



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΕΡΓΑΣΤΗΡΙΟ ΕΝΣΥΡΜΑΤΗΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ**

**Παραγωγή Κανόνων από Βάσεις Δεδομένων
Σκακιστικών Φινάλε με Μεθόδους Μηχανικής
Μάθησης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΦΡΑΓΚΟΥΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ: ΣΓΑΡΜΠΑΣ ΚΥΡΙΑΚΟΣ

ΠΑΤΡΑ - ΙΟΥΝΙΟΣ 2026

Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

Φραγκούδης Κωνσταντίνος

© 2026 – Με την επιφύλαξη παντός δικαιώματος

Το σύνολο της εργασίας αποτελεί πρωτότυπο έργο, παραχθέν από τον Φραγκούδη Κωνσταντίνο, και δεν παραβιάζει δικαιώματα τρίτων καθ' οιονδήποτε τρόπο. Αν η εργασία περιέχει υλικό, το οποίο δεν έχει παραχθεί από τον ίδιο, αυτό είναι ευδιάκριτο και αναφέρεται ρητώς εντός του κειμένου της εργασίας ως προϊόν εργασίας τρίτου, σημειώνοντας με παρομοίως σαφή τρόπο τα στοιχεία ταυτοποίησής του, ενώ παράλληλα βεβαιώνει πως στην περίπτωση χρήσης αυτούσιων γραφικών αναπαραστάσεων, εικόνων, γραφημάτων κ.λπ., έχει λάβει τη χωρίς περιορισμούς άδεια του κατόχου των πνευματικών δικαιωμάτων για την συμπερίληψη και επακόλουθη δημοσίευση του υλικού αυτού.

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με τίτλο

Παραγωγή Κανόνων από Βάσεις Δεδομένων Σκακιστικών Φινάλε με Μεθόδους Μηχανικής Μάθησης

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

ΚΩΝΣΤΑΝΤΙΝΟΥ ΦΡΑΓΚΟΥΔΗ ΤΟΥ ΔΗΜΗΤΡΙΟΥ

Αριθμός Μητρώου: 1083873

Παρουσιάστηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών στις

6/7/2026

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Κυριάκος Σγάρμπας, Αν. Καθηγητής, ΗΜ&ΤΥ (επιβλέπων)

Κωνσταντίνος Μουστάκας, Καθηγητής, ΗΜ&ΤΥ (μέλος επιτροπής)

Κωνσταντίνος Χατζηλυγερούδης, Επ. Καθηγητής, ΗΜ&ΤΥ (μέλος
επιτροπής)

Ο Επιβλέπων

Κυριάκος Σγάρμπας
Αναπληρωτής Καθηγητής

Ο Διευθυντής του Τομέα

Σταύρος Κουλουρίδης
Καθηγητής

ΠΕΡΙΛΗΨΗ

Παραγωγή Κανόνων από Βάσεις Δεδομένων Σκακιστικών Φινάλε με Μεθόδους Μηχανικής Μάθησης

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΟΣ:

ΦΡΑΓΚΟΥΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΚΥΡΙΑΚΟΣ ΣΓΑΡΜΠΑΣ

Οι σκακιστικές βάσεις δεδομένων φινάλε περιέχουν την ακριβή θεωρητική αξιολόγηση κάθε νόμιμης θέσης για συγκεκριμένες διατάξεις υλικού. Παρότι η πληροφορία αυτή είναι πλήρης και ακριβής, παρέχεται σε αδιαφανή μορφή: οι tablebases απαντούν στην ερώτηση «τι είναι αυτή η θέση» χωρίς να εξηγούν «γιατί». Η παρούσα διπλωματική εργασία διερευνά τη δυνατότητα μετατροπής αυτής της τέλει γνώσης σε ερμηνεύσιμους σκακιστικούς κανόνες, με τη χρήση μεθόδων μηχανικής μάθησης και συγκεκριμένα δέντρων απόφασης.

Η προσέγγιση εφαρμόζεται σε τρία θεμελιώδη σκακιστικά φινάλε: το KPK (Βασιλιάς και Πιόνι εναντίον Βασιλιά), το KRK (Βασιλιάς και Πύργος εναντίον Βασιλιά), και το KQK (Βασιλιάς και Βασίλισσα εναντίον Βασιλιά). Για κάθε φινάλε αναπτύχθηκε ενιαία υποδομή που περιλαμβάνει την εξαντλητική παραγωγή όλων των νόμιμων θέσεων, την αξιολόγησή τους μέσω probing στις Syzygy tablebases, την εξαγωγή σκακιστικών χαρακτηριστικών, την εκπαίδευση και αξιολόγηση δέντρων απόφασης, και τη μετατροπή των παραγόμενων κανόνων σε ονοματισμένα σκακιστικά μοτίβα.

Πέρα από την ίδια την κατασκευή των μοντέλων, η εργασία εξετάζει συστηματικά τη σχέση ανάμεσα στην αναπαράσταση των δεδομένων και στην ποιότητα των παραγόμενων κανόνων. Για κάθε φινάλε συγκρίνονται τρία διαφορετικά επίπεδα χαρακτηριστικών — από τις ωμές συντεταγμένες των κομματιών, μέσω παράγωγων γεωμετρικών μεγεθών, έως πλήρη χαρακτηριστικά που ενσωματώνουν έννοιες της κλασικής θεωρίας φινάλε. Η σύγκριση αυτή απαντά σε ένα ερώτημα με ευρύτερη σημασία στη μηχανική μάθηση: πόση ανθρώπινη γνώση χρειάζεται να κωδικοποιηθεί εκ των προτέρων στα χαρακτηριστικά, και πόση μπορεί ένα μοντέλο να «ανακαλύψει» μόνο του από τα δεδομένα.

Τα αποτελέσματα δείχνουν ότι η ενσωμάτωση σκακιστικής γνώσης στα χαρακτηριστικά οδηγεί σε μοντέλα με σαφώς πιο συμπαγή και ερμηνεύσιμη δομή, ενώ οι κανόνες που παράγονται αντιστοιχούν σε αναγνωρίσιμες έννοιες της θεωρίας φινάλε: αγώνας δρόμου για την προαγωγή, οδήγηση του βασιλιά προς την άκρη, περιορισμός μέσω πύργου, αναγνώριση κινδύνου πατ. Η ενιαία μεθοδολογία εφαρμόζεται με συνέπεια και στα τρία endgames, εξασφαλίζοντας τη συγκρισιμότητα

των αποτελεσμάτων και ανοίγοντας τον δρόμο για την επέκταση σε πιο σύνθετα φινάλε.

Ο κώδικας της εργασίας είναι διαθέσιμος στο Github μέσω του συνδέσμου:
<https://github.com/GatitoMafia/chess-endgames-rules>

Η παρουσίαση που πραγματοποιήθηκε στα πλαίσια του " The 3 Minute Thesis " είναι διαθέσιμη στο YouTube μέσα από το σύνδεσμο:
<https://www.youtube.com/watch?v=IUyxTaBNqE>

EXTENSIVE ENGLISH SUMMARY

Derivation of Rules from Chess Endgame Databases with Machine Learning Methods

STUDENT NAME, SURNAME:

SUPERVISOR NAME, SURNAME:

KONSTANTINOS, FRAGKOUDIS

KYRIAKOS, SGARBAS

Chess endgame tablebases provide the exact theoretical evaluation of every legal position for specific material configurations. Although this information is complete and reliable, it is returned as a direct look-up result: tablebases indicate what the theoretical outcome of a position is, without explaining why. The present diploma thesis investigates whether this perfect tablebase knowledge can be transformed into interpretable chess rules using machine learning methods and, more specifically, decision trees.

The proposed approach is applied to three fundamental chess endgames: KPK (King and Pawn versus King), KRK (King and Rook versus King), and KQK (King and Queen versus King). For each endgame, a unified infrastructure was developed, including the exhaustive generation of legal positions, their evaluation through Syzygy tablebase probing, the extraction of chess-informed features, the training and evaluation of decision tree models, and the transformation of the produced rules into named chess motifs.

Beyond the construction of the models themselves, the thesis systematically examines the relationship between the representation of a chess position and the quality of the extracted rules. For each endgame, three different levels of features are compared: raw coordinates of the pieces, derived geometric features, and full strategic features that encode concepts from endgame theory. This comparison makes it possible to study how much chess knowledge needs to be encoded explicitly in the features, and how much structure a model can discover from the data alone.

The results show that incorporating chess-informed knowledge into the feature representation leads to models with more compact and interpretable rule structures. At the same time, the produced rules correspond to recognizable concepts from endgame theory, such as the race for promotion in KPK, the restriction of the black king and the creation of a mating net in KRK, and the queen's control of the opponent king's mobility in KQK. The proposed methodology is applied consistently across all three endgames, allowing the results to be compared under a common experimental framework and providing a basis for future extension to more complex endgames.

The code for this thesis is available on GitHub at the following link:
<https://github.com/GatitoMafia/chess-endgames-rules>

"The 3 Minute Thesis" presentation is available on YouTube at:
<https://www.youtube.com/watch?v=IUyixTaBNqE>

Περιεχόμενα

1. Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής εργασίας	1
1.2 Στόχος της εργασίας	2
1.3 Χρησιμότητα της εργασίας	3
1.4 Πρακτικές δυσκολίες και προκλήσεις	3
1.5 Σχετικές εργασίες.....	5
2. Γενικές Γνώσεις	6
2.1 Το σκάκι.....	6
2.2 Σκακιστικά φινάλε	7
2.3 Μηχανική μάθηση	8
2.3.1 Βασικές κατηγορίες μηχανικής μάθησης.....	8
2.3.2 Δεδομένα εισόδου και εξαγωγή χαρακτηριστικών	9
2.3.3 Δέντρα Απόφασης.....	10
2.3.4 Εξαγωγή κανόνων και ερμηνευσιμότητα.....	12
3. Endgame Tablebases	13
3.1 Εισαγωγή και ιστορική επισκόπηση.....	13
3.2 Τύποι endgame tablebases.....	14
3.2.1 Nalimov tablebases.....	14
3.2.2 Gaviota tablebases	15
3.2.3 Syzygy tablebases	15
3.2.4 Lomonosov tablebases	16
3.2.5 Scorpio bitbases και άλλες ειδικές μορφές	16
3.3 Εσωτερική δομή και αποθήκευση	17

3.3.1 Αντιστοίχιση θέσεων σε δείκτες	17
3.3.2 Εκμετάλλευση συμμετριών	17
3.3.3 Συμπίεση των τιμών.....	18
3.3.4 Πρακτικές συνέπειες της δομής.....	19
3.4 Πρόσβαση και εργαλεία.....	19
3.4.1 Τοπική εγκατάσταση και χρήση	20
3.4.2 Online υπηρεσίες.....	20
3.4.3 Προγραμματιστική πρόσβαση μέσω βιβλιοθηκών	21
3.5 Οι Syzygy tablebases στην παρούσα εργασία	21
3.5.1 Λόγοι επιλογής.....	21
3.5.2 Εγκατάσταση και τοπική ρύθμιση.....	22
3.5.3 Μηχανισμός probing μέσω python-chess	22
3.5.4 Διαφορετική αξιοποίηση WDL και DTZ ανά endgame	23
4. Η εφαρμογή	25
4.1 Επισκόπηση του pipeline	25
4.2 Φινάλε KPK (King and Pawn vs King)	26
4.2.1 Παραγωγή του dataset	26
4.2.2 Σχεδίαση των χαρακτηριστικών	26
4.2.3 Εκπαίδευση δέντρων απόφασης	31
4.2.4 Σύγκριση επιπέδων αναπαράστασης.....	32
4.2.5 Ομαδοποίηση κανόνων σε motifs	33
4.2.6 Οπτικοποίηση και τεκμηρίωση	34
4.2.7 Αποτελέσματα για το φινάλε KPK	35
4.3 Φινάλε KRK (King and Rook vs King).....	39

4.3.1 Παραγωγή του dataset	39
4.3.2 Σχεδίαση των χαρακτηριστικών	41
4.3.3 Εκπαίδευση δέντρων απόφασης	43
4.3.4 Σύγκριση επιπέδων αναπαράστασης.....	43
4.3.5 Ομαδοποίηση κανόνων σε motifs	44
4.3.6 Οπτικοποίηση και τεκμηρίωση	45
4.3.7 Αποτελέσματα για το φινάλε KRK.....	45
4.4 Φινάλε KQK (King and Queen vs King).....	51
4.4.1 Παραγωγή του dataset	51
4.4.2 Σχεδίαση των χαρακτηριστικών	52
4.4.3 Εκπαίδευση δέντρων απόφασης	53
4.4.4 Σύγκριση επιπέδων αναπαράστασης.....	54
4.4.5 Ομαδοποίηση κανόνων σε motifs	54
4.4.6 Οπτικοποίηση και τεκμηρίωση	55
4.4.7 Αποτελέσματα για το φινάλε KQK	55
5. Συμπεράσματα & Μελλοντικές Βελτιώσεις	60
5.1 Συμπεράσματα	60
5.2 Μελλοντικές βελτιώσεις.....	61
6. Βιβλιογραφία	63

1. Εισαγωγή

1.1 Αντικείμενο της διπλωματικής εργασίας

Το σκάκι αποτελεί ένα από τα πιο χαρακτηριστικά παραδείγματα παιχνιδιού πλήρους πληροφόρησης, στο οποίο δεν υπάρχει τυχαιότητα και κάθε θέση καθορίζεται αποκλειστικά από τη διάταξη των κομματιών και τη σειρά κίνησης. Παρά τους απλούς βασικούς κανόνες του, ο συνολικός αριθμός πιθανών θέσεων και ακολουθιών κινήσεων είναι εξαιρετικά μεγάλος, γεγονός που καθιστά το σκάκι ιδιαίτερα ενδιαφέρον αντικείμενο για την επιστήμη των υπολογιστών, την τεχνητή νοημοσύνη και τη μηχανική μάθηση.

Τα σκακιστικά φινάλε, δηλαδή οι θέσεις με μικρό αριθμό πεσσών στη σκακιέρα, καταλαμβάνουν μια ιδιαίτερη θέση στη θεωρία του σκακιού. Παρά το μειωμένο υλικό — ή ίσως ακριβώς εξαιτίας του — η ακρίβεια εδώ είναι καθοριστική: μια διαφορά ενός τετραγώνου στη θέση των βασιλιάδων μπορεί να μετατρέψει μια κερδισμένη θέση σε ισόπαλη. Για τον λόγο αυτό, τα φινάλε αποτελούν ένα πεδίο όπου η ανθρώπινη θεωρία και η υπολογιστική ανάλυση συναντιούνται με ξεκάθαρο τρόπο.

Η υπολογιστική πλευρά αυτής της συνάντησης έχει σήμερα μια εδραιωμένη μορφή. Τα endgame tablebases, βάσεις δεδομένων που περιέχουν την πλήρη ανάλυση όλων των νόμιμων θέσεων για συγκεκριμένα φινάλε, παρέχουν την «τέλεια» αξιολόγηση κάθε θέσης. Γνωρίζουμε για κάθε διάταξη αν είναι κερδισμένη, ισόπαλη ή χαμένη, καθώς και πόσες κινήσεις απαιτούνται μέχρι το αποτέλεσμα. Το ζητούμενο, ωστόσο, δεν είναι η ίδια η πληροφορία — αυτή είναι ήδη διαθέσιμη. Το ζητούμενο είναι η μετατροπή της σε κατανοητούς κανόνες.

Σε αυτό ακριβώς εστιάζει η παρούσα εργασία. Χρησιμοποιώντας τις Syzygy tablebases ως πηγή αξιόπιστων ετικετών, εκπαιδεύονται δέντρα απόφασης πάνω σε τρία θεμελιώδη φινάλε: KPK, KRK και KQK. Η επιλογή των Syzygy έναντι των κλασικών Nalimov δεν είναι αυθαίρετη — οι Syzygy διακρίνουν τα WDL (win/draw/loss) από τα DTZ (depth to zeroing) δεδομένα, παρέχοντας έτσι δύο διαφορετικούς τύπους πληροφορίας που αξιοποιούνται με διαφορετικό τρόπο σε κάθε φινάλε.

Πράγματι, τα τρία φινάλε δεν αντιμετωπίζονται με τον ίδιο τρόπο. Το KPK τίθεται ως δυαδικό πρόβλημα Win/Draw, καθώς η θεμελιώδης ερώτηση εδώ είναι αν ο λευκός μπορεί καν να προαγάγει το πιόνι. Αντίθετα, στα KRK και KQK οι περισσότερες θέσεις είναι κερδισμένες και το πραγματικό ερώτημα μετατοπίζεται στο πόσο γρήγορα: οι θέσεις ομαδοποιούνται σε φάσεις μετατροπής (immediate, short, medium, long) με βάση τις DTZ τιμές. Η διαφορετική αυτή διατύπωση ανά endgame αντικατοπτρίζει τη διαφορετική σκακιστική φύση τους.

Πέρα όμως από την κατασκευή των μοντέλων, ένα ξεχωριστό σκέλος της εργασίας αφορά τη σύγκριση διαφορετικών τρόπων αναπαράστασης των θέσεων. Για κάθε φινάλε δοκιμάζονται τρία επίπεδα χαρακτηριστικών: σκέτες συντεταγμένες των κομματιών, γεωμετρικά χαρακτηριστικά (αποστάσεις, σχετικές θέσεις), και στρατηγικά χαρακτηριστικά που κωδικοποιούν έννοιες της θεωρίας φινάλε όπως opposition, key squares, rook cutoff και queen box. Η σύγκριση αυτή απαντά σε ένα ερώτημα που υπερβαίνει την ακρίβεια του μοντέλου: μαθαίνει το δέντρο μόνο του τη σκακιστική

θεωρία από τα ωμά δεδομένα, ή χρειάζεται να του την κωδικοποιήσουμε εμείς στα features;

Τα δέντρα απόφασης επιλέχθηκαν συνειδητά ως κεντρικό μοντέλο, γιατί κάθε μονοπάτι από τη ρίζα σε ένα φύλλο μετατρέπεται απευθείας σε κανόνα «αν–τότε». Έτσι, οι προβλέψεις του μοντέλου μπορούν να συνδεθούν με σκακιστικές έννοιες — περιορισμός του αντίπαλου βασιλιά, opposition, αποφυγή πατ — αντί να παραμείνουν αδιαφανείς αριθμητικές εκτιμήσεις. Η εργασία δεν περιορίζεται λοιπόν στην ταξινόμηση θέσεων· καλύπτει ολόκληρη τη διαδρομή από την παραγωγή δεδομένων μέχρι την τελική εξαγωγή ερμηνεύσιμης γνώσης.

1.2 Στόχος της εργασίας

Ο στόχος της εργασίας δεν είναι η μεγιστοποίηση της ακρίβειας ταξινόμησης. Αν αυτό ήταν το ζητούμενο, δεν θα χρειαζόταν καν μοντέλο μηχανικής μάθησης: τα tablebases επιστρέφουν ήδη τη σωστή απάντηση για κάθε νόμιμη θέση, με ακρίβεια 100%. Η εργασία ξεκινά από μια διαφορετική παραδοχή, ότι η τέλεια αλλά αδιαφανής πληροφορία των tablebases έχει περιορισμένη αξία για όποιον θέλει να κατανοήσει ένα φινάλε, να το διδάξει ή να το μελετήσει θεωρητικά.

Το κεντρικό ερευνητικό ερώτημα διατυπώνεται έτσι: μπορούν τα δέντρα απόφασης, εκπαιδευμένα πάνω σε αυτή την τέλεια πληροφορία, να την «αποσυμπιέσουν» σε σύνολο κανόνων που να αναγνωρίζονται από έναν σκακιστή ως γνωστές έννοιες της θεωρίας φινάλε; Οι έννοιες αυτές διαφέρουν ανά endgame: στο KPK αφορούν τα key squares, την opposition και τον αγώνα δρόμου για την προαγωγή· στο KRK τον περιορισμό του βασιλιά με χρήση πύργου και βασιλιά μαζί, την opposition και τη ζώνη μάτ· στο KQK τη συστηματική στένωση του «κουτιού» της βασίλισσας γύρω από τον μαύρο βασιλιά, σε συνδυασμό με αποφυγή πατ.

Για να απαντηθεί το ερώτημα αυτό, η εργασία οργανώνεται γύρω από τρεις άξονες. Πρώτον, την κατασκευή ενιαίας υποδομής — από την παραγωγή θέσεων και την αξιολόγησή τους μέσω Syzygy probing, μέχρι την εκπαίδευση και αξιολόγηση των δέντρων — που εφαρμόζεται με ομοιόμορφο τρόπο και στα τρία endgames. Δεύτερον, τη σύγκριση τριών διαφορετικών αναπαραστάσεων της θέσης, ώστε να εκτιμηθεί πόση σκακιστική γνώση χρειάζεται να ενσωματωθεί στα ίδια τα features και πόση μπορεί το δέντρο να «ανακαλύψει» μόνο του. Και τρίτον, τη μετατροπή των παραγόμενων leaf rules σε ευρύτερες ερμηνευτικές ομάδες (motifs) που να αντιστοιχούν σε ονοματισμένα στρατηγικά μοτίβα της θεωρίας φινάλε.

