξεργασία χρησιμοποιώντας τον αγωγό επεξεργασίας κειμένου του Spacy, ο οποίος περιλαμβάνει ένα ευρύ φάσμα χαρακτηριστικών για την επεξεργασία και ανάλυση δεδομένων κειμένου, όπως κωδικοποίηση, επισήμανση μέρους του λόγου και ανάλυση εξαρτήσεων. Επιπλέον, οι δυνατότητες αναγνώρισης οντοτήτων του SpaCy μπορούν επίσης να χρησιμοποιηθούν για τον εντοπισμό συστατικών λογισμικού και αριθμών έκδοσης που αναφέρονται στις αναφορές σφαλμάτων, οι οποίες μπορούν να είναι χρήσιμες στη διαδικασία ταξινόμησης. Τα προεπεξεργασμένα δεδομένα που αποτελούνται από αναφορές σφαλμάτων και τις σχετικές κατηγορίες τους χρησιμοποιούνται στη συνέχεια για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης, όπως μια Μηχανή Διανυσμάτων Υποστήριξης ή ένα Νευρωνικό Δίκτυο, για την πρόβλεψη της κατηγορίας μιας νέας αναφοράς σφάλματος. Στο πλαίσιο της ταξινόμησης σφαλμάτων με χρήση NLP/AI, το SpaCy μπορεί να χρησιμοποιηθεί για διάφορες εργασίες, όπως προεπεξεργασία κειμένου, εξαγωγή χαρακτηριστικών και ανάπτυξη μοντέλων. Όπως αναφέρθηκε παραπάνω, ορισμένα από τα βασικά χαρακτηριστικά του SpaCy που το καθιστούν δημοφιλή επιλογή για εργασίες NLP περιλαμβάνουν [11][12] :

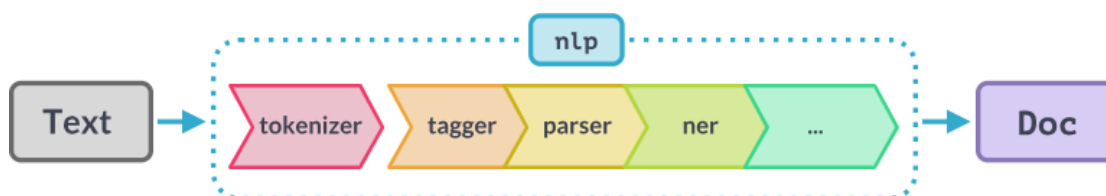
- Κωδικοποίηση (Tokenization): Το SpaCy παρέχει στιβαρή και παραμετροποιήσιμη κωδικοποίηση που μπορεί να χειριστεί σύνθετες δομές όπως συστολές, σύνθετες λέξεις και μη τυποποιημένα σημεία στίξης. Αυτό το χαρακτηριστικό μπορεί να είναι ιδιαίτερα καθοριστικό σε εργασίες ταξινόμησης σφαλμάτων όπου ο εντοπισμός μεμονωμένων λέξεων και φράσεων είναι σημαντικός.
- Επισήμανση μέρους του λόγου (Part-of-speech tagging): Το SpaCy μπορεί να αντιστοιχίσει αυτόματα μέρη του λόγου σε κάθε σύμβολο σε ένα κείμενο, το οποίο μπορεί να είναι χρήσιμο για τον εντοπισμό σχετικών λέξεων-κλειδιών και φράσεων που μπορεί να είναι ενδεικτικές ενός συγκεκριμένου τύπου σφάλματος.
- Αναγνώριση ονομαστικών οντοτήτων (NER): Το SpaCy μπορεί να εντοπίσει και να επισημάνει επώνυμες οντότητες όπως άτομα, οργανισμούς και τοποθεσίες σε ένα

κείμενο, γεγονός που μπορεί να είναι ωφέλιμο για τον εντοπισμό του σχετικού πλαισίου για την ταξινόμηση σφαλμάτων.

- Ανάλυση εξαρτήσεων (Dependency Parsing): Το SpaCy μπορεί να αναλύσει τη γραμματική δομή μιας πρότασης και να εντοπίσει τις σχέσεις μεταξύ των λέξεων, γεγονός που μπορεί να είναι χρήσιμο για την κατανόηση του πλαισίου και του νοήματος μιας αναφοράς σφαλμάτων.
- Προσαρμογή (Customization): Το SpaCy παρέχει μια σειρά επιλογών προσαρμογής για την εκπαίδευση και τη λεπτομερή ρύθμιση των μοντέλων NLP. Αυτό το χαρακτηριστικό μπορεί να είναι ιδιαίτερα βοηθητικό σε εργασίες ταξινόμησης σφαλμάτων όπου απαιτείται ειδική γνώση του τομέα.

Στην ταξινόμηση σφαλμάτων με χρήση NLP/AI, το SpaCy μπορεί να χρησιμοποιηθεί με διάφορους τρόπους. Για παράδειγμα, το SpaCy μπορεί να χρησιμοποιηθεί για εργασίες προεπεξεργασίας κειμένου, όπως tokenization, lemmatization, και αφαίρεση stopwords. Μπορεί επίσης να χρησιμοποιηθεί για την εξαγωγή σχετικών χαρακτηριστικών από τα δεδομένα κειμένου, όπως ετικέτες μέρους του λόγου και ονομαστικές οντότητες. Επιπλέον, το SpaCy μπορεί να χρησιμοποιηθεί για την εκπαίδευση και την τελειοποίηση μοντέλων μηχανικής μάθησης για εργασίες ταξινόμησης σφαλμάτων, όπως η ταξινόμηση κειμένου και η επισήμανση ακολουθιών [13]. Ενδεικτικά, ακολουθεί μία απεικόνιση της λειτουργικότητας της SpaCy:

Εικόνα 1: Χαρακτηριστικά της βιβλιοθήκης SpaCy



Συμπερασματικά, το SpaCy είναι μια ισχυρή και ευέλικτη βιβλιοθήκη για εργασίες NLP που μπορεί να αποτελέσει ένα πολύτιμο εργαλείο για την ταξινόμηση σφαλμάτων με χρήση NLP/AI. Τα χαρακτηριστικά της για την επεξεργασία κειμένου, την εξαγωγή χαρακτηριστικών και την ανάπτυξη μοντέλων την καθιστούν μια δημοφιλή επιλογή μεταξύ των ερευνητών και των επαγγελματιών της κοινότητας NLP.

4. Μεθοδολογία Μηχανικής Μάθησης

4.1 Εισαγωγή

Η μηχανική μάθηση (ML) έχει μετασηματίσει σημαντικά πολλούς τομείς τα τελευταία χρόνια, ενδυναμώνοντας τους υπολογιστές να αποκτούν γνώσεις από δεδομένα και να λαμβάνουν αυτόνομα αποφάσεις ή προβλέψεις. Αυτό το κεφάλαιο διερευνά τη μεθοδολογία της μηχανικής μάθησης, συμπεριλαμβανομένης της κατηγοριοποίησης των προβλημάτων, δυσκολίες ταξινόμησης, και την ταξινόμηση των μεθόδων μηχανικής εκμάθησης με βάση τις τεχνικές εκπαίδευσης.

4.2 Κατηγορίες προβλημάτων στην εφαρμογή της μηχανικής μάθησης

Οι μέθοδοι μηχανικής μάθησης εφαρμόζονται σε ένα ευρύ φάσμα κατηγοριών ζητημάτων, καθένα από τα οποία έχει ξεχωριστά χαρακτηριστικά και προκλήσεις:

- **Ανάλυση παλινδρόμησης:** Η ανάλυση παλινδρόμησης είναι ένας τύπος εποπτευόμενης μάθησης όπου ο στόχος είναι η πρόβλεψη συνεχών αριθμητικών τιμών χρησιμοποιώντας τα χαρακτηριστικά εισόδου. Κάποιες από τις εφαρμογές της παλινδρόμησης περιλαμβάνουν την εκπόνηση προβλέψεων σχετικά με τις τιμές των κατοικιών χρησιμοποιώντας παράγοντες όπως το μέγεθος, η τοποθεσία και ο αριθμός των υπνοδωματίων, ή την πρόβλεψη των τιμών των μετοχών χρησιμοποιώντας δεδομένα του παρελθόντος. [14]
- **Προβλήματα ταξινόμησης:** Τα προβλήματα ταξινόμησης περιλαμβάνουν την προσπάθεια κατηγοριοποίησης των εισερχόμενων στοιχείων σε ήδη υπάρχουσες κατηγορίες. Είναι ένας τύπος εποπτευόμενης μάθησης και χρησιμοποιείται ευρέως σε διάφορους τομείς, όπως η υγειονομική περίθαλψη (για παράδειγμα, ανίχνευση ασθενειών), η χρηματοδότηση (π.χ. αξιολόγηση πιστωτικού κινδύνου) και η επεξεργασία φυσικής γλώσσας (όπως ανάλυση συναισθήματος). [14]
- **Ανάλυση Ομαδοποίησης:** Η ομαδοποίηση είναι μια μη εποπτευόμενη εργασία μάθησης που συνεπάγεται την ομαδικότητα σχετικών στοιχείων δεδομένων με βάση τις κοινές τους ιδιότητες. Συχνά χρησιμοποιείται για την κατανομή καταναλωτών, την ανίχνευση ανωμαλιών και την αναγνώριση μοτίβων. [14]
- **Μείωση των διαστάσεων:** Η μείωση των διαστάσεων αναφέρεται στην εφαρμογή προσεγγίσεων που προσπαθούν να μειώσουν τον αριθμό των χαρακτηριστικών ή των μεταβλητών σε ένα σύνολο δεδομένων, διατηρώντας παράλληλα σημαντικές πληροφορίες. Η ανάλυση κύριων συστατικών (PCA) και η Στοχαστική ενσωμάτωση γειτόνων με κατανομή t (t-SNE) είναι τεχνικές που χρησιμοποιούνται ευρέως για τη μείωση του αριθμού των διαστάσεων στα δεδομένα.
- **Αναγνώριση ανωμαλιών:** Η αναγνώριση ανωμαλιών, ή ανίχνευση ακραίων τιμών, είναι η διαδικασία της ανακάλυψης ασυνήθιστων μοτίβων ή περιπτώσεων στα δεδομένα που αποκλίνουν από την αναμενόμενη τυπική συμπεριφορά. Βρίσκει χρήση στους τομείς της ανίχνευσης απάτης, της ασφάλειας δικτύου και της προγνωστικής συντήρησης.
- **Συστήματα συστάσεων:** Τα συστήματα συστάσεων χρησιμοποιούν τις προτιμήσεις και τη συμπεριφορά των χρηστών για να προσφέρουν προσαρμοσμένες προτάσεις για

προϊόντα, υπηρεσίες ή περιεχόμενο. Το συνεργατικό φιλτράρισμα και το φιλτράρισμα με βάση το περιεχόμενο είναι οι διαδεδομένες μεθοδολογίες που χρησιμοποιούνται στα συστήματα συστάσεων.

4.3 Θέματα κατηγοριοποίησης

Η ταξινόμηση είναι μια βασική δραστηριότητα στη μηχανική μάθηση, με κύριο στόχο την ανάθεση δεδομένων σε προκαθορισμένες κατηγορίες. Πιο συγκεκριμένα, παρακάτω παρουσιάζονται διάφορες κατηγορίες προβλημάτων ταξινόμησης [15] :

- Δυαδική ταξινόμηση: Η δυαδική ταξινόμηση αναφέρεται στη διαδικασία ταξινόμησης δεδομένων σε δύο ξεχωριστές κατηγορίες. Οι περιπτώσεις χρήσης μπορεί να περιλαμβάνουν την κατηγοριοποίηση των μηνυμάτων ηλεκτρονικού ταχυδρομείου σε spam ή μη spam, την πρόβλεψη της πιθανότητας ενός ασθενούς να έχει μια συγκεκριμένη ασθένεια, ή την αξιολόγηση του αν μια συναλλαγή είναι απάτη ή όχι.
- Ταξινόμηση πολλαπλών τάξεων: Η ταξινόμηση πολλαπλών τάξεων είναι η διαδικασία ταξινόμησης δεδομένων σε περισσότερες από δύο διαφορετικές κατηγορίες. Για παράδειγμα, η διαδικασία της κατηγοριοποίησης φωτογραφιών ζώων σε συγκεκριμένες ομάδες όπως σκύλος, γάτα, πουλί, και ούτω καθεξής. Η λογιστική παλινδρόμηση, τα δέντρα αποφάσεων και τα νευρωνικά δίκτυα είναι συχνά χρησιμοποιούμενες μέθοδοι για την κατηγοριοποίηση πολλών τάξεων.
- Ταξινόμηση πολλαπλών ετικετών: Η ταξινόμηση πολλαπλών ετικετών είναι ένας τύπος ταξινόμησης όπου οντότητες μπορούν να εκχωρηθούν σε πολλές κατηγορίες ταυτόχρονα. Για παράδειγμα, μία περίπτωση εφαρμογής περιλαμβάνει την κατηγοριοποίηση ειδησεογραφικών άρθρων σε τομείς όπως η πολιτική, ο αθλητισμός, η ψυχαγωγία κ.λπ., με τη δυνατότητα ενός άρθρου να ανήκει σε πολλές κατηγορίες.
- Μη ισορροπημένη ταξινόμηση: Η μη ισορροπημένη ταξινόμηση αναφέρεται στην ανάλυση των συνόλων δεδομένων στις οποίες μία τάξη είναι ουσιαστικά πιο κυρίαρχη από τις άλλες. Παρουσιάζει δυσκολίες όπως μεροληπτικές προβλέψεις που ευνοούν την πλειοψηφική τάξη και την απαίτηση για εξειδικευμένες στρατηγικές για την αντιμετώπιση των ανισορροπιών στην ταξική κατανομή. Για την αντιμετώπιση των δυσκολιών της ταξικής ανισορροπίας, μπορεί κανείς να χρησιμοποιήσει τεχνικές όπως η επαναδειγματοληψία, η μάθηση ευαίσθητη στο κόστος και οι μέθοδοι συνόλου.

