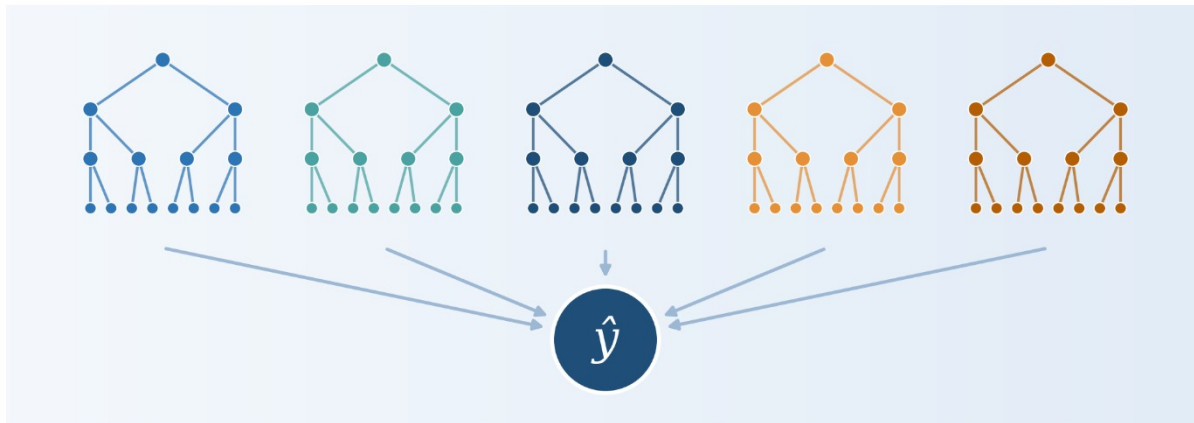


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανασκόπηση Βιβλιογραφίας και Πειραματική
Συγκριτική Μελέτη Bagging και Boosting Ensemble
Αλγορίθμων Κατηγοριοποίησης»



Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ιωάννη Παπαδόπουλου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο των σπουδών μου στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος. Το θέμα της — η συγκριτική μελέτη των ensemble αλγορίθμων κατηγοριοποίησης — επιλέχθηκε επειδή συνδυάζει δύο στοιχεία που θεωρώ πολύτιμα: ένα στέρεο θεωρητικό υπόβαθρο, που ανάγεται σε ορισμένες από τις σημαντικότερες ιδέες της μηχανικής μάθησης, και μια έντονα πρακτική διάσταση, καθώς οι εξεταζόμενοι αλγόριθμοι αποτελούν καθημερινά εργαλεία της σύγχρονης επιστήμης δεδομένων. Μέσα από την εκπόνησή της απέκτησα ουσιαστική εμπειρία στον σχεδιασμό και την εκτέλεση ελεγχόμενων υπολογιστικών πειραμάτων, στη στατιστική αποτίμηση αποτελεσμάτων και στη συστηματική μελέτη επιστημονικής βιβλιογραφίας.

Περίληψη

Η εργασία μελετά συγκριτικά τις δύο κυρίαρχες οικογένειες μεθόδων συνόλου (ensembles) για την κατηγοριοποίηση: το Bagging και το Boosting. Στο βιβλιογραφικό σκέλος παρουσιάζονται το θεωρητικό υπόβαθρο των δέντρων αποφάσεων και του διλήμματος μεροληψίας-διακύμανσης, οι θεμελιώδεις αλγόριθμοι κάθε οικογένειας — Bagging, Random Forest, Extra-Trees, AdaBoost, Gradient Boosting και οι σύγχρονες ιστογραμμικές υλοποιήσεις του — καθώς και οι σημαντικότερες εμπειρικές συγκριτικές μελέτες. Στο πειραματικό σκέλος, οι έξι αντίστοιχοι ταξινομητές της βιβλιοθήκης scikit-learn αξιολογούνται σε δεκατρία σύνολα δεδομένων του αποθετηρίου KEEL, κλιμακούμενου μεγέθους (768–57.999 δείγματα) και πλήθους κλάσεων (2–26), με δεκαπλή στρωματοποιημένη διασταυρούμενη επικύρωση, προεπιλεγμένες παραμέτρους και μέτρηση καθαρού χρόνου CPU, συνολικά 780 εκπαιδεύσεις. Τα αποτελέσματα αναδεικνύουν μια στατιστικά μη διακρίσιμη κορυφή προβλεπτικής απόδοσης (Extra-Trees, ιστογραμμικό Gradient Boosting, Random Forest), τη σημαντική υστέρηση του AdaBoost με προεπιλεγμένες ρυθμίσεις στα πολυκλασικά προβλήματα, και διαφορές έως τεσσάρων τάξεων μεγέθους στο κόστος εκπαίδευσης, με το ιστογραμμικό Gradient Boosting έως και 70 φορές ταχύτερο του κλασικού. Τον καλύτερο συμβιβασμό ποιότητας-κόστους προσφέρουν τα Extra-Trees και το ιστογραμμικό Gradient Boosting, ενώ το πλήθος των κλάσεων αναδεικνύεται καθοριστικότερος παράγοντας επιλογής και από το μέγεθος του προβλήματος.

Literature Review and Experimental Comparative Study of Bagging and Boosting Ensemble Classification Algorithms

Abstract

This thesis presents a comparative study of the two dominant families of ensemble classification methods: Bagging and Boosting. The literature review covers the theoretical background of decision trees and the bias-variance trade-off, the fundamental algorithms of each family — Bagging, Random Forest, Extra-Trees, AdaBoost, Gradient Boosting and its modern histogram-based implementations — as well as the most influential empirical comparative studies. In the experimental part, the six corresponding classifiers of the scikit-learn library are evaluated on thirteen datasets from the KEEL repository, spanning 768 to 57,999 instances and 2 to 26 classes, using stratified 10-fold cross validation, default hyperparameters and CPU process-time measurements, for a total of 780 model trainings. The results reveal a statistically indistinguishable top tier in predictive performance (Extra-Trees, histogram-based Gradient Boosting, Random Forest), a significant degradation of AdaBoost with default settings on multi-class problems, and training-cost differences of up to four orders of magnitude, with histogram-based Gradient Boosting being up to 70 times faster than the classical implementation. Extra-Trees and histogram-based Gradient Boosting offer the best quality-cost trade-off, while the number of classes emerges as a more decisive selection factor than dataset size.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου για την καθοδήγηση, τις εύστοχες παρατηρήσεις και τη συνεχή διαθεσιμότητά του καθ' όλη τη διάρκεια εκπόνησης της εργασίας, καθώς και την οικογένειά μου για τη στήριξή της σε όλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract.....	vii
Ευχαριστίες.....	viii
Περιεχόμενα.....	ix
Κατάλογος Σχημάτων.....	xii
Κατάλογος Πινάκων.....	xii
Κατάλογος Αλγορίθμων.....	xii
Συντομογραφίες.....	xiv
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Εισαγωγή.....	1
1.2 Κίνητρο: από τον μεμονωμένο ταξινομητή στα σύνολα μοντέλων.....	1
1.3 Στόχοι και ερευνητικά ερωτήματα.....	2
1.4 Συνεισφορά της εργασίας.....	3
1.5 Δομή της εργασίας.....	3
1.6 Επίλογος.....	3
Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο.....	4
2.1 Εισαγωγή.....	4
2.2 Επιβλεπόμενη μάθηση και το πρόβλημα της κατηγοριοποίησης.....	4
2.3 Δέντρα αποφάσεων.....	4
2.3.1 Δομή και λειτουργία.....	5
2.3.2 Η αναδρομική διαδικασία κατασκευής.....	5
2.3.3 Κριτήρια επιλογής διαχωρισμών.....	6
2.3.4 Οι βασικοί αλγόριθμοι επαγωγής: ID3, C4.5 και CART.....	6
2.3.5 Υπερπροσαρμογή και αστάθεια.....	7
2.4 Το δίλημμα μεροληψίας-διακύμανσης.....	7
2.5 Αξιολόγηση ταξινομητών.....	8
2.5.1 Πίνακας σύγχυσης και μετρικές ποιότητας.....	8
2.5.2 Διασταυρούμενη επικύρωση.....	9
2.5.3 Μετρικές υπολογιστικού κόστους.....	10
2.6 Επίλογος.....	10

Κεφάλαιο 3ο:	Μέθοδοι Συνόλου Ταξινόμητων: Ανασκόπηση Βιβλιογραφίας.....	11
3.1	Εισαγωγή.....	11
3.2	Η ιδέα των μεθόδων συνόλου.....	11
3.2.1	Τρόποι συνδυασμού των προβλέψεων.....	12
3.3	Bagging.....	12
3.3.1	Bootstrap δειγματοληψία και ο αλγόριθμος.....	12
3.3.2	Γιατί το Bagging μειώνει τη διακύμανση.....	13
3.4	Random Forests.....	13
3.5	Extremely Randomized Trees (Extra-Trees).....	14
3.6	Boosting.....	14
3.6.1	Από τους αδύναμους στους ισχυρούς ταξινομητές.....	14
3.6.2	AdaBoost.....	14
3.6.3	Μηχανές Gradient Boosting.....	15
3.6.4	Ιστογραμμικές υλοποιήσεις: XGBoost, LightGBM και HistGradientBoosting.....	16
3.7	Θεωρητική αντιπαράβολή Bagging και Boosting.....	16
3.8	Εμπειρικές συγκριτικές μελέτες στη βιβλιογραφία.....	18
3.9	Επίλογος.....	18
Κεφάλαιο 4ο:	Πειραματική Μεθοδολογία.....	19
4.1	Εισαγωγή.....	19
4.2	Πειραματικός σχεδιασμός.....	19
4.3	Σύνολα δεδομένων.....	19
4.4	Αλγόριθμοι και υλοποίηση.....	21
4.5	Πρωτόκολλο αξιολόγησης.....	21
4.6	Περιβάλλον εκτέλεσης.....	22
4.7	Επίλογος.....	22
Κεφάλαιο 5ο:	Αποτελέσματα και Συζήτηση.....	23
5.1	Εισαγωγή.....	23
5.2	Προβλεπτική απόδοση.....	23
5.3	Χρόνος εκπαίδευσης.....	25
5.4	Χρόνος ταξινόμησης.....	27
5.5	Συμβιβασμός ποιότητας και κόστους.....	27
5.6	Στατιστικός έλεγχος σημαντικότητας.....	28
5.7	Επίδραση των χαρακτηριστικών των προβλημάτων.....	29
5.8	Συζήτηση.....	29

5.9	Περιορισμοί της μελέτης.....	30
5.10	Επίλογος.....	30
Κεφάλαιο 6ο:	Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	31
6.1	Εισαγωγή.....	31
6.2	Σύνοψη συμπερασμάτων.....	31
6.3	Μελλοντικές επεκτάσεις.....	31
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		33
ΠΑΡΑΡΤΗΜΑ Α : ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΠΕΙΡΑΜΑΤΩΝ.....		36
ΠΑΡΑΡΤΗΜΑ Β : ΑΝΑΛΥΤΙΚΟΙ ΠΙΝΑΚΕΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....		39

Κατάλογος Σχημάτων

Σχήμα 2.1: Δέντρο απόφασης βάθους 3 για το σύνολο pima.....	5
Σχήμα 2.2: Σφάλματα εκπαίδευσης και ελέγχου συναρτήσει του μέγιστου βάθους (spambase).....	7
Σχήμα 2.3: Το δίλημμα μεροληψίας-διακύμανσης.....	8
Σχήμα 3.1: Σχηματική αναπαράσταση του Bagging.....	13
Σχήμα 3.2: Σχηματική αναπαράσταση του Boosting.....	15
Σχήμα 5.1: Χάρτης θερμότητας της μέσης ορθότητας.....	24
Σχήμα 5.2: Μέση κατάταξη των αλγορίθμων (μικρότερη τιμή = καλύτερη).....	25
Σχήμα 5.3: Μέσος χρόνος εκπαίδευσης ανά fold συναρτήσει του πλήθους δειγμάτων (λογαριθμικοί άξονες).....	26
Σχήμα 5.4: Συμβιβασμός ποιότητας-κόστους των έξι αλγορίθμων.....	28

Κατάλογος Πινάκων

Πίνακας 2.1: Παράδειγμα πίνακα σύγκρισης δυαδικού προβλήματος ($N = 1.000$, 100 θετικά).....	9
Πίνακας 3.1: Ποιοτική σύγκριση των χαρακτηριστικών των έξι αλγορίθμων της μελέτης.....	17
Πίνακας 4.1: Τα δεκατρία σύνολα δεδομένων της μελέτης (πηγή: αποθετήριο KEEL).....	20
Πίνακας 4.2: Κύριες προεπιλεγμένες υπερπαραμέτροι των έξι αλγορίθμων (scikit-learn 1.7).....	21
Πίνακας 5.1: Μέση ορθότητα ανά σύνολο δεδομένων και αλγόριθμο.....	23
Πίνακας 5.2: Μέσο μακρο-συναθροισμένο F-score ανά σύνολο δεδομένων και αλγόριθμο.....	24
Πίνακας 5.3: Μέσος χρόνος εκπαίδευσης ανά fold, σε δευτερόλεπτα CPU (έντονα η μικρότερη τιμή).	26
Πίνακας 5.4: Μέσος χρόνος ταξινόμησης ανά fold, σε δευτερόλεπτα CPU (έντονα η μικρότερη τιμή).	27
Πίνακας 5.5: Συγκεντρωτική εικόνα: μέσες κατατάξεις, νίκες και μέσοι χρόνοι.....	28

Κατάλογος Αλγορίθμων

Αλγόριθμος 2.1: Αναδρομική κατασκευή δέντρου απόφασης.....	6
Αλγόριθμος 3.1: AdaBoost (δυαδική μορφή).....	15
Αλγόριθμος 3.2: Η γενική μορφή του Gradient Boosting.....	16
Απόσπασμα κώδικα 4.1: Συνάρτηση ανάγνωσης αρχείων KEEL.....	21
Απόσπασμα κώδικα 4.2: Πυρήνας του πειραματικού βρόχου αξιολόγησης.....	22

Συντομογραφίες

CART	Classification and Regression Trees
CD	Critical Difference (κρίσιμη διαφορά)
CPU	Central Processing Unit
CV	Cross Validation (διασταυρούμενη επικύρωση)
Δ.Ε.	Διπλωματική Εργασία
FN	False Negatives (ψευδώς αρνητικά)
FP	False Positives (ψευδώς θετικά)
GB	Gradient Boosting
GBDT	Gradient Boosted Decision Trees
HistGB	Histogram-based Gradient Boosting
KEEL	Knowledge Extraction based on Evolutionary Learning
OOB	Out-Of-Bag
PAC	Probably Approximately Correct
RF	Random Forest
SAMME	Stagewise Additive Modeling using a Multi-class Exponential loss
TN	True Negatives (αληθώς αρνητικά)
TP	True Positives (αληθώς θετικά)

Κεφάλαιο 1ο: Εισαγωγή

Το κεφάλαιο αυτό αποτελεί την αφηρητή της εργασίας και έχει ως σκοπό να εισαγάγει τον αναγνώστη στο αντικείμενο της μελέτης. Παρουσιάζεται το πρόβλημα της κατηγοριοποίησης, γύρω από το οποίο περιστρέφεται ολόκληρη η εργασία, καθώς και οι λόγοι για τους οποίους οι μέθοδοι συνόλου ταξινομητών έχουν αναδειχθεί σε ένα από τα πλέον αποτελεσματικά εργαλεία της σύγχρονης μηχανικής μάθησης. Στη συνέχεια διατυπώνονται το κίνητρο της ενασχόλησης με το συγκεκριμένο θέμα, οι στόχοι που τίθενται και τα ερευνητικά ερωτήματα στα οποία επιχειρείται να δοθεί απάντηση, ενώ συνοψίζεται και η συνεισφορά της εργασίας. Το κεφάλαιο ολοκληρώνεται με την περιγραφή της συνολικής δομής του κειμένου, ώστε ο αναγνώστης να αποκτήσει από νωρίς μια σαφή εικόνα της πορείας που ακολουθείται στα επόμενα κεφάλαια.

1.1 Εισαγωγή

Η κατηγοριοποίηση (classification) αποτελεί ένα από τα θεμελιώδη προβλήματα της μηχανικής μάθησης και της εξόρυξης δεδομένων: με βάση ένα σύνολο παραδειγμάτων για τα οποία είναι γνωστή η κλάση στην οποία ανήκουν, ζητείται η κατασκευή ενός μοντέλου ικανού να προβλέπει την κλάση νέων, άγνωστων παραδειγμάτων [1], [2]. Εφαρμογές της συναντώνται σε κάθε σχεδόν πεδίο της σύγχρονης τεχνολογίας: από την ιατρική διάγνωση και την ανίχνευση ανεπιθύμητης αλληλογραφίας, έως την αξιολόγηση πιστοληπτικής ικανότητας, την αναγνώριση χειρόγραφων χαρακτήρων και την ανίχνευση εισβολών σε δίκτυα υπολογιστών.

Στο παρόν κεφάλαιο οριοθετείται το αντικείμενο της εργασίας. Παρουσιάζεται το κίνητρο για την ενασχόληση με τις μεθόδους συνόλου ταξινομητών (ensemble methods), διατυπώνονται οι στόχοι και τα ερευνητικά ερωτήματα, συνοψίζεται η συνεισφορά της εργασίας και, τέλος, περιγράφεται η δομή των κεφαλαίων που ακολουθούν.

1.2 Κίνητρο: από τον μεμονωμένο ταξινομητή στα σύνολα μοντέλων

Παρά την πληθώρα διαθέσιμων αλγορίθμων κατηγοριοποίησης, κανένας μεμονωμένος ταξινομητής δεν αποδίδει βέλτιστα σε όλα τα προβλήματα. Τα δέντρα αποφάσεων, για παράδειγμα, αν και διαισθητικά και υπολογιστικά αποδοτικά, είναι διαβόητα ασταθή: μικρές μεταβολές στο σύνολο εκπαίδευσης μπορούν να οδηγήσουν σε ριζικά διαφορετικά μοντέλα και, κατά συνέπεια, σε υψηλή διακύμανση του σφάλματος γενίκευσης [3]. Η διαπίστωση αυτή οδήγησε, από τη δεκαετία του 1990 και έπειτα, στην ανάπτυξη των μεθόδων συνόλου (ensemble methods), οι οποίες, αντί να επιλέγουν ένα μόνο μοντέλο, εκπαιδεύουν πολλά επιμέρους μοντέλα και συνδυάζουν τις προβλέψεις τους. Η θεωρητική και εμπειρική έρευνα έχει δείξει ότι, υπό την προϋπόθεση οι επιμέρους ταξινομητές να είναι ακριβέστεροι από την τυχαία πρόβλεψη και επαρκώς διαφορετικοί μεταξύ τους, το σύνολο επιτυγχάνει συστηματικά χαμηλότερο σφάλμα από τα μέλη του [4], [5].

Δύο οικογένειες μεθόδων συνόλου κυριαρχούν στη βιβλιογραφία και στην πράξη. Η πρώτη, το Bagging (Bootstrap Aggregating) [3] και οι απόγονοί του, όπως τα Random Forests [6] και τα Extremely Randomized Trees [7], εκπαιδεύει τα επιμέρους μοντέλα ανεξάρτητα και παράλληλα, εισάγοντας τυχαιότητα στα δεδομένα ή στη διαδικασία κατασκευής, με κύριο στόχο τη μείωση της διακύμανσης. Η δεύτερη, το Boosting, με κυριότερους εκπροσώπους τον AdaBoost [8] και τις μηχανές Gradient Boosting [9], κατασκευάζει τα μοντέλα διαδοχικά, με κάθε νέο μοντέλο να εστιάζει στα σφάλματα των προηγούμενων, στοχεύοντας πρωτίστως στη μείωση της μεροληψίας. Οι μέθοδοι αυτές δεν αποτελούν

απλώς αντικείμενο ακαδημαϊκού ενδιαφέροντος: οι σύγχρονες υλοποιήσεις gradient boosting, όπως το XGBoost [10] και το LightGBM [11], κυριαρχούν επί σειρά ετών στους διαγωνισμούς ανάλυσης δεδομένων και χρησιμοποιούνται ευρύτατα στη βιομηχανία, ιδίως σε προβλήματα με πινακοποιημένα (tabular) δεδομένα [12].

Το εύλογο πρακτικό ερώτημα που ανακύπτει είναι ποιο μέλος αυτής της μεγάλης οικογένειας αλγορίθμων οφείλει να επιλέξει ένας μηχανικός για ένα νέο πρόβλημα, ιδίως όταν δεν διαθέτει τον χρόνο ή τα δεδομένα για εκτεταμένη βελτιστοποίηση υπερπαραμέτρων. Η απάντηση δεν είναι προφανής, καθώς οι αλγόριθμοι διαφέρουν όχι μόνο ως προς την προβλεπτική τους ικανότητα, αλλά και ως προς το υπολογιστικό τους κόστος, το οποίο κλιμακώνεται με πολύ διαφορετικό τρόπο ως προς το μέγεθος του συνόλου δεδομένων και το πλήθος των κλάσεων. Η παρούσα εργασία επιχειρεί να απαντήσει σε αυτό το ερώτημα με συστηματικό, εμπειρικό τρόπο.