Επιτυχία της εργασίας, με αυτή τη διατύπωση, δεν είναι ένας υψηλός αριθμός ακρίβειας. Είναι ένα σύνολο κανόνων που ένας σκακιστής θα αναγνώριζε ως «αυτό που πραγματικά συμβαίνει» σε κάθε φινάλε.

1.3 Χρησιμότητα της εργασίας

Η χρησιμότητα μιας εργασίας που εξάγει κανόνες από tablebases μπορεί εύλογα να αμφισβητηθεί: αν τα tablebases ήδη δίνουν την τέλεια απάντηση, γιατί χρειάζονται «κανόνες»; Η απάντηση βρίσκεται στη διαφορά ανάμεσα στο να ξέρεις την απάντηση και στο να την καταλαβαίνεις. Ένα tablebase probe επιστρέφει «κερδισμένη σε 12 κινήσεις». Δεν λέει γιατί. Δεν λέει ποια χαρακτηριστικά της θέσης την κάνουν κερδισμένη, ούτε ποια θα την έκαναν ισόπαλη. Αυτή τη διαφορά καλούνται να καλύψουν οι ερμηνεύσιμοι κανόνες.

Σε εκπαιδευτικό επίπεδο, οι κανόνες που εξάγονται μπορούν να λειτουργήσουν συμπληρωματικά προς τα κλασικά εγχειρίδια φινάλε. Τα εγχειρίδια παρουσιάζουν επιλεγμένες «τυπικές» θέσεις και τις σχετικές τεχνικές, αλλά αναγκαστικά καλύπτουν ένα κλάσμα του χώρου. Ένα δέντρο απόφασης εκπαιδευμένο πάνω σε ολόκληρο τον χώρο ενός φινάλε δίνει κανόνες που ισχύουν παντού — όχι μόνο στις θέσεις που επέλεξε ο συγγραφέας του εγχειριδίου. Σε έναν αρχάριο ή ενδιάμεσο σκακιστή που μελετά το KPK, ένας κανόνας της μορφής «αν ο μαύρος βασιλιάς μπορεί να μπλοκάρει την προαγωγή και κερδίζει τον αγώνα για το key square, η θέση είναι ισόπαλη» είναι πιο άμεσα αξιοποιήσιμος από μια λίστα είκοσι θεωρητικών θέσεων προς απομνημόνευση.

Σε ερευνητικό επίπεδο, η εργασία έχει αξία περισσότερο ως μεθοδολογικό υπόδειγμα παρά ως αυτοτελής συνεισφορά στη θεωρία φινάλε. Το ερώτημα «πώς εξάγουμε ερμηνεύσιμη γνώση από μια πλήρως υπολογισμένη βάση γνώσης;» εμφανίζεται σε αρκετά πεδία πέρα από το σκάκι — από ιατρικά συστήματα υποστήριξης αποφάσεων (όπου ένα μοντέλο μπορεί να προβλέπει σωστά αλλά πρέπει να εξηγήσει γιατί), μέχρι πεδία όπως η ανάλυση παιχνιδιών στρατηγικής, όπου υπάρχει βέλτιστη λύση αλλά λείπει η ανθρώπινη αναπαράσταση. Τα σκακιστικά φινάλε προσφέρουν ένα καθαρό πειραματικό περιβάλλον για αυτό το ερώτημα: ο χώρος είναι πεπερασμένος, η αλήθεια είναι γνωστή, και η ποιότητα των κανόνων μπορεί να ελεγχθεί τόσο αριθμητικά (ακρίβεια) όσο και ποιοτικά (αναγνωρίζονται ως γνωστά μοτίβα από έναν ειδικό).

Ένα ιδιαίτερο σημείο που η εργασία επιχειρεί να φωτίσει είναι η σχέση ανάμεσα στην αναπαράσταση των δεδομένων και στην ποιότητα των κανόνων. Η σύγκριση raw coordinates, geometric και full strategic features δίνει συγκεκριμένη απάντηση στο ερώτημα «πόση ανθρώπινη γνώση πρέπει να κωδικοποιηθεί εκ των προτέρων;». Είναι ερώτημα που στη ML κοινότητα παραμένει ανοιχτό και η απάντηση εξαρτάται από το πεδίο.

1.4 Πρακτικές δυσκολίες και προκλήσεις

Η εκπόνηση της παρούσας εργασίας συνοδεύτηκε από αρκετές πρακτικές και μεθοδολογικές δυσκολίες, που σχετίζονται τόσο με τη φύση των σκακιστικών φινάλε όσο και με τον στόχο της παραγωγής ερμηνεύσιμων αποτελεσμάτων. Παρότι τα endgame tablebases προσφέρουν ακριβή αξιολόγηση κάθε θέσης, η μετατροπή αυτής της πληροφορίας σε οργανωμένα datasets, ουσιαστικά χαρακτηριστικά, αξιόπιστα μοντέλα ταξινόμησης και τελικά, σε κατανοητούς κανόνες, αποδείχθηκε πολύ πιο σύνθετη απ' όσο θα περίμενε κανείς εκ πρώτης όψεως.

Μια πρώτη δυσκολία αφορούσε τον ορισμό των κατηγοριών ταξινόμησης, ιδιαίτερα στα φινάλε KRK και KQK. Οι θέσεις σε αυτά τα φινάλε δεν ταξινομήθηκαν απλώς ως

κερδισμένες ή ισόπαλες, αλλά ομαδοποιήθηκαν σε φάσεις μετατροπής — immediate, short, medium και long conversion — με βάση τιμές που προκύπτουν από το DTZ. Η επιλογή των ορίων ανάμεσα στις φάσεις αυτές είναι ιδιαίτερα κρίσιμη, καθώς οι θέσεις κοντά στα σημεία μετάβασης συχνά παρουσιάζουν παρόμοια γεωμετρικά και στρατηγικά χαρακτηριστικά. Αυτό αντικατοπτρίζεται και στα αποτελέσματα: η ακρίβεια του μοντέλου επηρεάζεται περισσότερο, ακριβώς στα όρια μεταξύ γειτονικών φάσεων, όπου η διάκριση είναι δύσκολη ακόμη και με πλούσιο σύνολο χαρακτηριστικών.

Μια δεύτερη πρόκληση σχετίζεται με την επιλογή και τη σύγκριση των χαρακτηριστικών που περιγράφουν κάθε θέση. Η χρήση μόνο των συντεταγμένων των κομματιών δίνει στο μοντέλο τη βασική γεωμετρική πληροφορία, αλλά οδηγεί σε κανόνες που είναι δύσκολο να ερμηνευθούν σκακιστικά. Από την άλλη πλευρά, τα γεωμετρικά και στρατηγικά χαρακτηριστικά — αποστάσεις βασιλιάδων, περιορισμός του αντίπαλου βασιλιά, opposition, blockade, δυνατότητα προαγωγής — προσφέρουν σαφώς μεγαλύτερη ερμηνευσιμότητα, απαιτούν όμως προσεκτικό σχεδιασμό ώστε να μην ενσωματώνουν εκ των προτέρων την ίδια τη λύση του προβλήματος. Για τον λόγο αυτό, η εργασία δεν περιορίστηκε σε ένα μόνο σύνολο χαρακτηριστικών, αλλά συνέκρινε διαφορετικές αναπαραστάσεις, ώστε να φανεί καθαρά η επίδραση κάθε επιπέδου πληροφορίας στην απόδοση και στη σαφήνεια των κανόνων.

Ιδιαίτερη δυσκολία προέκυψε επίσης στην ισορροπία ανάμεσα στην ακρίβεια του μοντέλου και στην πολυπλοκότητα του δέντρου απόφασης. Ένα βαθύτερο δέντρο μπορεί να πετύχει υψηλότερη ακρίβεια, καθώς προσαρμόζεται καλύτερα στις λεπτομέρειες των δεδομένων, αλλά παράγει και πολύ περισσότερα leaves — και κατά συνέπεια πολύ περισσότερους κανόνες. Όσο πιο πολλοί οι κανόνες, τόσο πιο δύσκολη γίνεται η συνολική ερμηνεία του μοντέλου. Χρειάστηκε επομένως μια προσεκτική επιλογή των παραμέτρων εκπαίδευσης, ώστε το μοντέλο να διατηρεί ικανοποιητική απόδοση χωρίς να καταλήγει σε υπερβολικά πολύπλοκο ή δυσανάγνωστο σύνολο κανόνων.

Σημαντική πρόκληση αποτέλεσε τέλος το στάδιο του rule consolidation. Τα decision trees τείνουν να παράγουν πολλά διαφορετικά leaf rules, ειδικά όταν το δέντρο είναι βαθύ ή όταν το πρόβλημα έχει αρκετές οριακές περιπτώσεις. Εξετασμένοι μεμονωμένα, οι κανόνες αυτοί είναι συχνά υπερβολικά πολλοί και τεχνικοί για να οδηγήσουν άμεσα σε χρήσιμα συμπεράσματα. Αναπτύχθηκε για τον λόγο αυτό μια διαδικασία ομαδοποίησης των leaf rules σε ευρύτερες ερμηνευτικές κατηγορίες και σκακιστικά motifs. Με τον τρόπο αυτό, τα επιμέρους τεχνικά rules μετατρέπονται σε πιο συμπαγείς ομάδες κανόνων, που μπορούν να συνδεθούν με γνωστές έννοιες της θεωρίας φινάλε και να παρουσιαστούν σε μορφή αναγνωρίσιμη από έναν σκακιστή.

Συνολικά, οι βασικές δυσκολίες της εργασίας δεν περιορίστηκαν στην εκπαίδευση ενός μοντέλου ταξινόμησης. Αφορούσαν ολόκληρη τη διαδρομή: από τη δημιουργία αξιόπιστων datasets και τον ορισμό κατάλληλων κατηγοριών, μέχρι τη σύγκριση διαφορετικών αναπαραστάσεων, την επιλογή ισορροπημένης πολυπλοκότητας και την τελική μετατροπή των παραγόμενων κανόνων σε ερμηνεύσιμα σκακιστικά συμπεράσματα.

1.5 Σχετικές εργασίες

Ως σχετική προηγούμενη εργασία μπορεί να αναφερθεί η διπλωματική εργασία του Θάνου Αναστάσιου με τίτλο «Συμπύεση βάσης δεδομένων σκακιστικών φινάλε με μεθόδους data mining» [1]. Η εργασία αυτή ασχολείται με βάσεις δεδομένων σκακιστικών φινάλε και αξιοποιεί μεθόδους data mining για τη μελέτη και συμπύεση πληροφορίας που προέρχεται από τις βάσεις Nalimon. Στο πλαίσιο της εξετάζονται φινάλε όπως KPK, KQK, KRK και ορισμένα φινάλε τεσσάρων κομματιών, με χρήση του λογισμικού Weka και με στόχο την εκπαίδευση ταξινομητών που προσεγγίζουν την πληροφορία των βάσεων.

Η παρούσα εργασία έχει κοινό σημείο εκκίνησης με την παραπάνω διπλωματική, καθώς αξιοποιεί βάσεις δεδομένων σκακιστικών φινάλε και μεθόδους μηχανικής μάθησης. Ωστόσο, διαφοροποιείται ως προς τον βασικό της στόχο και τη μεθοδολογία. Η προηγούμενη εργασία επικεντρώνεται κυρίως στη συμπύεση της πληροφορίας των βάσεων και στη μελέτη της ακρίβειας ταξινομητών. Αντίθετα, η παρούσα εργασία δίνει έμφαση στη δημιουργία ενιαίας pipeline για τα φινάλε KPK, KRK και KQK, στη σύγκριση διαφορετικών συνόλων χαρακτηριστικών, στην αξιολόγηση της σταθερότητας των μοντέλων και στην εξαγωγή ερμηνεύσιμων κανόνων.

Επιπλέον, στην παρούσα εργασία χρησιμοποιούνται Syzygy tablebases αντί για Nalimon, ενώ τα πειράματα οργανώνονται γύρω από σύγχρονες διαδικασίες αξιολόγησης, όπως train/test split, cross-validation, multi-seed evaluation, feature-set comparison και rule consolidation. Με αυτόν τον τρόπο, το ενδιαφέρον δεν περιορίζεται στη μείωση ή συμπύεση της πληροφορίας, αλλά επεκτείνεται στη μετατροπή της tablebase πληροφορίας σε ανθρώπινα κατανοητά σκακιστικά motifs και κανόνες.

2. Γενικές Γνώσεις

2.1 Το σκάκι

Το σκάκι είναι ένα επιτραπέζιο παιχνίδι στρατηγικής, στο οποίο αντιπαρατίθενται δύο παίκτες σε μια τετράγωνη σκακιέρα 8×8, αποτελούμενη από εναλλασσόμενα φωτεινά και σκούρα τετράγωνα. Κάθε παίκτης ξεκινά με δεκαέξι πεσσούς έξι διαφορετικών τύπων (βασιλιάς, βασίλισσα, πύργοι, αξιωματικοί, ίπποι και πιόνια), οι οποίοι κινούνται σύμφωνα με συγκεκριμένους κανόνες. Σύμφωνα με τους κανονισμούς της FIDE [2], ο σκοπός κάθε παίκτη είναι να θέσει τον αντίπαλο βασιλιά υπό τέτοια απειλή (σαχ-ματ), ώστε ο αντίπαλος να μην έχει καμία νόμιμη κίνηση που να αποτρέπει την αιχμαλωσία του.

Το σύγχρονο σκάκι έχει ιστορία που εκτείνεται σε περισσότερους από δεκαπέντε αιώνες, με καταγωγή από παιχνίδια στρατηγικής που εμφανίστηκαν στην Ινδία τον 6ο αιώνα μ.Χ., όπως το chaturanga, και εξελίχθηκε σταδιακά στη μορφή που είναι γνωστή σήμερα στην Ευρώπη περίπου από τον 15ο–16ο αιώνα. Σήμερα θεωρείται κλασικό παράδειγμα παιχνιδιού πλήρους πληροφόρησης, χωρίς στοιχεία τύχης, με τεράστιο χώρο καταστάσεων και υψηλή υπολογιστική πολυπλοκότητα, γεγονός που το καθιστά προνομιακό αντικείμενο μελέτης στην επιστήμη των υπολογιστών και στη μηχανική μάθηση. Η ανάλυση σκακιστικών θέσεων και παρτίδων έχει οδηγήσει στην ανάπτυξη εξειδικευμένων αλγορίθμων αναζήτησης, σκακιστικών μηχανών και πιο πρόσφατα, σε εφαρμογές μεθόδων τεχνητής νοημοσύνης για την κατανόηση και αυτοματοποίηση της διαδικασίας λήψης αποφάσεων στο παιχνίδι [3].



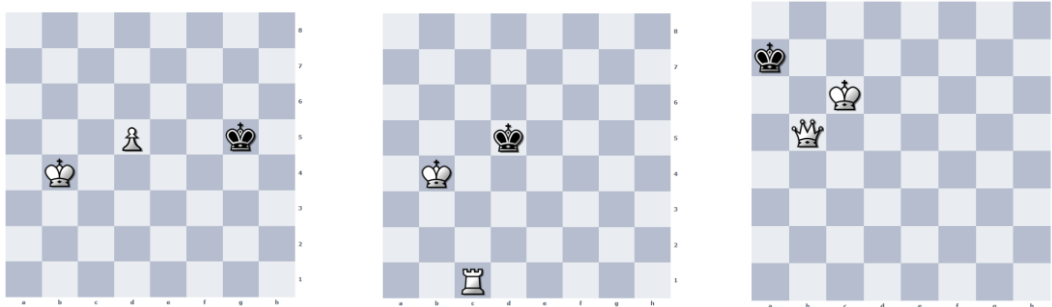
Εικόνα 2.1 Παραδοσιακή σκακιέρα

2.2 Σκακιστικά φινάλε

Το φινάλε (endgame) είναι η φάση της σκακιστικής παρτίδας που χαρακτηρίζεται από περιορισμένο αριθμό πεσσών στη σκακιέρα, συνήθως μετά την ανταλλαγή των περισσότερων κομματιών του ανοίγματος και του μεσαίου παιχνιδιού. Σε αυτή τη φάση, η δραστηριότητα των βασιλιάδων αυξάνεται, τα πιόνια αποκτούν κεντρικό ρόλο ως υποψήφια για προαγωγή και μικρές διαφορές στη θέση μπορούν να καθορίσουν αν μια παρτίδα είναι κερδισμένη, ισόπαλη ή χαμένη. Η θεωρία των φινάλε έχει αναπτυχθεί συστηματικά μέσα από αναλύσεις ισχυρών παικτών και θεωρητικών, με αποτέλεσμα να υπάρχουν κλασικά μοτίβα και «τεχνικές» που επαναλαμβάνονται σε πλήθος διαφορετικών θέσεων [4].

Στα φινάλε πιονιών, όπως τα κλασικά KPK (βασιλιάς–πιόνι εναντίον βασιλιά), κεντρικές έννοιες είναι τα «τετράγωνα-κλειδιά» (key squares), η έννοια του αγώνα δρόμου για την προαγωγή και η ορθή τοποθέτηση των βασιλιάδων, που καθορίζουν αν το επιθετικό πιόνι θα καταφέρει να προαχθεί ή αν η άμυνα μπορεί να κρατήσει την ισοπαλία. Στα φινάλε βαριών κομματιών, όπως το KRK (βασιλιάς–πύργος εναντίον βασιλιάς), η θεωρία παρέχει ολοκληρωμένες μεθόδους ματ, όπου η συνεργασία του πύργου με τον βασιλιά οδηγεί σταδιακά τον αντίπαλο βασιλιά στην άκρη της σκακιέρας και τελικά, σε θέση ματ μέσα από συγκεκριμένα μοτίβα περιορισμού. Αντίστοιχα, στα KQK (βασιλιάς–βασίλισσα εναντίον βασιλιάς), η πλευρά με τη βασίλισσα διαθέτει τεχνικά απλούστερη, αλλά όχι εντελώς προφανή διαδικασία ματ, η οποία βασίζεται σε συστηματικό περιορισμό των τετραγώνων διαφυγής του αντίπαλου βασιλιά και σε αποφυγή πατ, και έχει τυποποιηθεί σε σειρά κλασικών σχημάτων και τεχνικών. Παρά το ότι αυτά τα φινάλε θεωρούνται «θεμελιώδη» και συχνά διδάσκονται σε αρχικό στάδιο, εξακολουθούν να παρουσιάζουν πλούσια δομή σε επίπεδο στρατηγικών μοτίβων και ακριβών ακολουθιών κινήσεων, ιδιαίτερα όταν εξετάζονται σε πλήρη γενικότητα.

Λόγω του μειωμένου υλικού, τα φινάλε αποτέλεσαν από νωρίς πεδίο στο οποίο η ανθρώπινη θεωρία και η υπολογιστική ανάλυση μπορούσαν να συναντηθούν. Παραδοσιακά, η γνώση των τελικών βασιζόταν σε εγχειρίδια θεωρίας φινάλε και σε επιλεγμένες, «τυπικές» θέσεις, που θεωρούνται υποχρεωτικές για κάθε σοβαρό σκακιστή. Η εμφάνιση των endgame tablebases [4], δηλαδή πλήρως υπολογισμένων βάσεων δεδομένων φινάλε για συγκεκριμένους συνδυασμούς πεσσών, επέτρεψε να επιβεβαιωθούν ή και να αναθεωρηθούν πολλές κλασικές αξιολογήσεις θέσεων, αλλά ταυτόχρονα δημιούργησε έναν τεράστιο όγκο πληροφορίας δύσκολο να αποδοθεί σε συμπαγείς, ανθρώπινα ερμηνεύσιμους κανόνες. Η πρόκληση της σύνοψης αυτής της «τέλειας» πληροφορίας σε απλούς κανόνες και μοτίβα, κατανοητά από ανθρώπους, βρίσκεται στον πυρήνα πολλών σύγχρονων προσεγγίσεων που συνδέουν τη θεωρία φινάλε με μεθόδους μηχανικής μάθησης.