4.4 Κατηγορίες της μηχανικής μάθησης με βάση τη μέθοδο εκπαίδευσης

Οι αλγόριθμοι μηχανικής μάθησης μπορούν να ταξινομηθούν ανάλογα με τη μέθοδο εκπαίδευσης που χρησιμοποιούν, καθένας από τους οποίους έχει ξεχωριστές ιδιότητες και εφαρμογές:

- Εποπτευόμενη μάθηση: Η εποπτευόμενη μάθηση αναφέρεται στη διαδικασία με την οποία οι αλγόριθμοι αποκτούν γνώση από δεδομένα που έχουν επισημανθεί, πράγμα που σημαίνει ότι κάθε σημείο δεδομένων συνδέεται με μια συγκεκριμένη ετικέτα

στόχου. Στόχος τους είναι να μάθουν μια αντιστοίχιση από τα χαρακτηριστικά εισόδου στις ετικέτες εξόδου, επιτρέποντάς τους να κάνουν προβλέψεις σε νέα δεδομένα. Παραδείγματα περιλαμβάνουν τη γραμμική παλινδρόμηση, τη λογιστική παλινδρόμηση, τις μηχανές διανυσμάτων υποστήριξης και τα τυχαία δάση. [14]

- Μη εποπτευόμενη μάθηση: Η μη εποπτευόμενη μάθηση αναφέρεται στη διαδικασία κατά την οποία οι αλγόριθμοι εξάγουν μοτίβα ή δομές από δεδομένα που δεν συνδέονται με ετικέτες. Ο στόχος τους είναι να αποκαλύψουν κρυμμένα μοτίβα, να συγκεντρώσουν συγκρίσιμα σημεία δεδομένων ή να μειώσουν τις διαστάσεις των δεδομένων χωρίς ρητή καθοδήγηση. Η μη εποπτευόμενη μάθηση περιλαμβάνει αλγόριθμους ομαδοποίησης όπως K-means και ιεραρχική συσσώρευση, καθώς και τεχνικές μείωσης διαστάσεων όπως PCA και t-SNE. [14]
- Ημιοπτευόμενη μάθηση: Η ημιοπτευόμενη μάθηση είναι μια ενδιάμεση προσέγγιση που εμπίπτει μεταξύ της εποπτευμένης και της μη εποπτευόμενης μάθησης. Σε αυτή τη μέθοδο οι αλγόριθμοι μαθαίνουν από έναν συνδυασμό δεδομένων με και χωρίς ετικέτες. Αξιοποιεί την αφθονία των μη επισημειωμένων δεδομένων για να βελτιώσει την απόδοση του μοντέλου σε σενάρια όπου τα επισημειωμένα δεδομένα είναι σπάνια ή ακριβά για να αποκτηθούν. Οι αλγόριθμοι μάθησης με ημι-επίβλεψη περιλαμβάνουν μεθόδους αυτο-εκπαίδευσης, συν-εκπαίδευσης και διάδοσης ετικετών. [16]
- Ενισχυτική μάθηση: Η ενισχυτική μάθηση είναι ένας τύπος μάθησης όπου ένας πράκτορας μαθαίνει να λαμβάνει αποφάσεις αλληλεπιδρώντας με ένα περιβάλλον. Ο πράκτορας λαμβάνει ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών με βάση τις ενέργειές του, επιτρέποντάς του να μαθαίνει τις βέλτιστες στρατηγικές με την πάροδο του χρόνου. Οι αλγόριθμοι ενισχυτικής μάθησης, όπως η μάθηση Q και τα βαθιά δίκτυα Q (DQN), χρησιμοποιούνται ευρέως στη ρομποτική, τα τυχερά παιχνίδια και τα αυτόνομα συστήματα. [14]

Εν κατακλείδι, αυτό το κεφάλαιο παρουσιάζει μια λεπτομερή ανάλυση της μεθοδολογίας της μηχανικής μάθησης, που περιλαμβάνει την κατηγοριοποίηση των προβλημάτων, τα προβλήματα ταξινόμησης, και τις κατηγορίες μηχανικής μάθησης με βάση τις μεθόδους εκπαίδευσης. Η κατανόηση αυτών των εννοιών είναι απαραίτητη για την αποτελεσματική εφαρμογή τεχνικών μηχανικής μάθησης σε διάφορα προβλήματα του πραγματικού κόσμου, επιτρέποντας στους ερευνητές και τους επαγγελματίες να αντιμετωπίσουν σύνθετες προκλήσεις και να λάβουν αποφάσεις με βάση τα δεδομένα.

5. Βιβλιογραφική ανασκόπηση

Το θέμα της μελέτης του αυτοματοποιημένου ελέγχου λογισμικού με τεχνητή νοημοσύνη καλύπτεται λεπτομερώς σε αυτό το κεφάλαιο, το οποίο επικεντρώνεται στην ταξινόμηση των αναφορών σφαλμάτων Jira σε πολλαπλές κατηγορίες για την αξιολόγηση της προτεραιότητας και της σοβαρότητάς τους. Οι πηγές που περιλαμβάνονται εδώ καλύπτουν ένα ευρύ φάσμα ερευνητικών μελετών και ακαδημαϊκών εργασιών που εξετάζουν πολλές πτυχές της δοκιμής λογισμικού, προσεγγίσεις μηχανικής μάθησης, στρατηγικές ταξινόμησης σφαλμάτων καθώς και την επεξεργασία αναφορών σφαλμάτων Jira. Αυτό το κεφάλαιο καθορίζει τις βάσεις για τη δημιουργία και την αξιολόγηση του προτεινόμενου συστήματος ταξινόμησης πολλαπλών κατηγοριών για τις αναφορές σφαλμάτων Jira στα επόμενα κεφάλαια, συνδυάζοντας και εξετάζοντας την τρέχουσα βιβλιογραφία.

Η σημαντική μελέτη των Tabassum et al. [17], εισάγει, κατασκευάζει και αξιολογεί τέσσερις ταξινομητές: Multinomial Naive Bayes, Decision Tree, Logistic Regression και Random Forest. Μετά από προσαρμογή των υπερ-παραμέτρων των μοντέλων, διαπιστώθηκε ότι το Random Forest είχε την καλύτερη απόδοση, επιτυγχάνοντας ακρίβεια δοκιμής 91,73% και ακρίβεια εκπαίδευσης 100%. Η τεχνική SMOTE χρησιμοποιήθηκε για να διορθωθεί η σημαντική ανισορροπία στο σύνολο δεδομένων, η οποία δημιουργήθηκε αρχικά για τους σκοπούς της αιτιολογημένης ταξινόμησης. Η σύγκριση των μοντέλων που εκπαιδεύτηκαν σε ισορροπημένα και μη ισορροπημένα σύνολα δεδομένων κατέδειξε αναμφισβήτητα τη σημασία της χρήσης ενός ισοζυγισμένου συνόλου δεδομένων για την ταξινόμηση, καθώς συνεχώς έδινε τις βέλτιστες επιδόσεις σε όλα τα πειράματά τους.

Επιπροσθέτως, η έρευνα των Kaur & Jindal [18], αξιολογεί διάφορες μεθόδους μηχανικής μάθησης για την πρόβλεψη της ιεράρχησης ελαττωμάτων. Λαμβάνει υπόψη τις αναφορές χαρακτηριστικών σφαλμάτων που κατηγοριοποιούν τις σοβαρότητες των σφαλμάτων ως "Severe" για blocker, κρίσιμα και μείζονα ζητήματα και "Non-severe" για δευτερεύοντα και ασήμαντα σφάλματα. Οι στρατηγικές αξιολογούνται σε 13 έργα Apache που ανακτώνται αυτόματα χρησιμοποιώντας το εργαλείο Bug Report Collection System. Τα ευρήματα της μελέτης επιβεβαιώνουν ότι η προσέγγιση Boosting, ένα ensemble learner, ξεπερνά άλλες τεχνικές μηχανικής μάθησης, όπως Bayesian learners, decision trees, and support vector machines, που χρησιμοποιήθηκαν σε προηγούμενες μελέτες.

Παράλληλα, η έρευνα των Choudhary & Singh [19], επικεντρώνεται στην επιλογή αναφορών σφαλμάτων από τρία συγκεκριμένα components του Eclipse. Τα μοντέλα πρόβλεψης κατασκευάζονται με 10-fold cross-validation και αξιολογούνται σε πέντε επαναλήψεις του Eclipse. Τα αποτελέσματά τους δείχνουν την αποτελεσματική κατηγοριοποίηση των προτεραιοτήτων των σφαλμάτων λογισμικού χρησιμοποιώντας τους αλγόριθμους MLP (Multilayer Perceptron) και Naïve Bayes. Πιο συγκεκριμένα, το μοντέλο MLP ταξινομεί αποτελεσματικά διαφορετικές προτεραιότητες, με το ROC να κυμαίνεται από 87,6% έως 98,6%, καθώς και το μοντέλο Naïve Bayes έχει υψηλή αποδοτικότητα στην ταξινόμηση διαφορετικών προτεραιοτήτων, με το ROC να κυμαίνεται από το 89% έως 98%.

Ακόμη, οι Colavito et al. [20], παρέχοντας διορατικές πληροφορίες σχετικά με τις πιο πρόσφατες προσεγγίσεις που χρησιμοποιούνται στον κλάδο, εμβαθύνουν στην αυτοματοποιημένη κατηγοριοποίηση των ζητημάτων του GitHub. Οι ερευνητές διεξήγαγαν πειράματα χρησιμοποιώντας διάφορα προ-εκπαιδευμένα μοντέλα, συμπεριλαμβανομένων του BERT και

των παραλλαγών του ALBERT και RoBERTa. Η υψηλότερη απόδοση επιτεύχθηκε από την fine-tuned RoBERTa, με F1-score 0.8591.

Επιπλέον, η έρευνα των Pandey et al. [21], εμβαθύνει στη χρήση μεθόδων επεξεργασίας φυσικής γλώσσας (NLP) στην ανάλυση αναφορών σφαλμάτων Jira, τονίζοντας τη σημασία της προεπεξεργασίας δεδομένων κειμένου και της εξαγωγής χαρακτηριστικών για εργασίες κατηγοριοποίησης προβλημάτων. Τα αποτελέσματά τους υπογραμμίζουν τη σημασία των πληροφοριών που αφορούν συγκεκριμένο τομέα και των γλωσσικών παραγόντων για την ενίσχυση της ακρίβειας της πρόβλεψης της σοβαρότητας των σφαλμάτων.

Στη μελέτη τους, οι Kukkar et al. [22], εισήγαγαν ένα καινοτόμο πλαίσιο βαθιάς μάθησης για την ταξινόμηση της σοβαρότητας πολλαπλών κατηγοριών. Η προσέγγιση αυτή περιελάμβανε τη χρήση του Convolutional Neural Network (CNN) σε συνδυασμό με τις τεχνικές Random Forest και Boosting (BCR). Οι ερευνητές χρησιμοποίησαν την τεχνική της επεξεργασίας φυσικής γλώσσας για την προεπεξεργασία των αναφορών σφαλμάτων, ακολουθούμενη από τη χρήση του n-gram για την εξαγωγή χαρακτηριστικών. Στη συνέχεια, χρησιμοποιείται το CNN για την εξαγωγή σημαντικών προτύπων χαρακτηριστικών που αντιστοιχούν σε διάφορες κατηγορίες σοβαρότητας. Τέλος, ένα μοντέλο τυχαίου δάσους με boosting χρησιμοποιείται για την ταξινόμηση των πολλών κλάσεων σοβαρότητας σφαλμάτων. Η μέση ακρίβεια του προτεινόμενου μοντέλου είναι 96,34%.

Οι Kanwal & Maqbool [23], παρουσιάζουν στη μελέτη τους την ανάπτυξη ενός ταξινομητή προτεραιότητας σφαλμάτων που αυτοματοποιεί την απόδοση προτεραιότητας σφαλμάτων σε εισερχόμενες αναφορές σφαλμάτων σε ένα αποθετήριο. Ο ταξινομητής χρησιμοποιεί τις τεχνικές ταξινόμησης SVM και Naïve Bayes και τα αποτελέσματά τους συγκρίνονται. Η ακρίβεια των ταξινομητών καταδεικνύει ότι ο προτεινόμενος εισηγητής προτεραιοτήτων μπορεί να βοηθήσει τους δοκιμαστές λογισμικού στην ανάθεση του σωστού επιπέδου προτεραιότητας στις αναφορές σφαλμάτων, εξασφαλίζοντας την έγκαιρη επίλυσή τους. Τα πειραματικά τους αποτελέσματα, τα οποία αξιολογήθηκαν με τη χρήση της ακρίβειας, της ανάκλησης, των NFN (False Negatives) και NFP (False Positives), αποδεικνύουν ότι η απόδοση του SVM (Support Vector Machine) είναι ανώτερη από εκείνη του Naïve Bayes. Ο ταξινομητής προτεραιότητας σφαλμάτων για SVM αποδίδει καλύτερα αποτελέσματα όταν τα χαρακτηριστικά κατηγορίας και κειμένου συγχωνεύονται για την εκπαίδευση του ταξινομητή, ενώ η απόδοση του ταξινομητή Naïve Bayes βελτιώνεται όταν η μακροσκελής περιγραφή σφάλματος αποκλείεται ως χαρακτηριστικό. Αυτό αποδεικνύει ότι ο Naïve Bayes είναι λιγότερο αποτελεσματικός από τον SVM στη διαχείριση της μεγάλης διάστασης των χαρακτηριστικών κειμένου.

Ακόμη, η ανάλυση των Pandey et al. [24], αποκαλύπτει ότι ο αλγόριθμος Random Forest (RF) επιτυγχάνει σταθερά το μέγιστο F-measure, τη μέση ακρίβεια και το σταθμισμένο μέσο F-measure σε όλα τα μεγέθη του πίνακα όρων εγγράφων και τις τιμές κανονικοποίησης που εξέτασαν. Παράλληλα, προκύπτει από την έρευνά τους ότι τα μοντέλα μηχανών διανυσμάτων υποστήριξης (SVM) που χρησιμοποιούν πυρήνες sigmoid και rbf, καθώς και η γραμμική ανάλυση διάκρισης (LDA) σε κάποιο βαθμό, παρέχουν επιδόσεις συγκρίσιμες με αυτές του μοντέλου τυχαίου δάσους (RF). Τα ευρήματά τους υποδηλώνουν ότι ταξινομητές όπως το τυχαίο δάσος και το SVM μπορούν να επιτύχουν ακρίβεια που κυμαίνεται από 75% έως 83%, ανάλογα με το συγκεκριμένο έργο.

Επιπλέον, οι Tian et al. [25], παρουσίασαν ένα σύστημα που ονομάζεται DRONE, το οποίο αποσκοπεί στην πρόβλεψη των επιπέδων προτεραιότητας των αναφορών σφαλμάτων στο Bugzilla. Έλαβαν υπόψη διάφορα κριτήρια, όπως χρόνος, γλώσσα, συγγραφέα, σχετική αναφορά, σοβαρότητα και προϊόν. Τα χαρακτηριστικά που εξάγονται εισάγονται σε μια μηχανή

ταξινόμησης που ονομάζεται GRAY, η οποία δημιουργείται συνδυάζοντας γραμμική παλινδρόμηση με μια στρατηγική κατωφλίου, με σκοπό να αντιμετωπιστεί το πρόβλημα της ανισορροπίας των δεδομένων και να αποδοθούν ετικέτες προτεραιότητας στις αναφορές σφαλμάτων. Η ανάλυσή τους σε ένα σύνολο δεδομένων που περιλαμβάνει πάνω από 100.000 αναφορές σφαλμάτων από το Eclipse αποδεικνύει ότι η μέθοδός τους ξεπερνά τα baselines όσον αφορά το μέσο μέτρο F, με σχετική βελτίωση έως και 209%.