Η ιστορική διαδρομή του πεδίου είναι ενδεικτική της σημασίας του. Μέσα σε μία μόλις δεκαετία, από τις αρχές της δεκαετίας του 1990, διατυπώθηκαν οι θεμελιώδεις ιδέες που εξετάζει η παρούσα εργασία: η θεωρητική απόδειξη ότι οι αδύναμοι ταξινομητές μπορούν να «ενισχυθούν» σε ισχυρούς [13], το Bagging [3], ο AdaBoost [8], τα Random Forests [6] και οι μηχανές Gradient Boosting [9]. Οι εργασίες αυτές συγκαταλέγονται σήμερα στις πλέον ετεροαναφερόμενες της επιστήμης των υπολογιστών, ενώ οι απόγονοί τους — XGBoost, LightGBM και συναφείς υλοποιήσεις — αποτελούν τον πυρήνα των περισσότερων νικητήριων λύσεων σε διαγωνισμούς μηχανικής μάθησης και σημείο αναφοράς σε πρόσφατες συγκρίσεις έναντι ακόμη και βαθιών νευρωνικών δικτύων σε πινακοποιημένα δεδομένα [14].

Η ωρίμανση του πεδίου αποτυπώνεται και στο οικοσύστημα λογισμικού. Η βιβλιοθήκη scikit-learn [15], το καθιερωμένο σημείο εκκίνησης της μηχανικής μάθησης σε Python, προσφέρει σήμερα έξι διαφορετικούς ensemble ταξινομητές δέντρων, από την πιστή υλοποίηση του κλασικού Bagging έως τη σύγχρονη ιστογραμμική εκδοχή του gradient boosting. Η ίδια όμως αυτή πληθώρα επιλογών δημιουργεί το πρακτικό πρόβλημα που κινητοποιεί την παρούσα εργασία: η επιλογή μεταξύ τους σπανίως συνοδεύεται από συστηματικά, συγκρίσιμα στοιχεία τόσο για την ποιότητα όσο και για το υπολογιστικό κόστος καθενός.

1.3 Στόχοι και ερευνητικά ερωτήματα

Αντικείμενο της εργασίας είναι η βιβλιογραφική ανασκόπηση και η πειραματική συγκριτική αξιολόγηση έξι αντιπροσωπευτικών ensemble αλγορίθμων κατηγοριοποίησης των οικογενειών Bagging και Boosting, όπως αυτοί διατίθενται στη βιβλιοθήκη scikit-learn [15]: AdaBoost, Bagging, Extra-Trees, Gradient Boosting, Histogram-based Gradient Boosting και Random Forest. Η σύγκριση διενεργείται σε δεκατρία πραγματικά σύνολα δεδομένων του αποθετηρίου KEEL [16] με ενιαίο πρωτόκολλο δεκαπλής διασταυρούμενης επικύρωσης (10-fold cross validation) και προεπιλεγμένες (default) τιμές υπερπαραμέτρων. Ειδικότερα, η εργασία καλείται να απαντήσει στα ακόλουθα ερευνητικά ερωτήματα. Πρώτον (EE1), ποιος αλγόριθμος επιτυγχάνει την υψηλότερη προβλεπτική απόδοση, όπως αυτή αποτιμάται μέσω της ορθότητας (accuracy) και του μέσου αρμονικού ακρίβειας και ανάκλησης (F-score); Δεύτερον (EE2), πώς συγκρίνονται οι αλγόριθμοι ως προς το υπολογιστικό κόστος, δηλαδή τον χρόνο εκπαίδευσης και τον χρόνο ταξινόμησης; Τρίτον (EE3), ποιος αλγόριθμος προσφέρει τον καλύτερο συμβιβασμό μεταξύ προβλεπτικής απόδοσης και υπολογιστικού κόστους; Τέταρτον (EE4), πώς επιδρούν τα χαρακτηριστικά του εκάστοτε συνόλου δεδομένων, και ιδίως το πλήθος των δειγμάτων και των κλάσεων, στη συμπεριφορά κάθε αλγορίθμου;

1.4 Συνεισφορά της εργασίας

Η συνεισφορά της εργασίας είναι διττή. Σε θεωρητικό επίπεδο, παρέχεται μια συγκροτημένη, στα ελληνικά, επισκόπηση της θεωρίας των μεθόδων συνόλου, από τα θεμελιώδη αποτελέσματα των Breiman, Freund, Schapire και Friedman έως τις σύγχρονες ιστογραμμικές υλοποιήσεις gradient boosting, με έμφαση στον μηχανισμό με τον οποίο κάθε οικογένεια αντιμετωπίζει το δίλημμα μεροληψίας-διακύμανσης. Σε εμπειρικό επίπεδο, διενεργείται μια πλήρως αναπαραγώγιμη πειραματική μελέτη 780 εκτελέσεων εκπαίδευσης (13 σύνολα δεδομένων $\times 6$ αλγόριθμοι $\times 10$ folds), η οποία αποτιμά ταυτόχρονα τέσσερις μετρικές ποιότητας και δύο μετρικές υπολογιστικού κόστους, με μέτρηση καθαρού χρόνου επεξεργαστή (CPU process time) σε ενιαίο υπολογιστικό περιβάλλον. Τα πρωτογενή αποτελέσματα, ο πηγαίος κώδικας και τα σύνολα δεδομένων συνοδεύουν την εργασία, ώστε κάθε εύρημα να είναι επαληθεύσιμο.

1.5 Δομή της εργασίας

Η συνέχεια της εργασίας οργανώνεται ως εξής. Το Κεφάλαιο 2 παρουσιάζει το απαραίτητο θεωρητικό υπόβαθρο: το πρόβλημα της κατηγοριοποίησης, τα δέντρα αποφάσεων που αποτελούν το δομικό στοιχείο όλων των εξεταζόμενων μεθόδων, το δίλημμα μεροληψίας-διακύμανσης και τις μετρικές και διαδικασίες αξιολόγησης ταξινομητών. Το Κεφάλαιο 3 συνιστά τον πυρήνα της βιβλιογραφικής ανασκόπησης: αναλύει τη θεωρία των μεθόδων συνόλου και παρουσιάζει διεξοδικά τους έξι αλγόριθμους της μελέτης, καθώς και τις σημαντικότερες συγκριτικές εμπειρικές μελέτες της βιβλιογραφίας. Το Κεφάλαιο 4 περιγράφει την πειραματική μεθοδολογία: τα σύνολα δεδομένων, τις υλοποιήσεις, το πρωτόκολλο αξιολόγησης και το περιβάλλον εκτέλεσης. Το Κεφάλαιο 5 παρουσιάζει και σχολιάζει τα αποτελέσματα, συμπεριλαμβανομένου στατιστικού ελέγχου σημαντικότητας των διαφορών. Τέλος, το Κεφάλαιο 6 συνοψίζει τα συμπεράσματα και προτείνει κατευθύνσεις μελλοντικής επέκτασης.

1.6 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκε το πλαίσιο στο οποίο εντάσσεται η εργασία: το πρόβλημα της επιλογής ensemble αλγορίθμου κατηγοριοποίησης χωρίς εκ των προτέρων γνώση του προβλήματος, το κίνητρο πίσω από τη συγκριτική αξιολόγηση των οικογενειών Bagging και Boosting, τα τέσσερα ερευνητικά ερωτήματα, η συνεισφορά και η δομή της εργασίας. Το επόμενο κεφάλαιο θεμελιώνει τις έννοιες πάνω στις οποίες οικοδομείται η ανάλυση των κεφαλαίων που ακολουθούν.

Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο

Το κεφάλαιο αυτό θεμελιώνει τις βασικές έννοιες της μηχανικής μάθησης στις οποίες στηρίζεται η υπόλοιπη εργασία. Η μελέτη των μεθόδων συνόλου δεν είναι δυνατή χωρίς προηγούμενη εξοικείωση με το τυπικό πλαίσιο της επιβλεπόμενης μάθησης και με τα δέντρα αποφάσεων, τα οποία αποτελούν το δομικό υλικό όλων των αλγορίθμων που εξετάζονται στη συνέχεια. Ιδιαίτερη βαρύτητα δίνεται στο δίλημμα μεροληψίας-διακύμανσης, καθώς αυτό προσφέρει το εννοιολογικό πλαίσιο μέσα από το οποίο γίνεται κατανοητή η διαφορετική φιλοσοφία των δύο οικογενειών αλγορίθμων που συγκρίνονται στην εργασία. Τέλος, παρουσιάζονται οι μετρικές ποιότητας και υπολογιστικού κόστους, καθώς και οι διαδικασίες αξιολόγησης ταξινομητών, στοιχεία που αξιοποιούνται άμεσα στο πειραματικό μέρος της μελέτης.

2.1 Εισαγωγή

Η κατανόηση των μεθόδων συνόλου προϋποθέτει την εξοικείωση με μια σειρά θεμελιωδών εννοιών της μηχανικής μάθησης. Στο κεφάλαιο αυτό ορίζεται τυπικά το πρόβλημα της κατηγοριοποίησης, παρουσιάζονται τα δέντρα αποφάσεων, τα οποία χρησιμοποιούνται ως βασικοί ταξινομητές από όλους τους αλγορίθμους της μελέτης, αναλύεται το δίλημμα μεροληψίας-διακύμανσης, το οποίο αποτελεί το εννοιολογικό κλειδί για την κατανόηση του τρόπου λειτουργίας των Bagging και Boosting, και, τέλος, περιγράφονται οι μετρικές και οι διαδικασίες αξιολόγησης που χρησιμοποιούνται στο πειραματικό μέρος.

2.2 Επιβλεπόμενη μάθηση και το πρόβλημα της κατηγοριοποίησης

Στην επιβλεπόμενη μάθηση (supervised learning), ο αλγόριθμος μάθησης δέχεται ως είσοδο ένα σύνολο εκπαίδευσης $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, όπου κάθε $x_i \in X$ είναι ένα διάνυσμα χαρακτηριστικών (feature vector) και y_i η αντίστοιχη ετικέτα εξόδου. Όταν η ετικέτα λαμβάνει τιμές από ένα πεπερασμένο, μη διατεταγμένο σύνολο κλάσεων $Y = \{c_1, \dots, c_K\}$, το πρόβλημα ονομάζεται κατηγοριοποίηση· όταν $K = 2$ γίνεται λόγος για δυαδική (binary) κατηγοριοποίηση, ενώ για $K > 2$ για πολυκλασική (multi-class) [1]. Στόχος του αλγορίθμου είναι η επαγωγή μιας συνάρτησης $h: X \rightarrow Y$, η οποία ονομάζεται ταξινομητής (classifier) και ελαχιστοποιεί την πιθανότητα εσφαλμένης πρόβλεψης σε νέα, άγνωστα δεδομένα που προέρχονται από την ίδια, άγνωστη κατανομή με τα δεδομένα εκπαίδευσης. Η ικανότητα αυτή, γνωστή ως γενίκευση (generalization), διακρίνει τη μάθηση από την απλή απομνημόνευση: ένα μοντέλο που αποδίδει άριστα στα δεδομένα εκπαίδευσης αλλά αποτυγχάνει σε νέα δεδομένα είναι πρακτικά άχρηστο [17], [2].

Τα χαρακτηριστικά εισόδου μπορεί να είναι αριθμητικά (συνεχή ή διακριτά) ή ονομαστικά (κατηγορικά). Τα σύνολα δεδομένων της παρούσας μελέτης καλύπτουν προβλήματα δυαδικά και πολυκλασικά, με πλήθος κλάσεων από 2 έως 26, με κυρίως αριθμητικά χαρακτηριστικά, και με πλήθος δειγμάτων που εκτείνεται από μερικές εκατοντάδες έως δεκάδες χιλιάδες, στοιχεία που, όπως θα φανεί στο Κεφάλαιο 5, επηρεάζουν καθοριστικά τη συμπεριφορά των συγκρινόμενων αλγορίθμων.

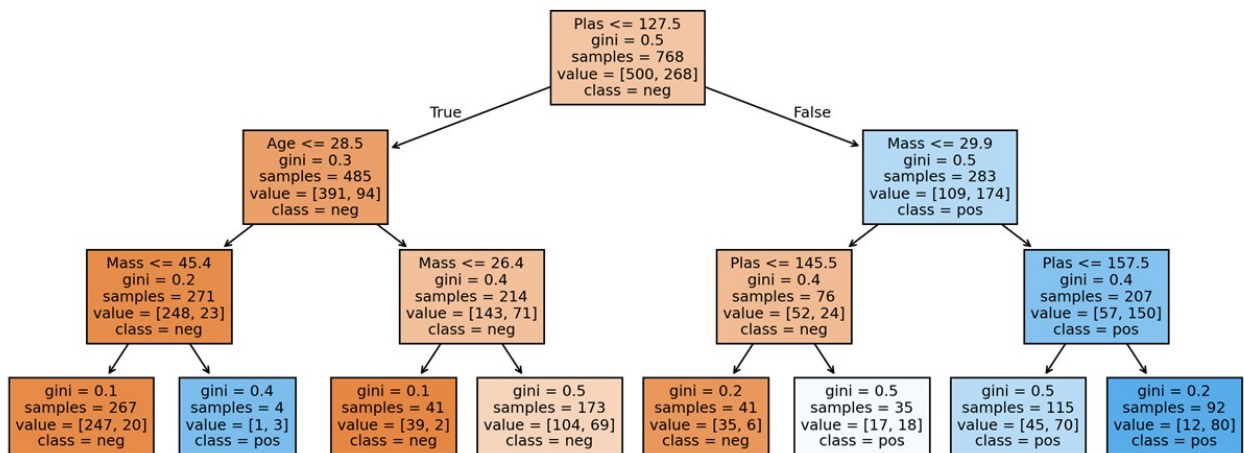
2.3 Δέντρα αποφάσεων

Τα δέντρα αποφάσεων επιλέχθηκαν ως αφετηρία της θεωρητικής παρουσίασης επειδή αποτελούν τον βασικό ταξινομητή (base learner) όλων των ensemble αλγορίθμων που εξετάζονται στην εργασία. Στις υποενότητες που ακολουθούν περιγράφονται η δομή και η λειτουργία τους, τα κριτήρια επιλογής διαχωρισμών, οι ιστορικά σημαντικότεροι αλγόριθμοι επαγωγής και, τέλος, τα φαινόμενα της

υπερπροσαρμογής και της αστάθειας, τα οποία εξηγούν γιατί τα δέντρα ωφελούνται τόσο πολύ από τη συμμετοχή τους σε σύνολα μοντέλων.

2.3.1 Δομή και λειτουργία

Ένα δέντρο απόφασης (decision tree) είναι ένα μοντέλο με δομή δέντρου, στο οποίο κάθε εσωτερικός κόμβος αντιστοιχεί σε έναν έλεγχο επί ενός χαρακτηριστικού (π.χ. « $\text{Plas} \leq 127,5$ »), κάθε ακμή στο αποτέλεσμα του ελέγχου και κάθε φύλλο σε μια πρόβλεψη κλάσης [18]. Η ταξινόμηση ενός νέου δείγματος γίνεται με διάσχιση του δέντρου από τη ρίζα προς κάποιο φύλλο, ακολουθώντας σε κάθε κόμβο την ακμή που αντιστοιχεί στην τιμή του εξεταζόμενου χαρακτηριστικού. Η κατασκευή του δέντρου γίνεται αναδρομικά, με στρατηγική «διαίρει και βασίλευε»: σε κάθε βήμα επιλέγεται ο διαχωρισμός που διαμερίζει «καλύτερα» το τρέχον υποσύνολο δεδομένων σε ομοιογενή ως προς την κλάση τμήματα, και η διαδικασία επαναλαμβάνεται στα τμήματα αυτά μέχρι την ικανοποίηση κάποιου κριτηρίου τερματισμού. Το Σχήμα 2.1 απεικονίζει ένα δέντρο βάθους 3 που επάγεται από το σύνολο δεδομένων *pima* της παρούσας μελέτης, στο οποίο ζητείται η διάγνωση διαβήτη από κλινικές μετρήσεις: είναι χαρακτηριστική η αναγνωσιμότητα του μοντέλου, καθώς κάθε μονοπάτι από τη ρίζα σε φύλλο συνιστά έναν ερμηνεύσιμο κανόνα.



Σχήμα Θεωρητικό Υπόβαθρο.1: Δέντρο απόφασης βάθους 3 για το σύνολο *pima*.

Σε κάθε εσωτερικό κόμβο του δέντρου αναγράφονται ο έλεγχος διαχωρισμού, ο δείκτης Gini, το πλήθος των δειγμάτων και η κατανομή τους στις κλάσεις, ενώ κάθε φύλλο αποδίδει την τελική πρόβλεψη (διάγνωση διαβήτη ή μη).

2.3.2 Η αναδρομική διαδικασία κατασκευής

Η επαγωγή του δέντρου από τα δεδομένα ακολουθεί την άπληστη (greedy), αναδρομική στρατηγική που σκιαγραφείται στον Αλγόριθμο 2.1. Σε κάθε κόμβο εξετάζονται όλοι οι εφικτοί διαχωρισμοί — για αριθμητικά χαρακτηριστικά, όλα τα ενδιάμεσα κατώφλια μεταξύ διαδοχικών διατεταγμένων τιμών — και επιλέγεται εκείνος που μεγιστοποιεί τη μείωση της μη καθαρότητας. Η αναδρομή τερματίζεται όταν ο κόμβος καταστεί αμιγής, όταν εξαντληθούν τα δείγματα ή τα χαρακτηριστικά, ή όταν ικανοποιηθεί κάποιο κριτήριο πρόωρου τερματισμού, όπως μέγιστο βάθος ή ελάχιστο πλήθος δειγμάτων ανά φύλλο. Η υπολογιστική πολυπλοκότητα της κατασκευής είναι της τάξης $O(p \cdot N \cdot \log N)$ για N δείγματα και p χαρακτηριστικά υπό ευνοϊκές συνθήκες — παρατήρηση κρίσιμη για το κόστος των μεθόδων συνόλου, που κατασκευάζουν δεκάδες ή εκατοντάδες δέντρα.

Κατασκευή Δέντρου(D):

```

αν όλα τα δείγματα του D ανήκουν στην ίδια κλάση ή πληρούται
κριτήριο τερματισμού:
    επέστρεψε φύλλο με την πλειοψηφούσα κλάση του D
για κάθε χαρακτηριστικό f και υποψήφιο κατώφλι t:
    υπολόγισε τη μείωση μη καθαρότητας του διαχωρισμού (f, t)
    επέλεξε τον βέλτιστο διαχωρισμό (f*, t*)
    διαμέρισε το D σε D_αριστ = {x: f*(x) ≤ t*} και D_δεξ = {x: f*(x) > t*}
    επέστρεψε κόμβο(f*, t*, ΚατασκευήΔέντρου(D_αριστ),
    ΚατασκευήΔέντρου(D_δεξ))

```

Αλγόριθμος Θεωρητικό Υπόβαθρο.1: Αναδρομική κατασκευή δέντρου απόφασης

Σημειώνεται ότι η άπληστη φύση της διαδικασίας δεν εγγυάται ολικά βέλτιστο δέντρο — η εύρεση του βέλτιστου δέντρου είναι NP-πλήρες πρόβλημα — και αποτελεί έναν από τους «υπολογιστικούς» λόγους του Dietterich για τους οποίους τα σύνολα μοντέλων υπερτερούν, όπως θα αναλυθεί στο Κεφάλαιο 3 [4].

2.3.3 Κριτήρια επιλογής διαχωρισμών

Η ποιότητα ενός υποψήφιου διαχωρισμού αποτιμάται μέσω ενός μέτρου μη καθαρότητας (impurity) των παραγόμενων κόμβων. Τα δύο επικρατέστερα μέτρα είναι η εντροπία και ο δείκτης Gini. Η εντροπία ενός κόμβου t με αναλογίες κλάσεων $p_1, \dots, p_K$ ορίζεται ως $H(t) = -\sum_k p_k \cdot \log_2 p_k$ και μεγιστοποιείται όταν οι κλάσεις είναι ομοιόμορφα αναμεμιγμένες, ενώ μηδενίζεται όταν ο κόμβος είναι αμιγής. Το κέρδος πληροφορίας (information gain) ενός διαχωρισμού ορίζεται ως η μείωση της εντροπίας που αυτός επιφέρει, δηλαδή η διαφορά της σταθμισμένης εντροπίας των θυγατρικών κόμβων από την εντροπία του γονικού [18]. Αντίστοιχα, ο δείκτης Gini ορίζεται ως $Gini(t) = 1 - \sum_k p_k^2$ και ερμηνεύεται ως η πιθανότητα εσφαλμένης ταξινόμησης ενός τυχαία επιλεγμένου δείγματος του κόμβου, αν αυτό ταξινομούταν τυχαία σύμφωνα με την κατανομή των κλάσεων του κόμβου [19]. Στην πράξη τα δύο κριτήρια οδηγούν σε παρόμοια δέντρα, με τον δείκτη Gini να προτιμάται συχνά για τον ελαφρώς μικρότερο υπολογιστικό του φόρτο· είναι και το προεπιλεγμένο κριτήριο των υλοποιήσεων δέντρων του scikit-learn που χρησιμοποιούνται στο πειραματικό μέρος.

Ένα σύντομο αριθμητικό παράδειγμα αποσαφηνίζει τα παραπάνω. Έστω κόμβος με 100 δείγματα, εκ των οποίων 60 της κλάσης A και 40 της κλάσης B: η εντροπία του είναι $H = -0,6 \cdot \log_2 0,6 - 0,4 \cdot \log_2 0,4 \approx 0,971$ bits και ο δείκτης Gini ίσος με $1 - (0,6^2 + 0,4^2) = 0,48$. Αν ένας υποψήφιος διαχωρισμός παράγει αριστερό παιδί με 50 δείγματα (45 A, 5 B) και δεξί με 50 (15 A, 35 B), η σταθμισμένη εντροπία των παιδιών είναι $0,5 \cdot 0,469 + 0,5 \cdot 0,881 \approx 0,675$, άρα το κέρδος πληροφορίας του διαχωρισμού είναι $0,971 - 0,675 \approx 0,296$ bits· αντίστοιχα, ο σταθμισμένος δείκτης Gini μειώνεται από 0,48 σε $0,5 \cdot 0,18 + 0,5 \cdot 0,42 = 0,30$. Μεταξύ όλων των υποψήφιων διαχωρισμών επιλέγεται εκείνος με τη μεγαλύτερη τέτοια μείωση.