Εικόνα 2.2 Παραδείγματα σκακιστικών φινάλε

2.3 Μηχανική μάθηση

2.3.1 Βασικές κατηγορίες μηχανικής μάθησης

Η μηχανική μάθηση (*machine learning* [9]) αποτελεί έναν από τους βασικούς κλάδους της τεχνητής νοημοσύνης και ασχολείται με την ανάπτυξη μεθόδων που επιτρέπουν σε ένα υπολογιστικό σύστημα να μαθαίνει από δεδομένα. Η βασική διαφορά της από έναν κλασικό, πλήρως χειροκίνητο προγραμματισμό είναι ότι ο σχεδιαστής δεν χρειάζεται να ορίσει ρητά όλους τους κανόνες που διέπουν το πρόβλημα. Αντίθετα, ο αλγόριθμος δέχεται παραδείγματα, αναζητά επαναλαμβανόμενα μοτίβα και δημιουργεί ένα μοντέλο που μπορεί να χρησιμοποιηθεί για πρόβλεψη, ταξινόμηση ή λήψη αποφάσεων σε νέα δεδομένα.

Η προσέγγιση αυτή είναι ιδιαίτερα χρήσιμη σε προβλήματα όπου η πλήρης αναλυτική περιγραφή είναι δύσκολη ή πρακτικά αδύνατη. Σε πολλές περιπτώσεις υπάρχουν διαθέσιμα πολλά δεδομένα, αλλά η εξαγωγή κανόνων από άνθρωπο είναι χρονοβόρα ή ανεπαρκής λόγω της πολυπλοκότητας του προβλήματος. Για τον λόγο αυτό, η μηχανική μάθηση εφαρμόζεται σήμερα σε πολλά πεδία, όπως η αναγνώριση προτύπων, η υπολογιστική όραση, η επεξεργασία φυσικής γλώσσας, η ανάλυση δεδομένων και η μελέτη παιχνιδιών στρατηγικής. Το σκάκι αποτελεί χαρακτηριστικό παράδειγμα τέτοιου πεδίου, καθώς ο αριθμός των πιθανών θέσεων και κινήσεων είναι εξαιρετικά μεγάλος.

Οι μέθοδοι μηχανικής μάθησης διακρίνονται συνήθως σε τρεις βασικές κατηγορίες: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση [9] [10]. Η διάκριση αυτή βασίζεται κυρίως στον τρόπο με τον οποίο παρέχονται τα δεδομένα στο σύστημα και στο είδος του προβλήματος που καλείται να λύσει το μοντέλο.

Η επιβλεπόμενη μάθηση (*supervised learning*) βασίζεται σε δεδομένα για τα οποία είναι γνωστή εκ των προτέρων η σωστή έξοδος. Κάθε παράδειγμα εκπαίδευσης συνοδεύεται από μια ετικέτα (*label*) και ο στόχος του αλγορίθμου είναι να μάθει τη σχέση ανάμεσα στα χαρακτηριστικά εισόδου και στην αντίστοιχη έξοδο. Τυπικά προβλήματα αυτής της κατηγορίας είναι η ταξινόμηση, όπου η έξοδος ανήκει σε μία από συγκεκριμένες κλάσεις, και η παλινδρόμηση, όπου η έξοδος είναι αριθμητική τιμή. Η επιβλεπόμενη μάθηση είναι η κατηγορία που συνδέεται άμεσα με την

παρούσα εργασία, καθώς οι θέσεις των σκακιστικών φινάλε συνοδεύονται από αξιόπιστες ετικέτες που προέρχονται από endgame tablebases.

Η μη επιβλεπόμενη μάθηση (*unsupervised learning*) εφαρμόζεται όταν τα δεδομένα δεν συνοδεύονται από προκαθορισμένες ετικέτες. Σε αυτή την περίπτωση, ο αλγόριθμος προσπαθεί να εντοπίσει εσωτερικές δομές, ομοιότητες ή ομάδες μέσα στα δεδομένα. Χαρακτηριστικό παράδειγμα είναι η συσταδοποίηση (*clustering*), όπου τα δεδομένα χωρίζονται σε ομάδες με βάση την ομοιότητά τους. Αν και η μη επιβλεπόμενη μάθηση δεν αποτελεί τον κύριο άξονα της παρούσας εργασίας, η αναφορά της είναι χρήσιμη για την κατανόηση της θέσης της επιβλεπόμενης μάθησης στο ευρύτερο πεδίο.

Η ενισχυτική μάθηση (*reinforcement learning*) βασίζεται στην αλληλεπίδραση ενός πράκτορα με ένα περιβάλλον. Ο πράκτορας επιλέγει ενέργειες, παρατηρεί τα αποτελέσματά τους και λαμβάνει ανταμοιβές ή ποινές, με στόχο να μάθει μια στρατηγική που μεγιστοποιεί τη συνολική ανταμοιβή σε βάθος χρόνου. Η προσέγγιση αυτή έχει ιδιαίτερο ενδιαφέρον σε προβλήματα παιχνιδιών και λήψης αποφάσεων, επειδή κάθε κίνηση μπορεί να επηρεάσει πολλές μελλοντικές καταστάσεις. Παρότι η ενισχυτική μάθηση συνδέεται στενά με το σκάκι, στην παρούσα εργασία δεν χρησιμοποιείται τέτοια προσέγγιση. Το ζητούμενο εδώ δεν είναι η εκμάθηση μιας στρατηγικής μέσω δοκιμών και ανταμοιβών, αλλά η εξαγωγή ερμηνεύσιμων κανόνων από ήδη αξιολογημένες θέσεις φινάλε.

Για τον λόγο αυτό, η επιβλεπόμενη μάθηση είναι η πιο κατάλληλη προσέγγιση για τη συγκεκριμένη μελέτη. Οι σκακιστικές θέσεις μπορούν να περιγραφούν με δομημένα χαρακτηριστικά, ενώ οι endgame tablebases παρέχουν τις αντίστοιχες ετικέτες με υψηλή αξιοπιστία. Έτσι, είναι δυνατή η εκπαίδευση μοντέλων που δεν στοχεύουν μόνο στην πρόβλεψη της σωστής κατηγορίας, αλλά και στην παραγωγή κανόνων που μπορούν να διαβαστούν και να ερμηνευθούν από άνθρωπο. Στα επόμενα τμήματα, η έμφαση δίνεται στα δεδομένα, στα χαρακτηριστικά που εξάγονται από τις σκακιστικές θέσεις και στα δέντρα απόφασης, τα οποία χρησιμοποιούνται ως ερμηνεύσιμα μοντέλα για την παραγωγή κανόνων.

2.3.2 Δεδομένα εισόδου και εξαγωγή χαρακτηριστικών

Η απόδοση ενός μοντέλου μηχανικής μάθησης εξαρτάται άμεσα από την ποιότητα και τη μορφή των δεδομένων που χρησιμοποιούνται στην εκπαίδευση. Σε πλαίσιο επιβλεπόμενης μάθησης, κάθε παράδειγμα αποτελείται από ένα διάνυσμα χαρακτηριστικών (*features*), που περιγράφουν την είσοδο, και από μια ετικέτα (*label*), που αντιστοιχεί στην επιθυμητή έξοδο. Στη συγκεκριμένη εργασία, κάθε παράδειγμα είναι μια σκακιστική θέση φινάλε, τα χαρακτηριστικά συνοψίζουν γεωμετρικές και στρατηγικές πληροφορίες της θέσης, ενώ οι ετικέτες προέρχονται από τις endgame tablebases (π.χ. αποτέλεσμα ή κατηγορία φάσης με βάση DTZ ή παρόμοια μέτρα).

Η επιλογή και ο σχεδιασμός των χαρακτηριστικών (*feature engineering*) [11] [12] είναι κρίσιμη, επειδή το μοντέλο «βλέπει» το πρόβλημα μόνο μέσα από αυτά. Πέρα από τις απλές συντεταγμένες των κομματιών, είναι συχνά χρήσιμο να δημιουργούνται παράγωγα χαρακτηριστικά, όπως αποστάσεις μεταξύ βασιλιάδων, σχετική θέση της βασίλισσας ή του πύργου, βαθμός περιορισμού του αντίπαλου βασιλιά, ή ενδείξεις για *opposition* και υποστήριξη πιονιών, ώστε η αναπαράσταση να αντανakλά καλύτερα έννοιες της θεωρίας φινάλε. Με αυτόν τον τρόπο διευκολύνεται

η εκμάθηση κανόνων που έχουν άμεση σκακιστική ερμηνεία και όχι μόνο γεωμετρικό χαρακτήρα.

Η προεπεξεργασία των δεδομένων περιλαμβάνει ένα σύνολο βημάτων με στόχο τα δεδομένα να γίνουν συνεπή και κατάλληλα για εκπαίδευση. Ενδεικτικά, μπορεί να απαιτείται καθαρισμός του συνόλου από διπλές ή λανθασμένες εγγραφές, χειρισμός ελλειπουσών τιμών, κανονικοποίηση ή διακριτοποίηση αριθμητικών μεταβλητών, καθώς και κωδικοποίηση κατηγορικών χαρακτηριστικών σε αριθμητική μορφή. Επιπλέον, συχνά εφαρμόζεται επιλογή χαρακτηριστικών (feature selection), ώστε να διατηρηθεί ένα μικρότερο σύνολο μεταβλητών που είναι επαρκές για την πρόβλεψη, αλλά ταυτόχρονα ευνοεί την ερμηνευσιμότητα του μοντέλου, ιδιαίτερα όταν χρησιμοποιούνται δέντρα απόφασης που παράγουν κανόνες πάνω σε συγκεκριμένα χαρακτηριστικά.

Τέλος, απαιτείται προσεκτικός χειρισμός των ετικετών-στόχων, ειδικά όταν οι αριθμητικές τιμές που παρέχουν οι tablebases (όπως DTZ ή DTM) μετατρέπονται σε διακριτές κατηγορίες. Η επιλογή του σχήματος αυτής της διακριτοποίησης επηρεάζει όχι μόνο τη διατύπωση του προβλήματος (ταξινόμηση αντί για παλινδρόμηση), αλλά και τον τύπο των κανόνων που μπορούν να εξαχθούν από τα μοντέλα, καθώς και τον τρόπο με τον οποίο ερμηνεύονται από σκακιστική σκοπιά.

2.3.3 Δέντρα Απόφασης

Τα δέντρα απόφασης (decision trees) [13] αποτελούν μια από τις πιο διαδεδομένες μεθόδους επιβλεπόμενης μάθησης για προβλήματα ταξινόμησης και παλινδρόμησης, ιδιαίτερα σε δεδομένα πίνακα (tabular data). Το μοντέλο αναπαρίσται ως μια δενδροειδής δομή, στην οποία κάθε εσωτερικός κόμβος αντιστοιχεί σε έναν έλεγχο πάνω σε κάποιο χαρακτηριστικό της εισόδου, τα κλαδιά αντιπροσωπεύουν τα δυνατά αποτελέσματα του ελέγχου και τα φύλλα αποδίδουν την τελική πρόβλεψη (κατηγορία ή αριθμητική τιμή). Η διαδικασία πρόβλεψης για ένα νέο παράδειγμα συνίσταται στην εκκίνηση από τη ρίζα του δέντρου και στη διάσχιση της διαδρομής που επιβάλλουν διαδοχικά οι λογικές συνθήκες μέχρι να καταλήξει σε κάποιο φύλλο.

όπως αποτέλεσμα ή φάση DTZ, μέσα από σαφείς, λογικούς κανόνες. Έτσι, τα decision trees μπορούν να λειτουργήσουν ως γέφυρα ανάμεσα στην «απόλυτη» αλλά δύσκολα ερμηνεύσιμη πληροφορία των tablebases και σε έναν πιο συνοπτικό, ανθρώπινα κατανοητό περιγραφικό λόγο για τα φινάλε KPK, KRK και KQK, με τη μορφή κανόνων που αναδεικνύουν βασικά στρατηγικά μοτίβα αυτών των φινάλε.

2.3.4 Εξαγωγή κανόνων και ερμηνευσιμότητα

Ένα από τα σημαντικότερα πλεονεκτήματα των δέντρων απόφασης είναι ότι επιτρέπουν την άμεση εξαγωγή κανόνων από το εκπαιδευμένο μοντέλο. Κάθε διαδρομή από τη ρίζα ενός δέντρου μέχρι ένα φύλλο μπορεί να αποδοθεί ως ένας κανόνας της μορφής «αν-τότε» (if-then rule), όπου οι ενδιάμεσες συνθήκες περιγράφουν περιορισμούς πάνω στα χαρακτηριστικά και το φύλλο αντιστοιχεί στην τελική πρόβλεψη. Με αυτόν τον τρόπο, το μοντέλο δεν λειτουργεί απλώς ως μηχανισμός πρόβλεψης, αλλά και ως μέσο αναπαράστασης γνώσης σε μορφή εύκολα αναγνώσιμη από τον άνθρωπο [16].

Η δυνατότητα αυτή συνδέεται άμεσα με την έννοια της ερμηνευσιμότητας, δηλαδή της ικανότητας ενός μοντέλου να εξηγεί με σαφή τρόπο πώς και γιατί καταλήγει σε μια συγκεκριμένη απόφαση. Σε αντίθεση με πιο σύνθετα μοντέλα που λειτουργούν συχνά ως «μαύρα κουτιά», τα δέντρα απόφασης και τα σύνολα κανόνων προσφέρουν διαφάνεια, αφού ο χρήστης μπορεί να εντοπίσει ποια χαρακτηριστικά επηρεάζουν την πρόβλεψη και με ποιον τρόπο. Αυτό είναι ιδιαίτερα σημαντικό σε εφαρμογές όπου ζητούμενο δεν είναι μόνο η ακρίβεια, αλλά και η κατανόηση των υποκείμενων σχέσεων που υπάρχουν στα δεδομένα.

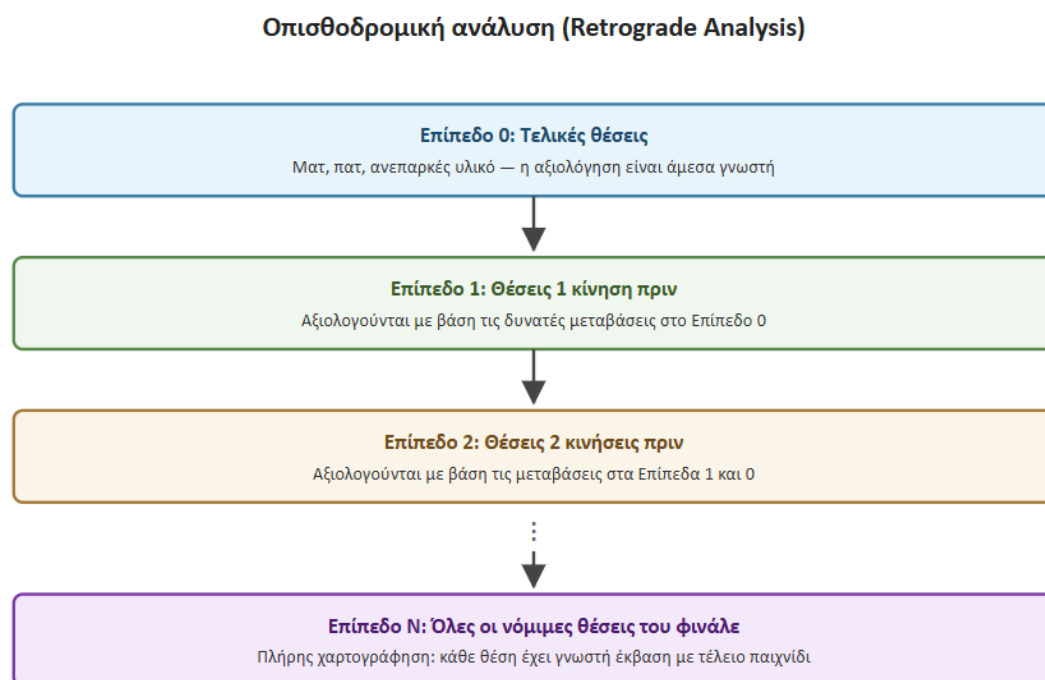
Στο πλαίσιο της παρούσας εργασίας, η εξαγωγή κανόνων αποκτά ιδιαίτερη σημασία, καθώς επιτρέπει τη μετατροπή της πληροφορίας που κωδικοποιείται στα endgame tablebases σε ένα σύνολο ανθρώπινα κατανοητών περιγραφών για τα φινάλε KPK, KRK και KQK. Με τον τρόπο αυτό, οι προβλέψεις του μοντέλου δεν περιορίζονται σε μία απλή κατηγοριοποίηση της θέσης, αλλά συνοδεύονται από λογικές συνθήκες που μπορούν να ερμηνευθούν σκακιστικά και να συνδεθούν με γνωστά στρατηγικά μοτίβα των τελικών.

3. Endgame Tablebases

3.1 Εισαγωγή και ιστορική επισκόπηση

Τα endgame tablebases αποτελούν μία από τις πιο χαρακτηριστικές εφαρμογές πεπερασμένης πλήρους ανάλυσης στο σκάκι. Πρόκειται για βάσεις δεδομένων που περιέχουν, για κάθε νόμιμη θέση ενός συγκεκριμένου φινάλε, την ακριβή θεωρητική της αξιολόγηση: αν είναι κερδισμένη, ισόπαλη ή χαμένη, και — ανάλογα με τον τύπο της βάσης — επιπλέον πληροφορία απόστασης μέχρι το αποτέλεσμα. Σε αντίθεση με τις σκακιστικές μηχανές, που στηρίζονται σε ευρετικές αξιολογήσεις και αναζήτηση περιορισμένου βάθους, οι tablebases δίνουν την αληθή απάντηση για κάθε θέση που καλύπτουν [5].

Η θεωρητική βάση των tablebases είναι η οπισθοδρομική ανάλυση (retrograde analysis), μια τεχνική που πρωτοεφαρμόστηκε συστηματικά στο σκάκι από τον Ken Thompson στη δεκαετία του 1980 [17]. Η ιδέα είναι αντισυμβατική: αντί να ξεκινήσει η ανάλυση από μια αρχική θέση και να προχωρήσει προς τα εμπρός, όπως κάνει μια κανονική μηχανή αναζήτησης, ξεκινά από όλες τις τελικές θέσεις του φινάλε — δηλαδή εκείνες που είναι άμεσα κερδισμένες (ματ) ή άμεσα ισόπαλες (πατ, ανεπαρκές υλικό) — και «προχωρώντας προς τα πίσω» στον χώρο των καταστάσεων, αποδίδει σε κάθε θέση την αξιολόγησή της με βάση τις θέσεις στις οποίες οδηγεί. Το αποτέλεσμα είναι μια πλήρης χαρτογράφηση: για κάθε θέση που ανήκει στο εξεταζόμενο σύνολο υλικού, είναι γνωστή εκ των προτέρων η έκβαση με τέλειο παιχνίδι.



Εικόνα 3.1 Retrograde Analysis

Ιστορικά, η εξέλιξη των tablebases ακολούθησε την αύξηση της διαθέσιμης υπολογιστικής ισχύος και της χωρητικότητας αποθήκευσης. Οι πρώτες ευρέως διαδεδομένες tablebases ήταν οι Nalimov, που αναπτύχθηκαν στα τέλη της δεκαετίας του 1990 και κάλυπταν φινάλε έως έξι κομμάτια [19]. Η επόμενη γενιά, οι Syzygy του Ronald de Man, εμφανίστηκε γύρω στο 2013 και σχεδιάστηκε ειδικά για ενσωμάτωση στις σκακιστικές μηχανές, λαμβάνοντας υπόψη και τον κανόνα των πενήντα κινήσεων [6][23]. Παράλληλα αναπτύχθηκαν εναλλακτικές μορφές όπως οι Gaviota [20] και οι Scorpio bitbases [24], ενώ η κάλυψη επεκτάθηκε στα επτά κομμάτια με τις Lomonosov tablebases, που υπολογίστηκαν το 2012 στον ομώνυμο υπερυπολογιστή του Πανεπιστημίου της Μόσχας [22].

Σε ό,τι αφορά την κάλυψη, η πρόοδος έχει πρακτικά όρια. Ο αριθμός των νόμιμων θέσεων αυξάνεται εκρηκτικά με τον αριθμό των κομματιών, και το ίδιο ισχύει για το χώρο αποθήκευσης που απαιτείται. Ενδεικτικά, οι πλήρεις Syzygy tablebases για όλα τα φινάλε έως πέντε κομμάτια καταλαμβάνουν περίπου 1 GB, ενώ η αντίστοιχη επέκταση στα έξι κομμάτια απαιτεί 150 GB και στα επτά κομμάτια ξεπερνά τα 16 TB [27]. Η μετάβαση στα οκτώ κομμάτια θεωρείται προς το παρόν ανέφικτη με τη σημερινή τεχνολογία.