6. Μεθοδολογία ταξινόμησης αναφορών σφαλμάτων Jira με χρήση NLP και AI

6.1 Εισαγωγή στην ταξινόμηση αναφορών σφαλμάτων Jira

Το Jira είναι ένα θεμελιώδες εργαλείο στις σύγχρονες μεθοδολογίες διαχείρισης έργων. Δημιουργήθηκε από την Atlassian Corporation το 2002 ως εξειδικευμένη λύση για την παρακολούθηση προβλημάτων σε έργα ανάπτυξης λογισμικού. Σχεδιάστηκε αρχικά για να χειριστεί την παρακολούθηση σφαλμάτων και την επίλυση προβλημάτων στις ομάδες ανάπτυξης, αλλά στη συνέχεια υπέστη σημαντική ανάπτυξη, διευρύνοντας το εύρος και τις δυνατότητές του. Το αρχικό απλό εργαλείο παρακολούθησης προβλημάτων έχει εξελιχθεί σε μια πλήρη πλατφόρμα διαχείρισης έργων που είναι ιδιαίτερα σεβαστή για την ευελιξία και την αποτελεσματικότητά της σε διάφορους οργανωτικούς τομείς. Η δημοτικότητα και η αξία του Jira προέρχονται κυρίως από τη λειτουργικότητά του, καθώς στο επίκεντρο του εύρους των δυνατοτήτων του είναι το ισχυρό σύστημα παρακολούθησης προβλημάτων, το οποίο επιτρέπει στους χρήστες να δημιουργούν αποτελεσματικά, να δίνουν προτεραιότητα, να αναθέτουν και να επιλύουν προβλήματα. Το Jira υπερβαίνει την απλή παρακολούθηση προβλημάτων και αντιμετωπίζει συγκεκριμένα τις περιπλοκές των ευέλικτων προσεγγίσεων διαχείρισης έργων. Έχει εξειδικευμένες δυνατότητες όπως οι Scrum και Kanban πίνακες. Αυτά τα εργαλεία επιτρέπουν στις ομάδες να βλέπουν αποτελεσματικά την πρόοδο των εργασιών, να οργανώνουν αποδοτικά τις περιόδους εργασίας και να αυξάνουν την αποτελεσματικότητα των επαναλαμβανόμενων διαδικασιών ανάπτυξης. Επιπλέον, οι προσαρμόσιμες ροές εργασίας του Jira επιτρέπουν στις εταιρείες να εξατομικεύουν τις ροές εργασιών έργου σύμφωνα με τις δικές τους ανάγκες, προάγοντας τη συμμόρφωση με τις καθιερωμένες διαδικασίες και βελτιστοποιώντας την υλοποίηση του έργου. Οι προηγμένες δυνατότητες αναφοράς και ανάλυσης του Jira παρέχουν στα ενδιαφερόμενα μέρη εκτεταμένες πληροφορίες σχετικά με την πρόοδο του έργου, την απόδοση της ομάδας και την κατανομή των πόρων. Αυτό επιτρέπει τη λήψη αποφάσεων που βασίζονται σε δεδομένα και τα προγράμματα συνεχούς βελτίωσης. Οι εγγενείς ιδιότητες του Jira ενισχύουν σημαντικά την αποτελεσματικότητά του ως εργαλείο διαχείρισης έργων. Ένα από τα πιο εξέχοντα χαρακτηριστικά είναι η διαισθητική διεπαφή του, που αναπτύχθηκε ειδικά για να καλύψει τους χρήστες με διαφορετικά επίπεδα τεχνικής εμπειρογνομosύνης. Επιπλέον, η ικανότητα επέκτασης της Jira την καθιστά κατάλληλη για έργα οποιουδήποτε μεγέθους, από πρωτοβουλίες μικρής κλίμακας έως μεγάλες επιχειρηματικές δραστηριότητες. Ακόμη, η δυνατότητα ενσωμάτωσής του με ένα ευρύ φάσμα εργαλείων και εφαρμογών βελτιώνει τη διαλειτουργικότητα και επεκτείνει τη χρησιμότητά του πέρα από βασικές εργασίες διαχείρισης έργων. Συμπερασματικά, το Jira διευκολύνει την άμεση και ανοιχτή επικοινωνία και ανταλλαγή γνώσεων μεταξύ των μελών της ομάδας, καλλιεργώντας έτσι μια κουλτούρα ευθύνης και κοινής ιδιοκτησίας, ενώ τηρεί τις αξίες της συνεργασίας και της διαφάνειας. Χρησιμοποιείται ευρέως σε διάφορες βιομηχανίες και οργανωτικές ρυθμίσεις λόγω της ευελιξίας και της ποικιλομορφίας του, ενώ διαδραματίζει κρίσιμο ρόλο στην παροχή αποτελεσματικών μεθόδων διαχείρισης έργων σε διάφορες βιομηχανίες, συμπεριλαμβανομένων των επιχειρήσεων ανάπτυξης λογισμικού, των οργανισμών μάρκετινγκ, των ακαδημαϊκών ιδρυμάτων και των οργανώσεων υγειονομικής περίθαλψης. Η χρησιμότητα αυτού του εργαλείου καλύπτει ένα ευρύ φάσμα εφαρμογών, όπως η ανάπτυξη λογισμικού, η διαχείριση προϊόντων, οι λειτουργίες πληροφορικής, η διοίκηση εκστρατειών μάρκετινγκ, ακόμη και ο συντονισμός ακαδημαϊκών ερευνητικών έργων. Συνοψίζοντας, ο μετασχηματισμός της Jira από ένα βασικό εργαλείο για την καταγραφή προβλημάτων σε μια ολοκληρωμένη πλατφόρμα για τη διαχείριση

έργων υπογραμμίζει τη σημασία της στις σύγχρονες μεθοδολογίες διαχείρισης έργων. Οι ποικίλες δυνατότητες του λογισμικού, η ευκολία χρήσης της διεπαφής, η ικανότητα προσαρμογής στις διαφορετικές ανάγκες και η ευελιξία το καθιστούν ένα απαραίτητο εργαλείο για τις επιχειρήσεις που στοχεύουν στη βελτίωση της υλοποίησης του έργου και την προώθηση μιας κουλτούρας ομαδικής εργασίας και δημιουργικότητας.

6.2 Επισκόπηση της εποπτευόμενης μάθησης και της ταξινόμησης πολλαπλών τάξεων
Η εποπτευόμενη μάθηση είναι ένα κυρίαρχο πλαίσιο στη μηχανική μάθηση, στο οποίο οι αλγόριθμοι αποκτούν γνώση από τα δεδομένα με ετικέτα προκειμένου να κάνουν προβλέψεις. Μέσα σε αυτό το πλαίσιο, ο αλγόριθμος υποβάλλεται σε εκπαίδευση χρησιμοποιώντας ένα σύνολο δεδομένων που αποτελείται από ζεύγη εισόδου-εξόδου. Η είσοδος αντιστοιχεί στις ιδιότητες των δεδομένων, ενώ η έξοδος είναι η μεταβλητή στόχου ή η ετικέτα. Ο στόχος της εποπτευόμενης μάθησης είναι να αποκτήσει μια λειτουργία που χαρτογραφεί με ακρίβεια τα δεδομένα εισόδου στις αντίστοιχες τιμές εξόδου, με την ικανότητα να εφαρμόσει αποτελεσματικά αυτή τη χαρτογράφηση σε νέα, προηγουμένως άορατα δεδομένα. Η ταξινόμηση πολλών τάξεων, ένας τύπος εποπτευόμενης μάθησης, περιλαμβάνει την κατηγοριοποίηση περιστατικών σε πολλές τάξεις ή κατηγορίες.

Στο πλαίσιο της μελέτης διαχείρισης σφαλμάτων, χρησιμοποιούμε την ταξινόμηση πολλαπλών κατηγοριών για να ταξινομήσουμε τις αναφορές σφάλματος του Jira σε διάφορες κατηγορίες που υποδεικνύουν το επίπεδο προτεραιότητας και σοβαρότητάς τους, όπως:

- **Σημαντικό (Major):** Τα μεγάλα σφάλματα ταξινομούνται ως ελαττώματα που έχουν ουσιαστική επίδραση στη λειτουργία ή τη χρησιμότητα του προϊόντος, αλλά δεν είναι αρκετά καταστροφικά για να σταματήσουν τη διαδικασία ανάπτυξης. Αυτά τα ζητήματα έχουν τη δυνατότητα να δημιουργήσουν σημαντικές διαταραχές για τους χρήστες ή να επηρεάσουν κρίσιμες λειτουργίες του προϊόντος.
- **Ελάσσων (Minor):** Τα μικρότερα σφάλματα χαρακτηρίζονται από τη μικρότερη σοβαρότητά τους σε σύγκριση με τα σημαντικά, με αποτέλεσμα έναν σχετικά μικρότερο αντίκτυπο στη λειτουργικότητα του λογισμικού ή την εμπειρία του χρήστη. Αυτά τα σφάλματα μπορεί να περιλαμβάνουν επιφανειακά ελαττώματα, μικρά λάθη ή δυσκολίες που έχουν ελάχιστη επίδραση στη λειτουργικότητα του λογισμικού.
- **Κρίσιμο (Critical):** Τα κρίσιμα σφάλματα ταξινομούνται ως σοβαρά ελαττώματα που επηρεάζουν σημαντικά τη λειτουργικότητα, την απόδοση ή την ασφάλεια του λογισμικού. Αυτά τα προβλήματα συχνά καθιστούν το λογισμικό μη λειτουργικό ή παρουσιάζουν σημαντικούς κινδύνους για τους χρήστες, την ακεραιότητα των δεδομένων ή τη σταθερότητα του συστήματος. Η επείγουσα προσοχή και η θέσπιση προτεραιοτήτων είναι απαραίτητες για την επίλυση κρίσιμων σφαλμάτων.
- **Ανασταλτικό (Blocker):** Τα σφάλματα "Blocker" είναι ο πιο κρίσιμος τύπος ελαττωμάτων, που σηματοδοτούν προβλήματα που παρεμποδίζουν κρίσιμες λειτουργίες ή σταματούν εντελώς την πρόοδο στις διαδικασίες ανάπτυξης. Αυτά τα σφάλματα χρησιμεύουν ως εμπόδια, παρεμποδίζοντας την εκπλήρωση κρίσιμων καθηκόντων ή της σωστής λειτουργίας του λογισμικού. Η επίλυση ανασταλτικών ζητημάτων λαμβάνει την υψηλότερη προτεραιότητα προκειμένου να αποκατασταθεί η λειτουργικότητα του λογισμικού και να διευκολυνθεί η πρόοδος.

-
- Ασήμαντο (Trivial): Τα σφάλματα που ταξινομούνται ως "Trivial" είναι ασήμαντα προβλήματα ή επιφανειακά ελαττώματα που έχουν ελάχιστη ή καμία επίδραση στη λειτουργικότητα, τη χρηστικότητα ή την απόδοση του λογισμικού. Αν και τα ασήμαντα σφάλματα μπορεί να μην απαιτούν επείγουσα φροντίδα, εξακολουθούν να καταγράφονται και να παρακολουθούνται για να παρέχουν εμπειριστατωμένη διαχείριση σφαλμάτων και διασφάλιση ποιότητας.

Ο στόχος αυτής της εργασίας είναι να κατηγοριοποιηθεί κάθε αναφορά σφαλμάτων σε μία από τις παραπάνω προκαθορισμένες κατηγορίες προτεραιότητας, χρησιμοποιώντας τις πληροφορίες που παρέχονται στην αναφορά των σφαλμάτων.

6.3 Συλλογή και προεπεξεργασία δεδομένων

Στην παρούσα εργασία χρησιμοποιείται ένα σύνολο δεδομένων αναφορών σφαλμάτων από το Jira που περιέχει αρκετές ιδιότητες και ετικέτες που σχετίζονται με τη διαχείριση και την επίλυση προβλημάτων. Το σύνολο δεδομένων περιλαμβάνει διάφορα βασικά χαρακτηριστικά των αναφορών σφαλμάτων, τα οποία παρουσιάζονται παρακάτω:

- `issue_id`: το αναγνωριστικό του προβλήματος (για μοναδική αναγνώριση)
- `type`: ο τύπος του σφάλματος που υποδεικνύει τη φύση του (π.χ., σφάλμα, εργασία, ή βελτίωση)
- `status`: η τρέχουσα κατάσταση σφάλματος που αντικατοπτρίζει την πρόοδό του στη διαδικασία επίλυσης
- `resolution`: η κατάσταση της λύσης που δείχνει το αποτέλεσμα των προσπαθειών επίλυσής του (π.χ. είναι σταθερή, άλυτη, ή αλλιώς)
- `component`: περιλαμβάνει λεπτομέρειες σχετικά με το συγκεκριμένο στοιχείο του λογισμικού που επηρεάζεται από το σφάλμα
- `priority`: η προτεραιότητα που αποδίδεται στο σφάλμα με βάση τη σοβαρότητα και τον αντίκτυπό του
- `reporter`: το άτομο που ανέφερε/δημιούργησε το θέμα
- `created`: η ημερομηνία κατά την οποία δημιουργήθηκε η αναφορά σφαλμάτων
- `assigned`: η ημερομηνία κατά την οποία ανατέθηκε η αναφορά σφαλμάτων
- `resolved`: η ημερομηνία κατά την οποία επιλύθηκε η αναφορά σφαλμάτων
- `summary`: περιληπτική περιγραφή
- `description`: περιεκτική περιγραφή
- `affected_version`: η έκδοση του λογισμικού που επηρεάζεται

-
- fixed_versions: η σταθερή έκδοση του λογισμικού
 - votes: ο αριθμός των ψήφων
 - watches: προβολές
 - commenter: σχολιαστές

Η ερευνητική πρόκληση περιλαμβάνει την κατηγοριοποίηση αναφορών σφαλμάτων σε ορισμένες ομάδες ανάλογα με τη σημασία και τα επίπεδα σοβαρότητας τους, χρησιμοποιώντας κριτήρια όπως η προτεραιότητα, το στοιχείο, ο τύπος και οι περιγραφές κειμένου. Οι σχετικές ονομασίες περιλαμβάνουν κατηγορίες προτεραιότητας όπως "Blocker", "Critical", "Major", "Minor" και "Trivial", οι οποίες αντιπροσωπεύουν με ακρίβεια το επίπεδο σοβαρότητας που σχετίζεται με κάθε αναφορά σφαλμάτων.

Κατά την ανάλυση του συνόλου δεδομένων, γίνεται προφανές ότι η πλειοψηφική κατηγορία (Major) περιέχει σημαντικά μεγαλύτερο αριθμό περιπτώσεων σε σύγκριση με τις μειονοτικές κατηγορίες (Minor, Trivial, Critical, Blocker). Συμπερασματικά, το σύνολο δεδομένων είναι μη ισορροπημένο, εφόσον ορίζεται από μια ουσιαστική απόκλιση στην κατανομή των περιπτώσεων σε διάφορες κατηγορίες. Αναλυτικότερα, η διαφορά στη συχνότητα των συμβάντων σε αυτές τις κατηγορίες υποδηλώνει μια άνιση κατανομή, η οποία έχει τη δυνατότητα να επηρεάσει την αποτελεσματικότητα των αλγορίθμων μηχανικής μάθησης που εκπαιδεύονται σε αυτό το σύνολο δεδομένων. Οι αλγόριθμοι μπορούν να εμφανίζουν προκατάληψη προς την κυρίαρχη τάξη, με αποτέλεσμα να επιτυγχάνεται λιγότερο από τη βέλτιστη απόδοση κατά την πρόβλεψη των μειονοτικών τάξεων. Για να μειωθεί αυτή η ανισορροπία και να ενισχυθεί η απόδοση του μοντέλου στις μειονοτικές τάξεις, εφαρμόστηκαν δύο ευρέως αναγνωρισμένες μέθοδοι:

- Συνθετική τεχνική υπερβολικής δειγματοληψίας μειονοτήτων (SMOTE - Synthetic Minority Oversampling Technique) : Μετριάζει την ταξική ανισορροπία με τη σύνθεση τεχνητών δειγμάτων για την υποεκπροσωπούμενη τάξη, εξισώνοντας έτσι την κατανομή των τάξεων και εξοπλίζοντας τον αλγόριθμο με πιο ακριβή δεδομένα κατάρτισης.
- RandomOverSampler : Αναπαράγει τις περιπτώσεις από την υποεκπροσωπούμενη τάξη για να εξισορροπήσει τον αριθμό των περιπτώσεων στην υπερεκπροσωποποιημένη τάξη, μετριάζοντας έτσι τον αντίκτυπο της ανισορρόπησης των τάξεων.