2.3.4 Οι βασικοί αλγόριθμοι επαγωγής: ID3, C4.5 και CART

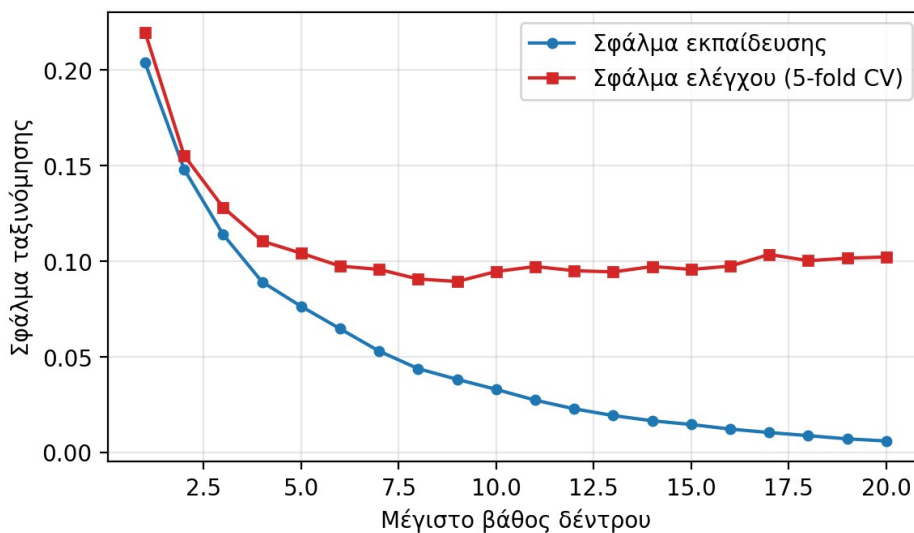
Ο ID3 του Quinlan [18] υπήρξε ο πρώτος ευρέως διαδεδομένος αλγόριθμος επαγωγής δέντρων: χειρίζεται ονομαστικά χαρακτηριστικά και επιλέγει διαχωρισμούς με βάση το κέρδος πληροφορίας. Ο διάδοχός του, C4.5 [20], ήρε τους βασικούς περιορισμούς του ID3, υποστηρίζοντας συνεχή χαρακτηριστικά μέσω δυαδικών κατωφλίων, ελλειπείς τιμές, καθώς και εκ των υστέρων κλάδεμα (post-pruning)· επιπλέον, αντικατέστησε το κέρδος πληροφορίας με τον λόγο κέρδους (gain ratio), που αμβλύνει τη συστηματική προτίμηση προς χαρακτηριστικά με πολλές διακριτές τιμές. Παράλληλα, ο CART των Breiman, Friedman, Olshen και Stone [19] εισήγαγε αυστηρά δυαδικά δέντρα με κριτήριο τον δείκτη Gini, υποστήριξη προβλημάτων παλινδρόμησης (regression trees) και κλάδεμα ελάχιστου κόστους-πολυπλοκότητας. Η υλοποίηση δέντρων του scikit-learn, την οποία χρησιμοποιούν ως δομικό

στοιχείο όλοι οι αλγόριθμοι της παρούσας μελέτης, είναι μια βελτιστοποιημένη παραλλαγή του CART [15].

2.3.5 Υπερπροσαρμογή και αστάθεια

Ένα δέντρο που αναπτύσσεται χωρίς περιορισμό μπορεί να διαμερίσει τον χώρο των χαρακτηριστικών μέχρι κάθε φύλλο να περιέχει ελάχιστα ή και μεμονωμένα δείγματα, απομονώνοντας έτσι και τον θόρυβο του συνόλου εκπαίδευσης. Το φαινόμενο αυτό, γνωστό ως υπερπροσαρμογή (overfitting), αποτυπώνεται χαρακτηριστικά στο Σχήμα 2.2, το οποίο προέκυψε από το σύνολο δεδομένων spambase της μελέτης: καθώς αυξάνεται το μέγιστο επιτρεπόμενο βάθος, το σφάλμα εκπαίδευσης μειώνεται μονότονα και τείνει στο μηδέν, ενώ το σφάλμα ελέγχου, εκτιμώμενο με πενταπλή διασταυρούμενη επικύρωση, σταθεροποιείται και στη συνέχεια αρχίζει να αυξάνεται ελαφρά — το μοντέλο πλέον μαθαίνει ιδιαιτερότητες του δείγματος και όχι τη γενική δομή του προβλήματος [17]. Οι κλασικές θεραπείες είναι ο πρόωρος τερματισμός της ανάπτυξης (pre-pruning) και το εκ των υστέρων κλάδεμα [20].

Εξίσου σημαντική για την παρούσα εργασία είναι η αστάθεια (instability) των δέντρων: επειδή η επιλογή κάθε διαχωρισμού εξαρτάται από τους προηγούμενους, μια μικρή διαταραχή των δεδομένων εκπαίδευσης — για παράδειγμα η αφαίρεση λίγων δειγμάτων — μπορεί να αλλάξει τον πρώτο διαχωρισμό και, αλυσιδωτά, ολόκληρη τη δομή του δέντρου. Ο Breiman [3] κατέταξε τα δέντρα στους «ασταθείς» αλγορίθμους μάθησης, μαζί με τα νευρωνικά δίκτυα, σε αντιδιαστολή με σταθερές μεθόδους όπως ο k-πλησιέστερος γείτονας. Όπως θα αναλυθεί στο Κεφάλαιο 3, η αστάθεια αυτή, αν και ανεπιθύμητη σε ένα μεμονωμένο μοντέλο, είναι ακριβώς η ιδιότητα που καθιστά τα δέντρα ιδανικούς βασικούς ταξινομητές για μεθόδους συνόλου: η υψηλή διακύμανσή τους μπορεί να εξουδετερωθεί μέσω μέσου όρου πολλών ανεξάρτητων εκδοχών τους.



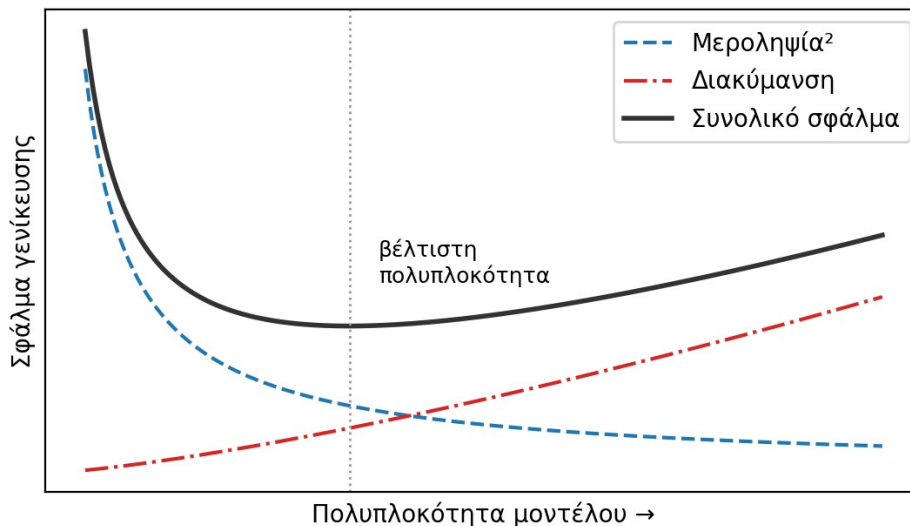
Σχήμα Θεωρητικό Υπόβαθρο.2: Σφάλματα εκπαίδευσης και ελέγχου συναρτήσεως του μέγιστου βάθους (spambase).

2.4 Το δίλημμα μεροληψίας-διακύμανσης

Το σφάλμα γενίκευσης ενός μοντέλου μάθησης μπορεί να αναλυθεί σε τρεις συνιστώσες: τη μεροληψία (bias), τη διακύμανση (variance) και το μη αναγώγιμο σφάλμα (irreducible error) [21]. Η μεροληψία εκφράζει το συστηματικό σφάλμα του μοντέλου, δηλαδή την αδυναμία της κλάσης υποθέσεων να

προσεγγίσει την πραγματική σχέση εισόδου-εξόδου· ένα ρηχό δέντρο-κολόβωμα (stump), για παράδειγμα, αδυνατεί εγγενώς να αναπαραστήσει σύνθετα όρια αποφάσεων. Η διακύμανση εκφράζει την ευαισθησία του μοντέλου στο συγκεκριμένο δείγμα εκπαίδευσης: ένα πλήρως ανεπτυγμένο δέντρο αλλάζει δραστικά από δείγμα σε δείγμα, άρα έχει υψηλή διακύμανση. Το μη αναγώγιμο σφάλμα, τέλος, οφείλεται στον εγγενή θόρυβο του προβλήματος και δεν μπορεί να εξαλειφθεί από κανένα μοντέλο. Αν και η αυστηρή αθροιστική ανάλυση ισχύει για το τετραγωνικό σφάλμα παλινδρόμησης, αντίστοιχες αναλύσεις έχουν προταθεί και για το σφάλμα ταξινόμησης [22].

Όπως απεικονίζεται σχηματικά στο Σχήμα 2.3, η αύξηση της πολυπλοκότητας του μοντέλου μειώνει τη μεροληψία αλλά αυξάνει τη διακύμανση, με το συνολικό σφάλμα να ελαχιστοποιείται σε ένα ενδιάμεσο σημείο ισορροπίας — το περίφημο δίλημμα μεροληψίας-διακύμανσης (bias-variance trade-off) [23]. Η ανάλυση αυτή προσφέρει το εννοιολογικό πλαίσιο για την κατανόηση των δύο οικογενειών μεθόδων συνόλου της εργασίας: το Bagging επιτίθεται στη συνιστώσα της διακύμανσης, υπολογίζοντας τον μέσο όρο πολλών μοντέλων υψηλής διακύμανσης και χαμηλής μεροληψίας (πλήρη δέντρα), ενώ το Boosting επιτίθεται πρωτίστως στη μεροληψία, συνθέτοντας διαδοχικά μοντέλα χαμηλής διακύμανσης και υψηλής μεροληψίας (ρηχά δέντρα) σε έναν ολοένα πλουσιότερο συνδυασμό [3], [9].



Σχήμα Θεωρητικό Υπόβαθρο.3: Το δίλημμα μεροληψίας-διακύμανσης.

2.5 Αξιολόγηση ταξινομητών

Η αξιόπιστη σύγκριση αλγορίθμων κατηγοριοποίησης προϋποθέτει σαφώς ορισμένες μετρικές απόδοσης και μια πειραματική διαδικασία που περιορίζει τη στατιστική αβεβαιότητα των εκτιμήσεων. Στις υποενότητες που ακολουθούν παρουσιάζονται οι μετρικές που χρησιμοποιούνται στην εργασία, η μέθοδος της διασταυρούμενης επικύρωσης και οι μετρικές υπολογιστικού κόστους.

2.5.1 Πίνακας σύγχυσης και μετρικές ποιότητας

Θεμέλιο της αξιολόγησης αποτελεί ο πίνακας σύγχυσης (confusion matrix), ο οποίος καταγράφει το πλήθος των δειγμάτων κάθε πραγματικής κλάσης που ταξινομήθηκαν σε κάθε προβλεπόμενη κλάση. Στη δυαδική περίπτωση, με θετική και αρνητική κλάση, τα στοιχεία του συνοψίζονται στα αληθώς θετικά (TP), αληθώς αρνητικά (TN), ψευδώς θετικά (FP) και ψευδώς αρνητικά (FN). Από αυτά ορίζονται οι βασικές μετρικές [24]: η ορθότητα $\text{accuracy} = (TP+TN)/(TP+TN+FP+FN)$, δηλαδή το ποσοστό συνολικά ορθών προβλέψεων· η ακρίβεια $\text{precision} = TP/(TP+FP)$, δηλαδή το ποσοστό των

προβλέψεων θετικής κλάσης που ήταν πράγματι θετικές· η ανάκληση $\text{recall} = \text{TP}/(\text{TP}+\text{FN})$, δηλαδή το ποσοστό των πραγματικά θετικών δειγμάτων που εντοπίστηκαν· και ο αρμονικός μέσος τους $F1 = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$, ο οποίος ισορροπεί τις δύο προηγούμενες και είναι πιο αντιπροσωπευτικός από την ορθότητα σε προβλήματα με ανισοκατανομημένες κλάσεις [25].

Σε πολυκλασικά προβλήματα, οι μετρικές precision, recall και F1 υπολογίζονται ανά κλάση και στη συνέχεια συναθροίζονται. Στη μακρο-συνάθροιση (macro-averaging) υπολογίζεται ο απλός μέσος όρος της μετρικής στις κλάσεις, ώστε κάθε κλάση να συμμετέχει ισότιμα ανεξαρτήτως πληθυσμού, ενώ στη μικρο-συνάθροιση (micro-averaging) συναθροίζονται πρώτα τα TP, FP, FN όλων των κλάσεων, οπότε κυριαρχούν οι πολυπληθείς κλάσεις [24]. Στην παρούσα εργασία υιοθετείται η μακρο-συνάθροιση, ακριβώς επειδή αρκετά από τα εξεταζόμενα σύνολα δεδομένων (π.χ. yeast, shuttle) εμφανίζουν έντονη ανισοκατανομή κλάσεων και είναι επιθυμητό η απόδοση στις σπάνιες κλάσεις να μη συγκαλύπτεται.

Για την εποπτεία των μετρικών, ο Πίνακας 2.1 παραθέτει ένα υποθετικό παράδειγμα δυαδικού πίνακα σύγχυσης για πρόβλημα ιατρικής διάγνωσης 1.000 ασθενών, με 100 πραγματικά θετικά περιστατικά. Ο ταξινομητής του παραδείγματος επιτυγχάνει $\text{accuracy} = (80+855)/1000 = 93,5\%$, τιμή που ηχεί ικανοποιητική· ωστόσο $\text{precision} = 80/(80+45) = 64\%$, $\text{recall} = 80/(80+20) = 80\%$ και $F1 \approx 71\%$, αποκαλύπτοντας ότι σχεδόν ένας στους τρεις συναγερμούς του είναι εσφαλμένος. Το παράδειγμα καθιστά απτό γιατί σε ανισοκατανομημένα προβλήματα η ορθότητα από μόνη της είναι παραπλανητικό μέτρο — ένας ταξινομητής που θα προέβλεπε πάντοτε την αρνητική κλάση θα επιτύγχανε $\text{accuracy} 90\%$ με μηδενική διαγνωστική αξία.

Πίνακας Θεωρητικό Υπόβαθρο.1: Παράδειγμα πίνακα σύγχυσης δυαδικού προβλήματος ($N = 1.000$, 100 θετικά).

	Πρόβλεψη: Θετικό	Πρόβλεψη: Αρνητικό	Σύνολο
Πραγματικά Θετικό	TP = 80	FN = 20	100
Πραγματικά Αρνητικό	FP = 45	TN = 855	900
Σύνολο	125	875	1.000

2.5.2 Διασταυρούμενη επικύρωση

Η εκτίμηση της απόδοσης στο ίδιο το σύνολο εκπαίδευσης είναι συστηματικά υπεραισιόδοξη· απαιτείται αποτίμηση σε δεδομένα που δεν συμμετείχαν στην εκπαίδευση. Έναντι του απλού διαχωρισμού train/test, η k-πλή διασταυρούμενη επικύρωση (k-fold cross validation) αξιοποιεί πληρέστερα τα διαθέσιμα δεδομένα: το σύνολο διαμερίζεται σε k ισομεγέθη τμήματα (folds) και εκτελούνται k επαναλήψεις, σε καθεμία από τις οποίες ένα διαφορετικό τμήμα χρησιμοποιείται για έλεγχο και τα υπόλοιπα k-1 για εκπαίδευση· η τελική εκτίμηση είναι ο μέσος όρος των k μετρήσεων. Ο Kohavi [26] έδειξε εμπειρικά ότι η στρωματοποιημένη (stratified) εκδοχή, στην οποία κάθε fold διατηρεί τις αναλογίες των κλάσεων του πλήρους συνόλου, παρέχει εκτιμήσεις με μικρότερη μεροληψία και διακύμανση, και συνέστησε τη στρωματοποιημένη δεκαπλή διασταυρούμενη επικύρωση ως πρότυπη πρακτική — τη σύσταση αυτή ακολουθεί και το πειραματικό πρωτόκολλο της παρούσας εργασίας.

2.5.3 Μετρικές υπολογιστικού κόστους

Πέραν της ποιότητας των προβλέψεων, κρίσιμη διάσταση της σύγκρισης αποτελεί το υπολογιστικό κόστος, το οποίο αναλύεται σε χρόνο εκπαίδευσης (training time), δηλαδή τον χρόνο κατασκευής του μοντέλου από τα δεδομένα εκπαίδευσης, και χρόνο ταξινόμησης (classification ή inference time), δηλαδή τον χρόνο παραγωγής προβλέψεων για το σύνολο ελέγχου. Η διάκριση είναι ουσιώδης, διότι οι δύο χρόνοι έχουν διαφορετική πρακτική βαρύτητα ανάλογα με την εφαρμογή: σε συστήματα που επανεκπαιδεύονται συχνά κυριαρχεί ο πρώτος, ενώ σε συστήματα πραγματικού χρόνου ο δεύτερος. Στην παρούσα εργασία οι χρόνοι μετρώνται ως καθαρός χρόνος επεξεργαστή της διεργασίας (CPU process time) και όχι ως χρόνος ρολογιού (wall-clock), ώστε οι μετρήσεις να μην αλλοιώνονται από τον διαμοιρασμό του επεξεργαστή με άλλες διεργασίες του συστήματος.

2.6 Επίλογος

Στο κεφάλαιο αυτό ορίστηκε το πρόβλημα της κατηγοριοποίησης και παρουσιάστηκαν τα δέντρα αποφάσεων, με έμφαση στα κριτήρια διαχωρισμού, στην υπερπροσαρμογή και στην αστάθειά τους, καθώς και η ανάλυση του σφάλματος γενίκευσης σε μεροληψία και διακύμανση. Περιγράφηκαν, επίσης, οι μετρικές ποιότητας και κόστους και η στρωματοποιημένη διασταυρούμενη επικύρωση που συγκροτούν το πλαίσιο αξιολόγησης της εργασίας. Με αυτά τα εφόδια, το επόμενο κεφάλαιο εισέρχεται στον πυρήνα της βιβλιογραφικής ανασκόπησης: τις μεθόδους συνόλου ταξινομητών.

Κεφάλαιο 3ο: Μέθοδοι Συνόλου Ταξινομητών: Ανασκόπηση Βιβλιογραφίας

Το κεφάλαιο αυτό ανασκοπεί τη βιβλιογραφία των μεθόδων συνόλου ταξινομητών, οι οποίες αποτελούν το κεντρικό αντικείμενο της εργασίας. Η ανασκόπηση ξεκινά από το θεμελιώδες ερώτημα γιατί, και υπό ποιες προϋποθέσεις, ο συνδυασμός πολλών μοντέλων μπορεί να υπερτερεί του καλύτερου μεμονωμένου μέλους του, και προχωρά στην αναλυτική παρουσίαση των δύο κυρίαρχων οικογενειών: του Bagging, με τους απογόνους του Random Forests και Extra-Trees, και του Boosting, από τον AdaBoost έως τις σύγχρονες ιστογραμμικές υλοποιήσεις gradient boosting. Μέσα από τη θεωρητική αντιπαραβολή των δύο προσεγγίσεων και την κριτική επισκόπηση των σημαντικότερων εμπειρικών συγκριτικών μελετών, αναδεικνύεται το ερευνητικό κενό που επιχειρεί να καλύψει η παρούσα πειραματική μελέτη.

3.1 Εισαγωγή

Το κεφάλαιο αυτό αποτελεί τον πυρήνα της βιβλιογραφικής ανασκόπησης της εργασίας. Αρχικά παρουσιάζεται η γενική θεωρία των μεθόδων συνόλου: γιατί και υπό ποιες προϋποθέσεις ο συνδυασμός πολλών ταξινομητών υπερτερεί του καλύτερου μεμονωμένου μέλους του. Στη συνέχεια αναλύονται διεξοδικά οι δύο μεγάλες οικογένειες που εξετάζονται πειραματικά: η οικογένεια του Bagging, με τους αλγόριθμους Bagging, Random Forest και Extra-Trees, και η οικογένεια του Boosting, με τους AdaBoost, Gradient Boosting και τις σύγχρονες ιστογραμμικές του υλοποιήσεις. Το κεφάλαιο ολοκληρώνεται με τη θεωρητική αντιπαραβολή των δύο οικογενειών και με την επισκόπηση των σημαντικότερων εμπειρικών συγκριτικών μελετών της βιβλιογραφίας, από την οποία προκύπτει και το κενό που επιχειρεί να καλύψει η παρούσα εργασία.

3.2 Η ιδέα των μεθόδων συνόλου

Μια μέθοδος συνόλου (ensemble method) κατασκευάζει ένα πλήθος βασικών ταξινομητών $h_1, h_2, \dots, h_M$ και συνδυάζει τις προβλέψεις τους, τυπικά μέσω (σταθμισμένης) ψηφοφορίας, ώστε να παραγάγει την τελική απόφαση. Οι Hansen και Salamon [5] έδειξαν από νωρίς ότι, εφόσον τα μέλη του συνόλου έχουν πιθανότητα σφάλματος μικρότερη του 0,5 και σφάλλουν ανεξάρτητα το ένα από το άλλο, η πιθανότητα σφάλματος της πλειοψηφικής ψήφου φθίνει καθώς αυξάνεται το πλήθος των μελών — μια άμεση συνέπεια του νόμου των μεγάλων αριθμών. Η πράξη βέβαια απέχει από την εξιδανίκευση της πλήρους ανεξαρτησίας: τα μέλη εκπαιδεύονται στα ίδια ή σε επικαλυπτόμενα δεδομένα, συνεπώς τα σφάλματά τους συσχετίζονται. Το ζητούμενο κάθε μεθόδου συνόλου είναι επομένως η ποικιλομορφία (diversity): η κατασκευή μελών επαρκώς ακριβών αλλά και επαρκώς διαφορετικών, ώστε τα σφάλματά τους να αλληλοαναιρούνται [27], [28].