Παρά αυτούς τους περιορισμούς, η ύπαρξη των tablebases έχει αλλάξει ουσιαστικά τον τρόπο μελέτης των φινάλε. Πολλές κλασικές αξιολογήσεις που είχαν διατυπωθεί από θεωρητικούς του σκακιού επιβεβαιώθηκαν ή — σε ορισμένες περιπτώσεις — αναθεωρήθηκαν, αναδεικνύοντας θέσεις που στην πράξη είναι κερδισμένες με μεγαλύτερο βάθος απ' όσο πίστευαν οι κορυφαίοι αναλυτές [4][18]. Ταυτόχρονα, η «τέλεια» αυτή πληροφορία δημιούργησε ένα νέο ερευνητικό πρόβλημα: τη μετάφρασή της σε μορφή κατανοητή από τον άνθρωπο. Αυτή ακριβώς η πρόκληση βρίσκεται στον πυρήνα της παρούσας εργασίας.

3.2 Τύποι endgame tablebases

Από τη δεκαετία του 1980 και μετά, αναπτύχθηκαν πολλαπλές οικογένειες endgame tablebases, η καθεμία με διαφορετικά σχεδιαστικά χαρακτηριστικά, διαφορετική κωδικοποίηση της πληροφορίας και διαφορετική κάλυψη ως προς τον αριθμό κομματιών. Παρότι όλες αποσκοπούν στον ίδιο στόχο — την πλήρη αξιολόγηση κάθε νόμιμης θέσης ενός φινάλε — οι μεταξύ τους διαφορές είναι σημαντικές, ιδιαίτερα σε ό,τι αφορά την πληροφορία απόστασης που αποθηκεύουν, την υποστήριξη του κανόνα των πενήντα κινήσεων και τις απαιτήσεις σε χώρο αποθήκευσης. Στις παρακάτω παραγράφους περιγράφονται οι σημαντικότερες από αυτές, με σειρά εμφάνισης στον χρόνο.

3.2.1 Nalimov tablebases

Οι Nalimov tablebases [19], που πήραν το όνομά τους από τον Eugene Nalimov, αποτέλεσαν την πρώτη ευρέως διαδεδομένη υλοποίηση πλήρους ανάλυσης φινάλε σε προγραμματιστικό περιβάλλον. Αναπτύχθηκαν στα τέλη της δεκαετίας του 1990 και κάλυψαν φινάλε έως έξι κομμάτια, παρέχοντας για κάθε θέση τον τύπο αποτελέσματος (νίκη, ισοπαλία, ήττα) καθώς και την πληροφορία depth to mate (DTM), δηλαδή τον ελάχιστο αριθμό κινήσεων μέχρι το ματ με τέλειο παιχνίδι.

Ένα σημαντικό χαρακτηριστικό των Nalimov tablebases είναι ότι δεν λαμβάνουν υπόψη τον κανόνα των πενήντα κινήσεων. Σύμφωνα με τους κανονισμούς της FIDE, αν παρέλθουν πενήντα κινήσεις χωρίς κίνηση πιονιού ή αιχμαλωσία, μπορεί να ζητηθεί ισοπαλία· οι Nalimov αγνοούν αυτόν τον περιορισμό, οπότε μια θέση μπορεί να χαρακτηρίζεται κερδισμένη παρόλο που η νίκη απαιτεί περισσότερες από πενήντα κινήσεις χωρίς zeroing move. Αυτό δημιούργησε στην πράξη αναντιστοιχίες ανάμεσα στη θεωρητική και τη «νόμιμα επιτρεπτή» αξιολόγηση. Παρά τον περιορισμό αυτό, οι Nalimov χρησιμοποιήθηκαν εκτενώς για πάνω από μία δεκαετία και αποτέλεσαν τη βάση για πολλές κλασικές αναλύσεις φινάλε.

3.2.2 Gaviota tablebases

Οι Gaviota tablebases [20][21] αναπτύχθηκαν από τον Miguel Ballicora ως εναλλακτική υλοποίηση που στοχεύει σε καλύτερη συμπίεση των δεδομένων και ταχύτερη ανάγνωση. Καλύπτουν φινάλε έως πέντε κομμάτια και αποθηκεύουν επίσης DTM πληροφορία. Χρησιμοποιήθηκαν κυρίως από τη σκακιστική μηχανή Houdini και θεωρούνται αξιόπιστη επιλογή σε συστήματα όπου ο χώρος αποθήκευσης ή η ταχύτητα πρόσβασης είναι κρίσιμα. Όπως και οι Nalimov, δεν χειρίζονται τον κανόνα των πενήντα κινήσεων, γεγονός που περιορίζει τη χρήση τους σε αυστηρά νόμιμα πλαίσια ανάλυσης.

3.2.3 Syzygy tablebases

Οι Syzygy tablebases [6][23], που δημιουργήθηκαν από τον Ronald de Man γύρω στο 2013, αποτελούν τη σύγχρονη επιλογή για χρήση σε σκακιστικές μηχανές και ερευνητικές εφαρμογές. Καλύπτουν φινάλε έως επτά κομμάτια και υιοθετούν μια διαφορετική φιλοσοφία αποθήκευσης: αντί να αποθηκεύουν την πληροφορία αποτελέσματος και την απόσταση σε ενιαία μορφή, τη διασπούν σε δύο τύπους αρχείων που χρησιμοποιούνται συμπληρωματικά.

Οι πρώτοι, οι WDL tables (win/draw/loss), περιέχουν για κάθε θέση την κωδικοποίηση του αποτελέσματος σε μερικά μόνο bits, με αποτέλεσμα να είναι πολύ μικροί σε μέγεθος και γρήγοροι στην ανάγνωση. Οι δεύτεροι, οι DTZ tables (depth to zeroing), αποθηκεύουν τον αριθμό κινήσεων μέχρι την επόμενη κίνηση που μηδενίζει τον μετρητή των πενήντα κινήσεων — δηλαδή κίνηση πιονιού ή αιχμαλωσία. Αυτή η σχεδιαστική επιλογή είναι κρίσιμη: επειδή η DTZ μέτρηση συμβαδίζει με τη λογική του κανόνα των πενήντα κινήσεων, οι Syzygy μπορούν να αποτυπώσουν με ακρίβεια αν μια θεωρητικά κερδισμένη θέση είναι πρακτικά κερδισμένη — δηλαδή εντός των επιτρεπτών ορίων του κανονισμού — ή αν στην πράξη καταλήγει σε ισοπαλία.

Λόγω αυτών των χαρακτηριστικών, οι Syzygy έχουν υιοθετηθεί από τις περισσότερες σύγχρονες ισχυρές μηχανές, μεταξύ των οποίων και το Stockfish από την έκδοση 5 και μετά [7]. Για την παρούσα εργασία επιλέχθηκαν επίσης οι Syzygy, και η σχετική τεκμηρίωση παρουσιάζεται αναλυτικά στην υποενότητα 3.5.

3.2.4 Lomonosov tablebases

Οι Lomonosov tablebases [22] υπολογίστηκαν το 2012 στον ομώνυμο υπερυπολογιστή του Πανεπιστημίου Lomonosov της Μόσχας και αποτελούν την πρώτη πλήρη ανάλυση όλων των φινάλε έως επτά κομμάτια με DTM πληροφορία. Λόγω του τεράστιου μεγέθους τους — που υπερβαίνει τα 140 TB — δεν διανέμονται για τοπική εγκατάσταση, αλλά είναι προσβάσιμες κυρίως μέσω διαδικτυακών υπηρεσιών, όπως το ChessOK Online Service. Η αξία τους είναι περισσότερο ερευνητική και θεωρητική παρά πρακτική: η ανάλυσή τους έχει αποκαλύψει εντυπωσιακά φινάλε με μεγάλο depth to mate, ορισμένα από τα οποία απαιτούν εκατοντάδες κινήσεις για να επιτευχθεί η νίκη.

3.2.5 Scorpio bitbases και άλλες ειδικές μορφές

Οι Scorpio bitbases [24], που αναπτύχθηκαν από τον Daniel Shawul για την ομώνυμη σκακιστική μηχανή, αποτελούν μια συμπυκνωμένη μορφή tablebases που κωδικοποιεί μόνο την πληροφορία WDL, χωρίς δεδομένα απόστασης. Αυτή η επιλογή κάνει τις Scorpio bitbases εξαιρετικά μικρές σε μέγεθος και ιδανικές για ενσωμάτωση σε συστήματα με περιορισμένους πόρους, με το κόστος ότι δεν παρέχουν καθόλου πληροφορία για το βάθος της νίκης ή της ήττας. Πέρα από αυτές, υπάρχουν επίσης ορισμένες παλαιότερες ή πειραματικές μορφές, όπως οι Edwards tablebases [5], που παρουσιάζουν κυρίως ιστορικό ενδιαφέρον.

Συγκριτική επισκόπηση

Στον παρακάτω πίνακα συνοψίζονται τα βασικά χαρακτηριστικά των πέντε οικογενειών που περιγράφηκαν παραπάνω.

Tablebase	Έτος	Μέγιστος αριθμός κομματιών	Πληροφορία απόστασης	Κανόνας 50 κινήσεων	Τυπικό μέγεθος
Nalimov	~1998	6	DTM	Όχι	~1TB (έως 6)
Gaviota	~2010	5	DTM	Όχι	~7GB (έως 5)
Syzygy	~2013	7	WDL + DTZ	Ναι	~1 GB (έως 5), ~150 GB (έως 6), ~18.4 TB (έως 7)
Lomonosov	2012	7	DTM	Όχι	~140TB
Scorpio bitbases	~2010	6	WDL	-	~10GB

Πίνακας 3.1 Συγκριτική επισκόπηση των κυριότερων οικογενειών endgame tablebases

Η επιλογή της κατάλληλης οικογένειας εξαρτάται από την εφαρμογή. Για ενσωμάτωση σε σκακιστικές μηχανές με γνώμονα τους κανονισμούς, οι Syzygy είναι

σχεδόν πάντα η προτιμώμενη επιλογή. Για θεωρητικές αναλύσεις όπου ενδιαφέρει το ακριβές DTM, οι Nalimov ή οι Lomonosov προσφέρουν την αντίστοιχη πληροφορία. Στην παρούσα εργασία, η επιλογή των Syzygy έγινε για λόγους που σχετίζονται με τη φύση των επιμέρους ταξινομητικών προβλημάτων και αναλύονται στην υποενότητα 3.5.

3.3 Εσωτερική δομή και αποθήκευση

Η βασική σχεδιαστική πρόκληση κάθε endgame tablebase είναι ότι ο αριθμός των νόμιμων θέσεων που πρέπει να αποθηκευτούν είναι τεράστιος, ενώ η πληροφορία που αντιστοιχεί σε κάθε θέση είναι μικρή — συχνά λίγα μόνο bits. Αυτή η ασυμμετρία ανάμεσα στον τεράστιο όγκο εγγραφών και τη μικρή πληροφορία ανά εγγραφή υπαγορεύει την όλη φιλοσοφία αποθήκευσης: οι tablebases δεν αποθηκεύουν τις θέσεις ρητά, αλλά τις **αντιστοιχίζουν σε δείκτες** μέσα σε μεγάλους πίνακες, ενώ οι ίδιες οι τιμές υπόκεινται σε εκτεταμένη συμπίεση [5][23].

3.3.1 Αντιστοίχιση θέσεων σε δείκτες

Κάθε νόμιμη θέση ενός φινάλε χαρακτηρίζεται πλήρως από τη θέση των κομματιών στα 64 τετράγωνα της σκακιέρας, την πλευρά που έχει τη σειρά κίνησης, και — όπου έχει νόημα — τυχόν επιπλέον πληροφορίες όπως δικαιώματα ροκέ ή δυνατότητα σύλληψης «εν διελεύσει». Η εσωτερική αναπαράσταση μιας tablebase δεν αποθηκεύει αυτές τις πληροφορίες ως κείμενο ή ως αντικείμενα θέσης, αλλά τις κωδικοποιεί σε έναν μοναδικό ακέραιο δείκτη, ο οποίος προσδιορίζει τη συγκεκριμένη θέση μέσα στον πίνακα δεδομένων.

Η συνάρτηση που μετατρέπει μια θέση σε δείκτη — γνωστή ως indexing function — είναι κρίσιμη: πρέπει να είναι αμφιμονοσήμαντη (κάθε νόμιμη θέση να αντιστοιχεί σε διαφορετικό δείκτη), αποδοτική (να υπολογίζεται γρήγορα), και κατά το δυνατόν συμπαγής (να μην αφήνει μεγάλα «κενά» στον πίνακα). Στην απλούστερη μορφή της, μια indexing function για ένα φινάλε με τρία κομμάτια θα μπορούσε να συνδυάζει τις τρεις θέσεις 0-63 σε έναν αριθμό από 0 έως περίπου 64^3 , ωστόσο μια τέτοια αφελή υλοποίηση θα ήταν εξαιρετικά σπάταλη, καθώς θα κάλυπτε και μη νόμιμες διατάξεις (π.χ. κομμάτια στο ίδιο τετράγωνο). Στην πράξη, οι σύγχρονες υλοποιήσεις [19][23] χρησιμοποιούν πιο εξελιγμένες indexing functions που εκμεταλλεύονται τους περιορισμούς της θέσης για να μειώσουν τον αριθμό των δεικτών.

3.3.2 Εκμετάλλευση συμμετριών

Ένα από τα σημαντικότερα εργαλεία συμπίεσης στις tablebases είναι η αξιοποίηση των συμμετριών της σκακιέρας. Η σκακιέρα διαθέτει αρκετές γεωμετρικές συμμετρίες — οριζόντιο και κατακόρυφο κατοπτρισμό, καθώς και διαγώνια συμμετρία — που μπορούν να μειώσουν δραστικά τον αριθμό των «μοναδικών» θέσεων που χρειάζεται να αποθηκευτούν. Για παράδειγμα, σε ένα φινάλε χωρίς πιόνια, μια θέση και η κατοπτρική της ως προς τον κατακόρυφο άξονα δίνουν το ίδιο αποτέλεσμα, οπότε αρκεί να αποθηκευτεί η μία.

Στα φινάλε χωρίς πιόνια, οι Syzygy εκμεταλλεύονται πλήρως αυτή την οκταπλή συμμετρία, ανάγοντας οποιαδήποτε θέση στην «κανονική» της μορφή όπου ο λευκός βασιλιάς βρίσκεται σε ένα από δέκα συγκεκριμένα τετράγωνα. Στα φινάλε με πιόνια η ευελιξία είναι μικρότερη, καθώς τα πιόνια έχουν προσανατολισμό κίνησης (κινούνται μόνο προς ορισμένη κατεύθυνση), οπότε χάνεται ο κατακόρυφος κατοπτρισμός — παραμένει όμως ο οριζόντιος, που υποδιπλασιάζει τον αριθμό των διακριτών θέσεων [23]. Αυτή ακριβώς η συμμετρία αξιοποιείται και στον κώδικα της παρούσας εργασίας για το φινάλε KPK, όπως περιγράφεται στο επόμενο κεφάλαιο.



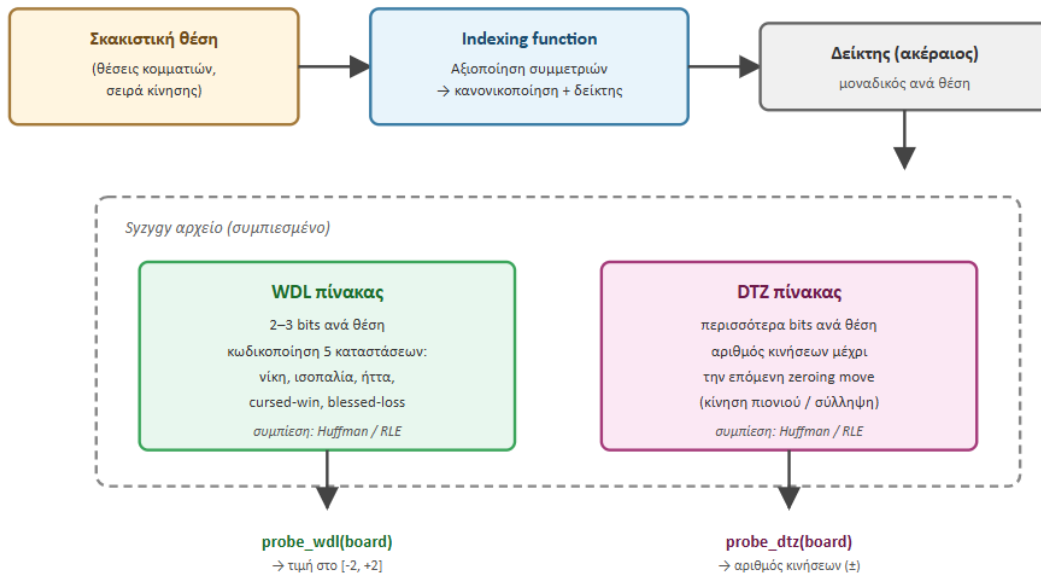
Εικόνα 3.2 Οριζόντια συμμετρία στο φινάλε KPK

3.3.3 Συμπίεση των τιμών

Πέρα από την αξιοποίηση συμμετριών για τη μείωση του πλήθους θέσεων, οι tablebases εφαρμόζουν επίσης συμπίεση πάνω στις τιμές που αποθηκεύονται. Στις WDL tables των Syzygy, η πληροφορία ανά θέση είναι μόλις δύο ή τρία bits — αρκούν για να κωδικοποιηθούν οι πέντε δυνατές καταστάσεις (νίκη, ισοπαλία, ήττα, και δύο ενδιάμεσες κατηγορίες για θέσεις που εξαρτώνται από τον κανόνα των πενήντα κινήσεων). Στις DTZ tables η απαιτούμενη πληροφορία είναι μεγαλύτερη, καθώς αποθηκεύεται αριθμός κινήσεων, αλλά παραμένει σχετικά μικρή λόγω του ότι οι περισσότερες θέσεις έχουν DTZ τιμές σε ένα περιορισμένο εύρος.

Για την περαιτέρω μείωση του μεγέθους, οι Syzygy χρησιμοποιούν τεχνικές όπως Huffman coding και run-length encoding [23], που εκμεταλλεύονται το γεγονός ότι γειτονικές θέσεις στον πίνακα συχνά έχουν ίδιες ή παρόμοιες τιμές. Το αποτέλεσμα είναι ότι ο τελικός χώρος αποθήκευσης μειώνεται σημαντικά σε σχέση με την «ωμή» αναπαράσταση, χωρίς απώλεια πληροφορίας. Η συμπίεση αυτή είναι αυτή που επιτρέπει τις πλήρεις Syzygy tablebases έως πέντε κομμάτια να καταλαμβάνουν μόλις 1 GB, ενώ μια ασυμπίεστη αναπαράσταση θα απαιτούσε δεκάδες ή εκατοντάδες φορές περισσότερο χώρο.

Εσωτερική δομή ενός Syzygy αρχείου



Εικόνα 3.3 Εσωτερική δομή ενός Syzygy αρχείου

3.3.4 Πρακτικές συνέπειες της δομής

Η συγκεκριμένη σχεδίαση έχει σαφείς πρακτικές συνέπειες για όποιον χρησιμοποιεί τις tablebases. Πρώτον, η ανάκτηση μιας τιμής δεν είναι απλή σειριακή ανάγνωση: απαιτείται υπολογισμός του δείκτη, εντοπισμός του αντίστοιχου block μέσα στο συμπίεσμένο αρχείο, αποσυμπίεσή του και τέλος ανάγνωση της τιμής. Δεύτερον, και πιο σημαντικό για την πρακτική χρήση, η ταχύτητα του μέσου αποθήκευσης επηρεάζει αισθητά την απόδοση. Για τον λόγο αυτό, οι κατασκευαστές και διανομείς των tablebases συνιστούν συστηματικά την αποθήκευση των αρχείων σε SSD ή τουλάχιστον σε γρήγορη USB μνήμη, καθώς η ανάγνωση από κλασικό μηχανικό δίσκο εισάγει σημαντικές καθυστερήσεις [27].

Στην παρούσα εργασία, οι Syzygy tablebases εγκαταστάθηκαν τοπικά σε SSD ώστε να εξασφαλιστεί επαρκής ταχύτητα probing για την παραγωγή των datasets, η οποία απαιτεί εκατοντάδες χιλιάδες αξιολογήσεις θέσεων ανά endgame.

3.4 Πρόσβαση και εργασία

Παρά τη συμπυκνωμένη εσωτερική δομή τους, η αξιοποίηση των endgame tablebases στην πράξη δεν είναι αυτονόητη: απαιτείται είτε τοπική εγκατάσταση των αρχείων και κατάλληλος μηχανισμός probing, είτε πρόσβαση μέσω διαδικτυακής υπηρεσίας. Στην ενότητα αυτή περιγράφονται οι τρεις βασικές προσεγγίσεις πρόσβασης που χρησιμοποιούνται σήμερα: η τοπική (offline) εγκατάσταση, η χρήση online υπηρεσιών και η προγραμματιστική πρόσβαση μέσω βιβλιοθηκών.