Η ενσωμάτωση των SMOTE και RandomOverSampler στην έρευνα μας είναι ζωτικής σημασίας για να διασφαλίσουμε ότι το μοντέλο ταξινόμησης σφαλμάτων μας μπορεί να αποκτήσει γνώσεις από το πλήρες φάσμα των επιπέδων προτεραιότητας και σοβαρότητας του προβλήματος, χωρίς να εκδηλώνει προκατάληψη προς την κυρίαρχη κατηγορία. Με την αύξηση του αριθμού των παραδειγμάτων στην τάξη μειοψηφίας και την εξίσωση της κατανομής των τάξεων, αυτές οι στρατηγικές βελτιώνουν την ικανότητα του μοντέλου να κάνει γενικεύσεις σε όλες τις τάξεις, με αποτέλεσμα τη βελτίωση της αποτελεσματικότητας και της ακρίβειας κατά την πρόβλεψη των προτεραιοτήτων των σφαλμάτων. Ως εκ τούτου, αυτή η προσέγγιση εγγυάται ότι το μοντέλο δεν εκθέτει προκατάληψη προς την κυρίαρχη κατηγορία και μπορεί να γενικευθεί με επιτυχία σε όλες τις κατηγορίες, βελτιώνοντας έτσι τη συνολική ακρίβεια και αξιοπιστία των συστημάτων διαχείρισης σφαλμάτων.

6.4 Εξαγωγή και επιλογή χαρακτηριστικών

Αναφορικά με την επεξεργασία του κειμένου, οι αναφορές σφαλμάτων περιείχαν χαρακτηριστικά κειμένου όπως η περίληψη του σφάλματος και η εκτεταμένη του περιγραφή. Στην παρούσα προσέγγιση, χρησιμοποιήθηκε αποκλειστικά η εκτεταμένη περιγραφή του σφάλματος καθώς περιέχει περισσότερη αξιοποιήσιμη πληροφορία, ωστόσο και πάλι υπάρχουν αρκετά χαρακτηριστικά που αποτελούνται από λέξεις που στερούνται σημασίας.

Επομένως, για την μετατροπή των κειμένων σε μια συνεκτική και σημαντική αναπαράσταση, χρησιμοποιήθηκε μια συμβατική μεθοδολογία κατηγοριοποίησης κειμένου. Αρχικά, το σύνολο του κειμένου μετατράπηκε σε μεμονωμένες λέξεις με την εξάλειψη της διατύπωσης, των φραγμών και των ειδικών συμβόλων όπως το @, το \$ και το %. Οι όροι που εξαλείφθηκαν από τη λίστα περιλαμβάνουν stopwords (όπως is, am, I, he), κοινές λέξεις (όπως because, everywhere), και μη αλφαβητικές λέξεις (όπως 12345, -359, 2015). Αυτές οι λέξεις αποκλείστηκαν επειδή είναι ασήμαντες και δεν συνεισφέρουν πολλές πληροφορίες στην περιγραφή του προβλήματος της αναφοράς σφαλμάτων. Επιπλέον, χρησιμοποιήθηκε η τεχνική Stemming για να μετατρέψει έναν όρο στη θεμελιώδη σημασιολογική του μορφή, με αποτέλεσμα την συμπίκνωση λέξεων με διαφορετικές γραμματικές δομές σε μία λέξη. Για παράδειγμα, οι λέξεις "πειραματικά" και "πειράματα" άλλαξαν σε "πείραμα". Τα ρήματα μετατράπηκαν επίσης στη βασική τους μορφή, όπως το "ήταν" και το "είναι" αλλάζοντας σε "είμαι". Στη συνέχεια, καθίσταται απαραίτητη η μετατροπή τους σε διανύσματα, κάτι το οποίο επιτυγχάνεται με τη χρήση του αλγορίθμου διανυσματοποίησης κειμένου TF-IDF Vectorizer.

6.4.1 TF-IDF Vectorizer

Η διαδικασία μετατροπής κειμένου σε διάνυσμα αναφέρεται συνήθως ως διανυσματοποίηση κειμένου (text vectorization). Είναι μια θεμελιώδης διαδικασία στην επεξεργασία φυσικής γλώσσας, επειδή κανένας από τους αλγορίθμους μηχανικής μάθησης δεν μπορεί να αντιληφθεί ένα κείμενο. Πιο αναλυτικά, η συχνότητα όρων - αντίστροφη συχνότητα εγγράφων είναι ένας διανυσματοποιητής κειμένου που μετατρέπει το κείμενο σε ένα αξιοποιήσιμο διάνυσμα.

Συνδυάζει δύο έννοιες, τη συχνότητα όρων (TF) και τη συχνότητα εγγράφων (DF). Η συχνότητα όρων είναι ο αριθμός των εμφανίσεων ενός συγκεκριμένου όρου σε ένα έγγραφο. Η συχνότητα όρων αναπαριστά κάθε κείμενο από τα δεδομένα ως έναν πίνακα, του οποίου οι γραμμές είναι ο αριθμός των εγγράφων και οι στήλες είναι ο αριθμός των διακριτών όρων σε όλα τα έγγραφα. Η συχνότητα εγγράφων είναι ο αριθμός των εγγράφων που περιέχουν έναν συγκεκριμένο όρο και δείχνει πόσο συχνός είναι αυτός ο όρος.

Η αντίστροφη συχνότητα εγγράφων (IDF) είναι το βάρος ενός όρου, έχει ως στόχο τη μείωση του βάρους του, εάν οι εμφανίσεις του όρου είναι διάσπαρτες σε όλα τα έγγραφα. Η IDF μπορεί να υπολογιστεί ως εξής:

$$idf_i = \log \left(\frac{n}{df_i} \right),$$

όπου idf_i είναι η βαθμολογία IDF για τον όρο i , df_i είναι ο αριθμός των εγγράφων που περιέχουν τον όρο i και n είναι ο συνολικός αριθμός των εγγράφων. Όσο υψηλότερο είναι το DF ενός όρου, τόσο χαμηλότερο είναι το IDF του όρου. Το σκορ TF-IDF, όπως υποδηλώνει το όνομα, είναι

απλώς ένας πολλαπλασιασμός του πίνακα συχνοτήτων όρων με το IDF του, μπορεί να υπολογιστεί ως εξής:

$$w_{i,j} = tf_{i,j} \times idf_i ,$$

όπου w_{ij} είναι η βαθμολογία TF-IDF για τον όρο i στο έγγραφο j , tf_{ij} είναι η συχνότητα όρων για τον όρο i στο έγγραφο j και idf_i είναι η βαθμολογία IDF για τον όρο i .

Με χρήση της παραπάνω μεθόδου, όλο το σύνολο των κριτικών μετατρέπεται σε διανύσματα, τα οποία θα χρησιμοποιηθούν στο στάδιο της υλοποίησης των μοντέλων ως χαρακτηριστικά.

Στην συγκεκριμένη εργασία, θα εξεταστεί επαρκώς η χρήση της παραμέτρου `max_features` του αλγορίθμου διανυσματοποίησης TF-IDF Vectorizer. Ειδικότερα, η παράμετρος `max_features` στο TF-IDF vectorization ελέγχει τον μέγιστο αριθμό χαρακτηριστικών (δηλαδή, μοναδικές λέξεις ή όρους) που πρέπει να ληφθούν υπόψη κατά την κατασκευή των διανυσμάτων. Ακολουθούν κάποιοι από τους λόγους για τη χρήση της παραμέτρου `max_features` στο TF-IDF vectorization:

- **Μείωση διαστάσεων:** Σε πολλές εργασίες επεξεργασίας φυσικής γλώσσας, το μέγεθος του λεξιλογίου ενδέχεται να είναι πολύ μεγάλο, να αποτελείται δηλαδή από χιλιάδες ή ακόμα και εκατομμύρια μοναδικές λέξεις. Περιορίζοντας τον αριθμό των χαρακτηριστικών χρησιμοποιώντας την παράμετρο `max_features`, μπορούμε να μειώσουμε τη διάσταση του χώρου χαρακτηριστικών, καθιστώντας το μοντέλο υπολογιστικά αποδοτικότερο και μειώνοντας τον κίνδυνο υπερπροσαρμογής, ειδικά όταν υπάρχουν περιορισμένοι υπολογιστικοί πόροι ή μικρότερα σύνολα δεδομένων.
- **Εστίαση στα πιο σημαντικά χαρακτηριστικά:** Ενδέχεται όλες οι λέξεις του λεξιλογίου να μην είναι εξίσου σημαντικές, καθώς ορισμένες μπορεί να είναι σπάνιες, θορυβώδεις ή άσχετες με την εργασία ταξινόμησης. Με τον καθορισμό μιας κατάλληλης τιμής για τα `max_features`, μπορούμε να επικεντρωθούμε στα πιο ενημερωτικά και διακριτικά χαρακτηριστικά, ενδεχομένως βελτιώνοντας την απόδοση του μοντέλου, φιλτράροντας λιγότερο σχετικούς όρους.
- **Αποδοτικότητα μνήμης:** Η αποθήκευση και επεξεργασία διανυσμάτων με μεγάλο αριθμό διαστάσεων μπορεί να απαιτήσει σημαντικούς πόρους μνήμης. Επομένως, ο περιορισμός του αριθμού των χαρακτηριστικών με `max_features` μπορεί να βοηθήσει στη διατήρηση της, ειδικά σε περιβάλλοντα με περιορισμένο χώρο ή σε μεγάλα σύνολα δεδομένων.
- **Επιτάχυνση της εκπαίδευσης:** Η εκπαίδευση των μοντέλων μηχανικής μάθησης σε υψηλών διαστάσεων χώρους χαρακτηριστικών μπορεί να είναι υπολογιστικά δαπανηρή, ειδικά για τους αλγόριθμους που είναι ευαίσθητοι στην κατάρρα της διαστασιακότητας. Με τη μείωση του αριθμού των χαρακτηριστικών με `max_features`, μπορούμε να επιταχύνουμε τη διαδικασία εκπαίδευσης, οδηγώντας σε ταχύτερη σύγκλιση μοντέλων και μικρότερους χρόνους εκπαίδευσης.

Συμπερασματικά, η χρήση της παραμέτρου `max_features` στην διανυσματοποίηση κειμένου με τον TF-IDF μας επιτρέπει να ελέγξουμε το μέγεθος και την πολυπλοκότητα του χώρου χαρακτηριστικών, οδηγώντας σε πιο αποτελεσματικά και δυνητικά καλύτερα μοντέλα σε εργασίες NLP.

6.5 Εκπαίδευση ταξινομητών

Αυτό το στάδιο αποτελείται από τρία ξεχωριστά επίπεδα μοντελοποίησης και δύο προσεγγίσεις για τον υπολογισμό της μέτρησης αξιολόγησης. Συγκεκριμένα, λαμβάνοντας υπόψη ότι υπάρχει μια ανισορροπημένη κατανομή περιπτώσεων σε κάθε κατηγορία, όπως εξηγήθηκε προηγουμένως, οι αλγόριθμοι εκπαιδεύτηκαν χρησιμοποιώντας δύο μεθοδολογίες για την αξιολόγηση της απόδοσης:

1. Μεθοδολογία `train-test split`
2. Μεθοδολογία διασταυρούμενης επικύρωσης (`Cross validation`)

εντός των ακόλουθων τριών κατηγοριών:

1. Χωρίς τη χρήση οποιασδήποτε τεχνικής υπερδειγματοληψίας.
2. Χρησιμοποιώντας τη μέθοδο υπερδειγματοληψίας `SMOTE`.
3. Χρησιμοποιώντας τη μέθοδο υπερδειγματοληψίας `RandomOverSampler`.

6.5.1 Η μέθοδος `Train-test split`

Η μέθοδος `Train-test split` είναι μια διαδεδομένη τεχνική μηχανικής μάθησης που χρησιμοποιείται για την αξιολόγηση της απόδοσης και των δυνατοτήτων γενίκευσης ενός μοντέλου. Η διαδικασία περιλαμβάνει τη διαίρεση των δεδομένων σε δύο ξεχωριστά υποσύνολα: ένα σύνολο εκπαίδευσης το οποίο χρησιμοποιείται για να διδάξει το μοντέλο μηχανικής μάθησης και ένα σύνολο δοκιμών που προορίζεται για την αξιολόγηση της απόδοσης του εκπαιδευμένου μοντέλου σε άγνωστα δεδομένα. Εκπαιδεύοντας το μοντέλο σε ένα υποσύνολο δεδομένων και αξιολογώντας την απόδοσή του σε ένα ξεχωριστό, μη συνδεδεμένο υποσύνολο, αποκτάται μια αμερόληπτη εκτίμηση της απόδοσης του μοντέλου σε "αόρατα" δεδομένα σε πραγματικές καταστάσεις. Αυτή η μεθοδολογία χρησιμεύει ως εγγύηση κατά του `overfitting`, μια κατάσταση στην οποία το μοντέλο απομνημονεύει υπερβολικά τα δεδομένα κατάρτισης αλλά αγωνίζεται να εφαρμόσει τις γνώσεις του σε νέες περιπτώσεις. Προστατεύει επίσης από το `underfitting`, το οποίο συμβαίνει όταν το μοντέλο αποτυγχάνει να συλλάβει με ακρίβεια τα βασικά πρότυπα που υπάρχουν στα δεδομένα. Στη συγκεκριμένη περίπτωση, το σύνολο εκπαίδευσης έλαβε το μεγαλύτερο ποσοστό των δεδομένων, το 80%, ενώ στο σύνολο δοκιμών αποδόθηκε το υπόλοιπο 20%. Αυτή η διαίρεση επιτυγχάνει μια αρμονική ισορροπία, εξασφαλίζοντας έναν άφθονο αριθμό δεδομένων για την κατάρτιση του μοντέλου αποτελεσματικά, ενώ παράλληλα παρέχει μια ικανοποιητική ποσότητα δεδομένων, για την αξιολόγηση της απόδοσής του. Χρησιμοποιώντας ένα σύνολο εκπαίδευσης 80%, το μοντέλο είναι σε θέση να έχει πρόσβαση σε μια σημαντική ποσότητα δεδομένων για τους σκοπούς της μάθησης. Αυτό επιτρέπει στο μοντέλο να συλλάβει αποτελεσματικά τα θεμελιώδη μοτίβα και συσχετισμούς που υπάρχουν μέσα στο σύνολο δεδομένων. Ταυτόχρονα, το σύνολο δοκιμών 20%

συνεισφέρει ένα σημαντικό ποσό δεδομένων για αυστηρή αξιολόγηση, εξασφαλίζοντας ότι η απόδοση του μοντέλου μετράται με ακρίβεια σε περιπτώσεις που δεν έχουν συναντηθεί προηγουμένως. Επομένως, ο στόχος είναι η εκπαίδευση του μοντέλου μηχανικής μάθησης σε ένα υποσύνολο δεδομένων που αντιπροσωπεύει με ακρίβεια το σύνολο, ενώ ταυτόχρονα διατίθεται ένα ξεχωριστό σύνολο δοκιμής για λεπτομερή αξιολόγηση.

Συνοψίζοντας, η μέθοδος διαίρεσης δοκιμών επιτρέπει την αμερόληπτη αξιολόγηση των μοντέλων μηχανικής μάθησης διαιρώντας το σύνολο δεδομένων σε εκπαιδευτικά και δοκιμαστικά σύνολα, με κατανομή 80% και 20% αντίστοιχα. Αυτή η κατανομή επιτυγχάνει μια αρμονική ισορροπία μεταξύ αποτελεσματικής κατάρτισης μοντέλων και ανθεκτικής αξιολόγησης, εξασφαλίζοντας μια ακριβή αξιολόγηση της απόδοσης του μοντέλου σε άγνωστα δεδομένα.