Ο Dietterich [4] συνόψισε σε τρεις λόγους το γιατί ένα σύνολο μπορεί να υπερτερεί του βέλτιστου μεμονωμένου μοντέλου. Ο στατιστικός λόγος: με περιορισμένα δεδομένα, πολλές διαφορετικές υποθέσεις εμφανίζουν παρόμοια απόδοση στο σύνολο εκπαίδευσης και ο μέσος όρος τους είναι ασφαλέστερος από την αυθαίρετη επιλογή μίας. Ο υπολογιστικός λόγος: οι άπληστοι αλγόριθμοι μάθησης, όπως η επαγωγή δέντρων, εγκλωβίζονται σε τοπικά βέλτιστα, και η εκκίνηση από πολλά διαφορετικά σημεία παρέχει καλύτερη προσέγγιση του ολικού βελτίστου. Ο αναπαραστασιακός λόγος: η πραγματική συνάρτηση-στόχος ενδέχεται να μην ανήκει καν στην κλάση υποθέσεων ενός μεμονωμένου μοντέλου, ενώ σταθμισμένοι συνδυασμοί πολλών μοντέλων διευρύνουν τον

αναπαραστάσιμο χώρο. Οι μέθοδοι συνόλου διακρίνονται συνήθως σε ομοιογενείς, όπου όλα τα μέλη προέρχονται από τον ίδιο αλγόριθμο μάθησης και η ποικιλομορφία εισάγεται μέσω τυχαιότητας στα δεδομένα ή στη διαδικασία, και σε ετερογενείς, όπου συνδυάζονται διαφορετικοί αλγόριθμοι, όπως στο stacking [29]. Οι έξι αλγόριθμοι της παρούσας μελέτης είναι όλοι ομοιογενείς, με βασικό ταξινομητή το δέντρο απόφασης, διαφέρουν όμως θεμελιωδώς στον τρόπο κατασκευής των μελών: παράλληλα και ανεξάρτητα στην οικογένεια του Bagging, σειριακά και προσαρμοστικά στην οικογένεια του Boosting [12], [30].

3.2.1 Τρόποι συνδυασμού των προβλέψεων

Ανεξάρτητα από τον τρόπο κατασκευής των μελών, κάθε μέθοδος συνόλου χρειάζεται έναν κανόνα συγχώνευσης των επιμέρους προβλέψεων. Ο απλούστερος είναι η σκληρή ψηφοφορία (hard voting): κάθε μέλος ψηφίζει μία κλάση και επικρατεί η πλειοψηφούσα. Όταν τα μέλη παρέχουν εκτιμήσεις πιθανοτήτων, προτιμάται η μαλακή ψηφοφορία (soft voting), δηλαδή ο μέσος όρος των κατανομών πιθανότητας και η επιλογή της κλάσης με τη μέγιστη μέση πιθανότητα, η οποία αξιοποιεί και την πεποίθηση κάθε μέλους, όχι μόνο την τελική του ψήφο — τα Random Forests του scikit-learn, για παράδειγμα, συνδυάζουν τα δέντρα τους ακριβώς έτσι [15]. Στο boosting, τα μέλη δεν είναι ισότιμα: οι ψήφοι σταθμίζονται με συντελεστές που αντανακλούν την ποιότητα κάθε μέλους (όπως τα *am* του AdaBoost) ή, στο gradient boosting, οι έξοδοι των μελών αθροίζονται ως διαδοχικές διορθώσεις μιας κοινής συνάρτησης πρόβλεψης.

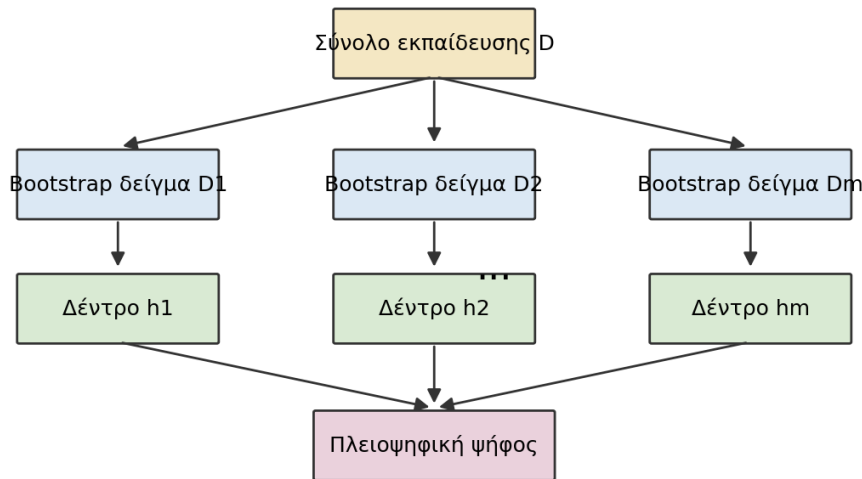
Πέρα από τους σταθερούς αυτούς κανόνες, ο συνδυασμός μπορεί και ο ίδιος να αποτελέσει αντικείμενο μάθησης: στο stacking (stacked generalization) του Wolpert [29], οι προβλέψεις ετερογενών βασικών μοντέλων τροφοδοτούν έναν μετα-ταξινομητή που μαθαίνει πώς να τις συνθέτει βέλτιστα. Οι ετερογενείς αυτές αρχιτεκτονικές, αν και ισχυρές, εκφεύγουν του πεδίου της παρούσας μελέτης, η οποία εστιάζει στις ομοιογενείς μεθόδους δέντρων — τη ραχοκοκαλιά της σύγχρονης πρακτικής σε πινακοποιημένα δεδομένα.

3.3 Bagging

Η ενότητα αυτή παρουσιάζει την πρώτη ιστορικά και εννοιολογικά απλούστερη οικογένεια: περιγράφεται πρώτα ο μηχανισμός bootstrap και ο αλγόριθμος Bagging και στη συνέχεια εξηγείται γιατί η μέθοδος μειώνει τη διακύμανση του σφάλματος.

3.3.1 Bootstrap δειγματοληψία και ο αλγόριθμος

Το Bagging (Bootstrap AGGREGatING), που προτάθηκε από τον Breiman το 1996 [3], εισάγει ποικιλομορφία μέσω της στατιστικής τεχνικής bootstrap: από το σύνολο εκπαίδευσης D με N δείγματα δημιουργούνται M «bootstrap δείγματα» $D_1, \dots, D_M$, καθένα από τα οποία προκύπτει με N τυχαίες επιλογές από το D με επανατοποθέτηση. Κάθε D_m περιέχει κατά μέσο όρο το 63,2% των διακριτών δειγμάτων του D (καθώς η πιθανότητα ένα δείγμα να μην επιλεγεί ποτέ είναι $(1-1/N)^N \approx e^{-1} \approx 0,368$), με ορισμένα δείγματα να εμφανίζονται πολλαπλές φορές. Σε κάθε D_m εκπαιδεύεται ανεξάρτητα ένας βασικός ταξινομητής h_m και η τελική πρόβλεψη προκύπτει με πλειοψηφική ψήφο, όπως απεικονίζεται στο Σχήμα 3.1. Παράπλευρο όφελος της διαδικασίας είναι η out-of-bag (OOB) εκτίμηση: κάθε μέλος μπορεί να αξιολογηθεί στα δείγματα που δεν συμμετείχαν στο δικό του bootstrap δείγμα, παρέχοντας αμερόληπτη εκτίμηση του σφάλματος γενίκευσης χωρίς χωριστό σύνολο επικύρωσης [31].



Σχήμα Μέθοδοι Συνόλου Ταξινομητών: Ανασκόπηση Βιβλιογραφίας.4: Σχηματική αναπαράσταση του Bagging.

3.3.2 Γιατί το Bagging μειώνει τη διακύμανση

Η θεωρητική εξήγηση της επιτυχίας του Bagging ανάγεται στην ιδιότητα του μέσου όρου: αν $Z_1, \dots, Z_M$ είναι τυχαίες μεταβλητές με κοινή διακύμανση σ^2 , ο μέσος όρος τους έχει διακύμανση σ^2/M όταν είναι ανεξάρτητες, και $\sigma^2 \cdot \rho + \sigma^2 \cdot (1-\rho)/M$ όταν είναι ανά δύο συσχετισμένες με συντελεστή ρ [23]. Εφαρμοζόμενο σε προβλέψεις μοντέλων, το αποτέλεσμα αυτό δείχνει ότι ο μέσος όρος πολλών εκδοχών ενός ασταθούς μοντέλου διατηρεί τη (χαμηλή) μεροληψία του, ενώ συρρικνώνει δραστικά τη συνιστώσα της διακύμανσης — τόσο περισσότερο όσο λιγότερο συσχετισμένα είναι τα μέλη. Ο ίδιος ο Breiman [3] τόνισε ότι το όφελος προϋποθέτει ασταθή βασικό ταξινομητή: εφαρμογή του Bagging σε σταθερές μεθόδους αφήνει την απόδοση ουσιαστικά αμετάβλητη ή και τη χειροτερεύει ελαφρά. Για τον λόγο αυτό, ιδανικά μέλη ενός Bagging συνόλου είναι τα πλήρως ανεπτυγμένα, ακλάδευτα δέντρα: χαμηλή μεροληψία, υψηλή διακύμανση, την οποία αναλαμβάνει να εξαλείψει ο μέσος όρος. Στην υλοποίηση BaggingClassifier του scikit-learn, που χρησιμοποιείται στο πειραματικό μέρος, βασικός ταξινομητής είναι ακριβώς ένα ακλάδευτο δέντρο CART, με προεπιλεγμένο πλήθος μελών $M = 10$ [15].

3.4 Random Forests

Τα Random Forests (τυχαία δάση), που εισήγαγε ο Breiman το 2001 [6], αποτελούν τη φυσική εξέλιξη του Bagging και ίσως τον δημοφιλέστερο αλγόριθμο μάθησης της τελευταίας εικοσιπενταετίας για πινακοποιημένα δεδομένα. Η καινοτομία τους έγκειται στην πρόσθεση ενός δεύτερου επιπέδου τυχειότητας: πέραν του bootstrap δείγματος ανά δέντρο, σε κάθε κόμβο κάθε δέντρου εξετάζεται για διαχωρισμό μόνο ένα τυχαίο υποσύνολο των χαρακτηριστικών, μεγέθους τυπικά $\sqrt{p}$ επί συνόλου p χαρακτηριστικών — ιδέα που προεικονίστηκε στη μέθοδο random subspace του Ho [32]. Η τυχαία επιλογή χαρακτηριστικών αποσυσχετίζει τα δέντρα μεταξύ τους: χωρίς αυτήν, ένα ισχυρό χαρακτηριστικό θα κυριαρχούσε στους πρώτους διαχωρισμούς όλων των δέντρων, καθιστώντας τα όμοια και περιορίζοντας το όφελος του μέσου όρου, σύμφωνα με την ανάλυση της προηγούμενης ενότητας.

Ο Breiman [6] απέδειξε ότι το σφάλμα γενίκευσης του δάσους συγκλίνει καθώς αυξάνεται το πλήθος των δέντρων — συνεπώς η προσθήκη δέντρων δεν προκαλεί υπερπροσαρμογή — και ότι φράσσεται από ποσότητα που εξαρτάται από την ισχύ (strength) των επιμέρους δέντρων και τη μεταξύ τους συσχέτιση (correlation): επιθυμητά είναι δάση ισχυρών και ασυσχέτιστων δέντρων, και η παράμετρος

του πλήθους εξεταζόμενων χαρακτηριστικών ρυθμίζει ακριβώς αυτή την ισορροπία. Τα τυχαία δάση προσφέρουν επιπλέον ενσωματωμένη εκτίμηση της σημαντικότητας των χαρακτηριστικών και OOB εκτίμηση σφάλματος, ενώ είναι εγγενώς παραλληλοποιήσιμα. Εκτενής επισκόπηση της σχετικής θεωρίας δίνεται από τους Biau και Scornet [33]. Η υλοποίηση RandomForestClassifier του scikit-learn χρησιμοποιεί προεπιλεγμένα $M = 100$ ακλάδευτα δέντρα με κριτήριο Gini και $\sqrt{p}$ χαρακτηριστικά ανά κόμβο [15].

3.5 Extremely Randomized Trees (Extra-Trees)

Τα Extremely Randomized Trees των Geurts, Ernst και Wehenkel [7] ωθούν την ιδέα της τυχαιοποίησης ένα βήμα παραπέρα. Διαφοροποιούνται από τα Random Forests σε δύο σημεία: πρώτον, κάθε δέντρο εκπαιδεύεται προεπιλεγμένα σε ολόκληρο το σύνολο εκπαίδευσης αντί σε bootstrap δείγμα· δεύτερον — και σημαντικότερο — για κάθε υπονήφιο χαρακτηριστικό ενός κόμβου δεν αναζητείται το βέλτιστο κατώφλι διαχωρισμού, αλλά επιλέγεται ένα τυχαίο κατώφλι εντός του εύρους τιμών του χαρακτηριστικού, και από τα τυχαία αυτά υπονήφια σημεία επιλέγεται το καλύτερο. Η πλήρης αυτή τυχαιοποίηση των κατωφλίων μειώνει περαιτέρω τη διακύμανση του συνόλου, με αντίτιμο μια ελαφρά αύξηση της μεροληψίας κάθε μέλους, και ταυτόχρονα καταργεί το υπολογιστικά ακριβότερο στάδιο της επαγωγής δέντρου — τη διάταξη των τιμών και την εξαντλητική αναζήτηση βέλτιστου κατωφλίου. Οι συγγραφείς ανέφεραν ακρίβεια συγκρίσιμη ή καλύτερη των Random Forests με αισθητά μικρότερο χρόνο εκπαίδευσης, εύρημα που, όπως θα φανεί στο Κεφάλαιο 5, επιβεβαιώνεται πλήρως και στη δική μας μελέτη.

3.6 Boosting

Η δεύτερη μεγάλη οικογένεια ακολουθεί ριζικά διαφορετική φιλοσοφία: αντί για ανεξάρτητη κατασκευή ισχυρών μοντέλων και μείωση της διακύμανσης, κατασκευάζει σειριακά αδύναμα μοντέλα, καθένα από τα οποία διορθώνει τα σφάλματα των προηγούμενων, μειώνοντας προοδευτικά τη μεροληψία. Στις υποενότητες που ακολουθούν παρουσιάζονται η θεωρητική καταγωγή της ιδέας, ο AdaBoost, η γενίκευσή του σε μηχανές gradient boosting και οι σύγχρονες ιστογραμμικές υλοποιήσεις.

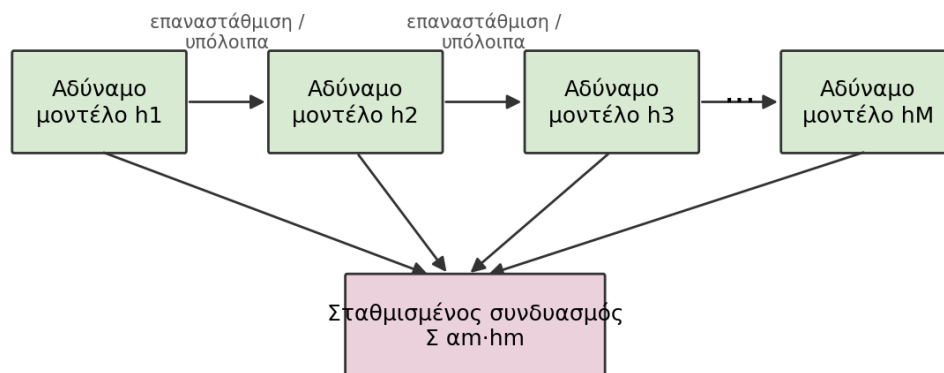
3.6.1 Από τους αδύναμους στους ισχυρούς ταξινομητές

Η καταγωγή του Boosting είναι σπάνια περίπτωση θεωρητικού ερωτήματος που γέννησε πρακτικό αλγόριθμο. Στο πλαίσιο της θεωρίας υπολογιστικής μάθησης PAC, οι Kearns και Valiant διατύπωσαν το ερώτημα αν ένας «αδύναμος» αλγόριθμος μάθησης, ελαφρώς μόνο καλύτερος από την τυχαία πρόβλεψη, μπορεί να «ενισχυθεί» (boosted) σε αυθαίρετα ακριβή. Ο Schapire [13] απάντησε καταφατικά με την πρώτη κατασκευαστική απόδειξη, δείχνοντας ότι η αδύναμη και η ισχυρή μαθησιμότητα είναι ισοδύναμες, και ο Freund [34] βελτίωσε την κατασκευή με ψηφοφορία πλειοψηφίας. Οι πρώτες αυτές κατασκευές είχαν κυρίως θεωρητικό χαρακτήρα· το πεδίο απογειώθηκε όταν οι Freund και Schapire παρουσίασαν έναν πρακτικό, προσαρμοστικό αλγόριθμο ενίσχυσης.

3.6.2 AdaBoost

Ο AdaBoost (Adaptive Boosting) [8] διατηρεί μια κατανομή βαρών επί των δειγμάτων εκπαίδευσης, αρχικά ομοιόμορφη. Σε κάθε γύρο m εκπαιδεύεται ένας αδύναμος ταξινομητής h_m στη σταθμισμένη κατανομή και υπολογίζεται το σταθμισμένο σφάλμα του ϵ_m · ο ταξινομητής λαμβάνει συντελεστή ψήφου $\alpha_m = \frac{1}{2} \cdot \ln((1-\epsilon_m)/\epsilon_m)$, τα βάρη των δειγμάτων που ταξινομήθηκαν εσφαλμένα αυξάνονται και των ορθών μειώνονται, και ο επόμενος γύρος εστιάζει έτσι στις «δύσκολες» περιπτώσεις, όπως απεικονίζεται στο Σχήμα 3.2. Η τελική απόφαση λαμβάνεται με σταθμισμένη ψήφο $\sum \alpha_m \cdot h_m$. Οι

συγγραφείς απέδειξαν ότι το σφάλμα εκπαίδευσης φθίνει εκθετικά ως προς το πλήθος των γύρων, ενώ η μεταγενέστερη ανάλυση των περιθωρίων (margins) [35] εξήγησε το εμπειρικά παρατηρούμενο παράδοξο ότι το σφάλμα γενίκευσης συχνά συνεχίζει να βελτιώνεται και μετά τον μηδενισμό του σφάλματος εκπαίδευσης: οι επιπλέον γύροι αυξάνουν τα περιθώρια ταξινόμησης. Ο αλγόριθμος αναδείχθηκε σε έναν από τους σημαντικότερους της μηχανικής μάθησης — οι δημιουργοί του τιμήθηκαν με το βραβείο Gödel το 2003 — έχει όμως τεκμηριωμένη ευαισθησία στον θόρυβο ετικετών, καθώς επιμένει να αυξάνει τα βάρη θορυβωδών δειγμάτων [36]. Για πολυκλασικά προβλήματα χρησιμοποιείται η επέκταση SAMME [37], την οποία υλοποιεί ο AdaBoostClassifier του scikit-learn με προεπιλεγμένο βασικό ταξινομητή δέντρο-κολόβωμα βάθους 1 και $M = 50$ γύρους — επιλογή που, όπως θα φανεί πειραματικά, περιορίζει σοβαρά την εκφραστική του ισχύ σε σύνθετα πολυκλασικά προβλήματα.



Σχήμα Μέθοδοι Συνόλου Ταξινομητών: Ανασκόπηση Βιβλιογραφίας.5: Σχηματική αναπαράσταση του Boosting.

Για λόγους πληρότητας της παρουσίασης, ο πυρήνας του αλγορίθμου AdaBoost — στη δυαδική του μορφή, από την οποία η πολυκλασική SAMME διαφέρει μόνο στον υπολογισμό των συντελεστών — δίνεται στον Αλγόριθμο 3.1.

```
AdaBoost(D = {(xi, yi)}, M):
    αρχικοποίησε βάρη wi = 1/N για κάθε δείγμα
    για m = 1 έως M:
        εκπαίδευσε αδύναμο ταξινομητή hm στην κατανομή w
        υπολόγισε σταθμισμένο σφάλμα εm = Σ wi · [hm(xi) ≠ yi]
        αν εm ≥ 0,5: σταμάτησε
        αm = ½ · ln((1 - εm)/εm)
        wi ← wi · exp(αm · [hm(xi) ≠ yi]) και κανονικοποίησε τα w
    επέστρεψε H(x) = argmaxy Σm αm · [hm(x) = y]
```

Αλγόριθμος Μέθοδοι Συνόλου Ταξινομητών: Ανασκόπηση Βιβλιογραφίας.2: AdaBoost (δυαδική μορφή)

3.6.3 Μηχανές Gradient Boosting

Ο Friedman [9] γενίκευσε το boosting αναδεικνύοντας μια βαθιά οπτική: η διαδοχική προσθήκη μοντέλων μπορεί να ιδωθεί ως κάθοδος κλίσης (gradient descent) σε συναρτησιακό χώρο. Το σύνολο κατασκευάζεται σταδιακά, $FM(x) = \sum m \cdot h_m(x)$, όπου σε κάθε βήμα το νέο μοντέλο h_m — τυπικά ένα ρηχό δέντρο παλινδρόμησης — προσαρμόζεται στις ψευδο-υπολειπόμενες τιμές (pseudo-residuals), δηλαδή στην αρνητική κλίση της συνάρτησης απώλειας ως προς τις τρέχουσες προβλέψεις. Η αφαίρεση αυτή αποδεσμεύει το boosting από τη συγκεκριμένη εκθετική απώλεια του AdaBoost — ο οποίος προκύπτει ως ειδική περίπτωση [38] — και επιτρέπει αυθαίρετες παραγωγίσιμες συναρτήσεις απώλειας.