3.4.1 Τοπική εγκατάσταση και χρήση

Η πιο διαδεδομένη μορφή πρόσβασης σε tablebases είναι η τοπική εγκατάσταση των αρχείων στον υπολογιστή του χρήστη. Αρκετές πηγές διαθέτουν τις Syzygy tablebases προς λήψη, μεταξύ των οποίων ο ιστότοπος Chess Endgame Database του Massimiliano Goï [27], ο οποίος προσφέρει άμεσο σύνδεσμο για το αρχείο των 3-4-5 κομματιών και torrent για το αντίστοιχο των 6 κομματιών.

Οι απαιτήσεις σε χώρο αποθήκευσης διαφέρουν δραστικά ανάλογα με την κάλυψη. Οι Syzygy για όλα τα φινάλε έως πέντε κομμάτια καταλαμβάνουν περίπου 1 GB, ποσότητα διαχειρίσιμη ακόμη και για μέσες εγκαταστάσεις. Η επέκταση στα έξι κομμάτια απαιτεί περίπου 150 GB, χωρισμένα σε WDL (~68 GB) και DTZ (~82 GB), ενώ η πλήρης κάλυψη επτά κομματιών ξεπερνά τα 16 TB — μέγεθος που πρακτικά καθιστά απαγορευτική την τοπική εγκατάσταση για τους περισσότερους χρήστες [27]. Όπως αναφέρθηκε στην προηγούμενη ενότητα, η αποθήκευση σε γρήγορο μέσο (SSD ή ταχεία USB μνήμη) είναι σχεδόν απαραίτητη για ικανοποιητική απόδοση.

Μετά την εγκατάσταση, η αξιοποίηση των tablebases γίνεται είτε αυτόματα από μια σκακιστική μηχανή που υποστηρίζει probing, είτε μέσω γραφικών περιβαλλόντων ανάλυσης. Οι σύγχρονες μηχανές όπως το Stockfish, όταν κατά την ανάλυση φτάσουν σε θέση με αρκετά λίγα κομμάτια, ελέγχουν πρώτα αν αυτή καλύπτεται από τα φορτωμένα tablebases. Αν ναι, αντί να συνεχίσουν την αναζήτηση, πραγματοποιούν μια κλήση στη βάση και λαμβάνουν την ακριβή αξιολόγηση, μαζί με ένα σύνολο βέλτιστων κινήσεων. Με αυτόν τον τρόπο, ολόκληρες υπο-περιοχές του δένδρου αναζήτησης «καταρρέουν» σε μία μόνο ανάκτηση, και ο υπολογιστικός χρόνος της μηχανής αφιερώνεται σε πιο πολύπλοκες θέσεις [7].

Παρόμοιος μηχανισμός χρησιμοποιείται και σε γραφικά περιβάλλοντα ανάλυσης όπως το ChessBase [7]. Ο χρήστης δηλώνει τη διαδρομή προς τα αρχεία Syzygy ή Nalimov, και η εφαρμογή τα αξιοποιεί στο παρασκήνιο: όταν η θέση που αναλύεται καλύπτεται από τις βάσεις, εμφανίζεται αυτόματα η ακριβής αξιολόγηση, ο ελάχιστος αριθμός κινήσεων μέχρι το αποτέλεσμα και οι κινήσεις που διατηρούν τη βέλτιστη έκβαση. Στην πράξη, αυτό σημαίνει ότι ο χρήστης μπορεί να «παιίζει» ενάντια στην τέλεια άμυνα ή επίθεση του tablebase και να βλέπει άμεσα ποιες κινήσεις οδηγούν σε απώλεια του θεωρητικά κερδισμένου αποτελέσματος.

3.4.2 Online υπηρεσίες

Για χρήστες που δεν επιθυμούν ή δεν μπορούν να εγκαταστήσουν τοπικά τις tablebases, υπάρχουν αρκετές διαδικτυακές υπηρεσίες που παρέχουν probing μέσω HTTP API. Η πιο γνωστή είναι το Lichess Tablebase API [25], το οποίο δέχεται μια θέση σε μορφή FEN και επιστρέφει σε JSON μορφή την WDL αξιολόγηση, την DTZ απόσταση, και τις βέλτιστες κινήσεις. Η υπηρεσία καλύπτει όλα τα φινάλε έως επτά κομμάτια — δηλαδή ολόκληρο το εύρος των διαθέσιμων Syzygy — χωρίς να απαιτεί από τον χρήστη οποιαδήποτε τοπική εγκατάσταση.

Το πλεονέκτημα των online υπηρεσιών είναι η άμεση πρόσβαση σε όλα τα διαθέσιμα φινάλε, συμπεριλαμβανομένων εκείνων που θα ήταν πρακτικά αδύνατο να εγκατασταθούν τοπικά. Το μειονέκτημα είναι ότι κάθε ερώτημα συνεπάγεται καθυστέρηση δικτύου και υπόκειται σε rate limits, γεγονός που τις καθιστά ακατάλληλες για εφαρμογές που απαιτούν μεγάλο όγκο probes σε σύντομο χρόνο —

όπως ακριβώς η μαζική παραγωγή datasets για εκπαίδευση μοντέλων μηχανικής μάθησης.

3.4.3 Προγραμματιστική πρόσβαση μέσω βιβλιοθηκών

Για ερευνητικές και προγραμματιστικές εφαρμογές, η πιο ευέλικτη επιλογή είναι η απευθείας πρόσβαση στα τοπικά αρχεία *tablebases* μέσω εξειδικευμένων βιβλιοθηκών. Σήμερα διατίθενται αρκετές τέτοιες βιβλιοθήκες, με διαφορετικές γλώσσες υλοποίησης και ομάδες χρηστών.

Η βιβλιοθήκη *Fathom* [26], γραμμένη σε C, αποτελεί το *de facto* πρότυπο για ενσωμάτωση *Syzygy probing* σε σκακιστικές μηχανές και χρησιμοποιείται μεταξύ άλλων από το *Stockfish*. Παρέχει χαμηλού επιπέδου πρόσβαση με υψηλές επιδόσεις και είναι κατάλληλη για εφαρμογές όπου ο χρόνος ανά *probe* είναι κρίσιμος. Η ενσωμάτωσή της απαιτεί όμως γνώσεις C και διαχείριση μνήμης, οπότε δεν είναι ιδανική για ερευνητικά *scripts* γρήγορης πρωτοτυποποίησης.

Η βιβλιοθήκη *python-chess* [8], γραμμένη σε Python, αποτελεί τη φυσική επιλογή για εφαρμογές που συνδυάζουν *probing tablebases* με ανάλυση δεδομένων ή μηχανική μάθηση. Παρέχει υψηλού επιπέδου διεπαφή για όλες τις σκακιστικές λειτουργίες (αναπαράσταση θέσεων, νόμιμες κινήσεις, μετατροπή FEN), και συγκεκριμένα για τις *Syzygy* διαθέτει το *module chess.syzygy*, μέσω του οποίου ένας χρήστης μπορεί να ανοίξει μια τοπική εγκατάσταση και να πραγματοποιήσει WDL ή DTZ *probing* με λίγες γραμμές κώδικα. Η ταχύτητα ανά *probe* είναι χαμηλότερη σε σχέση με την *Fathom*, παραμένει όμως επαρκής για παραγωγή datasets της τάξης εκατοντάδων χιλιάδων θέσεων σε λογικό χρόνο.

Στην παρούσα εργασία επιλέχθηκε η βιβλιοθήκη *python-chess* για την προγραμματιστική πρόσβαση στις *Syzygy tablebases*, επιλογή που δικαιολογείται από τη φύση της δουλειάς: η εφαρμογή δεν είναι σκακιστική μηχανή αλλά *pipeline* παραγωγής δεδομένων και εκπαίδευσης μοντέλων, οπότε η συμβατότητα με το ευρύτερο οικοσύστημα Python — *pandas*, *scikit-learn* — υπερτερεί έναντι της μέγιστης ταχύτητας ανά *probe*.

3.5 Οι *Syzygy tablebases* στην παρούσα εργασία

Με βάση τη συγκριτική παρουσίαση των προηγούμενων υποενοτήτων, οι *Syzygy tablebases* επιλέχθηκαν ως η πηγή ετικετών για την παρούσα εργασία. Η επιλογή αυτή δεν είναι αυθαίρετη: όπως αναλύεται στη συνέχεια, ο διαχωρισμός της πληροφορίας σε WDL και DTZ ταιριάζει απόλυτα με τη φύση των τριών διαφορετικών ταξινομητικών προβλημάτων (KPK, KRK, KQK), επιτρέποντας τη χρήση κατάλληλης ετικέτας ανά *endgame*.

3.5.1 Λόγοι επιλογής

Δύο ήταν οι κύριοι λόγοι που οδήγησαν στην επιλογή των *Syzygy* έναντι άλλων διαθέσιμων μορφών. Πρώτον, ο διαχωρισμός WDL και DTZ, που είναι μοναδικό

χαρακτηριστικό των Syzygy [6][23]. Στη συγκεκριμένη εργασία, το φινάλε KPK αντιμετωπίζεται ως δυαδικό πρόβλημα Win/Draw, για το οποίο η κατάλληλη πηγή ετικετών είναι ακριβώς η WDL μέτρηση. Αντίθετα, τα φινάλε KRK και KQK, στα οποία όλες οι μελετώμενες θέσεις είναι κερδισμένες για τον λευκό, αξιοποιούν την DTZ μέτρηση ώστε οι θέσεις να ομαδοποιηθούν σε φάσεις μετατροπής. Καμία άλλη οικογένεια tablebases δεν παρέχει αυτόν τον σαφή διαχωρισμό στην αποθηκευμένη πληροφορία.

Δεύτερον, η ενσωμάτωση με τον κανόνα των πενήντα κινήσεων, που εξασφαλίζει ότι οι αξιολογήσεις αντιστοιχούν σε πρακτικά επιτεύγματα αποτελέσματα και όχι μόνο σε θεωρητικά. Σε ένα ερευνητικό πλαίσιο εξαγωγής κανόνων που υποτίθεται ότι θα χρησιμοποιηθούν από σκακιστές, αυτή η συμβατότητα με τους κανονισμούς της FIDE είναι σημαντική.

Τέλος, η διαθεσιμότητα τεκμηριωμένης βιβλιοθήκης πρόσβασης μέσω python-chess αποτέλεσε καθοριστικό πρακτικό παράγοντα, καθώς επιτρέπει την αξιοποίηση των Syzygy χωρίς να απαιτείται χαμηλού επιπέδου υλοποίηση probing.

3.5.2 Εγκατάσταση και τοπική ρύθμιση

Για τις ανάγκες της εργασίας εγκαταστάθηκε τοπικά η έκδοση των Syzygy που καλύπτει όλα τα φινάλε με 3, 4 και 5 κομμάτια, η οποία επαρκεί πλήρως για τα τρία endgames που μελετώνται (KPK, KRK και KQK αντιστοιχούν όλα σε διατάξεις 3 κομματιών). Τα αρχεία ελήφθησαν από τη συλλογή του ιστότοπου Chess Endgame Database [27] και αποθηκεύτηκαν σε SSD ώστε να εξασφαλιστεί επαρκής ταχύτητα probing.

Το συνολικό μέγεθος της εγκατάστασης είναι περίπου 1 GB, διαχειρίσιμο για μια συνηθισμένη υπολογιστική διάταξη, και περιλαμβάνει τα αρχεία WDL και DTZ για κάθε υποστηριζόμενο συνδυασμό υλικού. Στον κώδικα της εφαρμογής, η διαδρομή προς τη συλλογή ορίζεται ως σταθερά διαμόρφωσης (SYZGY_PATH), επιτρέποντας την εύκολη μεταφορά της εφαρμογής σε διαφορετικό σύστημα.

3.5.3 Μηχανισμός probing μέσω python-chess

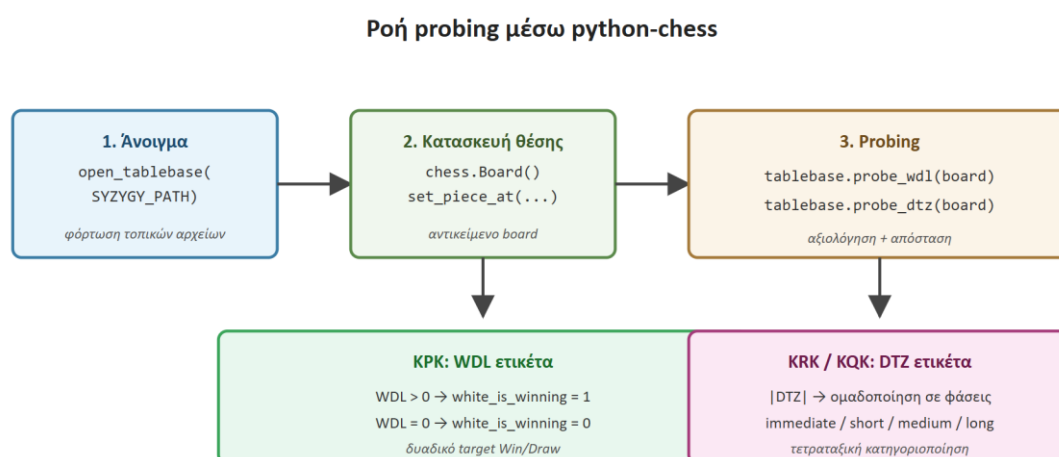
Η πρόσβαση στις τοπικά εγκατεστημένες Syzygy γίνεται μέσω του module chess.syzygy της βιβλιοθήκης python-chess [8]. Ο τυπικός κύκλος εργασίας ακολουθεί τρία βήματα, που αντιστοιχούν και στα τρία πρώτα κουτιά της Εικόνας 3.4: άνοιγμα της συλλογής, κατασκευή της προς εξέταση θέσης, και κλήση της κατάλληλης μεθόδου probing για ανάκτηση της αξιολόγησης.

Η συνάρτηση open_tablebase δέχεται τη διαδρομή προς τη συλλογή και επιστρέφει ένα αντικείμενο που διαχειρίζεται την πρόσβαση στα αρχεία. Στην εφαρμογή χρησιμοποιείται ως context manager, ώστε να εξασφαλίζεται η ορθή αποδέσμευση των πόρων μετά το πέρας της επεξεργασίας — λεπτομέρεια που έχει σημασία όταν εκτελείται κύκλος probing για εκατοντάδες χιλιάδες θέσεις, όπως απαιτείται για την παραγωγή των datasets.

Η θέση που πρόκειται να αξιολογηθεί κατασκευάζεται ως αντικείμενο chess.Board, στο οποίο τοποθετούνται τα κομμάτια του εκάστοτε φινάλε και ορίζεται η πλευρά που

έχει τη σειρά κίνησης. Στη συνέχεια καλείται η αντίστοιχη μέθοδος του tablebase: η `probe_wdl` επιστρέφει ακέραιη τιμή στο διάστημα $[-2, +2]$ από τη σκοπιά της πλευράς που έχει τη σειρά να κινήσει, με την τιμή $+2$ να αντιστοιχεί σε καθαρή νίκη, $+1$ σε νίκη που υπόκειται στον κανόνα των πενήντα κινήσεων, 0 σε ισοπαλία, -1 σε αντίστοιχη ήττα και -2 σε καθαρή ήττα. Η `probe_dtz` επιστρέφει ακέραιη τιμή της οποίας η απόλυτη τιμή αντιστοιχεί στον αριθμό κινήσεων μέχρι την επόμενη κίνηση πιονιού ή αιχμαλωσία υπό τέλει παιχνίδι, με θετικό πρόσημο σε κερδισμένες θέσεις και αρνητικό σε χαμένες.

Η ταχύτητα του probing εξαρτάται κυρίως από το μέσο αποθήκευσης. Με τη συλλογή τοποθετημένη σε SSD, η ροή των κλήσεων ανά θέση εκτελείται με επαρκή ταχύτητα ώστε η πλήρης παραγωγή των datasets για κάθε endgame να ολοκληρώνεται σε λογικό χρόνο, χωρίς το βήμα του probing να αποτελεί υπολογιστικό περιορισμό για την υπόλοιπη εφαρμογή. Η συγκεκριμένη ροή ενσωμάτωσης στον κώδικα της εφαρμογής παρουσιάζεται αναλυτικά στο επόμενο κεφάλαιο.



Εικόνα 3.4 Ροή probing

3.5.4 Διαφορετική αξιοποίηση WDL και DTZ ανά endgame

Παρότι ο μηχανισμός probing που περιγράφηκε είναι κοινός και για τα τρία endgames που μελετώνται, ο τρόπος με τον οποίο η ίδια πηγή πληροφορίας μετατρέπεται σε ετικέτα διαφέρει ουσιαστικά ανά περίπτωση. Η διαφοροποίηση αυτή αντικατοπτρίζει τη σκακιστική φύση κάθε φινάλε και αποτελεί κρίσιμη σχεδιαστική επιλογή της εργασίας.

Για το φινάλε KPK, χρησιμοποιείται η μέθοδος `probe_wdl`. Η επιστρεφόμενη τιμή κωδικοποιείται σε δυαδικό target `white_is_winning`, με τιμή 1 για $WDL > 0$ (νίκη) και 0 για $WDL \leq 0$ (ισοπαλία ή ήττα από τη σκοπιά του λευκού). Η επιλογή της WDL μέτρησης ως πηγής ετικέτας είναι φυσική για το KPK, καθώς το θεμελιώδες ερώτημα του φινάλε δεν αφορά την ταχύτητα της νίκης αλλά την ίδια την ύπαρξή της: ένα μόνο πiónι στη σκακιέρα μπορεί να επαρκεί για νίκη σε ορισμένες θέσεις και να μην επαρκεί σε άλλες, ανάλογα με τη γεωμετρική διάταξη των βασιλιάδων.

Για τα φινάλε KRK και KQK, χρησιμοποιείται η μέθοδος `probe_dtz`. Οι θέσεις που διατηρούνται στο dataset είναι αποκλειστικά κερδισμένες για τον λευκό, οπότε το ερώτημα μετατοπίζεται από το «κερδίζει ο λευκός;» στο «πόσο γρήγορα κερδίζει;». Η απόλυτη τιμή της DTZ μέτρησης ομαδοποιείται σε τέσσερις φάσεις μετατροπής — `immediate`, `short`, `medium` και `long conversion` — με βάση κατάλληλα όρια ορισμένα ξεχωριστά για κάθε endgame. Τα συγκεκριμένα όρια διαφέρουν επειδή η φύση των δύο φινάλε διαφέρει αισθητά: στο KQK οι θεωρητικές νίκες ολοκληρώνονται σε πολύ λιγότερες κινήσεις απ' ό,τι στα αντίστοιχα KRK, οπότε η ομαδοποίηση των φάσεων προσαρμόζεται αναλόγως ώστε να παράγει ισορροπημένες κλάσεις σε κάθε περίπτωση. Οι ακριβείς τιμές των ορίων και η δικαιολόγησή τους παρουσιάζονται στο επόμενο κεφάλαιο, όπου περιγράφεται και η συνολική διαδικασία παραγωγής των datasets.

Η συνδυασμένη χρήση WDL και DTZ probing αποτελεί τη βάση πάνω στην οποία στηρίζεται ολόκληρη η εφαρμογή, καθώς εξασφαλίζει ότι κάθε θέση συνοδεύεται από αξιόπιστη ετικέτα κατάλληλης μορφής για το αντίστοιχο πρόβλημα ταξινόμησης. Με αυτή τη βάση εξασφαλισμένη, το επόμενο κεφάλαιο παρουσιάζει το pipeline της εφαρμογής, από την παραγωγή των θέσεων και την εξαγωγή χαρακτηριστικών, μέχρι την εκπαίδευση των δέντρων απόφασης και την τελική εξαγωγή κανόνων.

4. Η εφαρμογή

4.1 Επισκόπηση του pipeline

Πριν παρουσιαστούν οι τρεις επιμέρους εφαρμογές, αξίζει να εξηγηθεί μια σχεδιαστική επιλογή που καθόρισε όλη τη δομή της δουλειάς: αντί να αναπτυχθεί ξεχωριστή υλοποίηση για κάθε φινάλε, σχεδιάστηκε ενιαία αρχιτεκτονική που εφαρμόζεται και στα τρία. Η ίδια ακολουθία βημάτων, η ίδια οργάνωση αρχείων, τα ίδια εργαλεία αξιολόγησης. Οι διαφορές περιορίζονται σε όσα πρέπει πραγματικά να διαφέρουν: τα features κάθε φινάλε και ο τρόπος με τον οποίο ορίζεται η ζητούμενη πρόβλεψη.