6.5.2 Η μέθοδος διασταυρούμενης επικύρωσης K-fold

Η διασταυρούμενη επικύρωση K-fold είναι μια διαδεδομένη μέθοδος στη μηχανική μάθηση για την αξιολόγηση της απόδοσης των μοντέλων. Η διαδικασία περιλαμβάνει τη διαίρεση του συνόλου δεδομένων σε k υποσύνολα του ίδιου μεγέθους, γνωστά ως folds. Το μοντέλο υποβάλλεται σε k -fold cross-validation, όπου εκπαιδεύεται k φορές. Σε κάθε επανάληψη, τα $k-1$ folds χρησιμοποιούνται ως τα δεδομένα εκπαίδευσης και τα υπόλοιπα χρησιμοποιούνται ως δεδομένα δοκιμών. Η λειτουργία επαναλαμβάνεται k φορές, με κάθε fold να χρησιμοποιείται ως δοκιμαστικό σύνολο ακριβώς μία φορά. Η τελική στατιστική απόδοσης υπολογίζεται λαμβάνοντας τον μέσο όρο των αποτελεσμάτων που ελήφθησαν από όλα τα k folds. Ο λόγος για τη χρήση του k -fold cross-validation είναι η ικανότητά του να προσφέρει μια πιο ανθεκτική αξιολόγηση της απόδοσης του μοντέλου σε σύγκριση με την μέθοδο Train-test split, καθώς η διασταυρούμενη επικύρωση K-fold επιτρέπει στο μοντέλο να εκπαιδευτεί και να αξιολογηθεί σε διάφορους συνδυασμούς περιπτώσεων δεδομένων κάνοντας κύκλους μέσα από διαφορετικά υποσύνολα δεδομένων τόσο ως σύνολα εκπαίδευσης όσο και ως σύνολα δοκιμών. Αυτό συμβάλλει στη μείωση των διακυμάνσεων στην εκτίμηση της απόδοσης και προσφέρει μια πιο αξιόπιστη αξιολόγηση της ικανότητας του μοντέλου να γενικεύσει. Η επιλογή της τιμής του k στην k -fold διασταυρούμενη επικύρωση καθορίζεται συνήθως λαμβάνοντας υπόψη το μέγεθος του συνόλου δεδομένων και τις διαθέσιμες υπολογιστικές δυνατότητες. Οι τυπικές επιλογές για την τιμή του k είναι 5, 10, ή ακόμη μεγαλύτερες τιμές για μικρότερα σύνολα δεδομένων. Η αύξηση της τιμής του k γενικά οδηγεί σε μια πιο ακριβή εκτίμηση της απόδοσης του μοντέλου, αλλά επίσης ενισχύει το υπολογιστικό βάρος της διαδικασίας. Από την άλλη πλευρά, η χρήση μιας μικρότερης τιμής του k μπορεί να είναι πιο αποτελεσματική όσον αφορά τον υπολογισμό, αλλά θα μπορούσε να οδηγήσει σε μεγαλύτερες διακυμάνσεις στην εκτίμηση της απόδοσης. Μέσω της χρήσης του k -fold cross-validation, ο στόχος είναι η απόκτηση μιας εμπειριστικώς και αξιόπιστης αξιολόγησης της απόδοσης του μοντέλου, εγγυώντας την ικανότητά του να εφαρμόζεται αποτελεσματικά σε νέα δεδομένα. Αυτή η μεθοδολογία προσφέρει μια ολοκληρωμένη αξιολόγηση της αποτελεσματικότητας του μοντέλου και βοηθά στη λήψη τεκμηριωμένων αποφάσεων σχετικά με την επιλογή μοντέλων και τη βελτιστοποίηση των παραμέτρων.

Συνοψίζοντας, η k -fold cross-validation είναι μια ισχυρή τεχνική για την αξιολόγηση της απόδοσης του μοντέλου, προσφέροντας μια πιο αξιόπιστη εκτίμηση των δυνατοτήτων γενίκευσης σε σύγκριση με μια ενιαία διαίρεση δοκιμής τρένου. Βελτιστοποιεί τη χρήση των προσβάσιμων δεδομένων, ενισχύει τη σταθερότητα των εκτιμήσεων απόδοσης και παρέχει πολύτιμες πληροφορίες σχετικά με την ανθεκτικότητα του μοντέλου σε διάφορα υποσύνολα δεδομένων.

6.6 Ταξινομητές

6.6.1 Βελτιστοποίηση των παραμέτρων με τη χρήση του GridSearchCV και του RandomizedSearchCV

Η προσαρμογή των παραμέτρων είναι μια ουσιαστική διαδικασία στη βελτίωση των μοντέλων μηχανικής μάθησης για να βελτιώσουν την απόδοσή τους και την ικανότητά τους να γενικεύουν. Περιλαμβάνει μια μεθοδολογική εξερεύνηση για τη βέλτιστη διαμόρφωση των υπερπαραμέτρων του μοντέλου, οι οποίοι είναι ρυθμιζόμενοι παράγοντες που διέπουν τη διαδικασία εκμάθησης και επηρεάζουν τη συμπεριφορά του. Οι GridSearchCV και RandomizedSearchCV χρησιμοποιούνται συχνά για την αυτοματοποίηση της διαδικασίας προσαρμογής παραμέτρων στη μηχανική μάθηση.

Το GridSearchCV, επίσης γνωστό ως Grid Search Cross-Validation, είναι μια ολοκληρωμένη μέθοδος αναζήτησης που αξιολογεί την απόδοση του μοντέλου δοκιμάζοντάς το σε ένα προκαθορισμένο πλέγμα τιμών υπερπαραμέτρων. Η διαδικασία εξετάζει εκτενώς όλους τους πιθανούς συνδυασμούς υπερπαραμέτρων μέσω της διασταυρούμενης επικύρωσης και προσδιορίζει τον συνδυασμό που επιτυγχάνει την υψηλότερη απόδοση σύμφωνα με ένα καθορισμένο κριτήριο αξιολόγησης. Το GridSearchCV κάνει μια εκτενή αναζήτηση όλων των εφικτών συνδυασμών των υπερπαραμέτρων, συμβάλλοντας σε υψηλές απαιτήσεις υπολογισμού, ιδιαίτερα για τα μοντέλα με πολυάριθμες υπερπαραμέτρους ή με ένα ευρύ φάσμα τιμών.

Αντίθετα, το RandomizedSearchCV είναι μια τεχνική αναζήτησης που επιλέγει τυχαία τιμές υπερπαραμέτρων πάνω από ένα καθορισμένο διάστημα παραμέτρων και αξιολογεί την απόδοσή τους χρησιμοποιώντας διασταυρούμενη επικύρωση. Το RandomizedSearchCV, σε αντίθεση με το GridSearch CV, χρησιμοποιεί μια τυχαία επιλογή ενός καθορισμένου αριθμού συνδυασμών υπερπαραμέτρων, με αποτέλεσμα τη βελτίωση της υπολογιστικής αποδοτικότητας, ειδικά για διαστήματα υπερπαραμέτρων με υψηλές διαστάσεις. Παράλληλα, διευκολύνει μια εμπειριστατωμένη έρευνα του υπερπαραμετρικού χώρου επιλέγοντας τυχαίες υπερπαραμέτρους, αποφεύγοντας ταυτόχρονα το υπολογιστικό φορτίο της εξαντλητικής αναζήτησης.

Ο σκοπός της προσαρμογής παραμέτρων χρησιμοποιώντας το GridSearchCV και το RandomizedSearchCV είναι να προσδιοριστεί ο ιδανικός συνδυασμός υπερπαραμέτρων που μεγιστοποιεί την απόδοση του μοντέλου σε νέα δεδομένα. Μέσω μιας μεθοδολογικής εξερεύνησης του υπερπαραμετρικού χώρου και της χρήσης της διασταυρούμενης επικύρωσης για την αξιολόγηση της απόδοσης, αυτές οι τεχνικές μας επιτρέπουν να προσαρμόσουμε με ακρίβεια τη συμπεριφορά του μοντέλου και να ενισχύσουμε την ικανότητά του να γενικεύεται σε νέες καταστάσεις. Αυτή η προσέγγιση έχει ως στόχο τη μεγιστοποίηση της απόδοσης του μοντέλου, τη βελτίωση της προβλεπόμενης ακρίβειας και, τελικά, την ενίσχυση της χρησιμότητας των μοντέλων σε πραγματικές εφαρμογές. Επιπρόσθετα, οι GridSearchCV και RandomizedSearchCV παρέχουν πολυάριθμα πλεονεκτήματα σε σύγκριση με τη χειροκίνητη προσαρμογή παραμέτρων ή ad hoc μεθόδους αναζήτησης. Αρχικά, προσφέρουν μια μεθοδική και αυτοματοποιημένη τεχνική για τη ρύθμιση των παραμέτρων, ελαχιστοποιώντας έτσι την ανάγκη για χειροκίνητα πειράματα. Επιπλέον, παρέχουν μια διεξοδική έρευνα του υπερπαραμετρικού χώρου, εξασφαλίζοντας ότι κανένας δυνητικά επιτυχημένος συνδυασμός δεν αγνοείται. Επιπλέον, ενσωματώνουν τη διασταυρούμενη επικύρωση, μια τεχνική που βοηθά στην επίτευξη αμερόληπτων αξιολογήσεων της απόδοσης του μοντέλου και μειώνει την πιθανότητα υπερβολικής συμμόρφωσης με τα δεδομένα κατάρτισης. Τέλος, ενισχύουν την

ικανότητα αναπαραγωγής και διαφάνειας της διαδικασίας προσαρμογής των παραμέτρων, επιτρέποντας την εύκολη αντιγραφή και σύγκριση των αποτελεσμάτων σε διάφορες δοκιμές.

6.6.2 Ταξινομητής Δέντρων Απόφασης (Decision Tree Classifier)

Τα δέντρα απόφασης είναι ανθεκτικές και προσαρμόσιμες μέθοδοι επιβλεπόμενης μηχανικής μάθησης που χρησιμοποιούνται τόσο για εργασίες ταξινόμησης όσο και παλινδρόμησης. Αυτά τα συστήματα λειτουργούν διαιρώντας τον χώρο χαρακτηριστικών σε περιοχές, με βάση τις τιμές χαρακτηριστικών εισόδου, με ιεραρχικό τρόπο χρησιμοποιώντας αναδρομική διαίρεση. Σε κάθε κόμβο στο δέντρο, γίνεται ένας προσδιορισμός με βάση την αξία ενός χαρακτηριστικού, με αποτέλεσμα τη διαίρεση των δεδομένων σε μικρότερες ομάδες. Αυτή η διαδικασία συνεχίζεται μέχρι να ικανοποιηθεί μια συνθήκη τερματισμού, όπως η επίτευξη του υψηλότερου επιπέδου βάθους δέντρου ή όταν περισσότεροι διαχωρισμοί δεν αυξάνουν σημαντικά την αποτελεσματικότητα του μοντέλου.

Τα δέντρα αποφάσεων είναι πολύτιμα λόγω της ερμηνευσιμότητάς τους και της διαφάνειάς τους, καθώς προσφέρουν μια συνοπτική περιγραφή της διαδικασίας λήψης αποφάσεως. Κάθε διαίρεση στο δέντρο αντιστοιχεί σε μια επιλογή που γίνεται σύμφωνα με ένα συγκεκριμένο χαρακτηριστικό, το οποίο απλοποιεί την κατανόηση της λογικής πίσω από τις προβλέψεις του μοντέλου. Επιπλέον, τα δέντρα αποφάσεων έχουν την ικανότητα να επεξεργάζονται τόσο αριθμητικά όσο και κατηγορικά δεδομένα, καθιστώντας τα κατάλληλα για ένα ευρύ φάσμα εφαρμογών.

Ωστόσο, τα δέντρα αποφάσεων είναι επιρρεπή σε υπερπροσαρμογή (overfitting), ειδικά όταν επιτρέπεται να φθάσουν σε υπερβολικό βάθος ή όταν εκπαιδεύονται σε δεδομένα με υψηλό επίπεδο θορύβου. Για την αντιμετώπιση αυτού του προβλήματος, οι στρατηγικές κοπής (pruning) μπορούν να χρησιμοποιηθούν για να περιορίσουν την επέκταση του δέντρου και να ενισχύσουν την ικανότητά του να γενικεύει. Παρά αυτόν τον περιορισμό, τα δέντρα αποφάσεων λειτουργούν ως τα θεμελιώδη συστατικά για πιο προηγμένες τεχνικές συνόλων.

6.6.3 Ταξινομητής Random Forest

Το Random Forest είναι μια τεχνική μάθησης που ενισχύει την προβλεπόμενη ακρίβεια και ανθεκτικότητα συνδυάζοντας πολλά δέντρα αποφάσεων. Η διαδικασία περιλαμβάνει τη δημιουργία αρκετών δέντρων αποφάσεων στη φάση της εκπαίδευσης. Κάθε δέντρο κατασκευάζεται χρησιμοποιώντας ένα τυχαίο υποσύνολο των δεδομένων εκπαίδευσης και μια τυχαία επιλογή των χαρακτηριστικών, ενώ οι προβλέψεις που παράγονται από κάθε μεμονωμένο δέντρο συνδυάζονται στη συνέχεια μέσω της διαδικασίας της μέτρησης ή της ψηφοφορίας για να δώσει την τελική πρόβλεψη.

Το Random Forest έχει πολλά πλεονεκτήματα σε σύγκριση με τα μεμονωμένα δέντρα αποφάσεων, καθώς ενισχύει τις επιδόσεις γενίκευσης και μετριάξει τον κίνδυνο υπερπροσαρμογής ενοποιώντας προβλέψεις από πολλά δέντρα. Επιπλέον, η αυθαίρετη επιλογή των χαρακτηριστικών σε κάθε κόμβο βοηθά στη μείωση της συσχέτισης μεταξύ των δέντρων, αυξάνοντας έτσι την ανθεκτικότητα του συστήματος ενάντια σε θορυβώδη ή άσχετα χαρακτηριστικά. Ακόμη, ο αλγόριθμος Random Forest είναι εξαιρετικά ευέλικτος και μπορεί να χρησιμοποιηθεί για διάφορες εφαρμογές, όπως η ταξινόμηση, η παλινδρόμηση και η ανίχνευση ακραίων τιμών. Επιπλέον, το Random Forest απαιτεί ελάχιστη ρύθμιση των υπερπαραμέτρων και ουσιαστικά δεν επηρεάζεται από την επιλογή των παραμέτρων, καθιστώντας το βολικό για άμεση χρήση.