όπως η λογιστική για ταξινόμηση. Κρίσιμη πρακτική συμβολή του Friedman είναι η συρρίκνωση (shrinkage): ο ρυθμός μάθησης $\nu < 1$ (τυπικά 0,1) περιορίζει τη συνεισφορά κάθε δέντρου, λειτουργεί ως ομαλοποίηση και βελτιώνει σταθερά τη γενίκευση με αντίτιμο περισσότερους γύρους. Η στοχαστική παραλλαγή [39], με υποδειγματοληψία των δεδομένων ανά γύρο, εισάγει επιπλέον τυχαιότητα που μειώνει τη συσχέτιση των διαδοχικών δέντρων. Σε πολυκλασικά προβλήματα K κλάσεων, η κλασική υλοποίηση εκπαιδεύει K δέντρα ανά γύρο — ένα ανά κλάση, σε σχήμα «έναν εναντίον όλων» — γεγονός που πολλαπλασιάζει το υπολογιστικό κόστος και εξηγεί, όπως θα φανεί στο Κεφάλαιο 5, τη δραματική επιβράδυνση του GradientBoostingClassifier στο πρόβλημα των 26 κλάσεων letter.

Η γενική δομή της μεθόδου, για αυθαίρετη παραγωγίσιμη συνάρτηση απώλειας L , συνοψίζεται στον Αλγόριθμο 3.2, όπου διακρίνονται καθαρά τα τρία συστατικά της: ο υπολογισμός των ψευδο-υπολοίπων ως αρνητικής κλίσης, η προσαρμογή δέντρου παλινδρόμησης σε αυτά και το συντηρητικό βήμα ενημέρωσης με ρυθμό μάθησης ν .

```
GradientBoosting(D, M,  $\nu$ , L):
    F0(x) = argminy Σi L(yi, y)           # αρχική σταθερή πρόβλεψη
    για m = 1 έως M:
        rm = -[∂L(yi, F(xi))/∂F(xi)]F=Fm-1 # ψευδο-υπόλοιπα
        προσάρμοσε δέντρο παλινδρόμησης hm στα {(xi, rm)}
        υπολόγισε βέλτιστα βάρη γ στα φύλλα του hm
        Fm(x) = Fm-1(x) +  $\nu \cdot h_m(x)$          # βήμα με συρρίκνωση  $\nu$ 
    επέστρεψε FM
```

Αλγόριθμος Μέθοδοι Συνόλου Ταξινόμητων: Ανασκόπηση Βιβλιογραφίας.3: Η γενική μορφή του Gradient Boosting

3.6.4 Ιστογραμμικές υλοποιήσεις: XGBoost, LightGBM και HistGradientBoosting

Η ευρεία υιοθέτηση του gradient boosting σε μεγάλης κλίμακας εφαρμογές οφείλεται σε μια γενιά υλοποιήσεων που επιτάχυναν δραστικά την εκπαίδευση. Το XGBoost των Chen και Guestrin [10] εισήγαγε ομαλοποιημένη συνάρτηση-στόχο με όρους ποινής στην πολυπλοκότητα των δέντρων, προσέγγιση δεύτερης τάξης της απώλειας, αποδοτικό χειρισμό αραιών δεδομένων και εκτεταμένες βελτιστοποιήσεις συστήματος. Το LightGBM των Ke et al. [11] καθιέρωσε την ιστογραμμική τεχνική ως πρότυπο: οι συνεχείς τιμές κάθε χαρακτηριστικού διακριτοποιούνται εκ των προτέρων σε μικρό πλήθος κάδων (τυπικά 255), οπότε η αναζήτηση βέλτιστου διαχωρισμού ανά κόμβο απαιτεί σάρωση ιστογραμμάτων αντί διατεταγμένων τιμών, μειώνοντας την πολυπλοκότητα από $O(N \cdot p)$ σε $O(\text{bins} \cdot p)$ ανά κόμβο· σε συνδυασμό με τεχνικές όπως η δειγματοληψία βάσει κλίσης (GOSS) και η ανάπτυξη δέντρων κατά φύλλο (leaf-wise), επιτεύχθηκαν επιταχύνσεις τάξης μεγέθους χωρίς απώλεια ακρίβειας. Ο HistGradientBoostingClassifier του scikit-learn, ο οποίος συμμετέχει στην πειραματική σύγκριση, είναι ιστογραμμική υλοποίηση εμπνευσμένη από το LightGBM [15], με προεπιλεγμένα 255 κάδους, ρυθμό μάθησης 0,1 και έως 100 επαναλήψεις, ενώ ενεργοποιεί αυτόματα πρόωρη διακοπή σε μεγάλα σύνολα δεδομένων. Αξίζει να σημειωθεί ότι σε πολυκλασικά προβλήματα διατηρεί την ανάγκη ενός δέντρου ανά κλάση ανά γύρο, όμως το ανά δέντρο κόστος είναι τόσο χαμηλότερο ώστε το συνολικό κόστος να παραμένει διαχειρίσιμο — αντίθεση που αναδεικνύεται καθαρά στα πειράματά μας. Στην ίδια γενιά ανήκει και το CatBoost [40], με έμφαση στον εγγενή χειρισμό κατηγορικών χαρακτηριστικών.

3.7 Θεωρητική αντιπαράβολή Bagging και Boosting

Η συγκριτική αποτίμηση των δύο οικογενειών συνοψίζεται στον Πίνακα 3.1. Η θεμελιώδης διαφορά έγκειται στον μηχανισμό βελτίωσης: το Bagging και οι παραλλαγές του μειώνουν τη διακύμανση ασταθών μοντέλων χαμηλής μεροληψίας μέσω ανεξάρτητης τυχαιοποίησης και μέσου όρου, ενώ το Boosting μειώνει πρωτίστως τη μεροληψία αδύναμων μοντέλων μέσω σειριακής, προσαρμοστικής

σύνθεσης [41], [23]. Από τη διαφορά αυτή απορρέουν οι πρακτικές τους ιδιότητες. Οι μέθοδοι Bagging είναι πλήρως παράλληλες — τα μέλη εκπαιδεύονται ανεξάρτητα — και σχεδόν αναισθητες στην αύξηση του πλήθους των μελών, το οποίο μόνο όφελος μπορεί να προσφέρει· αντίθετα, το Boosting είναι εγγενώς σειριακό, και η υπερβολική συνέχιση των γύρων μπορεί, ιδίως με θορυβώδη δεδομένα, να οδηγήσει σε υπερπροσαρμογή, με τον AdaBoost να είναι ο πιο ευαίσθητος στον θόρυβο ετικετών [36]. Ως αντιστάθμισμα, το Boosting επιτυγχάνει συχνά υψηλότερη οροφή ακρίβειας, καθώς βελτιστοποιεί άμεσα μια συνάρτηση απώλειας αντί να βασίζεται αποκλειστικά στην εξομάλυνση της τυχειότητας. Τέλος, ως προς την ευαισθησία στη ρύθμιση, τα τυχαία δάση και τα Extra-Trees θεωρούνται από τους πλέον εύρωστους «out-of-the-box» αλγορίθμους, ενώ οι μηχανές gradient boosting εξαρτώνται περισσότερο από τον συντονισμό ρυθμού μάθησης, πλήθους και βάθους δέντρων [42].

Πίνακας Μέθοδοι Συνόλου Ταξινόμητων: Ανασκόπηση Βιβλιογραφίας.2: Ποιοτική σύγκριση των χαρακτηριστικών των έξι αλγορίθμων της μελέτης.

Χαρακτηριστικό	Bagging	RF	Extra-Trees	AdaBoost	GB	HistGB
Οικογένεια	Bagging	Bagging	Bagging	Boosting	Boosting	Boosting
Κατασκευή μελών	Παράλληλη	Παράλληλη	Παράλληλη	Σειριακή	Σειριακή	Σειριακή
Κύριος στόχος	↓ διακύμανση	↓ διακύμανση	↓ διακύμανση	↓ μεροληψία	↓ μεροληψία	↓ μεροληψία
Βασικό δέντρο	Πλήρες	Πλήρες	Πλήρες	Κολόβωμα (β.1)	Ρηχό (β.3)	Ρηχό (φύλλα 31)
Τυχειότητα	Bootstrap	Bootstrap + χαρ/κά	Τυχαία κατώφλια	—	—	Κάδοι/υποδειγμ.
Πλήθος μελών (default)	10	100	100	50	100	≤100 ανά κλάση
Ευαισθησία σε θόρυβο	Χαμηλή	Χαμηλή	Χαμηλή	Υψηλή	Μέτρια	Μέτρια

Στην πρακτική αξιολόγηση των οικογενειών υπεισέρχονται και παράγοντες πέραν της ακρίβειας. Ως προς την παραλληλοποίηση, οι μέθοδοι bagging κλιμακώνονται σχεδόν γραμμικά με τους διαθέσιμους πυρήνες, αφού κάθε δέντρο εκπαιδεύεται ανεξάρτητα, ενώ στο boosting μόνο η κατασκευή του εκάστοτε δέντρου μπορεί να παραλληλιστεί εσωτερικά — διάκριση σημαντική σε πολυπύρρινα περιβάλλοντα, την οποία το πειραματικό πρωτόκολλο της εργασίας εξουδετερώνει σκόπιμα (μονονηματική εκτέλεση) για να απομονώσει το εγγενές αλγοριθμικό κόστος. Ως προς τη μνήμη, τα σύνολα εκατοντάδων πλήρων δέντρων μπορούν να καταλάβουν σημαντικό χώρο σε μεγάλα δεδομένα, ενώ τα ρηχά δέντρα του boosting είναι συμπαγέστερα. Τέλος, ως προς την ευαισθησία στη ρύθμιση, η μελέτη των Probst et al. [42] ποσοτικοποίησε ότι τα τυχαία δάση έχουν από τις μικρότερες δυνατότητες βελτίωσης μέσω συντονισμού — οι προεπιλογές τους είναι ήδη σχεδόν βέλτιστες — ενώ το gradient boosting εμφανίζει τη μεγαλύτερη «ρυθμισιμότητα» (tunability), στοιχείο που πρέπει να συνεκτιμάται κατά την ερμηνεία συγκρίσεων με προεπιλεγμένες τιμές, όπως η παρούσα.

3.8 Εμπειρικές συγκριτικές μελέτες στη βιβλιογραφία

Η εμπειρική σύγκριση μεθόδων συνόλου έχει μακρά ιστορία. Οι Bauer και Kohavi [41] συνέκριναν παραλλαγές Bagging και Boosting σε πλήθος συνόλων δεδομένων, διαπιστώνοντας ότι το Boosting επιτυγχάνει κατά κανόνα μεγαλύτερη μείωση σφάλματος, αλλά με σημαντικά υψηλότερη διακύμανση αποτελεσμάτων και ευπάθεια στον θόρυβο, ενώ το Bagging είναι σταθερά, αν και μετριοπαθώς, ωφέλιμο. Ο Dietterich [36] πρόσθεσε στη σύγκριση την τυχαιοποίηση των διαχωρισμών και έδειξε ότι με θόρυβο ετικετών το AdaBoost υποχωρεί αισθητά, ενώ το Bagging αναδεικνύεται ανθεκτικότερο. Οι Caruana και Niculescu-Mizil [43] σε ευρεία σύγκριση οικογενειών αλγορίθμων κατέταξαν τα boosted δέντρα και τα τυχαία δάση στις δύο πρώτες θέσεις συνολικής απόδοσης, προ των SVM και των νευρωνικών δικτύων της εποχής. Οι Fernández-Delgado et al. [44], αξιολογώντας 179 ταξινομητές σε 121 σύνολα δεδομένων του UCI, ανέδειξαν τα τυχαία δάση ως την οικογένεια με την υψηλότερη πιθανότητα να βρίσκεται στην κορυφή, με τα boosted δέντρα να ακολουθούν· η μελέτη τους, ωστόσο, προηγήθηκε της ωρίμανσης των σύγχρονων ιστογραμμικών υλοποιήσεων. Σε πιο πρόσφατες συγκρίσεις, οι Zhang et al. [45] επιβεβαίωσαν την ανταγωνιστικότητα των σύγχρονων GBDT βιβλιοθηκών, ενώ οι Grinsztajn et al. [14] έδειξαν ότι σε πινακοποιημένα δεδομένα μεσαίας κλίμακας οι μέθοδοι δέντρων-συνόλων εξακολουθούν να υπερτερούν και των βαθιών νευρωνικών δικτύων. Τέλος, ο Demšar [46] καθιέρωσε το στατιστικό πλαίσιο σύγκρισης ταξινομητών σε πολλαπλά σύνολα δεδομένων — έλεγχος Friedman με post-hoc Nemenyi — το οποίο υιοθετείται και στην παρούσα εργασία.

Από την επισκόπηση αναδεικνύεται το κενό που καλύπτει η παρούσα μελέτη: οι κλασικές συγκρίσεις προηγούνται των ιστογραμμικών υλοποιήσεων, ενώ οι νεότερες εστιάζουν σε εξωτερικές βιβλιοθήκες (XGBoost, LightGBM) και σπανίως αποτιμούν συστηματικά, με κοινό πρωτόκολλο και μετρήσεις καθαρού χρόνου CPU, τους έξι ensemble αλγορίθμους που προσφέρει σήμερα το scikit-learn — τη βιβλιοθήκη που αποτελεί το σημείο εκκίνησης της συντριπτικής πλειονότητας των εφαρμογών μηχανικής μάθησης σε Python. Επιπλέον, η ταυτόχρονη αποτίμηση ποιότητας και υπολογιστικού κόστους με προεπιλεγμένες ρυθμίσεις αντανakλά ακριβώς το σενάριο του μηχανικού που καλείται να επιλέξει αλγόριθμο χωρίς περιθώριο εκτεταμένου συντονισμού.

3.9 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκε το θεωρητικό οικοδόμημα των μεθόδων συνόλου: οι προϋποθέσεις υπό τις οποίες ο συνδυασμός ταξινομητών λειτουργεί, η οικογένεια του Bagging με τα Random Forests και τα Extra-Trees ως μηχανισμοί μείωσης της διακύμανσης, η οικογένεια του Boosting από τη θεωρητική της καταγωγή έως τον AdaBoost, τις μηχανές gradient boosting και τις σύγχρονες ιστογραμμικές υλοποιήσεις, καθώς και η μεταξύ τους θεωρητική και εμπειρική αντιπαραβολή στη βιβλιογραφία. Το επόμενο κεφάλαιο μεταφράζει το πλαίσιο αυτό σε συγκεκριμένη πειραματική μεθοδολογία.

Κεφάλαιο 4ο: Πειραματική Μεθοδολογία

Το κεφάλαιο αυτό γεφυρώνει τη βιβλιογραφική ανασκόπηση με το πειραματικό μέρος της εργασίας, περιγράφοντας τον τρόπο με τον οποίο σχεδιάστηκε και εκτελέστηκε η συγκριτική μελέτη. Η αξιοπιστία κάθε εμπειρικής σύγκρισης αλγορίθμων εξαρτάται καθοριστικά από τη διαφάνεια και την αυστηρότητα της μεθοδολογίας της· για τον λόγο αυτό, οι επιμέρους επιλογές τεκμηριώνονται ρητά και αιτιολογούνται. Παρουσιάζονται ο πειραματικός σχεδιασμός και τα κριτήρια επιλογής των συνόλων δεδομένων, οι συγκρινόμενοι αλγόριθμοι με τις ρυθμίσεις τους, καθώς και ο τρόπος μέτρησης τόσο της προβλεπτικής απόδοσης όσο και του υπολογιστικού κόστους. Στόχος είναι η μελέτη να είναι πλήρως αναπαραγώγιμη και τα συμπεράσματα που θα προκύψουν να μπορούν να ερμηνευθούν με ασφάλεια.

4.1 Εισαγωγή

Το κεφάλαιο αυτό περιγράφει με πληρότητα τη μεθοδολογία της πειραματικής μελέτης, ώστε τα αποτελέσματα του Κεφαλαίου 5 να είναι πλήρως αναπαραγώγιμα. Παρουσιάζονται διαδοχικά ο πειραματικός σχεδιασμός, τα σύνολα δεδομένων και τα κριτήρια επιλογής τους, οι αλγόριθμοι με τις προεπιλεγμένες παραμέτρους τους, το πρωτόκολλο αξιολόγησης με τον τρόπο μέτρησης των χρόνων, και το υπολογιστικό περιβάλλον εκτέλεσης.

4.2 Πειραματικός σχεδιασμός

Ο σχεδιασμός απορρέει ευθέως από τα ερευνητικά ερωτήματα του Κεφαλαίου 1. Για την αποτίμηση της προβλεπτικής απόδοσης (EE1) και του υπολογιστικού κόστους (EE2) κάθε συνδυασμός συνόλου δεδομένων και αλγορίθμου αξιολογείται με δεκαπλή στρωματοποιημένη διασταυρούμενη επικύρωση, καταγράφοντας τέσσερις μετρικές ποιότητας και δύο μετρικές χρόνου. Συνολικά εκτελούνται $13 \times 6 \times 10 = 780$ εκπαιδεύσεις μοντέλων. Για τη μελέτη του συμβιβασμού ποιότητας-κόστους (EE3) και της επίδρασης των χαρακτηριστικών των προβλημάτων (EE4), τα σύνολα δεδομένων επιλέχθηκαν ώστε να καλύπτουν ευρύ φάσμα μεγεθών και πλήθους κλάσεων, όπως αναλύεται στην επόμενη ενότητα. Όλοι οι αλγόριθμοι εκτελούνται με τις προεπιλεγμένες τιμές υπερπαραμέτρων: η επιλογή αυτή εξασφαλίζει ισότιμη μεταχείριση — κανένας αλγόριθμος δεν ευνοείται από ανισομερή συντονισμό — και αντανakλά το ρεαλιστικό σενάριο χρήσης «out-of-the-box», το οποίο έχει υιοθετηθεί και από προηγούμενες συγκριτικές μελέτες [44].

4.3 Σύνολα δεδομένων

Χρησιμοποιήθηκαν δεκατρία σύνολα δεδομένων κατηγοριοποίησης από το αποθετήριο KEEL (Knowledge Extraction based on Evolutionary Learning) [16], το οποίο διαθέτει επιμελημένες εκδοχές κλασικών προβλημάτων αναφοράς σε ενιαία μορφή αρχείων. Τα κριτήρια επιλογής ήταν τρία: πρώτον, κάλυψη ευρέος φάσματος μεγεθών, από 768 έως 57.999 δείγματα, με επτά από τα δεκατρία σύνολα να υπερβαίνουν τα 5.000 δείγματα και τρία τα 15.000 (magic, letter, shuttle)· δεύτερον, κάλυψη δυαδικών και πολυκλασικών προβλημάτων, με πλήθος κλάσεων από 2 έως 26· τρίτον, ποικιλία διαστατικότητας, από 8 έως 57 χαρακτηριστικά, και βαθμού ανισοκατανομής των κλάσεων, από πλήρως ισορροπημένα (π.χ. letter, penbased) έως έντονα ανισοκατανομημένα (π.χ. shuttle, όπου η πολυπληθέστερη κλάση καλύπτει σχεδόν το 80% των δειγμάτων και οι σπανιότερες λίγες δεκάδες). Τα χαρακτηριστικά των συνόλων συνοψίζονται στον Πίνακα 4.1. Η μόνη προεπεξεργασία που εφαρμόστηκε ήταν η κωδικοποίηση των ονομαστικών χαρακτηριστικών σε ακέραιες τιμές (label encoding), καθώς οι υλοποιήσεις δέντρων του scikit-learn απαιτούν αριθμητική είσοδο· δεν εφαρμόστηκε κανονικοποίηση,

δεδομένου ότι τα δέντρα απόφασης είναι αναλλοίωτα σε μονότονους μετασχηματισμούς των χαρακτηριστικών.

Πίνακας Πειραματική Μεθοδολογία.3: Τα δεκατρία σύνολα δεδομένων της μελέτης (πηγή: αποθετήριο KEEL).

Σύνολο δεδομένων	Δείγματα	Χαρακτηριστικά	Κλάσεις
pima	768	8	2
vehicle	846	18	4
contraceptive	1.473	9	3
yeast	1.484	8	10
segment	2.310	19	7
spambase	4.597	57	2
texture	5.500	40	11
satimage	6.435	36	6
ring	7.400	20	2
penbased	10.992	16	10
magic	19.020	10	2
letter	20.000	16	26
shuttle	57.999	9	7

Τα έξι μικρότερα σύνολα καλύπτουν κλασικά προβλήματα αναφοράς της βιβλιογραφίας. Το pima αφορά τη διάγνωση διαβήτη σε γυναίκες ινδιάνικης καταγωγής Pima από οκτώ κλινικές μετρήσεις. Το vehicle ζητά την αναγνώριση τεσσάρων τύπων οχημάτων από γεωμετρικά χαρακτηριστικά της σιλουέτας τους. Το contraceptive (CMC) προβλέπει την επιλογή αντισυλληπτικής μεθόδου από δημογραφικά και κοινωνικοοικονομικά χαρακτηριστικά — ένα δύσκολο πρόβλημα τριών κλάσεων με έντονη επικάλυψη, στο οποίο, όπως θα φανεί, καμία μέθοδος δεν υπερβαίνει το 57%. Το yeast προβλέπει τη θέση πρωτεϊνικού εντοπισμού σε κύτταρα ζύμης, με δέκα έντονα ανισοκατανεμημένες κλάσεις. Το segment αφορά την ταξινόμηση περιοχών εικόνων εξωτερικών σκηνών σε επτά κατηγορίες υφής, και το spambase τη διάκριση ανεπιθύμητης ηλεκτρονικής αλληλογραφίας από 57 χαρακτηριστικά συχνοτήτων λέξεων και χαρακτήρων.