Η επιλογή αυτή δεν ήταν αυτονόητη στην αρχή. Το KPK, το KRK και το KQK έχουν διαφορετική σκακιστική φύση, ιδιαίτερα αν συγκριθεί το KPK με τα KRK και KQK, ώστε να δικαιολογούνταν τρεις διακριτές υλοποιήσεις. Η ενιαία αρχιτεκτονική όμως πρόσφερε δύο πλεονεκτήματα. Πρώτον, μείωσε δραστικά την επαναληπτική εργασία: κάθε φορά που χρειαζόταν αλλαγή σε επίπεδο αξιολόγησης ή κατασκευής χαρακτηριστικών, η αντίστοιχη προσαρμογή γινόταν σε όλα τα endgames με ομοιόμορφο τρόπο. Δεύτερον, και πιο σημαντικό, εξασφάλισε συγκρισιμότητα. Τα αποτελέσματα των τριών φινάλε προέρχονται από την ίδια πειραματική διαδικασία και μπορούν να ερμηνευθούν μαζί χωρίς αμφιβολίες για αποκλίσεις στη μεθοδολογία.

Η αρχιτεκτονική υλοποιήθηκε σε έξι αρχεία Python ανά φινάλε, καθένα υπεύθυνο για ένα διακριτό βήμα. Ο Generator είναι το μοναδικό σημείο επαφής με τις Syzygy tablebases: παράγει εξαντλητικά όλες τις νόμιμες θέσεις του φινάλε, υπολογίζει για κάθε μία τα χαρακτηριστικά και την επικέτα της, και αποθηκεύει το αποτέλεσμα σε CSV. Από εκεί και πέρα, οι Syzygy δεν χρειάζονται ξανά. Ο DecTree εκπαιδεύει το δέντρο απόφασης πάνω στο dataset, με συστηματική σταθερότητα ως προς πολλαπλά seeds και επιλογή υπερπαραμέτρων, μέσω κριτηρίου που ισορροπεί ακρίβεια και πολυπλοκότητα. Ο Comp συγκρίνει τρία διαφορετικά επίπεδα αναπαράστασης των θέσεων, ώστε να φανεί καθαρά πόση σκακιστική γνώση χρειάζεται να ενσωματωθεί στα features και πόση μπορεί το δέντρο να «ανακαλύψει» μόνο του. Ο RuleConsolidation αναλαμβάνει το πιο πρωτότυπο και ίσως πιο δύσκολο βήμα: μετατρέπει τα πολλά τεχνικά leaf rules του δέντρου σε ένα μικρό σύνολο ονοματισμένων σκακιστικών μοτίβων. Τέλος, οι Visual και ReadMe παράγουν τα συνοδευτικά γραφήματα και την αυτοτελή τεκμηρίωση κάθε εκτέλεσης.

Ο σαφής αυτός διαχωρισμός ευθυνών δεν είναι μόνο σχεδιαστική κομψότητα. Επιτρέπει στον αναγνώστη ή τον επόμενο ερευνητή να εκτελέσει μόνο όσα βήματα τον ενδιαφέρουν — για παράδειγμα, να πάρει το έτοιμο CSV και να δοκιμάσει διαφορετικό μοντέλο ταξινόμησης, χωρίς να χρειάζεται να εγκαταστήσει τη συλλογή των Syzygy. Επιτρέπει επίσης την παράλληλη ανάπτυξη: αλλαγές στον RuleConsolidation δεν επηρεάζουν τον DecTree, και αντιστρόφως.

Η συνολική έκταση της εφαρμογής έφτασε τις δεκαπέντε χιλιάδες γραμμές κώδικα Python. Λόγω του μεγέθους αυτού, δεν θα ήταν λειτουργικό να συμπεριληφθεί ο πλήρης κώδικας στο κείμενο. Επιλέχθηκε αντ' αυτού η παρουσίαση επιλεγμένων αποσπασμάτων στα σημεία όπου ο κώδικας αναδεικνύει κάποια σχεδιαστική απόφαση ή λεπτομέρεια που αξίζει να επεξηγηθεί, ενώ ο πλήρης κώδικας διατίθεται μέσω συνδέσμου στην αρχή της εργασίας.

4.2 Φινάλε KPK (*King and Pawn vs King*)

4.2.1 Παραγωγή του dataset

Το πρώτο βήμα της εφαρμογής είναι η κατασκευή ενός πλήρους dataset που θα τροφοδοτήσει όλα τα επόμενα στάδια. Ο σχεδιασμός του βασίστηκε σε μια απλή αρχή: αφού ο χώρος των KPK[28] θέσεων είναι πεπερασμένος και σχετικά μικρός, δεν υπάρχει λόγος να αρκεστούμε σε δειγματοληψία. Το dataset περιλαμβάνει εξαντλητικά όλες τις νόμιμες θέσεις του φινάλε, ώστε το μοντέλο που θα προκύψει να μην εξαρτάται από την επιλογή του δείγματος.

Η διαδικασία υλοποιείται στο αρχείο `KPKGenerator.py` και ακολουθεί τέσσερα διαδοχικά βήματα: ορισμό του χώρου αναζήτησης, απαρίθμηση και νομιμοποίηση των θέσεων, αξιολόγηση κάθε θέσης μέσω probing στις Syzygy, και υπολογισμό των χαρακτηριστικών.

4.2.2 Σχεδίαση των χαρακτηριστικών

Η σχεδίαση των χαρακτηριστικών είναι από τις πιο σημαντικές αποφάσεις ολόκληρης της εργασίας. Σε μια διπλωματική με στόχο την εξαγωγή ερμηνεύσιμων κανόνων, τα features δεν είναι απλά είσοδος για το μοντέλο — είναι η γλώσσα στην οποία θα διατυπωθούν τελικά οι κανόνες. Ένα δέντρο εκπαιδευμένο πάνω σε ωμές συντεταγμένες θα παράγει κανόνες της μορφής «αν η στήλη του πιονιού είναι ≤ 2 και η σειρά του λευκού βασιλιά είναι ≥ 5 ...» — τεχνικά σωστούς, σκακιστικά άχρηστους. Ένα δέντρο εκπαιδευμένο πάνω σε χαρακτηριστικά όπως το «κερδίζει ο λευκός τον αγώνα για το key square» παράγει κανόνες που μπορούν να αναγνωριστούν από έναν σκακιστή και να συνδεθούν με γνωστές έννοιες της θεωρίας φινάλε.

Για το KPK σχεδιάστηκαν δεκατέσσερα χαρακτηριστικά, οργανωμένα σε τρεις θεματικές ομάδες με διαφορετικό ρόλο: τα γεωμετρικά που περιγράφουν την απλή δομή της θέσης, τα σκακιστικά κωδικοποιημένα που ενσωματώνουν συγκεκριμένες έννοιες της θεωρίας του φινάλε, και τα opposition και υποστήριξης που πιάνουν τη σχέση των δύο βασιλιάδων μεταξύ τους και με το πιόνι.

Γεωμετρικά χαρακτηριστικά

Η πρώτη ομάδα περιλαμβάνει οκτώ χαρακτηριστικά που περιγράφουν τη δομή της θέσης χωρίς να ενσωματώνουν σκακιστική θεωρία. Είναι, με μια έννοια, η ωμή γεωμετρία της θέσης: ποιος παίζει (`side_to_move`), πόσα βήματα απομένουν στο πιόνι μέχρι την προαγωγή (`steps_to_promotion`), αν το πιόνι είναι στη στήλη a — δηλαδή rook pawn (`is_rook_pawn`), αν ο μαύρος βασιλιάς βρίσκεται μπροστά από το πιόνι (`black_king_ahead_of_pawn`), και τέσσερα μεγέθη απόστασης που αποτυπώνουν τη σχετική θέση των κομματιών μεταξύ τους.

Οι αποστάσεις υπολογίζονται με τη μετρική Chebyshev — δηλαδή τον αριθμό κινήσεων που χρειάζεται ένας βασιλιάς για να καλύψει την απόσταση δύο τετραγώνων. Επιλέχθηκε έναντι της Manhattan απόστασης γιατί ταιριάζει με τον τρόπο που κινούνται πραγματικά τα κομμάτια στη σκακιά: ένας βασιλιάς που πρέπει να καλύψει δύο στήλες και δύο σειρές χρειάζεται δύο κινήσεις, όχι τέσσερις.

Από τα τέσσερα μεγέθη απόστασης, τα `white_king_pawn_distance` και `black_king_pawn_distance` αποτυπώνουν αν ο κάθε βασιλιάς είναι κοντά στο πιόνι — κρίσιμο για την υποστήριξη της προαγωγής από τη μία πλευρά και για τον αποκλεισμό της από την άλλη. Η τιμή `pawn_distance_diff` ποσοτικοποιεί τον σχετικό αγώνα των δύο βασιλιάδων προς το πιόνι: θετική τιμή σημαίνει ότι ο λευκός είναι πιο κοντά, αρνητική ότι ο μαύρος. Είναι μια από τις πιο φυσικές γεωμετρικές μετρικές για ένα φινάλε όπου ο αγώνας δρόμου είναι κεντρικός. Τέλος, το `kings_distance` καταγράφει την απόσταση μεταξύ των δύο βασιλιάδων, που σχετίζεται με την *opposition* και τις μεταξύ τους αλληλεπιδράσεις.

Η ομάδα αυτή είναι η πιο «αθώα» σκακιστικά — δεν προϋποθέτει γνώση θεωρίας. Είναι όμως αναγκαία: χωρίς αυτή, τα πιο εξελιγμένα *features* θα στερούνταν του βασικού γεωμετρικού πλαισίου μέσα στο οποίο νοηματοδοτούνται.

```
# Basic state / geometry.
"side_to_move": int(board.turn == chess.WHITE),
"steps_to_promotion": steps_to_promotion(white_pawn),
"is_rook_pawn": int(chess.square_file(white_pawn) == 0),
"black_king_ahead_of_pawn": black_king_ahead_of_pawn(black_king, white_pawn),
"white_king_pawn_distance": white_king_pawn_distance,
"black_king_pawn_distance": black_king_pawn_distance,
"pawn_distance_diff": black_king_pawn_distance - white_king_pawn_distance,
"kings_distance": kings_distance,
```

Εικόνα 4.1 Γεωμετρικά Χαρακτηριστικά

Σκακιστικά κωδικοποιημένα χαρακτηριστικά

Η δεύτερη ομάδα είναι η πιο πρωτότυπη και αποτελείται από τρία χαρακτηριστικά που ενσωματώνουν κωδικοποιημένη σκακιστική θεωρία. Σε αντίθεση με τα γεωμετρικά, αυτά τα *features* απαντούν σε σκακιστικά ερωτήματα: «προλαβαίνει ο βασιλιάς να πιάσει το πιόνι;», «μπλοκάρεται η προαγωγή;», «κερδίζει ο λευκός τον αγώνα δρόμου;». Είναι, με άλλα λόγια, η σκακιστική γνώση που μετατρέπεται σε αριθμό.

Το πρώτο, `black_king_inside_square_of_pawn`, υλοποιεί τον κλασικό κανόνα του τετραγώνου του πιονιού: ο μαύρος βασιλιάς πιάνει το πιόνι αν και μόνο αν βρίσκεται μέσα στο νοητό τετράγωνο που σχηματίζεται από το πιόνι και το τετράγωνο προαγωγής. Η μέτρηση γίνεται με Chebyshev απόσταση από το πιόνι προς το τετράγωνο προαγωγής, αλλά με μια κρίσιμη λεπτομέρεια.

```
promotion_square = chess.square(chess.square_file(white_pawn), 7)
steps = steps_to_promotion(white_pawn)

effective_steps = steps - 1 if board.turn == chess.WHITE else steps
effective_steps = max(effective_steps, 0)

black_king_to_promotion = chebyshev_distance(black_king, promotion_square)

return int(black_king_to_promotion <= effective_steps)
```

Εικόνα 4.2 *Black_king_inside_square_of_pawn*

Η αφαίρεση της μιας κίνησης όταν παίζει ο λευκός είναι σχεδόν αόρατη, αλλά σκακιστικά καθοριστική. Αν είναι η σειρά του λευκού να παίζει, το πιόνι θα προχωρήσει πρώτο — άρα ο μαύρος έχει μία λιγότερη κίνηση στη διάθεσή του για να πιάσει το «τετράγωνο». Χωρίς τη διόρθωση αυτή, το feature θα έδινε λάθος αποτέλεσμα ακριβώς στις πιο οριακές θέσεις, εκεί όπου η σωστή πληροφορία έχει τη μεγαλύτερη σημασία.

Το δεύτερο, `black_king_can_block_promotion`, ελέγχει αν ο μαύρος βασιλιάς προλαβαίνει να φτάσει σε ένα από τα τετράγωνα που μπλοκάρουν την προαγωγή — μια λιγότερο γνωστή αλλά εξίσου σημαντική έννοια. Στον υπολογισμό λαμβάνεται υπόψη ότι ο μαύρος βασιλιάς δεν μπορεί να σταθεί σε τετράγωνο γειτονικό του λευκού βασιλιά. Ο περιορισμός αυτός αφαιρεί αυτόματα κάποιους από τους υποψήφιους στόχους κι έτσι κάνει τον κανόνα ακριβέστερο.

Το τρίτο και πιο σύνθετο, `white_wins_key_square_race`, εφαρμόζει τη θεωρία των key squares. Στο KPK υπάρχει για κάθε θέση του πιονιού ένα σύνολο από τετράγωνα-κλειδιά: αν ο λευκός βασιλιάς προλάβει να φτάσει σε κάποιο από αυτά πριν τον μαύρο, η νίκη είναι θεωρητικά εξασφαλισμένη. Η συνάρτηση `white_wins_key_square_race` ελέγχει αυτή τη συνθήκη με τρόπο που λαμβάνει υπόψη και την opposition:

```
for key_square in key_squares:
    white_distance = chebyshev_distance(white_king, key_square)
    black_distance = chebyshev_distance(black_king, key_square)

    if white_distance < black_distance:
        return 1

    if white_distance == black_distance and has_direct_opposition and board.turn == chess.BLACK:
        return 1

return 0
```

Εικόνα 4.3 `White_wins_key_square_race`

Η πρώτη συνθήκη είναι απλή: ο λευκός είναι αυστηρά πιο κοντά σε κάποιο key square. Η δεύτερη όμως είναι λεπτή — όταν οι αποστάσεις είναι ίσες, ο λευκός εξακολουθεί να κερδίζει αν έχει direct opposition και ο μαύρος είναι αυτός που πρέπει να μετακινηθεί πρώτος (zugzwang). Αυτή η λεπτομέρεια του tempo είναι από τα πιο «λεπτεπίλεπτα» κομμάτια της θεωρίας του KPK, και η ενσωμάτωσή της εδώ δείχνει τη σκακιστική προσοχή με την οποία σχεδιάστηκε το feature.

Πρέπει να αναγνωριστεί όμως ένας περιορισμός: ο υπολογισμός των key squares για rook pawn είναι σκόπιμα απλοποιημένος, καθώς η πλήρης θεωρία της εξαίρεσης του πιονιού στη στήλη a απαιτεί ξεχωριστή μεταχείριση που ξεφεύγει από τα όρια ενός μεμονωμένου feature. Αυτή η απλοποίηση γίνεται μέσα σε ένα πλαίσιο όπου άλλα features (κυρίως το `is_rook_pawn`) δίνουν στο δέντρο τη δυνατότητα να «μάθει» τη διαφορετική συμπεριφορά αυτών των θέσεων.

```
# Chess-informed KPK features.
"black_king_inside_square_of_pawn": black_king_inside_square_of_pawn(
    board,
    white_pawn,
    black_king,
),
"black_king_can_block_promotion": black_king_can_block_promotion(
    board,
    white_pawn,
    white_king,
    black_king,
),
"white_wins_key_square_race": white_wins_key_square_race(
    board,
    white_king,
    black_king,
    white_pawn,
),
```

Εικόνα 4.4 Chess-informed KPK features

Opposition και υποστήριξη

Η τρίτη ομάδα αποτελείται από τρία features που αφορούν τη σχέση των δύο βασιλιάδων μεταξύ τους και με το πιόνι. Είναι μικρότερη σε πλήθος, αλλά πιάνει έννοιες που είναι κεντρικές στη θεωρία του φινάλε και που, χωρίς ξεχωριστή κωδικοποίηση, δύσκολα θα ανακαλύπτονταν από μια μηχανή με βάση τα απλά γεωμετρικά features.

Τα `kings_have_direct_opposition` και `kings_have_distant_opposition` πιάνουν δύο τύπους της κλασικής opposition: τη direct (οι βασιλιάδες σε απόσταση ακριβώς 2 σε ίδια γραμμή ή στήλη) και τη distant (απόσταση 4 ή 6 στην ίδια γραμμή ή στήλη). Στην opposition, η πλευρά που πρέπει να μετακινηθεί πρώτη βρίσκεται σε μειονεκτική θέση — το ξέρει κάθε σκακιστής που έχει μελετήσει KPK. Η ύπαρξη ή απουσία opposition είναι από τα πρώτα πράγματα που σκέφτεται κανείς όταν αναλύει αυτό το φινάλε.

Το `white_king_strongly_supports_pawn` ορίζει μια απλή αλλά χρήσιμη έννοια: ο λευκός βασιλιάς θεωρείται ότι «υποστηρίζει ισχυρά» το πιόνι όταν βρίσκεται τουλάχιστον δύο σειρές μπροστά του και όχι μακρύτερα από μία στήλη. Είναι η κλασική «υποστηρικτική» γεωμετρία: ο βασιλιάς ηγείται, το πιόνι ακολουθεί. Η διπλή συνθήκη ($\text{rank diff} \geq 2$ και $\text{file diff} \leq 1$) πιάνει ένα σαφές πρότυπο που αντιστοιχεί σε κερδισμένες θέσεις στις περισσότερες περιπτώσεις.

Αξίζει εδώ μια παρατήρηση που θα φανεί στα αποτελέσματα: τα features αυτής της ομάδας έχουν σαφή σκακιστική αξία, αλλά η αλληλεπικάλυψή τους με άλλα features (ιδιαίτερα τις αποστάσεις) είναι σημαντική. Στο τελικό μοντέλο, η συνεισφορά τους θα εξεταστεί σε σχέση με τα ισχυρότερα features των άλλων ομάδων.

```
# Opposition / support features.
"kings_have_direct_opposition": kings_have_direct_opposition(
    white_king,
    black_king,
),
"kings_have_distant_opposition": kings_have_distant_opposition(
    white_king,
    black_king,
),
"white_king_strongly_supports_pawn": white_king_strongly_supports_pawn(
    white_king,
    white_pawn,
),
```

Εικόνα 4.5 Opposition / support features

Η ετικέτα (label)

Πέρα από τα δεκατέσσερα features που τροφοδοτούν το δέντρο, το dataset κρατάει και τρεις στήλες σχετικές με την έκβαση κάθε θέσης. Η `wdl_raw` είναι η ωμή τιμή που επιστρέφει η Syzygy μέσω `probe_wdl` — διατηρείται κυρίως για διαγνωστικούς λόγους. Η `result_class` είναι η συμβολική κλάση ("Win" ή "Draw") από τη σκοπιά του λευκού. Και η `white_is_winning` είναι η τελική δυαδική στήλη (0 ή 1) που χρησιμοποιείται ως target του ταξινομητή.

Η διατήρηση των τριών αυτών στηλών δεν είναι περιττή. Η `wdl_raw` επιτρέπει σε μελλοντικές αναλύσεις να διακρίνουν λεπτότερες περιπτώσεις (όπως `cursed wins` ή `blessed losses`) χωρίς να ξαναγίνει probing. Η `result_class` παρέχει αναγνώσιμη μορφή για ταχείες οπτικές επιθεωρήσεις του dataset. Και η `white_is_winning` τροφοδοτεί απευθείας τον ταξινομητή.

```
# Syzygy WDL labels.
"wdl_raw": raw_wdl,
"result_class": result_class,
"white_is_winning": int(result_class == "Win"),
```

Εικόνα 4.6 Syzygy WDL labels

4.2.3 Εκπαίδευση δέντρων απόφασης

Μετά την κατασκευή του dataset και τη σχεδίαση των features, το επόμενο βήμα είναι η εκπαίδευση του ίδιου του δέντρου απόφασης. Παρά τη φαινομενική απλότητα του DecisionTreeClassifier της scikit-learn, η διαδικασία εκπαίδευσης στη συγκεκριμένη εφαρμογή ενσωματώνει αρκετές σχεδιαστικές επιλογές, οι οποίες αποσκοπούν στην εξασφάλιση δύο πραγμάτων: αξιόπιστης αξιολόγησης της απόδοσης και βελτιστοποιημένης ισορροπίας μεταξύ ακρίβειας και ερμηνευσιμότητας.

Διαχωρισμός δεδομένων

Η πρώτη απόφαση αφορά τον τρόπο με τον οποίο χωρίζονται τα δεδομένα. Επιλέχθηκε ο κλασικός διαχωρισμός 80/20 σε εκπαίδευση και έλεγχο, με στρωματοποίηση (stratification) ως προς το target. Η στρωματοποίηση εξασφαλίζει ότι η αναλογία Win προς Draw παραμένει η ίδια και στα δύο σύνολα — κρίσιμο σε ένα dataset με ανισόρροπη κατανομή κλάσεων όπως αυτό του KPK.