6.6.4 Πολυωνυμική Λογιστική Παλινδρόμηση (Multinomial Logistic Regression)

Η Πολυωνυμική Λογιστική Παλινδρόμηση είναι ένας τύπος εποπτευόμενης μεθόδου μάθησης που έχει σχεδιαστεί ειδικά για την επίλυση προβλημάτων ταξινόμησης πολλαπλών τάξεων. Διευρύνει τις δυνατότητες της δυαδικής λογιστικής παλινδρόμησης για να συμπεριλάβει πολλαπλές τάξεις, χρησιμοποιώντας τη λειτουργία softmax για να περιγράψει την κατανομή πιθανότητας σε όλες τις δυνητικές τάξεις. Σε κάθε κλάση αποδίδονται ξεχωριστά βάρη, και η πιθανότητα της κάθε κατηγορίας υπολογίζεται με την εφαρμογή της λειτουργίας softmax στον γραμμικό συνδυασμό των χαρακτηριστικών εισόδου και των αντίστοιχων βαρών τους. Επιπλέον, η Πολυωνυμική Λογιστική Παλινδρόμηση παρέχει διάφορα οφέλη, όπως απλότητα, ερμηνευσιμότητα και την ικανότητα να αντιπροσωπεύει μη γραμμικές συνδέσεις μέσω μετασχηματισμών χαρακτηριστικών. Ακόμη, έχει την ικανότητα να κάνει πιθανολογικές προβλέψεις, οι οποίες επιτρέπουν τον υπολογισμό της αβεβαιότητας και την αξιολόγηση του μοντέλου. Παρ' όλα αυτά, είναι επιρρεπής σε υπερπροσαρμογή όταν ο αριθμός των χαρακτηριστικών είναι σημαντικά μεγαλύτερος από τον αριθμό των περιπτώσεων, και μπορεί να επιδείξει υποδεέστερη απόδοση σε σύνολα δεδομένων με ουσιαστική μη γραμμική απόδοση.

6.6.5 Στοχαστική Κάθοδος Κλίσης (Stochastic Gradient Descent)

Η Στοχαστική Κάθοδος Κλίσης (SGD), είναι μια ευρέως χρησιμοποιούμενη τεχνική βελτιστοποίησης για την εκπαίδευση μοντέλων μηχανικής μάθησης, ειδικά σε σενάρια που περιλαμβάνουν μάθηση μεγάλης κλίμακας. Η λειτουργία εκτελείται με συνεχή ενημέρωση των παραμέτρων του μοντέλου χρησιμοποιώντας τις κλίσεις της συνάρτησης απώλειας σε σχέση με τα δεδομένα εκπαίδευσης. Το stochastic gradient descent, σε αντίθεση με την batch gradient descent, υπολογίζει τις κλίσεις χρησιμοποιώντας ένα ενιαίο τυχαίο σημείο δεδομένων ή ένα μικρό υποσύνολο δεδομένων. Αυτή η προσέγγιση ενισχύει την υπολογιστική αποδοτικότητα και είναι κατάλληλη για τη διαχείριση τεράστιων συνόλων δεδομένων. Επιπλέον, παρέχει διάφορα πλεονεκτήματα, όπως η ικανότητα να χειρίζεται μεγάλα σύνολα δεδομένων, αποτελεσματικό υπολογισμό και τη δυνατότητα επίλυσης προβλημάτων βελτιστοποίησης που είναι μη κυρτά και μη ομαλά. Επιπλέον, το SGD είναι ιδιαίτερα κατάλληλο για καταστάσεις που αφορούν την ηλεκτρονική μάθηση, όπου τα δεδομένα λαμβάνονται με διαδοχικό τρόπο και το μοντέλο απαιτεί συνεχείς ενημερώσεις. Ωστόσο, μπορεί να συναντήσει ουσιαστική μεταβλητότητα στις ενημερώσεις παραμέτρων, με αποτέλεσμα την ασταθή σύγκλιση και βραδύτερα ποσοστά σύγκλισης σε σύγκριση με την παρτίδα batch gradient descent.

6.6.6 Γραμμικός Ταξινομητής Διανυσμάτων Υποστήριξης (Linear Support Vector Classifier)

Ο Γραμμικός Ταξινομητής Διανυσμάτων Υποστήριξης (Linear SVC) είναι μια εποπτευόμενη προσέγγιση μάθησης που χρησιμοποιείται για προβλήματα ταξινόμησης, ειδικά όταν εργάζεται με δεδομένα που μπορούν να διαχωριστούν γραμμικά. Λειτουργεί προσδιορίζοντας το υπερπλάνο που διαιρεί πιο αποτελεσματικά τις τάξεις στο χώρο χαρακτηριστικών. Το Linear SVC διαφέρει από το κλασικό SVC χρησιμοποιώντας έναν γραμμικό πυρήνα αντί ενός μη γραμμικού για να μετατρέψει τα δεδομένα σε έναν χώρο υψηλότερων διαστάσεων. Αυτή η επιλογή του πυρήνα καθιστά το Linear SVC υπολογιστικά αποδοτικό και κατάλληλο για τη διαχείριση συνόλων δεδομένων μεγάλης κλίμακας. Ο στόχος του Linear SVC είναι να βελτιστοποιήσει το περιθώριο μεταξύ των φορέων υποστήριξης διαφόρων κατηγοριών, γεγονός που ενισχύει την ικανότητα γενίκευσης και αυξάνει την αντίσταση στον θόρυβο. Ο αλγόριθμος λειτουργεί με την

επίλυση ενός προβλήματος διπλής βελτιστοποίησης για να εντοπίσει το υπερπλάνο που επιτυγχάνει το υψηλότερο περιθώριο μεταξύ των τάξεων. Όταν ασχολείται με καταστάσεις όπου οι τάξεις δεν μπορούν εύκολα να διαχωριστούν με μια ευθεία γραμμή, το Linear SVC χρησιμοποιεί μια τεχνική μαλακού περιθωρίου που επιτρέπει έναν ορισμένο αριθμό λανθασμένων ταξινομήσεων με στόχο τη μεγιστοποίηση της απόστασης μεταξύ των τάξεων. Το Linear SVC παρέχει διάφορα πλεονεκτήματα, όπως η ικανότητα να χειρίζεται μεγάλα σύνολα δεδομένων, αποτελεσματική απόδοση υπολογισμού και αποτελεσματικότητα σε χώρους με μεγάλο αριθμό διαστάσεων. Επιπλέον, έχει όρια λήψης αποφάσεων που μπορούν να κατανοηθούν εύκολα, καθιστώντας το κατάλληλο για εφαρμογές που δίνουν προτεραιότητα στη διαφάνεια του μοντέλου. Τέλος, μπορεί να παρουσιάσει υποβέλτιστη απόδοση όταν εφαρμόζεται σε σύνολα δεδομένων που περιέχουν περίπλοκες μη γραμμικές σχέσεις, καθώς εξαρτάται σε μεγάλο βαθμό από τα γραμμικά όρια αποφάσεων.

7. Αποτελέσματα

7.1 Αποτελέσματα της ταξινόμησης των αναφορών σφαλμάτων Jira με χρήση NLP

Σε αυτό το κεφάλαιο, παρουσιάζονται τα αποτελέσματα των μοντέλων ταξινόμησης αναφορών σφαλμάτων Jira. Η απόδοση των μοντέλων αξιολογείται με τη χρήση της μετρικής της ακρίβειας τόσο στο σύνολο εκπαίδευσης όσο και στο σύνολο τεστ, καθώς χρησιμεύει ως πολύτιμος δείκτης απόδοσης για την ανίχνευση υπερπροσαρμογής (overfitting) και την αξιολόγηση της επίδοσης γενίκευσης σε ταξινομητές. Συγκρίνοντας την απόδοση των ταξινομητών και στα δύο σύνολα, προκύπτουν πληροφορίες για την ικανότητά του να μαθαίνει από τα δεδομένα εκπαίδευσης και να γενικεύει σε νέες περιπτώσεις, ενημερώνοντας τελικά τις αποφάσεις σχετικά με την ανάπτυξη και τη βελτίωση του μοντέλου.

7.1.1 Ανίχνευση υπερπροσαρμογής

Ο εντοπισμός του overfitting είναι απαραίτητος για να διασφαλιστεί ότι ένας ταξινομητής γενικεύεται αποτελεσματικά σε καινούργια δεδομένα. Αναλυτικότερα, η σύγκριση της ακρίβειας που αποκτήθηκε στο σετ εκπαίδευσης με την ακρίβεια που επιτεύχθηκε στο δοκιμαστικό είναι ένα βασικό σημάδι overfitting. Εάν η ακρίβεια στο σετ εκπαίδευσης είναι σημαντικά μεγαλύτερη από την ακρίβεια στο σύνολο δοκιμών, αυτό υποδηλώνει ότι το μοντέλο ενδέχεται να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης. Αυτή η ανισότητα υποδηλώνει ότι το μοντέλο έχει δεσμεύσει τα δεδομένα εκπαίδευσης στη μνήμη αντί να κατανοεί τα υποκείμενα μοτίβα, με αποτέλεσμα την ανεπαρκή εξάπλωσή του σε άγνωστα δεδομένα. Από την άλλη πλευρά, όταν η ακρίβεια και στα δύο σύνολα είναι παρόμοια, αυτό δείχνει ότι το μοντέλο έχει αποκτήσει αποτελεσματικά τα θεμελιώδη πρότυπα στα δεδομένα χωρίς να βιώνει υπερπροσαρμογή, επιδεικνύοντας καλή γενίκευση.

7.2 Περίληψη μεθόδου εκπαίδευσης των ταξινομητών

Παρακάτω παρουσιάζεται μια περίληψη των παραμέτρων που έχουν επιλεγεί για κάθε ταξινομητή, καθώς και οι δύο μέθοδοι για τον υπολογισμό των μέτρων αξιολόγησης.

Πίνακας 3: Παράμετροι που χρησιμοποιήθηκαν στον κάθε ταξινομητή

Ταξινομητές	Παράμετροι	Μέθοδος Train-Test Split	Μέθοδος Cross-Validation
Random Forest	Default	80% - 20%	k=5-fold
Random Forest GridSearchCV	max_depth=13, n_estimators=150	80% - 20%	k=5-fold
Random Forest RandomizedSearchCV	max_depth=9, n_estimators=150	80% - 20%	k=5-fold
Random Forest RandomOverSampler	Default	80% - 20%	k=5-fold
Random Forest SMOTE	Default	80% - 20%	k=5-fold
Decision Tree	Default	80% - 20%	k=5-fold
Decision Tree GridSearchCV	max_depth=9	80% - 20%	k=5-fold
Decision Tree RandomizedSearchCV	max_depth=9	80% - 20%	k=5-fold
Decision Tree RandomOverSampler	Default	80% - 20%	k=5-fold
Decision Tree SMOTE	Default	80% - 20%	k=5-fold
Linear SVC	Default	80% - 20%	k=5-fold
Stochastic Gradient Descent	loss="hinge", penalty="l2"	80% - 20%	k=5-fold
Multinomial Logistic Regression	Default	80% - 20%	k=5-fold
Multinomial Logistic Regression GridSearchCV	C=0.01, penalty="l2"	80% - 20%	k=5-fold

7.3 Παρουσίαση αποτελεσμάτων

Στους παρακάτω πίνακες παρουσιάζονται τα αποτελέσματα των ταξινομητών, οι οποίοι αξιολογούνται με τη χρήση της μετρικής της ακρίβειας στο σύνολο εκπαίδευσης και στο σύνολο δοκιμών, για διάφορα max_features του του αλγορίθμου διανυσματοποίησης κειμένου TF-IDF Vectorizer, όπως αναφέρθηκε και παραπάνω.

Πίνακας 4: Αποτελέσματα ταξινομητών για max_features = 100

Ταξινομητές	Ακρίβεια στο σύνολο εκπαίδευσης	Ακρίβεια στο σύνολο δοκιμών	Παράμετρος max_features του TF-IDF
Random Forest	98.375%	63.875%	100
Random Forest GridSearchCV	68.84375%	64.75%	100
Random Forest RandomizedSearchCV	80.0625%	65.75%	100
Random Forest RandomOverSampler	97.65625%	65.0%	100
Random Forest SMOTE	98.1875%	60.1249%	100
Decision Tree	98.3125%	57.375%	100
Decision Tree GridSearchCV	65.0625%	65.0%	100
Decision Tree RandomizedSearchCV	65.0625%	65.0%	100
Decision Tree RandomOverSampler	97.65625%	52.875%	100
Decision Tree SMOTE	97.65625%	53.625%	100
Linear SVC	67.21875%	64.875%	100
Stochastic Gradient Descent	66.46875%	66.0%	100
Multinomial Logistic Regression	67.3125%	64.375%	100
Multinomial Logistic Regression GridSearchCV	65.625%	64.625%	100

Σύμφωνα με τον παραπάνω πίνακα, για το Random Forest η ακρίβεια στο σύνολο εκπαίδευσης είναι πολύ υψηλή (98,375%), υποδεικνύοντας ότι το μοντέλο ταιριάζει πολύ καλά στα δεδομένα εκπαίδευσης. Ωστόσο, η ακρίβεια στο σύνολο δοκιμής είναι συγκριτικά χαμηλότερη (63,875%), γεγονός που υποδηλώνει ότι το μοντέλο μπορεί να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης. Αναφορικά με το Random Forest GridSearchCV, παρατηρείται ότι η ακρίβεια στο σύνολο εκπαίδευσης είναι χαμηλότερη σε σύγκριση με το μη βελτιστοποιημένο μοντέλο Random Forest (68,84375%), ενώ η ακρίβεια στο σύνολο δοκιμής είναι ελαφρώς καλύτερη από το μη βελτιστοποιημένο μοντέλο (64,75%). Παράλληλα, με τη χρήση του Random Forest RandomizedSearchCV, η ακρίβεια στο σύνολο εκπαίδευσης είναι χαμηλότερη σε σύγκριση με το μη βελτιστοποιημένο μοντέλο Random Forest (80,0625%) και η ακρίβεια στο σύνολο δοκιμής είναι ελαφρώς καλύτερη από το μη βελτιστοποιημένο μοντέλο (65,75%). Ακόμη, χρησιμοποιώντας το Random Forest RandomOverSampler φαίνεται πως η ακρίβεια στο σύνολο εκπαίδευσης είναι πολύ υψηλή (97,65625%), υποδεικνύοντας καλή προσαρμογή στα δεδομένα υπερδειγματοληψίας, με την ακρίβεια στο σύνολο δοκιμής να είναι λογική (65,0%), υποδηλώνοντας ότι το μοντέλο γενικεύει αρκετά καλά. Στη συνέχεια, με τη χρήση του Random Forest SMOTE Oversampling, η ακρίβεια στο σύνολο εκπαίδευσης είναι πολύ υψηλή (98,1875%), παρόμοια με το Random Forest RandomOverSampler. Ωστόσο, η ακρίβεια στο σύνολο δοκιμής είναι σχετικά χαμηλή (60,1249%), υποδεικνύοντας πιθανά προβλήματα με τη γενίκευση. Επιπρόσθετα, για το Decision Tree η ακρίβεια στο σύνολο εκπαίδευσης είναι πολύ υψηλή (98,3125%), παρόμοια με το Random Forest, ενώ η ακρίβεια στο σύνολο δοκιμής είναι σχετικά χαμηλή (57,375%), υποδεικνύοντας πιθανή υπερπροσαρμογή. Αναφορικά με τα Decision Tree GridSearchCV και Decision Tree RandomizedSearchCV, η ακρίβεια στο σύνολο εκπαίδευσης είναι σχετικά χαμηλή (65,0625%), υποδηλώνοντας ότι το μοντέλο ενδέχεται να μην προσαρμόζεται καλά στα δεδομένα και η ακρίβεια στο σύνολο δοκιμής είναι παρόμοια με το σύνολο εκπαίδευσης, υποδεικνύοντας ισορροπημένη απόδοση. Παράλληλα, για τα Decision Tree RandomOverSampler και Decision Tree SMOTE, η ακρίβεια στο σύνολο εκπαίδευσης είναι υψηλή (97,65625%), παρόμοια με τα μοντέλα Random Forest που χρησιμοποιούν τεχνικές υπερδειγματοληψίας, ενώ η ακρίβεια στο σύνολο δοκιμής είναι σχετικά χαμηλή (52,875% για το RandomOverSampler, 53,625% για το SMOTE), υποδεικνύοντας πιθανά προβλήματα με τη γενίκευση. Αναφορικά με τους ταξινομητές Linear SVC, Stochastic Gradient Descent, Multinomial Logistic Regression και Multinomial Logistic Regression GridSearchCV, είναι φανερό πως παρουσιάζουν διαφορετικά επίπεδα απόδοσης όσον αφορά την ακρίβεια τόσο στα σύνολα εκπαίδευσης όσο και στα σύνολα δοκιμής, με ορισμένα μοντέλα να παρουσιάζουν πιθανή υπερπροσαρμογή (π.χ. Linear SVC) και άλλα να παρουσιάζουν ισορροπημένη απόδοση.