Τα επτά μεγαλύτερα σύνολα επιλέχθηκαν για τον έλεγχο της κλιμάκωσης. Το texture ζητά την αναγνώριση έντεκα υφών Brodatz από συντελεστές τέταρτης τάξης, το satimage την ταξινόμηση εικονοστοιχείων δορυφορικών εικόνων Landsat σε έξι κατηγορίες εδαφοκάλυψης, και το ring είναι το συνθετικό πρόβλημα δύο ομόκεντρων κατανομών Gauss του Breiman. Το penbased αφορά την αναγνώριση χειρόγραφων ψηφίων από δυναμικά χαρακτηριστικά γραφίδας (10 κλάσεις, 10.992 δείγματα). Το magic (19.020 δείγματα) προέρχεται από το τηλεσκόπιο ακτινοβολίας Cherenkov MAGIC και ζητά τη διάκριση σημάτων ακτίνων γ από αδρονικό υπόβαθρο. Το letter (20.000 δείγματα) είναι το κλασικό πρόβλημα αναγνώρισης των 26 κεφαλαίων λατινικών χαρακτήρων από στατιστικά χαρακτηριστικά εικόνας — το απαιτητικότερο πολυκλασικό πρόβλημα της μελέτης. Τέλος, το shuttle (57.999 δείγματα, 7 κλάσεις) περιέχει τηλεμετρία του διαστημικού λεωφορείου της NASA και χαρακτηρίζεται από ακραία ανισοκατανομή: η πολυπληθέστερη κλάση καλύπτει το 78,6% των δειγμάτων, ενώ οι δύο σπανιότερες αριθμούν λίγες δεκάδες δείγματα.

4.4 Αλγόριθμοι και υλοποίηση

Συγκρίνονται και οι έξι ensemble ταξινομητές που προσφέρει η βιβλιοθήκη scikit-learn [15] (έκδοση 1.7): AdaBoostClassifier, BaggingClassifier, ExtraTreesClassifier, GradientBoostingClassifier, HistGradientBoostingClassifier και RandomForestClassifier. Η θεωρία καθενός παρουσιάστηκε στο Κεφάλαιο 3· ο Πίνακας 4.2 συνοψίζει τις σημαντικότερες προεπιλεγμένες τιμές υπερπαραμέτρων, οι οποίες και χρησιμοποιήθηκαν αμετάβλητες. Αξίζει να υπογραμμιστούν οι διαφορές στη «δύναμη πυρός» των προεπιλογών, καθώς ερμηνεύουν μέρος των αποτελεσμάτων: ο BaggingClassifier χρησιμοποιεί μόλις 10 πλήρη δέντρα έναντι 100 των RandomForest/ExtraTrees, ενώ ο AdaBoostClassifier χρησιμοποιεί 50 κολοβώματα βάθους 1, δηλαδή εξαιρετικά αδύναμα μέλη. Σε όλους τους αλγορίθμους ορίστηκε σταθερή γεννήτρια τυχαιότητας (random_state = 42) για λόγους αναπαραγωγιμότητας.

Πίνακας Πειραματική Μεθοδολογία.4: Κύριες προεπιλεγμένες υπερπαραμέτροι των έξι αλγορίθμων (scikit-learn 1.7).

Αλγόριθμος	Μέλη	Βασικό δέντρο	Λοιπές βασικές προεπιλογές
AdaBoostClassifier	50	βάθος 1 (stump)	SAMME, learning_rate = 1.0
BaggingClassifier	10	απεριόριστο βάθος	bootstrap δείγματα μεγέθους N
ExtraTreesClassifier	100	απεριόριστο βάθος	χωρίς bootstrap, $\sqrt{p}$ χαρ/κά, Gini
GradientBoostingClassifier	100	βάθος 3	learning_rate = 0.1, log-loss
HistGradientBoostingClassifier	≤100 επαν.	≤31 φύλλα	255 κάδοι, learning_rate = 0.1, early stopping (αυτόμ.)
RandomForestClassifier	100	απεριόριστο βάθος	bootstrap, $\sqrt{p}$ χαρ/κά, Gini

Η ανάγνωση των αρχείων KEEL υλοποιήθηκε με συνάρτηση ανάλυσης της επικεφαλίδας @attribute και του τμήματος @data, της οποίας η δομή δίνεται συνοπτικά στο Απόσπασμα κώδικα 4.1· ο πλήρης κώδικας παρατίθεται στο Παράρτημα Α.

```
def parse_keel(path):
    """Διαβάζει αρχείο KEEL .dat -> X (float array), y (ετικέτες)."""
    attrs, nominal, data_lines, in_data = [], {}, [], False
    with open(path) as f:
        for line in f:
            line = line.strip()
            low = line.lower()
            if low.startswith("@attribute"):
                ... # καταγραφή ονόματος και τύπου χαρακτηριστικού
            elif low.startswith("@data"):
                in_data = True
            elif in_data and line:
                data_lines.append([v.strip() for v in line.split(",")])
    ... # μετατροπή σε πίνακες X, y (label encoding στα ονομαστικά)
    return X, np.array(y)
```

Απόσπασμα κώδικα Πειραματική Μεθοδολογία.4: Συνάρτηση ανάγνωσης αρχείων KEEL

4.5 Πρωτόκολλο αξιολόγησης

Κάθε συνδυασμός συνόλου δεδομένων και αλγορίθμου αξιολογήθηκε με δεκαπλή στρωματοποιημένη διασταυρούμενη επικύρωση (StratifiedKFold με ανακάτεμα και σταθερό σπόρο 42), σύμφωνα με τη σύσταση του Kohavi [26]: η στρωματοποίηση διατηρεί την αναλογία των κλάσεων σε κάθε fold, στοιχείο κρίσιμο για τα ανισοκατανεμημένα σύνολα της μελέτης. Σε κάθε fold καταγράφηκαν έξι μεγέθη: ο χρόνος εκπαίδευσης, ο χρόνος ταξινόμησης του συνόλου ελέγχου, η ορθότητα, και οι μακρο-συναθροισμένες ακρίβεια, ανάκληση και F-score. Ως τελική τιμή κάθε μεγέθους αναφέρεται ο μέσος όρος των δέκα folds. Οι χρόνοι μετρήθηκαν με τη συνάρτηση `time.process_time()` της Python, η οποία επιστρέφει τον καθαρό χρόνο επεξεργαστή της τρέχουσας διεργασίας — άθροισμα χρόνου χρήστη και συστήματος — αγνοώντας διαστήματα αναμονής, ώστε οι μετρήσεις να είναι ανεξάρτητες από τον στιγμιαίο φόρτο του υπολογιστή. Ο πυρήνας του πειραματικού βρόχου δίνεται στο Απόσπασμα κώδικα 4.2.

```
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
for fold, (tr, te) in enumerate(skf.split(X, y)):
    clf = ALGOS[algo_name]()          # νέο μοντέλο με default παραμέτρους

    t0 = time.process_time()          # καθαρός χρόνος CPU
    clf.fit(X[tr], y[tr])
    train_time = time.process_time() - t0

    t0 = time.process_time()
    pred = clf.predict(X[te])
    classif_time = time.process_time() - t0

    acc = accuracy_score(y[te], pred)
    prec, rec, f1, _ = precision_recall_fscore_support(
        y[te], pred, average="macro", zero_division=0)
```

Απόσπασμα κώδικα Πειραματική Μεθοδολογία.5: Πυρήνας του πειραματικού βρόχου αξιολόγησης

Επισημαίνεται ότι, καθώς κάθε αλγόριθμος αξιολογείται στα ίδια ακριβώς folds κάθε συνόλου δεδομένων (κοινός σπόρος διαμέρισης), οι συγκρίσεις μεταξύ αλγορίθμων είναι κατά ζεύγη συζευγμένες, γεγονός που ενισχύει τη στατιστική ισχύ του ελέγχου σημαντικότητας του Κεφαλαίου 5.

4.6 Περιβάλλον εκτέλεσης

Όλα τα πειράματα εκτελέστηκαν στο ίδιο υπολογιστικό περιβάλλον: εικονική μηχανή Linux (Ubuntu 22.04) με επεξεργαστή x86-64 δύο πυρήνων, Python 3.10 και scikit-learn 1.7. Για τη διασφάλιση δίκαιης σύγκρισης των χρόνων, η εκτέλεση περιορίστηκε σε ένα νήμα ανά διεργασία (μεταβλητή περιβάλλοντος `OMP_NUM_THREADS = 1`), ώστε αλγόριθμοι με πολυνηματικές υλοποιήσεις (κυρίως ο ιστογραμμικός HistGradientBoosting) να μην εμφανίζουν πλασματικό πλεονέκτημα ή μειονέκτημα έναντι των υπολοίπων· καμία υλοποίηση δεν χρησιμοποίησε παραλληλισμό (`n_jobs = 1`). Οι απόλυτες τιμές των χρόνων εξαρτώνται φυσικά από το συγκεκριμένο υλικό· επειδή όμως όλοι οι συνδυασμοί μετρήθηκαν στο ίδιο μηχάνημα με τους ίδιους περιορισμούς, οι σχετικές συγκρίσεις — που είναι και το ζητούμενο της μελέτης — παραμένουν έγκυρες. Ο πλήρης κώδικας, τα σύνολα δεδομένων και τα πρωτογενή αποτελέσματα ανά fold συνοδεύουν την εργασία (Παραρτήματα Α και Β).

4.7 Επίλογος

Στο κεφάλαιο αυτό τεκμηριώθηκε η πειραματική μεθοδολογία: δεκατρία σύνολα δεδομένων του KEEL με κλιμακούμενο μέγεθος και πλήθος κλάσεων, έξι ensemble αλγόριθμοι του scikit-learn με προεπιλεγμένες παραμέτρους, δεκαπλή στρωματοποιημένη διασταυρούμενη επικύρωση με κοινά folds,

μέτρηση τεσσάρων μετρικών ποιότητας και δύο μετρικών καθαρού χρόνου CPU, και ενιαίο μονονηματικό περιβάλλον εκτέλεσης. Το επόμενο κεφάλαιο παρουσιάζει και αναλύει τα αποτελέσματα.

Κεφάλαιο 5ο: Αποτελέσματα και Συζήτηση

Το κεφάλαιο αυτό αποτελεί την καρδιά του πειραματικού μέρους της εργασίας: παρουσιάζει, αναλύει και ερμηνεύει τα αποτελέσματα της συγκριτικής μελέτης των αλγορίθμων. Τα ευρήματα δεν παρατίθενται απλώς ως αριθμητικοί πίνακες· εξετάζονται διαδοχικά ως προς την προβλεπτική απόδοση, το υπολογιστικό κόστος εκπαίδευσης και ταξινόμησης, και τον συμβιβασμό ανάμεσα στην ποιότητα και το κόστος, ο οποίος συχνά καθορίζει την επιλογή αλγορίθμου στην πράξη. Οι παρατηρούμενες διαφορές υποβάλλονται σε στατιστικό έλεγχο σημαντικότητας, ενώ διερευνάται και η επίδραση των χαρακτηριστικών των προβλημάτων στη συμπεριφορά των μεθόδων. Το κεφάλαιο ολοκληρώνεται με τη συζήτηση των ευρημάτων σε αντιπαραβολή με τη βιβλιογραφία των προηγούμενων κεφαλαίων και με την ειλικρινή καταγραφή των περιορισμών της μελέτης.

5.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα των 780 εκτελέσεων του πειραματικού πρωτοκόλλου. Η ανάλυση οργανώνεται γύρω από τα ερευνητικά ερωτήματα: εξετάζεται πρώτα η προβλεπτική απόδοση (ορθότητα και F-score), έπειτα το υπολογιστικό κόστος (χρόνοι εκπαίδευσης και ταξινόμησης), στη συνέχεια ο συμβιβασμός των δύο, και ακολουθεί στατιστικός έλεγχος σημαντικότητας των διαφορών κατά Demšar [46]. Το κεφάλαιο κλείνει με συζήτηση των ευρημάτων υπό το φως της βιβλιογραφίας και με καταγραφή των περιορισμών της μελέτης.

5.2 Προβλεπτική απόδοση

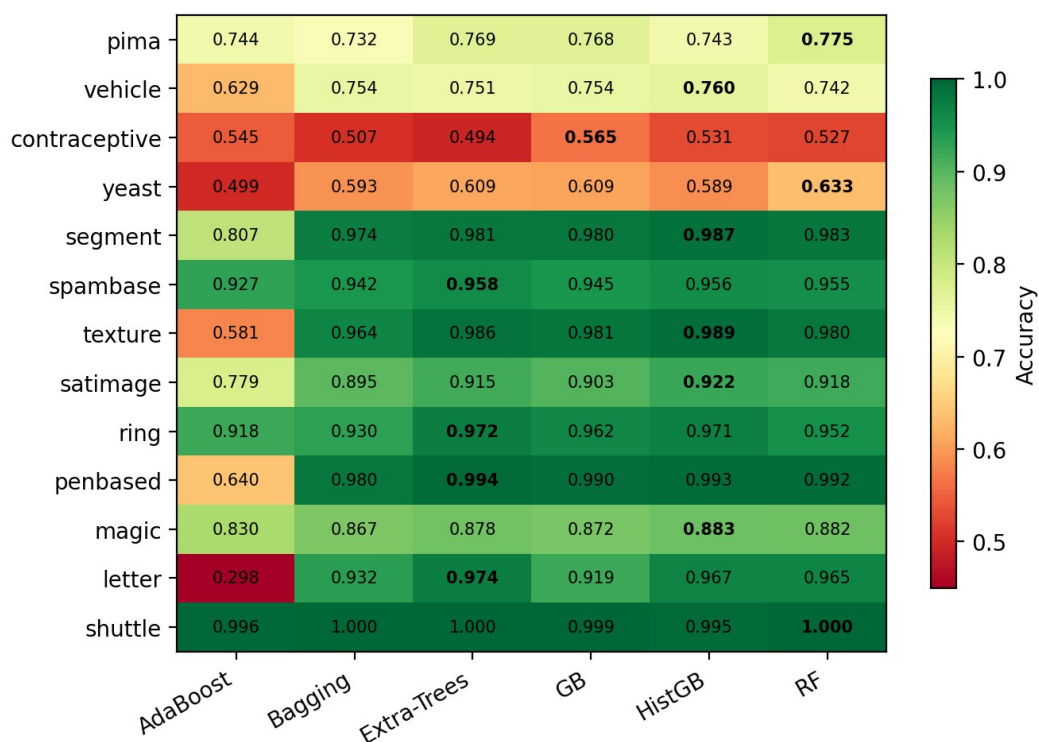
Ο Πίνακας 5.1 παρουσιάζει τη μέση ορθότητα (accuracy) κάθε αλγορίθμου σε κάθε σύνολο δεδομένων και ο Πίνακας 5.2 το αντίστοιχο μακρο-συναθροισμένο F-score· με έντονη γραφή σημειώνεται η καλύτερη τιμή κάθε γραμμής. Το Σχήμα 5.1 αποτυπώνει εποπτικά την ίδια εικόνα ως χάρτη θερμότητας.

Πίνακας Αποτελέσματα και Συζήτηση.5: Μέση ορθότητα ανά σύνολο δεδομένων και αλγόριθμο.

Dataset	AdaBoost	Bagging	Extra-Trees	GB	HistGB	RF
pima	0,7435	0,7318	0,7695	0,7683	0,7434	0,7747
vehicle	0,6289	0,7542	0,7506	0,7541	0,7601	0,7423
contraceptive	0,5452	0,5071	0,4942	0,5655	0,5309	0,5268
yeast	0,4987	0,5929	0,6091	0,6085	0,5889	0,6327
segment	0,8074	0,9736	0,9805	0,9801	0,9874	0,9827
spambase	0,9271	0,9421	0,9580	0,9452	0,9565	0,9548
texture	0,5811	0,9644	0,9856	0,9815	0,9895	0,9796
satimage	0,7795	0,8949	0,9155	0,9030	0,9221	0,9178
ring	0,9184	0,9300	0,9723	0,9622	0,9705	0,9524
penbased	0,6400	0,9796	0,9941	0,9897	0,9928	0,9915
magic	0,8301	0,8673	0,8785	0,8717	0,8826	0,8818
letter	0,2980	0,9323	0,9741	0,9191	0,9670	0,9649
shuttle	0,9958	0,9997	0,9997	0,9993	0,9955	0,9999

Πίνακας Αποτελέσματα και Συζήτηση.6: Μέσο μακρο-συναθροισμένο F-score ανά σύνολο δεδομένων και αλγόριθμο.

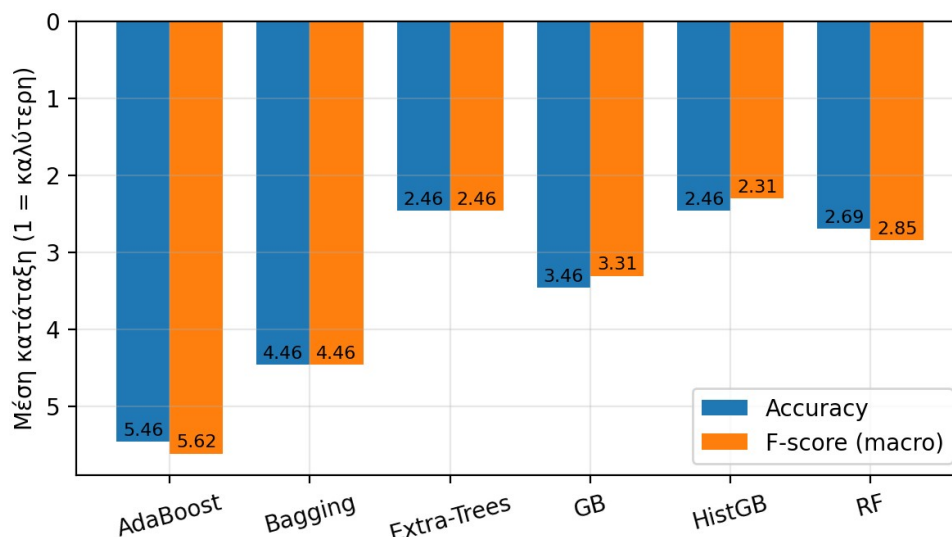
Dataset	AdaBoost	Bagging	Extra-Trees	GB	HistGB	RF
pima	0,7074	0,6866	0,7349	0,7390	0,7140	0,7407
vehicle	0,6216	0,7499	0,7451	0,7541	0,7604	0,7378
contraceptive	0,5140	0,4822	0,4701	0,5481	0,5093	0,5026
yeast	0,3310	0,4825	0,5357	0,5093	0,4419	0,5540
segment	0,7799	0,9735	0,9805	0,9801	0,9875	0,9827
spambase	0,9229	0,9390	0,9559	0,9422	0,9544	0,9524
texture	0,5413	0,9644	0,9856	0,9815	0,9894	0,9797
satimage	0,7361	0,8731	0,8967	0,8826	0,9060	0,8999
ring	0,9182	0,9300	0,9723	0,9621	0,9705	0,9524
penbased	0,6286	0,9797	0,9941	0,9898	0,9928	0,9916
magic	0,8069	0,8488	0,8611	0,8527	0,8669	0,8664
letter	0,2574	0,9322	0,9740	0,9190	0,9670	0,9648
shuttle	0,5021	0,9779	0,9750	0,9525	0,6362	0,9657



Σχήμα Αποτελέσματα και Συζήτηση.6: Χάρτης θερμότητας της μέσης ορθότητας.

Από τους πίνακες προκύπτει καταρχάς ότι οι πέντε από τους έξι αλγορίθμους κινούνται σε παραπλήσια, υψηλά επίπεδα: η μέση ορθότητα στο σύνολο των δεκατριών προβλημάτων είναι 86,9% για τον RandomForest, 86,8% για τους ExtraTrees και HistGradientBoosting, 86,5% για τον GradientBoosting και 85,2% για τον Bagging, ενώ ο AdaBoost υπολείπεται αισθητά με 70,7%. Σε επίπεδο «νικών» ανά σύνολο δεδομένων, ο HistGradientBoosting επιτυγχάνει την υψηλότερη ορθότητα σε πέντε από τα δεκατρία προβλήματα, ο ExtraTrees σε τέσσερα, ο RandomForest σε τρία και ο GradientBoosting σε

ένα, ενώ οι AdaBoost και Bagging σε κανένα. Η μέση κατάταξη (mean rank), που παρουσιάζεται στο Σχήμα 5.2 και αναλύεται στατιστικά στην ενότητα 5.7, συνοψίζει την ίδια ιεραρχία: ExtraTrees και HistGradientBoosting μοιράζονται την πρώτη θέση ως προς την ορθότητα (2,46), ακολουθεί πολύ κοντά ο RandomForest (2,69), έπειτα ο GradientBoosting (3,46), ο Bagging (4,46) και τελευταίος ο AdaBoost (5,46)· ως προς το F-score η εικόνα είναι σχεδόν ταυτόσημη, με τον HistGradientBoosting οριακά πρώτο (2,31).



Σχήμα Αποτελέσματα και Συζήτηση.7: Μέση κατάταξη των αλγορίθμων (μικρότερη τιμή = καλύτερη).