Επιλογή υπερπαραμέτρων

Η πιο ενδιαφέρουσα σχεδιαστική επιλογή αφορά τον τρόπο με τον οποίο επιλέγονται οι κρίσιμες υπερπαραμέτροι: το μέγιστο βάθος του δέντρου και η στάθμιση κλάσεων (class weight). Αντί για μία αυθαίρετη επιλογή, υλοποιήθηκε συστηματική αναζήτηση πάνω σε grid.

Δοκιμάστηκαν τρία βάθη (7, 9, 12) και δύο επιλογές class weight (καμία στάθμιση και "balanced"). Κάθε συνδυασμός εκπαιδεύεται με τρία διαφορετικά seeds (42, 123, 456) ώστε να εκτιμηθεί η σταθερότητα των αποτελεσμάτων ως προς τη στοχαστικότητα του train/test split. Συνολικά, αυτό σημαίνει 18 διαφορετικές εκπαιδεύσεις ($3 \times 2 \times 3$) που αξιολογούνται πριν την τελική επιλογή.

Η επιλογή του βέλτιστου συνδυασμού δεν γίνεται με βάση τη σκέτη ακρίβεια, αλλά μέσω ενός adjusted score που τιμωρεί την πολυπλοκότητα.

```
scored["complexity_penalty"] = penalty_weight * np.sqrt(
    scored["number_of_leaves_mean"] / max_leaves
)

scored["adjusted_score"] = (
    scored["balanced_accuracy_mean"] - scored["complexity_penalty"]
)
```

Εικόνα 4.7 Adjusted score

Η μέση balanced accuracy στα τρία seeds είναι το κύριο μέγεθος, αλλά αφαιρείται μια ποινή ανάλογη του πλήθους φύλλων του δέντρου, κανονικοποιημένου ως προς το μέγιστο πλήθος φύλλων που παρατηρείται σε ολόκληρο το grid. Το PENALTY_WEIGHT = 0.01 είναι αρκετά μικρό ώστε η ακρίβεια να παραμένει το κυρίαρχο κριτήριο, αλλά αρκετά μη μηδενικό ώστε όταν δύο μοντέλα έχουν παρόμοια απόδοση, να προτιμάται το απλούστερο.

Διασταυρωμένη επαλήθευση

Παράλληλα με την αξιολόγηση στο test set, εκτελείται και 5-fold cross-validation πάνω στο σύνολο των δεδομένων. Η balanced accuracy μετράται σε κάθε fold και αναφέρεται ο μέσος όρος μαζί με την τυπική απόκλιση. Η διπλή αυτή αξιολόγηση δίνει πιο αξιόπιστη εικόνα της γενικευτικής ικανότητας του μοντέλου από έναν μόνο διαχωρισμό.

Για να εξασφαλιστεί η αξιοπιστία της αξιολόγησης, στην παρουσίαση των αποτελεσμάτων η κύρια μετρική που αναφέρεται είναι η cross-validation accuracy, η οποία υπολογίζεται ανεξάρτητα από τη διαδικασία επιλογής υπερπαραμέτρων. Η τιμή στο test split αναφέρεται συμπληρωματικά, ως δευτερεύουσα ένδειξη. Με τον τρόπο αυτό, η αναφερόμενη επίδοση δεν επηρεάζεται από την αλληλεπίδραση μεταξύ της επιλογής βέλτιστου μοντέλου και του τελικού ελέγχου.

Περιορισμοί κατά την εκπαίδευση

Πέρα από τις παραπάνω επιλογές, στην εκπαίδευση κάθε δέντρου επιβάλλονται και κάποιοι περιορισμοί κανονικοποίησης: ελάχιστο πλήθος δειγμάτων ανά εσωτερικό κόμβο `min_samples_split = 50` και ελάχιστο πλήθος δειγμάτων ανά φύλλο `min_samples_leaf = 25`. Οι τιμές αυτές είναι αρκετά μεγάλες ώστε να αποτρέπουν την υπερπροσαρμογή σε μεμονωμένες οριακές θέσεις, αλλά αρκετά μικρές ώστε να επιτρέπουν στο δέντρο να εντοπίζει πραγματικά μοτίβα. Έχουν ως αποτέλεσμα ένα δέντρο όπου κάθε φύλλο αντιστοιχεί σε τουλάχιστον 25 πραγματικές θέσεις — όριο που διασφαλίζει ότι οι κανόνες που θα εξαχθούν αργότερα έχουν στατιστική σημασία.

4.2.4 Σύγκριση επιπέδων αναπαράστασης

Το πλήρες σύνολο των δεκατεσσάρων features περιλαμβάνει δύο διαφορετικούς τύπους πληροφορίας: παράγωγα γεωμετρικά μεγέθη και σκακιστικά κωδικοποιημένες έννοιες. Πρακτικά, αυτό σημαίνει ότι το δέντρο εκπαιδεύεται πάνω σε ένα μείγμα γνώσης διαφορετικού επιπέδου. Η συνεισφορά κάθε τύπου ξεχωριστά παραμένει όμως αδιευκρίνιστη χωρίς ξεχωριστή ανάλυση.

Για τον σκοπό αυτό, υλοποιήθηκε στο αρχείο `KPKComp.py` μια συστηματική σύγκριση τριών διαφορετικών επιπέδων αναπαράστασης της θέσης. Στόχος της σύγκρισης είναι να εκτιμηθεί πόσο πραγματικά συνεισφέρει η ενσωματωμένη σκακιστική γνώση στην απόδοση και στην ερμηνευσιμότητα του μοντέλου, σε σχέση με μια απλούστερη αναπαράσταση που στηρίζεται μόνο σε γεωμετρικά μεγέθη ή ακόμα και σε ωμές συντεταγμένες των κομματιών.

Τα τρία επίπεδα

Το πρώτο επίπεδο, `raw coordinates`, περιλαμβάνει μόνο τις ωμές γεωμετρικές συντεταγμένες των κομματιών: τη στήλη και τη σειρά καθενός από τους δύο βασιλιάδες και του πιονιού, καθώς και ποιος έχει τη σειρά κίνησης. Το δέντρο δεν λαμβάνει καμία υπόθεση για το τι σημαίνουν αυτές οι συντεταγμένες· πρέπει να ανακαλύψει μόνο του οποιαδήποτε δομή.

Το δεύτερο επίπεδο, `geometric features`, προσθέτει παράγωγα μεγέθη που εκφράζουν σχέσεις μεταξύ των κομματιών: αποστάσεις βασιλιάδων από το πιόνι,

μεταξύ τους, βήματα μέχρι την προαγωγή, η ένδειξη rook pawn, και η διαφορά αποστάσεων. Είναι ακόμα γεωμετρική πληροφορία, αλλά πιο επεξεργασμένη — αντί για «πού είναι κάθε κομμάτι», η ερώτηση γίνεται «πώς σχετίζονται μεταξύ τους».

Το τρίτο επίπεδο, full strategic features, είναι το πλήρες σύνολο που χρησιμοποιήθηκε στην 4.2.3. Προσθέτει στα γεωμετρικά τα σκακιστικά κωδικοποιημένα features (rule of the square, key square race, block promotion) και τις opposition/support έννοιες. Είναι το επίπεδο που ενσωματώνει ουσιαστική σκακιστική θεωρία.

Μεθοδολογία σύγκρισης

Για να είναι η σύγκριση δίκαιη, και τα τρία επίπεδα εκπαιδεύονται με την ίδια ακριβώς διαδικασία: ίδιο train/test split, ίδιο βάθος και class weight (αυτό που επελέγη ως βέλτιστο για το full set), ίδιο πλήθος seeds. Η μόνη διαφορά είναι ο πίνακας X των features. Αυτό εξασφαλίζει ότι όποιες διαφορές προκύπτουν στις μετρικές οφείλονται αποκλειστικά στην αναπαράσταση των δεδομένων.

Συγκρίνονται τρία μεγέθη: η balanced accuracy (πόσο καλά προβλέπει), το πλήθος φύλλων του δέντρου (πόσο σύνθετοι είναι οι κανόνες που χρειάζονται), και το feature importance ranking (ποια features κυριαρχούν στις αποφάσεις).

4.2.5 Ομαδοποίηση κανόνων σε motifs

Η συγκεκριμένη ενότητα αποτελεί την πιο πρωτότυπη ενότητα της εργασίας. Το πρόβλημα που λύνει είναι το εξής: ένα εκπαιδευμένο δέντρο απόφασης για το KPK παράγει συνήθως εκατοντάδες leaf rules — δηλαδή ένα μονοπάτι από τη ρίζα μέχρι κάθε φύλλο αντιστοιχεί σε έναν διακριτό κανόνα. Από τη μία πλευρά, αυτή είναι η δύναμη των δέντρων: παράγουν ρητούς κανόνες. Από την άλλη, εκατοντάδες κανόνες δεν είναι ερμηνεύσιμοι από άνθρωπο. Οπότε σκοπός είναι η εξαγωγή ενός σχετικά μικρού συνόλου αναγνωρίσιμων μοτίβων — όχι μια λίστα 300 τεχνικών συνθηκών.

Το βήμα του rule consolidation, υλοποιημένο στο KPKRuleConsolidation.py, αναλαμβάνει αυτή τη συμπίεση. Παίρνει την έξοδο του DecTree και τη μετατρέπει σε ένα μικρό σύνολο ονοματισμένων motifs που μπορούν να αναγνωριστούν ως γνωστές έννοιες της θεωρίας φινάλε.

Το πρώτο βήμα είναι η μετατροπή κάθε leaf rule σε μια κανονικοποιημένη συμβολοσειρά που λέγεται signature. Η ιδέα είναι ότι πολλά leaf rules διαφέρουν μεταξύ τους μόνο σε λεπτομέρειες — ένα όριο 4.5 αντί 5.5, για παράδειγμα — ενώ ουσιαστικά περιγράφουν το ίδιο μοτίβο. Αν μπορέσουμε να συμπτύξουμε τις παραλλαγές αυτές σε μια κοινή signature, μειώνεται δραστικά το πλήθος των διακριτών κανόνων.

Η κανονικοποίηση γίνεται μέσω συστηματικού binning των αριθμητικών χαρακτηριστικών. Για κάθε feature ορίζεται μια διατεταγμένη λίστα από bins με σκακιστικά νοηματοδοτημένες ετικέτες

Με τις λίστες αυτές, κάθε όριο στο εκάστοτε leaf rule μεταφράζεται σε διακριτή ετικέτα. Π.χ. η συνθήκη $\text{kings_distance} \leq 2$ γίνεται CLOSE, η $\text{white_king_pawn_distance} \leq 1$ γίνεται TOUCHING. Όλα τα leaf rules με τις ίδιες

ετικέτες σε όλα τα features συγκεντρώνονται κάτω από την ίδια signature. Έτσι, δέκα παραλλαγές που διέφεραν μόνο σε δευτερεύοντα όρια συμπύσσονται σε μία σαφή περιγραφή.

To interpretive layer

Η συμπίεση σε signatures είναι μεν πιο διαχειρίσιμη από εκατοντάδες leaf rules, αλλά οι signatures εξακολουθούν να είναι τεχνικές. Μια signature όπως «TOUCHING + ADJACENT + INSIDE_SQUARE + BLOCK + WINS_RACE» είναι μεν συμπυκνωμένη, αλλά δεν είναι σκακιστική περιγραφή.

Το επόμενο βήμα είναι η αντιστοίχιση signatures σε ονοματισμένα motifs, που είναι ονόματα γνωστών σκακιστικών εννοιών. Η αντιστοίχιση γίνεται μέσω της συνάρτησης `identify_interpretive_motif`, η οποία ελέγχει για κάθε signature αν ταιριάζει με ένα από τα προκαθορισμένα πρότυπα. Παραδείγματα τέτοιων motifs στο KPK είναι: «king-pawn-race won by white», «blocked promotion in drawn position», «opposition wins for white in critical phase».

Πρέπει εδώ να γίνει μια αναφορά σε ένα μεθοδολογικό χαρακτηριστικό αυτού του βήματος: η αντιστοίχιση signatures σε motifs δεν είναι αυτόματη. Έγινε χειροκίνητα, με βάση σκακιστική παρατήρηση των κυρίαρχων signatures στο εκπαιδευμένο δέντρο. Δηλαδή, αφού παρήχθησαν οι signatures από τα leaf rules, εξετάστηκαν χειροκίνητα ποιες από αυτές αντιστοιχούσαν σε καθαρά αναγνωρίσιμα σκακιστικά μοτίβα, και ορίστηκαν προγραμματιστικά οι αντίστοιχοι κανόνες αναγνώρισης.

Αυτό είναι ένα semi-handcrafted interpretation layer. Δεν είναι μια αυτόνομα μαθημένη ομαδοποίηση μέσω clustering ή frequent itemset mining — είναι σκακιστικά καθοδηγούμενη ερμηνεία. Η δικαιολόγηση αυτής της επιλογής είναι ότι ο στόχος δεν είναι η ανακάλυψη άγνωστης δομής, αλλά η σύνδεση του δέντρου με γνωστή σκακιστική θεωρία. Όταν το δέντρο «ανακαλύπτει» έναν κανόνα που είναι ουσιαστικά η οριοθέτηση τετραγώνου του πιονιού, αξίζει να ονομαστεί έτσι αντί να μείνει ως ανώνυμη signature.

Catch-all και κάλυψη

Τέλος, δεν αντιστοιχίζονται όλες οι signature σε motifs. Όσες δεν ταιριάζουν με κανένα από τα προκαθορισμένα πρότυπα ομαδοποιούνται σε μια κατηγορία «catch-all». Το ποσοστό των leaf rules που πέφτει στο catch-all είναι ουσιαστικός δείκτης ποιότητας του interpretation layer: όσο μικρότερο, τόσο πιο πλήρης η σκακιστική κάλυψη. Για το KPK, το catch-all ποσοστό είναι ουσιαστικά κάτω από 1% και για τις δύο κλάσεις, αποτέλεσμα που δείχνει σχεδόν πλήρη σκακιστική κάλυψη.

4.2.6 Οπτικοποίηση και τεκμηρίωση

Τα δύο τελευταία αρχεία του pipeline έχουν περισσότερο υποστηρικτικό παρά μεθοδολογικό ρόλο και αναφέρονται για πληρότητα.

Το `KPKVisual.py` παράγει τα γραφήματα που συνοδεύουν την ανάλυση: την οπτικοποίηση του δέντρου απόφασης (περιορισμένη στα πρώτα επίπεδα για λόγους αναγνωσιμότητας), τα feature importance bar plots, τους confusion matrices, τα συγκριτικά διαγράμματα των τριών επιπέδων features, και τα διαγράμματα

σταθερότητας ως προς το seed. Όλα παράγονται με matplotlib και αποθηκεύονται ως αρχεία PNG σε δομημένο φάκελο εξόδου.

Το KPKReadMe.py δημιουργεί ένα αυτοτελές αρχείο README.md που τεκμηριώνει την εκτέλεση: τι παρήχθη, ποια υπερπαραμέτροι επιλέχθηκαν, ποιες είναι οι κύριες μετρικές, και ποια είναι η ερμηνεία των motifs.

Η συμβολή των δύο αυτών αρχείων στη μεθοδολογία είναι έμμεση: εξυπηρετούν την αναπαραγωγιμότητα και τη διαφάνεια, όχι την παραγωγή των αποτελεσμάτων.

4.2.7 Αποτελέσματα για το φινάλε KPK

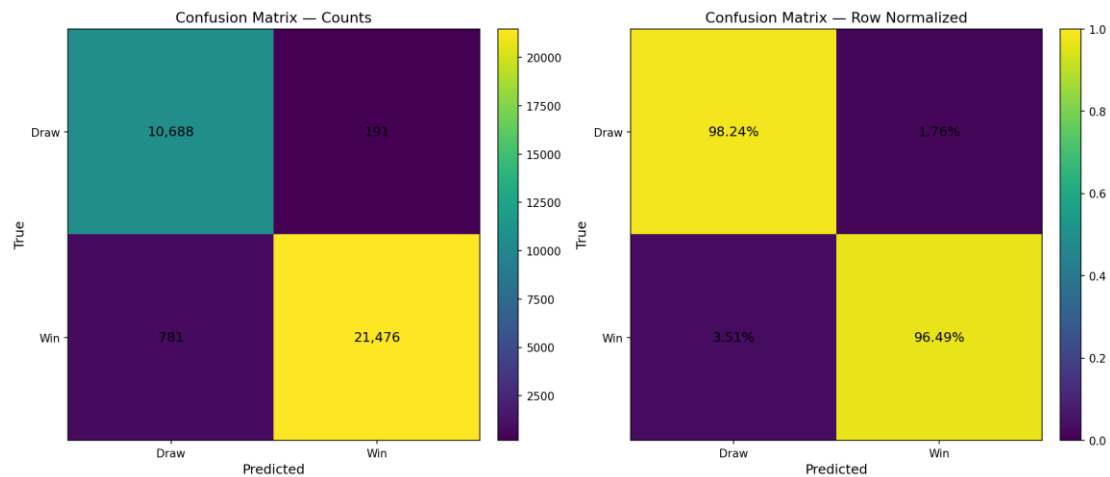
Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα της εφαρμογής του παραπάνω pipeline στο φινάλε KPK. Η παρουσίαση οργανώνεται σε τέσσερις άξονες: γενική απόδοση του τελικού μοντέλου, ανάλυση της συμβολής κάθε feature, σύγκριση των τριών επιπέδων αναπαράστασης, και ανάλυση των παραγόμενων κανόνων.

Απόδοση του τελικού μοντέλου

Ο βέλτιστος συνδυασμός υπερπαραμέτρων που επιλέχθηκε μέσω του adjusted score grid search ήταν βάθος 12 με class weight balanced. Το αντίστοιχο μοντέλο επιτυγχάνει balanced accuracy 0.9737 στο test set και αντίστοιχη τιμή 0.9728 ± 0.0008 στην 5-fold cross-validation, με 338 φύλλα. Η εξαιρετική συμφωνία ανάμεσα στις δύο μετρικές υποδεικνύει ότι η επίδοση είναι σταθερή και δεν προκύπτει από τυχαίες ιδιαιτερότητες του διαχωρισμού.

Από τα συνολικά 972 λάθη του μοντέλου στο test set, 191 είναι false positives (το μοντέλο προβλέπει νίκη ενώ η θέση είναι ισοπαλία) και 781 είναι false negatives (προβλέπει ισοπαλία ενώ η θέση είναι νίκη). Δηλαδή το μοντέλο είναι συντηρητικό: σπάνια ισχυρίζεται ψευδή νίκη, αλλά αρκετά συχνά χάνει πραγματικές νίκες.

Η ασυμμετρία αυτή έχει σκακιστική σημασία. Ένα false positive σημαίνει στην πράξη ότι ένας σκακιστής που εμπιστεύτηκε το μοντέλο θα έπαιζε για νίκη σε ισόπαλη θέση. Ένα false negative σημαίνει ότι ο σκακιστής θα εγκατέλειπε ή θα δεχόταν ισοπαλία σε κερδισμένη θέση — απογοητευτικό, αλλά όχι καταστροφικό. Το ότι το μοντέλο γέρνει προς τα false negatives σημαίνει ότι, αν χρησιμοποιηθεί πρακτικά, είναι ασφαλέστερο προς τη συντηρητική κατεύθυνση.