Πίνακας 5: Αποτελέσματα ταξινομητών για max_features = 200

Ταξινομητές	Ακρίβεια στο σύνολο εκπαίδευσης	Ακρίβεια στο σύνολο δοκιμών	Παράμετρος max_features του TF-IDF
Random Forest	99.375%	66.375%	200
Random Forest GridSearchCV	68.4375%	65.125%	200
Random Forest RandomizedSearchCV	74.8125%	65.25%	200
Random Forest RandomOverSampler	99.15625%	66.625%	200
Random Forest SMOTE	99.15625%	63.5%	200
Decision Tree	99.375%	55.5%	200
Decision Tree GridSearchCV	65.0625%	65.0%	200
Decision Tree RandomizedSearchCV	65.0625%	65.0%	200
Decision Tree RandomOverSampler	99.15625%	52.375%	200
Decision Tree SMOTE	99.15625%	50.249%	200
Linear SVC	70.0%	65.375%	200
Stochastic Gradient Descent	69.875%	65.125%	200
Multinomial Logistic Regression	65.125%	65.0%	200
Multinomial Logistic Regression GridSearchCV	65.6875%	65.0%	200

Αναλύοντας τα παραπάνω αποτελέσματα, ο ταξινομητής Random Forest παρέχει πολύ υψηλή ακρίβεια στο σύνολο εκπαίδευσης (99,375%), υποδεικνύοντας ότι ήταν σε θέση να μάθει αποτελεσματικά από τα δεδομένα εκπαίδευσης. Ωστόσο, η ακρίβεια στο σύνολο δοκιμής έπεσε στο 66,375%, γεγονός που υποδηλώνει ότι το μοντέλο μπορεί να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης και να μην γενικεύει καλά σε άορατα δεδομένα. Η χρήση του Random Forest με GridSearchCV επιτυγχάνει σημαντική μείωση της ακρίβειας στο σύνολο εκπαίδευσης σε σύγκριση με το τυπικό Random Forest, υποδεικνύοντας ότι ο συντονισμός των υπερπαραμέτρων ενδέχεται να περιορίσει την ικανότητα μάθησης του μοντέλου, ενώ η ακρίβεια στο σύνολο δοκιμής παρέμεινε σχετικά παρόμοια, υποδηλώνοντας ότι οι υπερπαραμέτροι που επιλέχθηκαν μέσω του GridSearchCV δεν βελτίωσαν την απόδοση γενίκευσης. Στη συνέχεια, για το Random Forest με RandomizedSearchCV η ακρίβεια στο σύνολο εκπαίδευσης μειώθηκε σε σύγκριση με το τυπικό Random Forest, αλλά όχι τόσο πολύ όσο με το GridSearchCV, υποδεικνύοντας μια πιο ευέλικτη αναζήτηση υπερπαραμέτρων. Παρόμοια με το GridSearchCV, η ακρίβεια στο σύνολο δοκιμής δεν παρουσίασε σημαντική βελτίωση σε σύγκριση με το τυπικό Random Forest. Ακόμη, χρησιμοποιώντας τον ταξινομητή Random Forest με RandomOverSampler, φαίνεται πως η ακρίβεια στο σύνολο εκπαίδευσης παρέμεινε πολύ υψηλή, γεγονός που υποδηλώνει ότι η υπερδειγματοληψία βοήθησε στην καλύτερη σύλληψη των περιπτώσεων της μειονοτικής κλάσης, καθώς και πως η ακρίβεια στο σύνολο δοκιμής βελτιώθηκε ελαφρώς σε σύγκριση με το τυπικό Random Forest, υποδεικνύοντας καλύτερη απόδοση γενίκευσης. Παράλληλα, η χρήση του Random Forest με SMOTE Oversampling, παρόμοια με το RandomOverSampler, βοήθησε στη βελτίωση της απόδοσης του μοντέλου στο σύνολο εκπαίδευσης. Ωστόσο, η ακρίβεια στο σύνολο δοκιμής μειώθηκε, γεγονός που υποδηλώνει ότι η τεχνική υπερδειγματοληψίας ενδέχεται να έχει εισάγει θόρυβο ή ότι το μοντέλο προσαρμόζεται υπερβολικά στα επαυξημένα δεδομένα. Στη συνέχεια, ο ταξινομητής Decision Tree, πέτυχε πολύ υψηλή ακρίβεια στο σύνολο εκπαίδευσης (99,375%), παρόμοια με το Random Forest, υποδεικνύοντας ότι μπορούσε να απομνημονεύσει τέλεια τα δεδομένα εκπαίδευσης. Ωστόσο, η ακρίβεια στο σύνολο δοκιμής μειώθηκε σημαντικά (55,5%), υποδεικνύοντας σοβαρή υπερπροσαρμογή και κακή γενίκευση. Ακόμη, είναι φανερό πως για τον ταξινομητή Decision Tree με χρήση των GridSearchCV και RandomizedSearchCV, η ακρίβεια στο σύνολο εκπαίδευσης μειώθηκε σε σύγκριση με το τυπικό Decision Tree, αλλά η διαφορά δεν ήταν σημαντική, ενώ τόσο το GridSearchCV όσο και το RandomizedSearchCV δεν βελτίωσαν σημαντικά την ακρίβεια στο σύνολο δοκιμής σε σύγκριση με το τυπικό Δέντρο Απόφασης. Επιπλέον, προκύπτει από τη χρήση του ταξινομητή Decision Tree με RandomOverSampler και SMOTE Oversampling ότι, παρόμοια με το Random Forest, και οι δύο τεχνικές υπερδειγματοληψίας βελτίωσαν την ακρίβεια στο σύνολο εκπαίδευσης, ωστόσο, η ακρίβεια στο σύνολο δοκιμής μειώθηκε σημαντικά, υποδεικνύοντας πιθανή υπερπροσαρμογή ή εισαγωγή θορύβου. Αναφορικά με τους ταξινομητές Linear SVC, Stochastic Gradient Descent, Multinomial Logistic Regression, και οι τρεις ταξινομητές πέτυχαν μέτρια ακρίβεια στο σύνολο εκπαίδευσης, ενώ η ακρίβεια στο σύνολο δοκιμής ήταν σχετικά σταθερή αλλά όχι πολύ υψηλή. Τέλος, για τον ταξινομητή Multinomial Logistic Regression με χρήση GridSearchCV, η ακρίβεια στο σύνολο εκπαίδευσης βελτιώθηκε ελαφρώς σε σύγκριση με την τυπική Multinomial Logistic Regression, ωστόσο, η βελτίωση δεν αντανακλάται στην ακρίβεια στο σύνολο δοκιμής.

Πίνακας 6: Αποτελέσματα ταξινομητών για max_features = 500

Ταξινομητές	Ακρίβεια στο σύνολο εκπαίδευσης	Ακρίβεια στο σύνολο δοκιμών	Παράμετρος max_features του TF-IDF
Random Forest	99.875%	66.125%	500
Random Forest GridSearchCV	65.75%	65.0%	500
Random Forest RandomizedSearchCV	65.5625%	65.125%	500
Random Forest RandomOverSampler	99.875%	67.625%	500
Random Forest SMOTE	99.59375%	57.625%	500
Decision Tree	99.875%	57.375%	500
Decision Tree GridSearchCV	65.0625%	65.0%	500
Decision Tree RandomizedSearchCV	99.875%	65.0%	500
Decision Tree RandomOverSampler	97.65625%	55.5%	500
Decision Tree SMOTE	99.875%	54.5%	500
Linear SVC	76.09375%	56.75%	500
Stochastic Gradient Descent	67.375%	58.625%	500
Multinomial Logistic Regression	77.78125%	58.625%	500
Multinomial Logistic Regression GridSearchCV	71.28125%	49.75%	500

Σύμφωνα με τα παραπάνω ευρήματα, ο ταξινομητής Random Forest έχει πολύ υψηλή ακρίβεια στο σύνολο εκπαίδευσης, υποδεικνύοντας ότι το μοντέλο ταιριάζει σχεδόν απόλυτα με τα δεδομένα εκπαίδευσης, ενώ η ακρίβεια στο σύνολο δοκιμών χαρακτηρίζεται ως μέτρια, γεγονός που υποδηλώνει κάποιο βαθμό υπερπροσαρμογής, καθώς το μοντέλο αποδίδει σημαντικά χειρότερα σε νέα δεδομένα. Το RandomOverSampler βελτιώνει ελαφρώς την απόδοση στο σύνολο δοκιμών σε σύγκριση με το βασικό Random Forest, υποδεικνύοντας ότι η αντιμετώπιση της ανισορροπίας των κλάσεων μπορεί να είναι επωφελής. Ωστόσο, η τεχνική SMOTE Oversampling φαίνεται να επηρεάζει αρνητικά την απόδοση. Όσον αφορά τον ταξινομητή Decision Tree, προσδίδει πολύ υψηλή ακρίβεια στο σύνολο εκπαίδευσης και μέτρια έως χαμηλή ακρίβεια στο σύνολο δοκιμών, ενώ οι τεχνικές GridSearchCV και RandomizedSearchCV δεν φαίνεται να βελτιώνουν σημαντικά την απόδοση. Αναφορικά με τους ταξινομητές Linear SVC, Stochastic Gradient Descent, Multinomial Logistic Regression, είναι προφανές ότι έχουν μέτρια ακρίβεια στο σύνολο εκπαίδευσης και μέτρια έως χαμηλή ακρίβεια στο σύνολο δοκιμών. Ακόμη, η χρήση του GridSearchCV δεν φαίνεται να παρέχει ουσιαστικές βελτιώσεις στην περίπτωση της Multinomial Logistic Regression.

Πίνακας 7: Αποτελέσματα ταξινομητών για max_features = 800

Ταξινομητές	Ακρίβεια στο σύνολο εκπαίδευσης	Ακρίβεια στο σύνολο δοκιμών	Παράμετρος max_features του TF-IDF
Random Forest	99.96875%	66.375%	800
Random Forest GridSearchCV	65.8125%	65.0%	800
Random Forest RandomizedSearchCV	65.5625%	65.125%	800
Random Forest RandomOverSampler	99.96875%	66.625%	800
Random Forest SMOTE	99.96875%	65.5%	800
Decision Tree	99.96875%	57.49%	800
Decision Tree GridSearchCV	65.0625%	65.0%	800
Decision Tree RandomizedSearchCV	65.0625%	65.0%	800
Decision Tree RandomOverSampler	99.96875%	55.125%	800
Decision Tree SMOTE	99.96875%	55.0%	800
Linear SVC	78.03125%	63.625%	800
Stochastic Gradient Descent	78.875%	64.875%	800
Multinomial Logistic Regression	75.0625%	65.125%	800
Multinomial Logistic Regression GridSearchCV	65.28125%	65.0%	800

Συγκρίνοντας τις παραπάνω μετρήσεις, οι ταξινομητές παρουσιάζουν ένα ευρύ φάσμα ακρίβειας τόσο στα σύνολα εκπαίδευσης όσο και στα σύνολα δοκιμής. Ειδικότερα, ο ταξινομητής Random Forest επιτυγχάνει υψηλή ακρίβεια 99,97% στο σύνολο εκπαίδευσης, αλλά αυτή πέφτει σημαντικά στο 66,38% στο σύνολο δοκιμής, αναδεικνύοντας ένα σημαντικό βαθμό υπερπροσαρμογής. Ομοίως, το μοντέλο Decision Tree υποφέρει επίσης από υπερπροσαρμογή, με ακρίβεια 99,97% στο σύνολο εκπαίδευσης και σημαντικά χαμηλότερη ακρίβεια 57,49% στο σύνολο δοκιμής. Αντιθέτως, τα μοντέλα Linear SVC, Stochastic Gradient Descent και Multinomial Logistic Regression επιδεικνύουν πιο ισορροπημένες επιδόσεις. Το Linear SVC επιτυγχάνει ακρίβεια 78,03% στο σύνολο εκπαίδευσης, η οποία πέφτει ελαφρώς στο 63,63% στο σύνολο δοκιμής, υποδεικνύοντας κάποια υπερπροσαρμογή αλλά καλύτερη ικανότητα γενίκευσης σε σύγκριση με τα προηγούμενα μοντέλα που βασίζονται σε δέντρα. Ακόμη, οι ταξινομητές Stochastic Gradient Descent και Multinomial Logistic Regression παρουσιάζουν επίσης μέτρια πτώση της ακρίβειας από το σύνολο εκπαίδευσης στο σύνολο δοκιμής, με ακρίβειες 78,88% και 75,06% στο σύνολο εκπαίδευσης, αντίστοιχα, και ακρίβειες 64,88% και 65,13% στο σύνολο δοκιμής, αντίστοιχα. Παρά τις προσπάθειες βελτιστοποίησης των υπερπαραμέτρων μέσω τεχνικών όπως το GridSearchCV και το RandomizedSearchCV, παρατηρείται περιορισμένη βελτίωση σε όλα τα μοντέλα, γεγονός που υποδηλώνει ότι είναι απαραίτητες οι προσαρμογές των μοντέλων.