Η ανάλυση ανά τύπο προβλήματος (EE4) είναι αποκαλυπτική. Στα δυαδικά και ολιγοκλασικά προβλήματα οι διαφορές των πέντε κορυφαίων αλγορίθμων είναι μικρές — στο magic, για παράδειγμα, το εύρος ορθότητας μεταξύ Bagging και HistGradientBoosting είναι μόλις 1,7 ποσοστιαίες μονάδες. Στα πολυκλασικά προβλήματα, αντίθετα, οι αποκλίσεις διευρύνονται δραματικά, με χαρακτηριστικότερη περίπτωση το letter (26 κλάσεις): ο AdaBoost, της οποίας τα 50 κολοβώματα βάθους 1 αδυνατούν εγγενώς να διαχωρίσουν 26 κλάσεις, καταρρέει στο 29,8%, ο GradientBoosting με τα ρηχά δέντρα βάθους 3 περιορίζεται στο 91,9%, ενώ οι μέθοδοι με πλήρη δέντρα — ExtraTrees (97,4%), RandomForest (96,5%) — και ο HistGradientBoosting (96,7%) υπερβαίνουν το 96%. Το εύρημα συνάδει με τη θεωρία του Κεφαλαίου 3: σε προβλήματα με σύνθετα, πολυκλασικά όρια αποφάσεων, η χαμηλή εκφραστική ισχύς των αδύναμων μελών του boosting με προεπιλεγμένο, μικρό αριθμό γύρων μετατρέπεται σε ανυπέρβλητη μεροληψία, ενώ τα πλήρη δέντρα των μεθόδων bagging διατηρούν χαμηλή μεροληψία και αφήνουν στον μέσο όρο τη μείωση της διακύμανσης. Αξιοσημείωτη είναι επίσης η επίδοση όλων των μεθόδων στο ανισοκατανεμημένο shuttle, όπου οι RandomForest και Bagging αγγίζουν ορθότητα 99,99%· ωστόσο το μακρο-F-score του ίδιου προβλήματος είναι αισθητά χαμηλότερο (π.χ. 93,6% για τον Bagging), υπενθυμίζοντας ότι στις σπάνιες κλάσεις η εικόνα είναι λιγότερο ειδυλλιακή — ακριβώς ο λόγος για τον οποίο υιοθετήθηκε η μακρο-συνάθροιση.

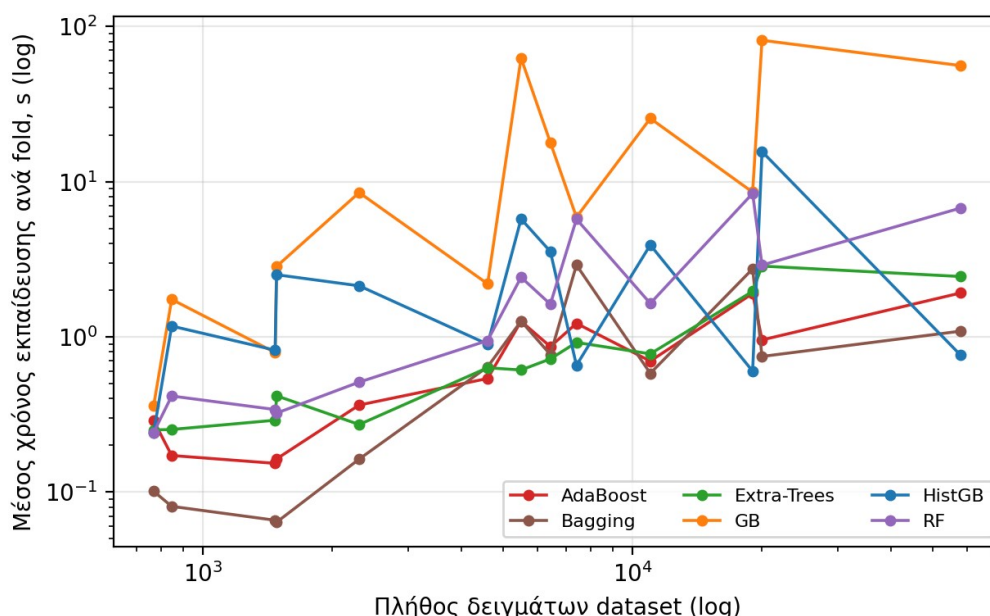
5.3 Χρόνος εκπαίδευσης

Ο Πίνακας 5.3 παρουσιάζει τον μέσο χρόνο εκπαίδευσης ανά fold και το Σχήμα 5.3 την κλιμάκωσή του ως προς το μέγεθος του συνόλου δεδομένων. Η εικόνα εδώ είναι εντυπωσιακά ανομοιογενής: οι χρόνοι εκτείνονται σε τέσσερις τάξεις μεγέθους, από εκατοστά του δευτερολέπτου έως και άνω του λεπτού ανά fold. Ταχύτερος συνολικά αναδεικνύεται ο Bagging (μέση κατάταξη ταχύτητας 1,92) — αναμενόμενο, αφού εκπαιδεύει μόλις 10 δέντρα — με τους AdaBoost (2,46) και ExtraTrees (2,69) να ακολουθούν·

έπεται ο HistGradientBoosting (3,85), ο RandomForest (4,15), και μακράν τελευταίος ο GradientBoosting (5,92), ο οποίος είναι ο αργότερος αλγόριθμος σε έντεκα από τα δεκατρία προβλήματα. Η σύγκριση των δύο «αδελφών» μεθόδων πλήρων δέντρων επιβεβαιώνει τη θεωρητική πρόβλεψη των Geurts et al. [7]: τα Extra-Trees, αποφεύγοντας την εξαντλητική αναζήτηση βέλτιστων καταωφλίων, εκπαιδεύονται συστηματικά ταχύτερα από τον RandomForest — στο magic, ενδεικτικά, 2,0 έναντι 8,4 δευτερολέπτων ανά fold.

Πίνακας Αποτελέσματα και Συζήτηση.7: Μέσος χρόνος εκπαίδευσης ανά fold, σε δευτερόλεπτα CPU (έντονα η μικρότερη τιμή).

Dataset	AdaBoost	Bagging	Extra-Trees	GB	HistGB	RF
pima	0,288	0,101	0,251	0,359	0,240	0,242
vehicle	0,172	0,081	0,253	1,741	1,172	0,415
contraceptive	0,153	0,066	0,290	0,791	0,819	0,340
yeast	0,163	0,064	0,417	2,838	2,510	0,322
segment	0,363	0,163	0,272	8,494	2,130	0,510
spambase	0,539	0,640	0,632	2,194	0,896	0,945
texture	1,259	1,261	0,611	62,514	5,741	2,418
satimage	0,864	0,768	0,720	17,840	3,543	1,621
ring	1,220	2,900	0,919	5,894	0,655	5,730
penbased	0,697	0,581	0,776	25,523	3,905	1,640
magic	1,901	2,738	1,962	8,545	0,600	8,391
letter	0,954	0,747	2,855	81,336	15,608	2,896
shuttle	1,916	1,087	2,444	55,848	0,760	6,755



Σχήμα Αποτελέσματα και Συζήτηση.8: Μέσος χρόνος εκπαίδευσης ανά fold συναρτήσει του πλήθους δειγμάτων (λογαριθμικοί άξονες).

Το κρίσιμότερο εύρημα αφορά την αλληλεπίδραση αλγορίθμου και πλήθους κλάσεων (EE4). Ο κλασικός GradientBoosting, που εκπαιδεύει ένα δέντρο ανά κλάση ανά γύρο, εμφανίζει εκρηκτική

επιβράδυνση στα πολυκλασικά προβλήματα μεγάλης κλίμακας: στο letter (20.000 δείγματα, 26 κλάσεις) απαιτεί 81,3 δευτερόλεπτα ανά fold και στο shuttle (58.000 δείγματα, 7 κλάσεις) 55,8, όταν ο ιστογραμμικός HistGradientBoosting — που επιλύει το ίδιο πρόβλημα βελτιστοποίησης με διακριτοποιημένα χαρακτηριστικά — χρειάζεται 15,6 και 0,8 δευτερόλεπτα αντίστοιχα, δηλαδή είναι από 5 έως 70 φορές ταχύτερος, με ίση ή καλύτερη ποιότητα προβλέψεων. Η δραστική αυτή διαφορά αποτελεί την πλέον απτή πειραματική επιβεβαίωση της συνεισφοράς των ιστογραμμικών τεχνικών [11] και εξηγεί γιατί οι κλασικές υλοποιήσεις gradient boosting έχουν πρακτικά εκτοπιστεί από τις ιστογραμμικές σε δεδομένα μεσαίας και μεγάλης κλίμακας.

5.4 Χρόνος ταξινόμησης

Ο Πίνακας 5.4 παρουσιάζει τον μέσο χρόνο ταξινόμησης του συνόλου ελέγχου ανά fold. Οι απόλυτες τιμές είναι μικρές — το ακριβότερο κελί δεν υπερβαίνει το μισό δευτερόλεπτο — η σχετική εικόνα όμως αντιστρέφει εν μέρει εκείνη της εκπαίδευσης. Ταχύτεροι στην πρόβλεψη είναι οι αλγόριθμοι boosting με ρηχά μέλη: ο GradientBoosting, ακριβότερος στην εκπαίδευση, είναι σταθερά από τους φθηνότερους στην ταξινόμηση, αφού η πρόβλεψη διασχίζει 100 δέντρα βάθους μόλις 3, και ο AdaBoost αντίστοιχα 50 κολοβώματα. Στον αντίποδα, οι μέθοδοι με 100 πλήρη δέντρα (RandomForest, ExtraTrees) πληρώνουν στην πρόβλεψη το βάθος των μελών τους, με τους χρόνους τους να παραμένουν πάντως πρακτικά αμελητέοι για τις περισσότερες εφαρμογές. Η παρατήρηση έχει πρακτική αξία για συστήματα με αυστηρές απαιτήσεις καθυστέρησης ή με τεράστιους όγκους προβλέψεων, όπου το κόστος ταξινόμησης, και όχι εκπαίδευσης, είναι το κυρίαρχο (EE2).

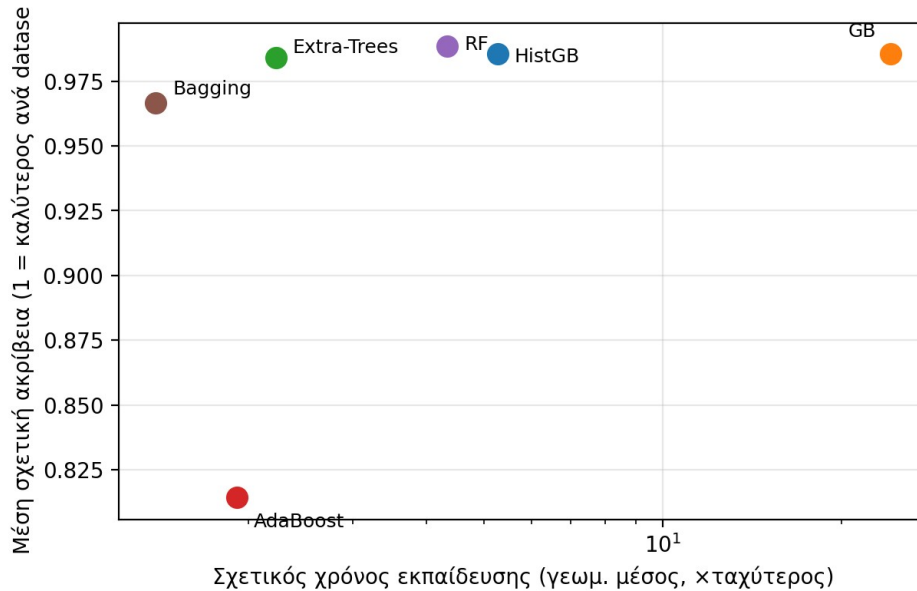
Πίνακας Αποτελέσματα και Συζήτηση.8: Μέσος χρόνος ταξινόμησης ανά fold, σε δευτερόλεπτα CPU (έντονα η μικρότερη τιμή).

Dataset	AdaBoost	Bagging	Extra-Trees	GB	HistGB	RF
pima	0,0178	0,0054	0,0112	0,0008	0,0023	0,0070
vehicle	0,0096	0,0025	0,0116	0,0027	0,0106	0,0130
contraceptive	0,0101	0,0037	0,0136	0,0028	0,0100	0,0155
yeast	0,0112	0,0023	0,0183	0,0048	0,0296	0,0100
segment	0,0111	0,0024	0,0124	0,0042	0,0240	0,0084
spambase	0,0110	0,0046	0,0166	0,0013	0,0082	0,0144
texture	0,0156	0,0034	0,0175	0,0092	0,0625	0,0142
satimage	0,0223	0,0052	0,0223	0,0074	0,0479	0,0195
ring	0,0116	0,0055	0,0352	0,0019	0,0131	0,0173
penbased	0,0178	0,0064	0,0255	0,0205	0,1023	0,0198
magic	0,0136	0,0072	0,0660	0,0035	0,0211	0,0383
letter	0,0397	0,0086	0,0845	0,0818	0,5023	0,0650
shuttle	0,0359	0,0087	0,0661	0,0576	0,0651	0,0449

5.5 Συμβιβασμός ποιότητας και κόστους

Το Σχήμα 5.4 συνοψίζει την απάντηση στο EE3, απεικονίζοντας για κάθε αλγόριθμο τη μέση σχετική ορθότητα (ως ποσοστό της βέλτιστης ανά πρόβλημα) έναντι του γεωμετρικού μέσου του σχετικού χρόνου εκπαίδευσης (ως πολλαπλάσιο του ταχύτερου ανά πρόβλημα). Στο βορειοδυτικό, επιθυμητό άκρο του διαγράμματος τοποθετούνται οι ExtraTrees και HistGradientBoosting, που συνδυάζουν

κορυφαία ακρίβεια με μέτριο κόστος· ο RandomForest προσφέρει ισοδύναμη σχεδόν ακρίβεια με αισθητά υψηλότερο χρόνο εκπαίδευσης· ο Bagging είναι ο φθηνότερος αλλά με μετρήσιμη υστέρηση ακρίβειας· ενώ ο GradientBoosting κυριαρχείται πλήρως — ίση ή χειρότερη ακρίβεια με πολλαπλάσιο κόστος — και ο AdaBoost, αν και φθηνός, αποκλείεται λόγω χαμηλής ποιότητας. Υπό το πρίσμα Pareto, οι μη κυριαρχούμενες επιλογές της μελέτης είναι ουσιαστικά τρεις: Bagging για ελάχιστο κόστος, ExtraTrees και HistGradientBoosting για μέγιστη ποιότητα.



Σχήμα Αποτελέσματα και Συζήτηση.9: Συμβιβασμός ποιότητας-κόστους των έξι αλγορίθμων.

5.6 Στατιστικός έλεγχος σημαντικότητας

Για τον έλεγχο της στατιστικής σημαντικότητας των διαφορών ακολουθήθηκε η μεθοδολογία του Demšar [46] για σύγκριση πολλαπλών ταξινομητών σε πολλαπλά σύνολα δεδομένων. Ο μη παραμετρικός έλεγχος Friedman επί των κατατάξεων ορθότητας απέρριψε την υπόθεση ισοδυναμίας των έξι αλγορίθμων με $\chi^2 = 28,16$ και $p = 3,4 \times 10^{-5}$ (αντίστοιχα, για το F-score, $\chi^2 = 31,07$ και $p = 9,1 \times 10^{-6}$). Ο post-hoc έλεγχος Nemenyi με επίπεδο σημαντικότητας 0,05 δίνει κρίσιμη διαφορά μέσων κατατάξεων $CD = 2,09$ για $k = 6$ αλγορίθμους και $N = 13$ σύνολα δεδομένων. Συνεπώς, οι ExtraTrees (2,46), HistGradientBoosting (2,46) και RandomForest (2,69) υπερτερούν στατιστικώς σημαντικά του AdaBoost (5,46), ενώ οι δύο πρώτοι και του Bagging στο F-score· οι μεταξύ των τριών κορυφαίων διαφορές, όπως και εκείνη από τον GradientBoosting (3,46), δεν είναι στατιστικώς σημαντικές στο διαθέσιμο πλήθος προβλημάτων. Η εικόνα αυτή — κορεσμός της κορυφής και σαφής απόσταση της ουράς — είναι απολύτως συμβατή με τα ευρήματα ευρύτερων συγκρίσεων [44].

Πίνακας Αποτελέσματα και Συζήτηση.9: Συγκεντρωτική εικόνα: μέσες κατατάξεις, νίκες και μέσοι χρόνοι.

Αλγόριθμος	Κατάταξη Acc.	Κατάταξη F1	Νίκες Acc.	Κατάταξη χρόνου εκπ.	Χρόνος εκπ. (s)	Χρόνος ταξ. (s)
AdaBoost	5,46	5,62	0	2,46	0,807	0,0175
Bagging	4,46	4,46	0	1,92	0,861	0,0051
Extra-Trees	2,46	2,46	4	2,69	0,954	0,0308
GB	3,46	3,31	1	5,92	21,071	0,0153
HistGB	2,46	2,31	5	3,85	2,968	0,0692

Αλγόριθμος	Κατάταξη Acc.	Κατάταξη F1	Νίκες Acc.	Κατάταξη χρόνου εκπ.	Χρόνος εκπ. (s)	Χρόνος ταξ. (s)
RF	2,69	2,85	3	4,15	2,479	0,0221

5.7 Επίδραση των χαρακτηριστικών των προβλημάτων

Η συνολική εικόνα των προηγούμενων ενοτήτων αποκτά μεγαλύτερη διαγνωστική αξία όταν αναλυθεί κατά ομάδες προβλημάτων, απαντώντας ευθέως στο EE4. Στα έξι δυαδικά και ολιγοκλασικά προβλήματα (prima, spambase, ring, magic, contraceptive, vehicle), οι πέντε κορυφαίοι αλγόριθμοι είναι ουσιαστικά δυσδιάκριτοι — οι διαφορές ορθότητας σπανίως υπερβαίνουν τις δύο ποσοστιαίες μονάδες — και ακόμη και ο AdaBoost παραμένει αξιοπρεπής όπου το πρόβλημα είναι σχεδόν γραμμικά διαχωρίσιμο, όπως στο ring (97,0%). Η επιλογή αλγορίθμου σε τέτοια προβλήματα μπορεί συνεπώς να γίνει σχεδόν αποκλειστικά με κριτήρια κόστους.

Στα πολυκλασικά προβλήματα η εικόνα αντιστρέφεται και οι δομικές διαφορές των αλγορίθμων έρχονται στο προσκήνιο. Καθώς το πλήθος των κλάσεων αυξάνεται από 7 (segment, shuttle) σε 10-11 (yeast, penbased, texture) και τέλος σε 26 (letter), το χάσμα μεταξύ των μεθόδων πλήρων δέντρων και των boosting μεθόδων με ρηγά, προεπιλεγμένα μέλη διευρύνεται συστηματικά: στο penbased ο AdaBoost υπολείπεται της κορυφής κατά 55 μονάδες ορθότητας και στο letter κατά 68. Ταυτόχρονα, το πολυκλασικό σχήμα «ένα δέντρο ανά κλάση ανά γύρο» καθιστά τον κλασικό GradientBoosting απαγορευτικά αργό ακριβώς εκεί όπου θα χρειαζόταν περισσότερους πόρους. Αντίθετα, το μέγεθος του συνόλου καθαυτό αποδεικνύεται ήπιος παράγοντας: όλες οι μέθοδοι κλιμακώνονται σχεδόν γραμμικά με το πλήθος των δειγμάτων (Σχήμα 5.3), και το ιστογραμμικό gradient boosting υπογραμμικά, χάρη στη σταθερή διακριτοποίηση των χαρακτηριστικών.

Τέλος, η ανισοκατανομή των κλάσεων αναδεικνύεται σε παράγοντα που διαφοροποιεί όχι τόσο τη μέση απόδοση όσο τη συμπεριφορά στις σπάνιες κλάσεις: στο shuttle, όλες οι μέθοδοι υπερβαίνουν το 99,8% σε ορθότητα, αλλά το μακρο-F-score κυμαίνεται από 63,6% (HistGradientBoosting, του οποίου η αυτόματη πρόωρη διακοπή στα μεγάλα σύνολα φαίνεται να θυσιάζει τις κλάσεις των λίγων δεκάδων δειγμάτων) έως 97,8% (Bagging). Το εύρημα συνιστά πρακτική προειδοποίηση: σε εφαρμογές όπου οι σπάνιες κλάσεις είναι κρίσιμες, η αξιολόγηση οφείλει να γίνεται με μακρο-συναθροισμένες μετρικές και οι αυτοματισμοί των υλοποιήσεων να ελέγχονται προσεκτικά.

5.8 Συζήτηση

Συνοψίζοντας τις απαντήσεις στα ερευνητικά ερωτήματα: ως προς την ποιότητα (EE1), τρεις αλγόριθμοι — ExtraTrees, HistGradientBoosting, RandomForest — συγκροτούν μια στατιστικά μη διακρίσιμη κορυφή, με τον AdaBoost με τις προεπιλεγμένες ρυθμίσεις του να είναι ακατάλληλος για σύνθετα, πολυκλασικά προβλήματα. Ως προς το κόστος (EE2), η εκπαίδευση κυμαίνεται σε τέσσερις τάξεις μεγέθους με τον κλασικό GradientBoosting συστηματικά ακριβότερο και τους Bagging/AdaBoost/ExtraTrees φθηνότερους, ενώ στην ταξινόμηση η ιεραρχία αναστρέφεται υπέρ των boosting μεθόδων με ρηγά μέλη. Ως προς τον συμβιβασμό (EE3), κυρίαρχες επιλογές αναδεικνύονται οι ExtraTrees και HistGradientBoosting. Ως προς την επίδραση των χαρακτηριστικών του προβλήματος (EE4), το πλήθος των κλάσεων αναδείχθηκε καθοριστικότερο και από το μέγεθος: πολλαπλασιάζει το

κόστος του κλασικού gradient boosting και τιμωρεί τα αδύναμα μέλη του AdaBoost, ενώ αφήνει σχεδόν ανεπηρέαστες τις μεθόδους πλήρων δέντρων.