Πίνακας 8: Αποτελέσματα ταξινομητών για max_features = 2000

Ταξινομητές	Ακρίβεια στο σύνολο εκπαίδευσης	Ακρίβεια στο σύνολο δοκιμών	Παράμετρος max_features του TF-IDF
Random Forest	99.96875%	66.375%	2000
Random Forest GridSearchCV	65.09375%	65.0%	2000
Random Forest RandomizedSearchCV	65.0625%	65.0%	2000
Random Forest RandomOverSampler	99.96875%	66.125%	2000
Random Forest SMOTE	99.96875%	64.25%	2000
Decision Tree	99.96875%	57.875%	2000
Decision Tree GridSearchCV	65.0625%	65.0%	2000
Decision Tree RandomizedSearchCV	65.0625%	65.0%	2000
Decision Tree RandomOverSampler	99.96875%	54.8749%	2000
Decision Tree SMOTE	99.96875%	55.5%	2000
Linear SVC	87.4375%	63.875%	2000
Stochastic Gradient Descent	86.09375%	64.125%	2000
Multinomial Logistic Regression	78.6875%	64.625%	2000
Multinomial Logistic Regression GridSearchCV	65.0625%	65.0%	2000

Από τον παραπάνω πίνακα, παρατηρείται ότι το Random Forest επιτυγχάνει υψηλή ακρίβεια 99,97% στο σύνολο εκπαίδευσης, αλλά αυτή πέφτει στο 66,38% στο σύνολο δοκιμής, παρουσιάζοντας ξανά σημάδια υπερπροσαρμογής. Η χρήση των GridSearchCV και RandomizedSearchCV για το Random Forest αποδίδουν παρόμοιες ακρίβειες, με οριακές βελτιώσεις στο σύνολο δοκιμής σε σύγκριση με το βασικό μοντέλο. Επιπλέον, οι τεχνικές RandomOverSampler και SMOTE Oversampling που εφαρμόζονται στο Random Forest παρουσιάζουν μικρές διαφοροποιήσεις στις ακρίβειες του συνόλου δοκιμής, υποδηλώνοντας περιορισμένη αποτελεσματικότητα στην αντιμετώπιση της ανισορροπίας των κλάσεων. Ομοίως, ο ταξινομητής Decision Tree εμφανίζει υπερπροσαρμογή, με υψηλή ακρίβεια στο σετ εκπαίδευσης 99,97% , αλλά πολύ χαμηλότερη ακρίβεια στο σετ δοκιμής 57,88%. Η ρύθμιση των υπερπαραμέτρων μέσω των GridSearchCV και RandomizedSearchCV δεν οδηγεί σε σημαντικές βελτιώσεις στην απόδοση του Decision Tree. Παράλληλα, οι τεχνικές RandomOverSampler και SMOTE Oversampling αποτυγχάνουν να βελτιώσουν σημαντικά την απόδοση του Decision Tree. Στη συνέχεια, το Linear SVC παρουσιάζει σχετικά ισορροπημένη απόδοση, με ακρίβεια 87,44% στο σύνολο εκπαίδευσης και 63,88% στο σύνολο δοκιμής, δείχνοντας κάποιο βαθμό υπερπροσαρμογής. Η Stochastic Gradient Descent παρουσιάζει επίσης ισορροπημένη απόδοση, με ακρίβεια 86,09% στο σύνολο εκπαίδευσης και 64,13% στο σύνολο δοκιμής. Η Multinomial Logistic Regression επιτυγχάνει ακρίβεια 78,69% στο σετ εκπαίδευσης και ακρίβεια στο σετ δοκιμής 64,63%, υποδεικνύοντας μέτρια υπερπροσαρμογή. Το GridSearchCV για την πολυωνυμική λογιστική παλινδρόμηση αποδίδει παρόμοια αποτελέσματα με το βασικό μοντέλο, υποδηλώνοντας περιορισμένη βελτίωση μέσω της ρύθμισης των υπερπαραμέτρων.

8. Συζήτηση

Από τους παραπάνω πίνακες, αρχικά είναι φανερό ότι υπάρχει υπερπροσαρμογή στα μοντέλα Random Forest και Decision Tree, καθώς επιδεικνύουν εξαιρετική ακρίβεια στο σύνολο εκπαίδευσης, γεγονός που υποδηλώνει ότι έχουν απομνημονεύσει επιτυχώς τα δεδομένα εκπαίδευσης. Ωστόσο, στο σύνολο δοκιμής η ακρίβεια είναι σημαντικά μειωμένη (π.χ. περίπου 65% για το Random Forest), γεγονός που υποδηλώνει ότι τα μοντέλα δυσκολεύονται να εφαρμόσουν αποτελεσματικά τη γνώση τους σε άγνωστα δεδομένα.

Η διαφορά στην ακρίβεια μεταξύ των συνόλων εκπαίδευσης και δοκιμής είναι μια τυπική ένδειξη υπερπροσαρμογής, κατά την οποία το μοντέλο εστιάζει υπερβολικά στην προσαρμογή των τυχαίων μεταβολών στα δεδομένα εκπαίδευσης αντί να καταγράφει τα θεμελιώδη πρότυπα.

Αναφορικά με την αποτελεσματικότητα της ρύθμισης των υπερπαραμέτρων, οι GridSearchCV και RandomizedSearchCV χρησιμοποιούνται για τη βελτιστοποίηση των υπερπαραμέτρων των ταξινομητών, όπως η ποσότητα των δέντρων στο Random Forest ή το μέγιστο βάθος των δέντρων απόφασης. Ακόμη και μετά την προσαρμογή των υπερπαραμέτρων, οι επιδόσεις των μοντέλων στο σύνολο δοκιμών δεν παρουσιάζουν ουσιαστική βελτίωση σε σύγκριση με τα βασικά μοντέλα με προεπιλεγμένες υπερπαραμέτρους. Αυτό υποδηλώνει ότι οι υπερπαραμέτροι που διερευνήθηκαν σε αυτές τις μεθόδους μπορεί να μην είναι οι κύριοι καθοριστικοί παράγοντες που επηρεάζουν την ικανότητα γενίκευσης των μοντέλων. Είναι πιθανό ότι άλλες μεταβλητές, όπως η επιλογή των χαρακτηριστικών ή οι στρατηγικές για την προετοιμασία των δεδομένων, μπορεί να έχουν σημαντικότερο αντίκτυπο στη βελτίωση των επιδόσεων.

Στη συνέχεια, σχετικά με τις επιπτώσεις της αντιμετώπισης μη ισορροπημένων δεδομένων, οι προσεγγίσεις RandomOverSampler και SMOTE Oversampling χρησιμοποιούνται για τον μετριασμό των προβλημάτων μη ισορροπημένων δεδομένων, όπου μια κλάση παρουσιάζει πολύ μεγαλύτερη επικράτηση από τις άλλες. Αν και το Random Forest παρουσιάζει μια μέτρια βελτίωση στην ακρίβεια του συνόλου δοκιμών ως αποτέλεσμα της χρήσης αυτών των τεχνικών, η απόδοση του Decision Tree μειώνεται στην πραγματικότητα όταν χρησιμοποιούνται μέθοδοι υπερδειγματοληψίας. Η διαφορετική αποτελεσματικότητα των στρατηγικών χειρισμού μη ισορροπημένων δεδομένων υποδεικνύεται από αυτό το μικτό αποτέλεσμα, το οποίο εξαρτάται από τον ταξινομητή και τα συγκεκριμένα χαρακτηριστικά του συνόλου δεδομένων.

Συνοψίζοντας, οι παρατηρήσεις υποδηλώνουν ότι παρόλο που ορισμένοι ταξινομητές παρουσιάζουν εξαιρετική ακρίβεια στο σύνολο εκπαίδευσης, έχουν δυσκολίες στη γενίκευση σε άγνωστα δεδομένα, γεγονός που υποδηλώνει την πιθανότητα υπερπροσαρμογής. Επιπλέον, αξίζει να σημειωθεί ότι η βελτιστοποίηση των υπερπαραμέτρων και η εφαρμογή μεθόδων για την αντιμετώπιση των ανισόρροπων δεδομένων δεν βελτιώνουν ομοιόμορφα την απόδοση όλων των ταξινομητών. Αυτό υπογραμμίζει τη σημασία της διεξαγωγής σχολαστικών πειραματισμών και αναλύσεων για τον προσδιορισμό των πιο αποτελεσματικών προσεγγίσεων για την ενίσχυση της γενίκευσης και της απόδοσης του μοντέλου. Για τον μετριασμό της υπερπροσαρμογής και τη βελτίωση της συνολικής απόδοσης του μοντέλου, ενδέχεται να απαιτείται πρόσθετη ανάλυση, όπως η μηχανική των χαρακτηριστικών, η προετοιμασία των δεδομένων και η αξιολόγηση εναλλακτικών αλγορίθμων.

9. Συμπεράσματα και μελλοντικές εργασίες

Από την ανάλυση των επιδόσεων των ταξινομητών, μπορούν να εξαχθούν διάφορα συμπεράσματα σχετικά με την αποτελεσματικότητά τους και τους τομείς που θα μπορούσαν να βελτιωθούν. Πρώτον, οι ταξινομητές, συγκεκριμένα οι Random Forest και Decision Tree, παρουσιάζουν ενδείξεις υπερπροσαρμογής, όπως υποδεικνύεται από την αυξημένη ακρίβεια στο σύνολο εκπαίδευσης αλλά τη μειωμένη ακρίβεια στο σύνολο δοκιμής. Αυτό υπογραμμίζει τη σημασία του μετριασμού των προβλημάτων υπερπροσαρμογής με τη χρήση τεχνικών κανονικοποίησης ή προσεγγίσεων επιλογής χαρακτηριστικών για την ενίσχυση της γενίκευσης του μοντέλου. Επιπλέον, η αποτελεσματικότητα της ρύθμισης των υπερπαραμέτρων μέσω των GridSearchCV και RandomizedSearchCV φαίνεται να είναι περιορισμένη, υποδεικνύοντας ότι εναλλακτικές μέθοδοι ή μια πιο εξαντλητική εξέταση του χώρου υπερπαραμέτρων μπορεί να είναι απαραίτητη για τη βελτίωση της απόδοσης του μοντέλου. Επιπλέον, τα ασυνεπή αποτελέσματα που προκύπτουν από μη ισορροπημένες τεχνικές χειρισμού δεδομένων, όπως οι RandomOverSampler και SMOTE, υπογραμμίζουν την ανάγκη για μια εξελιγμένη κατανόηση του τρόπου με τον οποίο αυτές οι προσεγγίσεις αλληλεπιδρούν με διάφορους ταξινομητές και σύνολα δεδομένων. Η περαιτέρω έρευνα θα πρέπει να θέσει ως προτεραιότητα τη διερεύνηση εναλλακτικών αλγορίθμων, τακτικών σχεδιασμού χαρακτηριστικών και τεχνικών συνόλου για την αντιμετώπιση της υπερπροσαρμογής και την ενίσχυση της συνολικής ανθεκτικότητας και αποτελεσματικότητας των μοντέλων μηχανικής μάθησης σε πρακτικά σενάρια.

Επιπλέον, η διερεύνηση της ενσωμάτωσης της εμπειρογνωμοσύνης στον τομέα και η χρήση εξελιγμένων μεθόδων για την αξιολόγηση των μοντέλων, συμπεριλαμβανομένης της διασταυρούμενης επικύρωσης και της ανάλυσης των καμπυλών μάθησης, μπορεί να αποφέρει βαθιά κατανόηση της απόδοσης των μοντέλων και να διευκολύνει τη δημιουργία πιο αποτελεσματικών και αξιόπιστων μοντέλων πρόβλεψης.

Συμπερασματικά, τα αποτελέσματα της παρούσας μελέτης θέτουν τα θεμέλια για μελλοντικές ερευνητικές προσπάθειες που επικεντρώνονται στην ανάπτυξη τεχνικών μηχανικής μάθησης και στη βελτίωση της πρακτικής αποτελεσματικότητάς τους σε διάφορους τομείς.

10. Αναφορές

- [1] Anwar, N. (2019, May 15). Review Paper on Various Software Testing Techniques | Global Journal of Computer Science and Technology. <http://dx.doi.org/10.34257/GJCSTCVOL19IS2PG43>
- [2] Software Testing Techniques: A Literature Review. (2016, November 1). IEEE Conference Publication | IEEE Xplore. <http://dx.doi.org/10.1109/ICT4M.2016.045>
- [3] F. Okezie et al 2019 J. Phys.: Conf. Ser. 1378 042030. DOI 10.1088/1742-6596/1378/4/042030
- [4] Ateşoğullar, D., & Mishra, A. (2020, December 15). AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW available in https://www.researchgate.net/publication/346109409_AUTOMATION_TESTING_TOOLS_A_COMPARATIVE_VIEW
- [5] Advaita Aditya Chevuturu et al A Comparative Survey on Software Testing Tools - International Journal of Engineering and Advanced Technology (IJEAT). (2023, May 10). International Journal of Engineering and Advanced Technology (IJEAT). <https://www.doi.org/10.35940/ijeat.F3664.0811622>
- [6] Battina, D. S. (2019, December 12). Artificial Intelligence in Software Test Automation: A Systematic Literature Review. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4004324
- [7] Job, M. A. (2021, January 1). Automating and Optimizing Software Testing using Artificial Intelligence Techniques. International Journal of Advanced Computer Science and Applications. <https://doi.org/10.14569/ijacsa.2021.0120571>
- [8] Artificial Intelligence in Software Testing: A Systematic Review. (2023, October 31). IEEE Conference Publication | IEEE Xplore. <http://dx.doi.org/10.1109/TENCON58879.2023.10322349>
- [9] AI-based Test Automation: A Grey Literature Analysis. (2021, April 1). IEEE Conference Publication | IEEE Xplore. <https://doi.org/10.1109/ICSTW52544.2021.00051>
- [10] Hu, C., Gong, H., & He, Y. (2022, October 12). Data driven identification of international cutting edge science and technologies using SpaCy. <https://doi.org/10.1371/journal.pone.0275872>
- [11] Reyna, R., & Simske, S. J. (2022, June 7). How to Generate and Import Functional Test Cases into a Project Management Software System using Natural Language Processing. <https://doi.org/10.2352/issn.2168-3204.2022.19.1.9>
- [12] Garousi V., Felderer M. (2020, May). NLP-assisted software testing: A systematic mapping of the literature. DOI:10.1016/j.infsof.2020.106321
- [13] Gupta Alok (2023). Test Case Selection from Test Specifications using Natural Language Processing. <http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-223665>
- [14] J. Alzubi, A. Nayyar, A. Kumar, "Machine Learning from Theory to Algorithms: An Overview", Journal of Physics: Conference Series, Volume 1142, Second National Conference on Computational Intelligence (NCCI 2018) 5 December 2018, Bangalore, India
- [15] J. Brownlee: "4 Types of Classification Tasks in Machine Learning", available in <https://machinelearningmastery.com/types-of-classification-in-machine-learning/>

-
- [16] O. Chapelle, B. Scholkopf, A. Zien, Eds: "Semi-Supervised Learning", IEEE Transactions on Neural Networks, 2009
- [17] Tabassum, N., Namoun, A., Alyas, T., Tufail, A., Taqi, M., & Kim, K. H. (2023, February 23). Classification of Bugs in Cloud Computing Applications Using Machine Learning Techniques. *Applied Sciences*, 13(5), 2880. <https://doi.org/10.3390/app13052880>
- [18] Kaur, A., & Jindal, S. G. (2019, June 3). Text analytics based severity prediction of software bugs for apache projects. *International Journal of Systems Assurance Engineering and Management*. <https://doi.org/10.1007/s13198-019-00807-8>
- [19] Choudhary A. & Singh S. (2017, June 20). Neural Network Based Bug Priority Prediction Model Using Text Classification Techniques. <https://doi.org/10.26483/ijarcs.v8i5.3559>
- [20] Colavito, G., Lanubile, F., & Novielli, N. (2022, May 21). Issue report classification using pre-trained language models. <https://doi.org/10.1145/3528588.3528659>
- [21] Pandey, N., Hudait, A., Sanyal, D. K., & Sen, A. (2017, July 13). Automated Classification of Issue Reports from a Software Issue Tracker. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-981-10-3373-5_42
- [22] A. Kukkar, R. Mohana, A. Nayyar, J. Kim, B.-G. Kang, and N. Chilamkurti, "A novel Deep-Learning-Based bug severity classification technique using convolutional neural networks and random forest with boosting," *Sensors*, vol. 19, no. 13, p. 2964, Jul. 2019.
- [23] Kanwal, J., & Maqbool, O. (2012, March 1). Bug Prioritization to Facilitate Bug Report Triage. *Journal of Computer Science and Technology*. <https://doi.org/10.1007/s11390-012-1230-3>
- [24] Pandey, N., Sanyal, D. K., Hudait, A., & Sen, A. (2017, July 24). Automated classification of software issue reports using machine learning techniques: an empirical study. *Innovations in Systems and Software Engineering*. <https://doi.org/10.1007/s11334-017-0294-1>
- [25] Tian, Y., Lo, D., Xia, X., & Sun, C. (2014, August 3). Automated prediction of bug report priority using multi-factor analysis. *Empirical Software Engineering*. <https://doi.org/10.1007/s10664-014-9331-y>