Τα ευρήματα εναρμονίζονται με τη βιβλιογραφία, την οποία και επικαιροποιούν. Η κυριαρχία των τυχαίων δασών των κλασικών μελετών [44], [43] επιβεβαιώνεται, πλην όμως με δύο ουσιώδεις προσθήκες: πρώτον, τα συχνά παραγνωρισμένα Extra-Trees αποδεικνύονται πλήρως ισοδύναμα σε ποιότητα και ταχύτερα στην εκπαίδευση, καθιστάμενα ίσως η καλύτερη «πρώτη επιλογή» μεταξύ των μεθόδων bagging· δεύτερον, το σύγχρονο ιστογραμμικό gradient boosting προσφέρει την ποιότητα της κορυφής με κόστος τάξεις μεγέθους μικρότερο του κλασικού, επιβεβαιώνοντας σε ελεγχόμενες συνθήκες τη συνεισφορά των Ke et al. [11]. Πρακτικά, για τον μηχανικό που επιλέγει αλγόριθμο χωρίς περιθώριο συντονισμού, τα αποτελέσματα υποδεικνύουν ExtraTrees ή HistGradientBoosting ως προεπιλογή, RandomForest ως ισοδύναμη συντηρητική εναλλακτική, και αποθαρρύνουν τη χρήση των AdaBoost και GradientBoosting με προεπιλεγμένες ρυθμίσεις — του πρώτου για λόγους ποιότητας, του δεύτερου για λόγους κόστους.

5.9 Περιορισμοί της μελέτης

Τα συμπεράσματα τελούν υπό ορισμένους περιορισμούς. Πρώτον, η σύγκριση αφορά προεπιλεγμένες ρυθμίσεις· με συντονισμό υπερπαραμέτρων η εικόνα θα μπορούσε να μεταβληθεί, ιδίως για τους AdaBoost (βαθύτερα μέλη, περισσότεροι γύροι) και GradientBoosting, και η σχετική διερεύνηση αποτελεί φυσική συνέχεια της εργασίας. Δεύτερον, τα δεκατρία προβλήματα, αν και επιλεγμένα με κριτήρια ποικιλίας, δεν εξαντλούν τον χώρο των εφαρμογών — απουσιάζουν, ενδεικτικά, προβλήματα με χιλιάδες χαρακτηριστικά ή με κυρίαρχα κατηγορικά χαρακτηριστικά. Τρίτον, οι χρόνοι μετρήθηκαν σε συγκεκριμένο υλικό και σε μονονηματική εκτέλεση· η άρση του περιορισμού του ενός νήματος θα ευνοούσε διαφορετικά τους αλγορίθμους, καθώς οι μέθοδοι bagging παραλληλοποιούνται πληρέστερα. Τέταρτον, ο έλεγχος Nemenyi με $N = 13$ έχει περιορισμένη ισχύ, συνεπώς η μη ανίχνευση διαφοράς μεταξύ των κορυφαίων δεν συνιστά απόδειξη ισοδυναμίας.

5.10 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν και αναλύθηκαν τα αποτελέσματα της πειραματικής σύγκρισης: η στατιστικά μη διακρίσιμη κορυφή των ExtraTrees, HistGradientBoosting και RandomForest, η κατάρρευση του AdaBoost στα πολυκλασικά προβλήματα, η δραματική υπεροχή της ιστογραμμικής υλοποίησης έναντι του κλασικού gradient boosting στο κόστος εκπαίδευσης, και η ανάδειξη του πλήθους των κλάσεων ως καθοριστικού παράγοντα. Το επόμενο κεφάλαιο συμπυκνώνει τα συμπεράσματα και σκιαγραφεί τις μελλοντικές επεκτάσεις.

Κεφάλαιο 6ο: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Το κεφάλαιο αυτό κλείνει την εργασία, αποτιμώντας συνολικά όσα προηγήθηκαν, τόσο ως προς τη βιβλιογραφική ανασκόπηση όσο και ως προς την πειραματική συγκριτική μελέτη. Τα κύρια συμπεράσματα συμπυκνώνονται σε πρακτικές κατευθύνσεις, χρήσιμες για τον μηχανικό που καλείται να επιλέξει αλγόριθμο συνόλου για ένα νέο πρόβλημα κατηγοριοποίησης, λαμβάνοντας υπόψη τόσο την προβλεπτική απόδοση όσο και το υπολογιστικό κόστος. Παράλληλα, αναγνωρίζονται τα ανοιχτά ζητήματα που δεν κατέστη δυνατό να καλυφθούν στο πλαίσιο της παρούσας μελέτης, τα οποία συνιστούν πρόσφορο έδαφος για μελλοντική έρευνα και συγκεκριμένες επεκτάσεις του πειραματικού πρωτοκόλλου.

6.1 Εισαγωγή

Το κεφάλαιο αυτό ολοκληρώνει την εργασία: συνοψίζει τα κύρια συμπεράσματα που προέκυψαν από τη βιβλιογραφική ανασκόπηση και την πειραματική μελέτη, και προτείνει συγκεκριμένες κατευθύνσεις για μελλοντική επέκταση.

6.2 Σύνοψη συμπερασμάτων

Η εργασία μελέτησε, βιβλιογραφικά και πειραματικά, τις δύο κυρίαρχες οικογένειες μεθόδων συνόλου ταξινομητών. Η βιβλιογραφική ανασκόπηση ανέδειξε το δίλημμα μεροληψίας-διακύμανσης ως το ενοποιητικό πλαίσιο κατανόησής τους: οι μέθοδοι της οικογένειας Bagging (Bagging, Random Forest, Extra-Trees) εκπαιδεύουν παράλληλα και ανεξάρτητα ισχυρά, ασταθή μοντέλα και μειώνουν τη διακύμανση μέσω τυχαιοποίησης και μέσου όρου, ενώ οι μέθοδοι Boosting (AdaBoost, Gradient Boosting και οι ιστογραμμικές υλοποιήσεις του) συνθέτουν σειριακά αδύναμα μοντέλα που εστιάζουν στα σφάλματα των προηγούμενων, μειώνοντας προοδευτικά τη μεροληψία.

Η πειραματική σύγκριση των έξι υλοποιήσεων του scikit-learn σε δεκατρία σύνολα δεδομένων του KEEL, με δεκαπλή στρωματοποιημένη διασταυρούμενη επικύρωση και προεπιλεγμένες παραμέτρους, κατέληξε σε σαφή και στατιστικά τεκμηριωμένα ευρήματα. Την κορυφή της προβλεπτικής απόδοσης μοιράζονται, χωρίς στατιστικώς σημαντική μεταξύ τους διαφορά, οι Extra-Trees, HistGradientBoosting και Random Forest· ο AdaBoost με τις προεπιλεγμένες ρυθμίσεις του υπολείπεται σημαντικά, με χαρακτηριστική κατάρρευση στα πολυκλασικά προβλήματα. Στο υπολογιστικό σκέλος, ο χρόνος εκπαίδευσης εκτείνεται σε τέσσερις τάξεις μεγέθους, με τον κλασικό GradientBoosting συστηματικά ακριβότερο — έως και 70 φορές αργότερο από την ιστογραμμική του εκδοχή σε μεγάλα πολυκλασικά προβλήματα — και με τα Extra-Trees σταθερά ταχύτερα του Random Forest, όπως προβλέπει η θεωρία τους. Συνολικά, τον καλύτερο συμβιβασμό ποιότητας-κόστους προσφέρουν τα Extra-Trees και ο HistGradientBoosting, οι οποίοι και προτείνονται ως προεπιλογή για προβλήματα κατηγοριοποίησης πινακοποιημένων δεδομένων χωρίς περιθώριο συντονισμού· καθοριστικός παράγοντας για την επιλογή αναδείχθηκε το πλήθος των κλάσεων του προβλήματος, περισσότερο και από το μέγεθός του.

6.3 Μελλοντικές επεκτάσεις

Η παρούσα μελέτη επιδέχεται φυσικές επεκτάσεις προς τέσσερις κατευθύνσεις. Η πρώτη και αμεσότερη είναι η μελέτη της επίδρασης των υπερπαραμέτρων: με διαχωρισμό των δεδομένων σε σύνολα εκπαίδευσης και ελέγχου, μπορεί να διερευνηθεί συστηματικά η ευαισθησία κάθε αλγορίθμου στις σημαντικότερες παραμέτρους του — πλήθος και βάθος μελών, ρυθμός μάθησης για τις μεθόδους boosting, πλήθος εξεταζόμενων χαρακτηριστικών για τα δάση — και να ελεγχθεί αν ο συντονισμός

μεταβάλλει την ιεραρχία των προεπιλογών, ιδίως για τους AdaBoost και GradientBoosting που εμφανίστηκαν αδικημένοι από αυτές. Η δεύτερη κατεύθυνση είναι η διεύρυνση των συγκρινόμενων υλοποιήσεων με τις εξωτερικές βιβλιοθήκες XGBoost, LightGBM και CatBoost, ώστε να αποτιμηθεί η θέση των υλοποιήσεων του scikit-learn έναντι του εξειδικευμένου οικοσυστήματος gradient boosting. Η τρίτη αφορά τη διεύρυνση των προβλημάτων: σύνολα δεδομένων με έντονη ανισοκατανομή κλάσεων σε συνδυασμό με τεχνικές εξισορρόπησης, με πολύ υψηλή διαστατικότητα, ή με κυρίαρχα κατηγορικά χαρακτηριστικά. Η τέταρτη, τέλος, αφορά την ερμηνευσιμότητα: η συγκριτική μελέτη των εκτιμήσεων σημαντικότητας χαρακτηριστικών και σύγχρονων εργαλείων ερμηνείας, όπως οι τιμές SHAP [47], θα προσέθετε στη σύγκριση μια διάσταση ολοένα κρισιμότερη για τις πραγματικές εφαρμογές.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, *Introduction to Data Mining*, 2nd ed. New York, NY, USA: Pearson, 2018.
- [2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [3] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [4] T. G. Dietterich, “Ensemble methods in machine learning,” in *Proc. 1st Int. Workshop on Multiple Classifier Systems (MCS)*, Cagliari, Italy, 2000, pp. 1–15.
- [5] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [6] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [7] P. Geurts, D. Ernst and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.
- [8] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [9] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [10] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
- [11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems 30 (NIPS)*, Long Beach, CA, USA, 2017, pp. 3146–3154.
- [12] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.
- [13] R. E. Schapire, “The strength of weak learnability,” *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [14] L. Grinsztajn, E. Oyallon and G. Varoquaux, “Why do tree-based models still outperform deep learning on typical tabular data?,” in *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, LA, USA, 2022, pp. 507–520.
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [16] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez and F. Herrera, “KEEL data-mining software tool: Data set repository, integration of algorithms and experimental

- analysis framework,” *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2–3, pp. 255–287, 2011.
- [17] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.
 - [18] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
 - [19] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*. Belmont, CA, USA: Wadsworth, 1984.
 - [20] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.
 - [21] S. Geman, E. Bienenstock and R. Doursat, “Neural networks and the bias/variance dilemma,” *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.
 - [22] P. Domingos, “A unified bias-variance decomposition and its applications,” in *Proc. 17th Int. Conf. on Machine Learning (ICML)*, Stanford, CA, USA, 2000, pp. 231–238.
 - [23] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.
 - [24] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
 - [25] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
 - [26] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proc. 14th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Montreal, Canada, 1995, pp. 1137–1143.
 - [27] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2004.
 - [28] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.
 - [29] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
 - [30] X. Dong, Z. Yu, W. Cao, Y. Shi and Q. Ma, “A survey on ensemble learning,” *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, 2020.
 - [31] L. Breiman, “Out-of-bag estimation,” *Statistics Department, University of California, Berkeley, CA, USA, Tech. Rep.*, 1996.
 - [32] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
 - [33] G. Biau and E. Scornet, “A random forest guided tour,” *TEST*, vol. 25, no. 2, pp. 197–227, 2016.
 - [34] Y. Freund, “Boosting a weak learning algorithm by majority,” *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995.

- [35] R. E. Schapire, Y. Freund, P. Bartlett and W. S. Lee, “Boosting the margin: A new explanation for the effectiveness of voting methods,” *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.
- [36] T. G. Dietterich, “An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization,” *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.
- [37] J. Zhu, H. Zou, S. Rosset and T. Hastie, “Multi-class AdaBoost,” *Statistics and Its Interface*, vol. 2, no. 3, pp. 349–360, 2009.
- [38] J. Friedman, T. Hastie and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [39] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.
- [40] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, “CatBoost: Unbiased boosting with categorical features,” in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, Montreal, Canada, 2018, pp. 6638–6648.
- [41] E. Bauer and R. Kohavi, “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants,” *Machine Learning*, vol. 36, no. 1–2, pp. 105–139, 1999.
- [42] P. Probst, A.-L. Boulesteix and B. Bischl, “Tunability: Importance of hyperparameters of machine learning algorithms,” *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.
- [43] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proc. 23rd Int. Conf. on Machine Learning (ICML)*, Pittsburgh, PA, USA, 2006, pp. 161–168.
- [44] M. Fernández-Delgado, E. Cernadas, S. Barro and D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems?,” *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.
- [45] C. Zhang, C. Liu, X. Zhang and G. Almpanidis, “An up-to-date comparison of state-of-the-art classification algorithms,” *Expert Systems with Applications*, vol. 82, pp. 128–150, 2017.
- [46] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [47] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30 (NIPS)*, Long Beach, CA, USA, 2017, pp. 4765–4774.

ΠΑΡΑΡΤΗΜΑ Α : ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΠΕΙΡΑΜΑΤΩΝ

Παρατίθεται ο πλήρης πηγαίος κώδικας εκτέλεσης των πειραμάτων (αρχείο `experiment.py`, Python 3, βιβλιοθήκες `scikit-learn` και `NumPy`).

```

"""
Πειραματική σύγκριση Bagging & Boosting ensemble αλγορίθμων κατηγοριοποίησης.
10-fold stratified cross validation, default παράμετροι, μέτρηση CPU time.

Χρήση: python3 experiment.py <dataset_name> <algo_name>
Αποτέλεσμα: results/<dataset>__<algo>.json
"""

import sys, os, json, time
import numpy as np
from sklearn.ensemble import (AdaBoostClassifier, BaggingClassifier,
                              ExtraTreesClassifier, GradientBoostingClassifier,
                              HistGradientBoostingClassifier, RandomForestClassifier)
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

DATA_DIR = os.path.join(os.path.dirname(os.path.abspath(__file__)), "keel_data")
RESULTS_DIR = os.path.join(os.path.dirname(os.path.abspath(__file__)), "results")
SEED = 42

ALGOS = {
    "AdaBoost": lambda: AdaBoostClassifier(random_state=SEED),
    "Bagging": lambda: BaggingClassifier(random_state=SEED),
    "ExtraTrees": lambda: ExtraTreesClassifier(random_state=SEED),
    "GradientBoosting": lambda: GradientBoostingClassifier(random_state=SEED),
    "HistGradientBoosting": lambda: HistGradientBoostingClassifier(random_state=SEED),
    "RandomForest": lambda: RandomForestClassifier(random_state=SEED),
}

def parse_keel(path):
    """Διαβάζει KEEL .dat αρχείο -> X (float array), y (labels)."""
    attrs, nominal = [], {}
    data_lines = []
    in_data = False
    with open(path, "r") as f:
        for line in f:
            line = line.strip()
            if not line:
                continue
            low = line.lower()
            if low.startswith("@attribute"):
                # όνομα attribute
                rest = line.split(None, 1)[1]
                name = rest.split("{")[0].split("[")[0].split()[0].strip()
                attrs.append(name)
                if "{" in rest: # nominal
                    vals = rest[rest.index("{") + 1: rest.index("}")]
                    nominal[len(attrs) - 1] = [v.strip() for v in vals.split(",")]
            elif low.startswith("@data"):
                in_data = True
            elif low.startswith("@"):
                continue
            elif in_data:
                data_lines.append([v.strip() for v in line.split(",")])

    n_attr = len(attrs)
    cls_idx = n_attr - 1 # στο KEEL η κλάση είναι το τελευταίο attribute
    X = np.empty((len(data_lines), n_attr - 1), dtype=float)
    y = []
    for i, row in enumerate(data_lines):
        for j in range(n_attr - 1):
            if j in nominal: # label encoding για nominal χαρακτηριστικά
                X[i, j] = nominal[j].index(row[j])
            else:
                X[i, j] = float(row[j])
    y.append(row[cls_idx])

```

```

    return X, np.array(y)

def run(dataset, algo_name):
    X, y = parse_keel(os.path.join(DATA_DIR, dataset + ".dat"))
    skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=SEED)
    folds = []
    for fold_i, (tr, te) in enumerate(skf.split(X, y)):
        clf = ALGOS[algo_name]()
        t0 = time.process_time()
        clf.fit(X[tr], y[tr])
        train_t = time.process_time() - t0
        t0 = time.process_time()
        pred = clf.predict(X[te])
        classif_t = time.process_time() - t0
        acc = accuracy_score(y[te], pred)
        prec, rec, f1, _ = precision_recall_fscore_support(
            y[te], pred, average="macro", zero_division=0)
        folds.append(dict(fold=fold_i, train_time=train_t, classif_time=classif_t,
            accuracy=acc, precision=prec, recall=rec, fscore=f1))
    print(f"{dataset}/{algo_name} fold {fold_i}: acc={acc:.4f} "
        f"train={train_t:.2f}s classif={classif_t:.3f}s", flush=True)

    mean = lambda k: float(np.mean([f[k] for f in folds]))
    out = dict(dataset=dataset, algorithm=algo_name,
        n_samples=int(X.shape[0]), n_features=int(X.shape[1]),
        n_classes=int(len(set(y))),
        train_time=mean("train_time"), classif_time=mean("classif_time"),
        accuracy=mean("accuracy"), precision=mean("precision"),
        recall=mean("recall"), fscore=mean("fscore"), folds=folds)
    os.makedirs(RESULTS_DIR, exist_ok=True)
    with open(os.path.join(RESULTS_DIR, f"{dataset}__{algo_name}.json"), "w") as f:
        json.dump(out, f, indent=1)
    print(f"DONE {dataset}/{algo_name}", flush=True)

DATASETS = ["pima", "vehicle", "contraceptive", "yeast", "seg

```

ΠΑΡΑΡΤΗΜΑ Β : ΑΠΟΤΕΛΕΣΜΑΤΩΝ

ΑΝΑΛΥΤΙΚΟΙ

ΠΙΝΑΚΕΣ

Παρατίθενται οι αναλυτικοί πίνακες της μακρο-συναθροισμένης ακρίβειας (precision) και ανάκλησης (recall) ανά σύνολο δεδομένων και αλγόριθμο. Με έντονη γραφή σημειώνεται η καλύτερη τιμή κάθε γραμμής. Τα πλήρη αποτελέσματα ανά fold συνοδεύουν την εργασία σε ηλεκτρονική μορφή.

Μακρο-συναθροισμένη ακρίβεια (precision):

Dataset	AdaBoost	Bagging	Extra-Trees	GB	HistGB	RF
pima	0,7261	0,7137	0,7620	0,7533	0,7246	0,7648
vehicle	0,6243	0,7502	0,7410	0,7573	0,7615	0,7350
contraceptive	0,5276	0,4885	0,4751	0,5582	0,5136	0,5096
yeast	0,3550	0,5148	0,5624	0,5317	0,4583	0,5901
segment	0,7995	0,9744	0,9811	0,9807	0,9878	0,9833
spambase	0,9279	0,9426	0,9573	0,9455	0,9550	0,9548
texture	0,6014	0,9650	0,9860	0,9821	0,9898	0,9801
satimage	0,7555	0,8778	0,9102	0,8916	0,9123	0,9104
ring	0,9210	0,9317	0,9726	0,9634	0,9706	0,9536
penbased	0,6693	0,9800	0,9942	0,9899	0,9929	0,9917
magic	0,8222	0,8672	0,8821	0,8759	0,8835	0,8809
letter	0,3331	0,9335	0,9744	0,9209	0,9675	0,9655
shuttle	0,5147	0,9824	0,9815	0,9697	0,6214	0,9674

Μακρο-συναθροισμένη ανάκληση (recall):

Dataset	AdaBoost	Bagging	Extra-Trees	GB	HistGB	RF
pima	0,7007	0,6798	0,7250	0,7327	0,7109	0,7316
vehicle	0,6340	0,7572	0,7540	0,7564	0,7627	0,7453
contraceptive	0,5133	0,4812	0,4693	0,5461	0,5086	0,5002
yeast	0,3491	0,4790	0,5393	0,5135	0,4471	0,5528
segment	0,8074	0,9736	0,9805	0,9801	0,9874	0,9827
spambase	0,9192	0,9361	0,9547	0,9398	0,9541	0,9503
texture	0,5811	0,9644	0,9856	0,9815	0,9895	0,9796
satimage	0,7313	0,8703	0,8889	0,8776	0,9023	0,8940
ring	0,9180	0,9303	0,9722	0,9619	0,9705	0,9527
penbased	0,6381	0,9797	0,9941	0,9897	0,9928	0,9916
magic	0,7975	0,8372	0,8483	0,8391	0,8559	0,8564
letter	0,2983	0,9319	0,9740	0,9188	0,9668	0,9647
shuttle	0,4935	0,9753	0,9700	0,9443	0,6593	0,9648