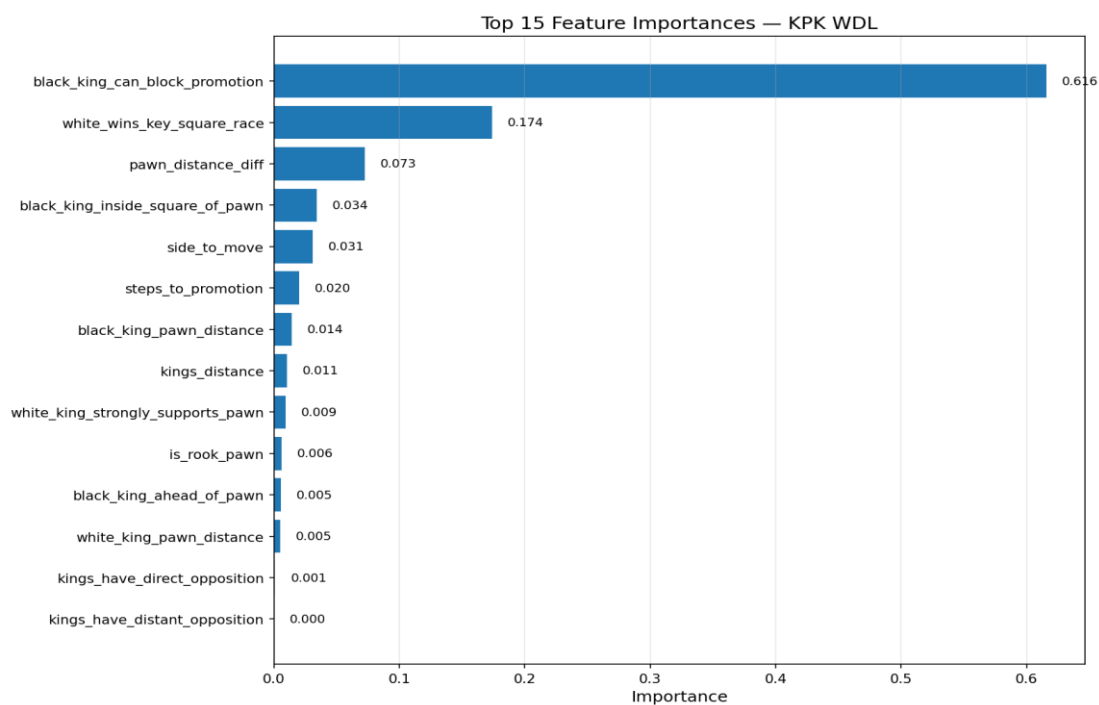
Εικόνα 4.8 KPK Confusion matrix

Feature importances

Η ανάλυση της συμβολής κάθε feature στις αποφάσεις του δέντρου αποκαλύπτει ότι δύο features κυριαρχούν: το `black_king_can_block_promotion` με importance 0.616, και το `white_wins_key_square_race` με 0.174. Μαζί αυτά τα δύο εξηγούν περίπου το 79% των αποφάσεων του δέντρου. Σε μεγαλύτερη απόσταση ακολουθούν το `pawn_distance_diff` (0.073), το `black_king_inside_square_of_pawn` (0.034), και το `side_to_move` (0.031), ενώ όλα τα υπόλοιπα features συνεισφέρουν λιγότερο από 2% το καθένα.

Σκακιστικά αυτό βγάζει νόημα. Στο KPK, οι δύο πιο κρίσιμες ερωτήσεις είναι ακριβώς αυτές: «μπορεί ο μαύρος να μπλοκάρει την προαγωγή;» και «κερδίζει ο λευκός τον αγώνα για το key square;». Το δέντρο, με βάση τα δεδομένα από τις Syzygy, ανακάλυψε ακριβώς αυτή την ιεραρχία.

Έγινε διαγνωστικός έλεγχος για το ενδεχόμενο το `black_king_can_block_promotion` να αποτελεί proxy του target — δηλαδή να κωδικοποιεί σχεδόν πλήρως την απάντηση. Η εμπειρική ανάλυση έδειξε ότι μέσα στις θέσεις όπου το feature είναι 1, μόνο το 66.7% είναι Draw — δηλαδή το ένα τρίτο των θέσεων που μπλοκάρουν την προαγωγή είναι παρ' όλα αυτά κερδισμένες. Άρα το feature από μόνο του δεν αρκεί για να προβλέψει την έκβαση: χρειάζεται τα υπόλοιπα features για να διακρίνει τις κερδισμένες θέσεις από τις ισόπαλες μέσα στις 79.6 χιλιάδες θέσεις όπου ο μαύρος μπορεί να μπλοκάρει. Το δέντρο πραγματικά λύνει ένα μη τετριμμένο πρόβλημα.



Εικόνα 4.9 KPK Feature importances

Σύγκριση των τριών επιπέδων αναπαράστασης

Το πιο ενδιαφέρον εύρημα της εργασίας για το KPK προκύπτει από τη σύγκριση των τριών επιπέδων features.

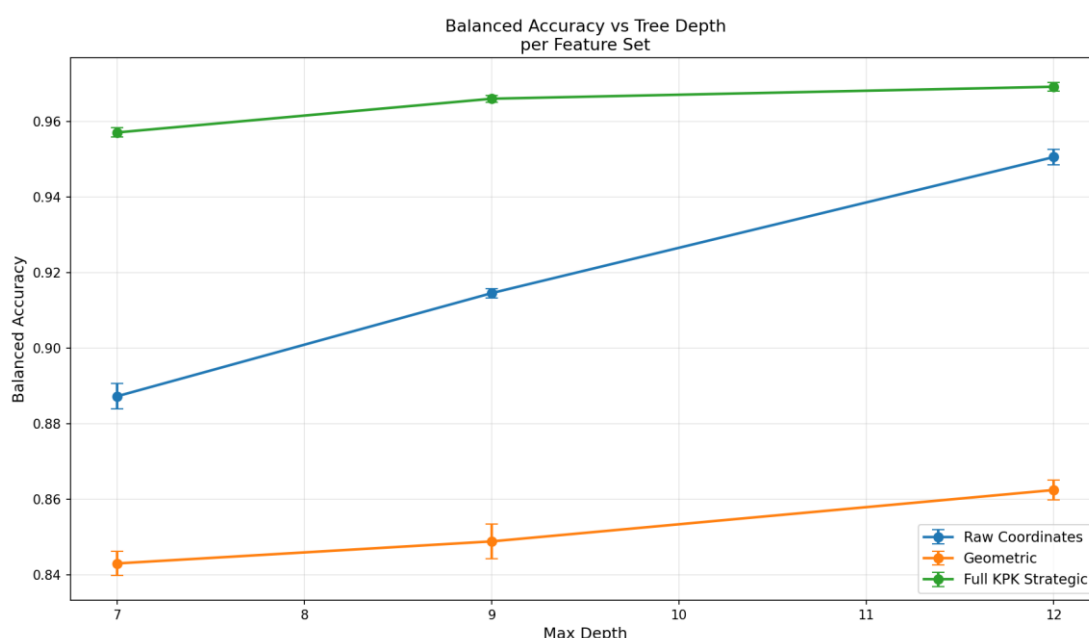
Επίπεδο	Balanced Accuracy	Πλήθος φύλλων	Λάθη στο test
Raw coordinates	0.9542	713	1.711
Geometric features	0.8821	766	4.378
Full strategic	0.9733	338	972

Πίνακας 4.1 Σύγκριση των τριών επιπέδων αναπαράστασης στο KPK

Δύο πράγματα ξεχωρίζουν. Πρώτον, το geometric επίπεδο τα πάει αισθητά χειρότερα από το raw, παρότι περιέχει πιο επεξεργασμένη πληροφορία. Δεύτερον, το full strategic επίπεδο πετυχαίνει αξιοσημείωτη βελτίωση accuracy έναντι του raw (από 0.9542 σε 0.9733) και μειώνει τα λάθη κατά 43% (από 1711 σε 972) και το πλήθος φύλλων στο μισό (από 713 σε 338).

Η ερμηνεία αυτών των ευρημάτων δίνει την απάντηση σε ένα από τα κεντρικά ερευνητικά ερωτήματα της εργασίας. Το *geometric* επίπεδο τα πάζει χειρότερα από το *raw* γιατί οι αποστάσεις, ως σχετική πληροφορία, χάνουν την απόλυτη θέση — που στο KPK είναι κρίσιμη, αφού η συμπεριφορά μιας θέσης εξαρτάται ουσιαστικά από το αν τα κομμάτια βρίσκονται στο κέντρο ή στην άκρη της σκακιέρας. Το *strategic* επίπεδο δίνει σαφή βελτίωση *accuracy*, και παράγει πολύ πιο συμπαγείς κανόνες, ουσιαστικά με τα μισά φύλλα. Δηλαδή η σκακιστική γνώση δεν αξίζει μόνο επειδή κάνει το μοντέλο πιο ακριβές, αλλά κυρίως επειδή κάνει τους κανόνες πολύ πιο ερμηνεύσιμους.

Αυτό είναι ακριβώς το είδος συμπεράσματος που δικαιολογεί τον τίτλο της διπλωματικής: η εξαγωγή κανόνων δεν είναι ζήτημα μέγιστης ακρίβειας, αλλά της σχέσης μεταξύ ακρίβειας και πολυπλοκότητας. Με σκακιστικά κωδικοποιημένα *features*, μεγαλύτερη ακρίβεια επιτυγχάνεται με τους μισούς κανόνες — και αυτοί οι κανόνες έχουν σκακιστικό νόημα.



Εικόνα 4.10 Συγκριτικό γράφημα *balanced accuracy*

Ανάλυση των παραγόμενων κανόνων

Από τις 338 *leaf rules* του τελικού δέντρου, το *rule consolidation* παράγει συνολικά 11 ονοματισμένα *motifs* — 5 για τη Draw κλάση και 6 για τη Win κλάση — που καλύπτουν τη συντριπτική πλειονότητα των θέσεων. Το ποσοστό κανόνων που πέφτει στο *catch-all* είναι μόλις 0.75% για τη Draw κλάση και 0.31% για τη Win κλάση — δηλαδή ουσιαστικά κάτω από 1% συνολικά. Αυτό σημαίνει ότι σχεδόν κάθε κανόνας που παράγει το δέντρο αναγνωρίζεται ως μια συγκεκριμένη σκακιστική έννοια.

Τα κυρίαρχα *motifs* αντιστοιχούν ακριβώς στις θεωρητικές έννοιες του KPK φινάλε. Για τις θέσεις νίκης, το πιο συχνό *motif* (που καλύπτει 77.2% των κερδισμένων θέσεων) είναι το *win_promotion_race_signature*, που περιγράφεται ως «Νίκη: ο μαύρος βασιλιάς δεν βρίσκεται μέσα στο τετράγωνο του πιονιού και δεν προλαβαίνει

να δημιουργήσει blockade». Ακολουθεί το `win_key_square_race_only_signature` (14.9%) — «Νίκη: ο λευκός βασιλιάς κερδίζει τον αγώνα προς τα κρίσιμα τετράγωνα». Για τις θέσεις ισοπαλίας, το πιο συχνό είναι το `draw_promotion_blockade_signature` (54.7%) — «Ισοπαλία: ο μαύρος βασιλιάς προλαβαίνει να μπλοκάρει την προαγωγή» — ακολουθούμενο από το `draw_square_rule_and_blockade_signature` (36.3%) — «Ισοπαλία: ο μαύρος βρίσκεται μέσα στο τετράγωνο του πιονιού και μπορεί να οργανώσει blockade».

Είναι αξιοσημείωτο ότι τα παραπάνω motifs αντιστοιχούν σε έννοιες που χρησιμοποιούνται φυσικά στη θεωρία του KPK. Ένας σκακιστής θα περιέγραφε το φινάλε με παρόμοιους όρους, όπως αγώνας για το τετράγωνο του πιονιού, αγώνας για τα key squares, blockade και υποστήριξη από τον βασιλιά. Το σημαντικό είναι ότι το δέντρο δεν είχε άμεση πρόσβαση στους ίδιους τους θεωρητικούς κανόνες του KPK, αλλά εκπαιδεύτηκε πάνω σε αξιολογήσεις Syzygy και σε χαρακτηριστικά που είχαν εξαχθεί από τις θέσεις. Παρ' όλα αυτά, οι κανόνες που παρήγαγε αντιστοιχούν καθαρά σε γνωστές σκακιστικές έννοιες. Το αποτέλεσμα αυτό αποτελεί ισχυρή ένδειξη ότι ο βασικός στόχος της εργασίας είναι εφικτός: η πληροφορία των tablebases μπορεί να μετατραπεί, μέσω δέντρων απόφασης, σε ερμηνεύσιμους κανόνες με πραγματικό σκακιστικό νόημα.

4.3 Φινάλε KRK (*King and Rook vs King*)

Η μελέτη του φινάλε KRK ακολουθεί την ίδια αρχιτεκτονική με το KPK: έξι αρχεία Python, ίδια ροή επεξεργασίας, ίδιες μεθοδολογικές αρχές. Οι διαφορές εντοπίζονται σε δύο επίπεδα. Στο επίπεδο της πηγής ετικέτας, η αξιολόγηση κάθε θέσης δεν προέρχεται πλέον από WDL αλλά από DTZ probing, με την τιμή που επιστρέφει η Syzygy να ομαδοποιείται σε τέσσερις φάσεις μετατροπής. Στο επίπεδο των χαρακτηριστικών, η σκακιστική φύση του KRK απαιτεί εντελώς διαφορετικές έννοιες: αντί για rule of the square και key squares, εδώ κυριαρχούν ο περιορισμός του βασιλιά μέσω του πύργου, η οδήγησή του προς την άκρη, και η δημιουργία mating net.

4.3.1 Παραγωγή του dataset

Το αρχείο `KRKGenerator.py` υλοποιεί την παραγωγή του dataset ακολουθώντας τα ίδια τέσσερα διαδοχικά βήματα με το KPK, αλλά με τρεις ουσιαστικές διαφοροποιήσεις.

Μη εφαρμογή συμμετρίας

Η πρώτη διαφορά αφορά τον τρόπο με τον οποίο αντιμετωπίζεται ο χώρος αναζήτησης. Σε αντίθεση με το KPK, όπου αξιοποιήθηκε οριζόντια συμμετρία λόγω της ύπαρξης πιονιού, στο KRK δεν εφαρμόστηκε κανονικοποίηση συμμετρίας στο dataset. Η επιλογή αυτή δεν σημαίνει ότι το φινάλε KRK δεν διαθέτει γεωμετρικές συμμετρίες. Αντίθετα, επειδή δεν υπάρχει πιόνι με συγκεκριμένη κατεύθυνση κίνησης, το φινάλε παρουσιάζει συμμετρικές διατάξεις ως προς άξονες και περιστροφές της σκακιέρας.

Στην παρούσα εργασία, όμως, οι συμμετρίες αυτές δεν αξιοποιήθηκαν στη φάση παραγωγής του dataset. Οι θέσεις διατηρήθηκαν στην αρχική τους μορφή, ώστε το μοντέλο να εκπαιδευτεί σε όλες τις δυνατές γεωμετρικές διατάξεις του φινάλε. Με αυτόν τον τρόπο, η σύγκριση των feature sets παραμένει πιο άμεση, καθώς το raw coordinate επίπεδο, το geometric επίπεδο και το full strategic επίπεδο αξιολογούνται πάνω στον ίδιο πλήρη χώρο θέσεων, χωρίς πρόσθετη κανονικοποίηση.

Φιλτράρισμα μόνο winning θέσεων

Η δεύτερη σημαντική διαφορά είναι η φύση του προβλήματος που λύνεται. Στο KPK, το ερώτημα ήταν αν ο λευκός κερδίζει — ερώτημα που έχει νόημα να τεθεί για κάθε θέση, ανεξάρτητα από το αποτέλεσμα. Στο KRK, αντίθετα, η παρούσα εργασία δεν εξετάζει όλες τις δυνατές διατάξεις ανεξαιρέτως, αλλά τις θέσεις που, σύμφωνα με τις Syzygy tablebases, είναι θεωρητικά κερδισμένες για τον λευκό. Υπάρχουν φυσικά διατάξεις, στις οποίες ο λευκός πύργος μπορεί να βρίσκεται εκτεθειμένος δίπλα στον μαύρο βασιλιά χωρίς επαρκή υποστήριξη από τον λευκό βασιλιά, με αποτέλεσμα ο μαύρος να μπορεί να τον αιχμαλωτίσει και η θέση να οδηγείται πρακτικά σε ισοπαλία. Οι περιπτώσεις αυτές δεν αποτελούν το αντικείμενο της συγκεκριμένης ταξινόμησης. Εφόσον λοιπόν στο dataset διατηρούνται οι θέσεις που έχουν ήδη αξιολογηθεί ως κερδισμένες για τον λευκό, το ενδιαφέρον δεν βρίσκεται στο ερώτημα «κερδίζει;», αλλά στο «πόσο άμεσα μετατρέπεται η θέση;».

Για τον λόγο αυτό, το dataset περιορίζεται αποκλειστικά σε θέσεις όπου παίζει ο λευκός και όπου η Syzygy επιβεβαιώνει ότι η θέση είναι κερδισμένη ($WDL = 2$). Η επιλογή αυτή απλοποιεί την ερμηνεία των αποτελεσμάτων: κάθε παραγόμενος κανόνας του μοντέλου θα αφορά την ταξινόμηση κερδισμένων θέσεων σε φάσεις προόδου προς τη νίκη, χωρίς επικαλύψεις με το ζήτημα της ίδιας της έκβασης.

Μετάβαση από WDL σε DTZ

Η τρίτη και πιο ουσιαστική διαφορά αφορά την ετικέτα. Αντί για το probe_wdl, η αξιολόγηση γίνεται μέσω probe_dtz, που επιστρέφει τον αριθμό κινήσεων μέχρι την επόμενη κίνηση πιονιού ή αιχμαλωσία υπό τέλειο παιχνίδι. Στο KRK, όπου δεν υπάρχουν πιόνια, η μόνη zeroing move είναι η αιχμαλωσία του μαύρου βασιλιά — δηλαδή το ματ. Η DTZ τιμή αντιστοιχεί επομένως πρακτικά στον αριθμό ημικινήσεων μέχρι το ματ.

Οι DTZ τιμές του dataset κυμαίνονται από 1 έως 31, με μέση τιμή 18.7 και διάμεσο 19. Οι πραγματικές τιμές mate-in-N κυμαίνονται από 1 έως 16 κινήσεις, με μέση τιμή 9.9. Για να μετατραπεί αυτή η συνεχής τιμή σε ταξινομητικό πρόβλημα, η απόλυτη DTZ τιμή ομαδοποιείται σε τέσσερις φάσεις μετατροπής, με όρια που επιλέχθηκαν ώστε να αντικατοπτρίζουν σκακιστικά διακριτές φάσεις της νίκης:

Φάση	Όριο DTZ	Πλήθος θέσεων	Ποσοστό
Immediate (mate or near)	≤ 2	1.512	0.9%
Short conversion	≤ 8	10.428	6.0%
Medium conversion	≤ 16	42.048	24.0%
Long conversion	> 16	121.180	69.1%

Πίνακας 4.2 Ομαδοποίηση των KRK θέσεων σε φάσεις μετατροπής με βάση την απόλυτη DTZ τιμή

Τα όρια (≤ 2 , ≤ 8 , ≤ 16) δεν αποτελούν καθιερωμένο σκακιστικό διαχωρισμό, αλλά σχεδιαστική επιλογή της παρούσας εργασίας. Χρησιμοποιούνται για να μετατρέψουν τις αριθμητικές DTZ τιμές σε τέσσερις διακριτές και πρακτικά ερμηνεύσιμες κατηγορίες μετατροπής. Η επιλογή τους βασίζεται στη διαισθητική ερμηνεία της απόστασης DTZ: όσο μικρότερη είναι η τιμή, τόσο πιο κοντά βρίσκεται η θέση στο τελικό στάδιο της τεχνικής, ενώ όσο μεγαλώνει η τιμή, τόσο περισσότερο η θέση αντιστοιχεί σε προηγούμενη φάση περιορισμού και βελτίωσης.

Με αυτή τη λογική, η κλάση `immediate_mate_or_near` περιλαμβάνει θέσεις όπου η τελική μετατροπή βρίσκεται πολύ κοντά. Η `short_conversion` καλύπτει θέσεις με σύντομη απόσταση μέχρι την αποφασιστική πρόοδο. Η `medium_conversion` περιγράφει ενδιάμεσες θέσεις, όπου ο λευκός έχει σαφές θεωρητικό πλεονέκτημα αλλά απαιτείται ακόμη οργανωμένη τεχνική. Τέλος, η `long_conversion` αντιστοιχεί σε πιο πρώιμες θέσεις, όπου ο περιορισμός του μαύρου βασιλιά και η συνεργασία των λευκών κομματιών βρίσκονται ακόμη σε εξέλιξη.

4.3.2 Σχεδίαση των χαρακτηριστικών

Για το KRK σχεδιάστηκαν έντεκα χαρακτηριστικά, οργανωμένα ξανά σε τρεις ομάδες: γεωμετρικά, σκακιστικά κωδικοποιημένα, και `opposition`. Ο συνολικός αριθμός είναι μικρότερος από τα δεκατέσσερα του KPK, καθώς αρκετές έννοιες του KPK (`rule of the square`, `key squares`, `promotion race`) δεν έχουν αντιστοιχία στο KRK.

Γεωμετρικά χαρακτηριστικά

Η ομάδα των γεωμετρικών περιλαμβάνει επτά χαρακτηριστικά: `wk_bk_chebyshev_distance`, `wk_bk_manhattan_distance`, `wk_distance_to_edge`, `bk_distance_to_edge`, `bk_distance_to_corner`, `wr_bk_rank_distance` και `wr_bk_file_distance`. Ανήκουν στην ίδια λογική κατηγορία με τα γεωμετρικά του KPK — αποστάσεις και σχετικές θέσεις κομματιών — αλλά έχουν εντελώς διαφορετική σκακιστική σημασία.

Οι δύο αποστάσεις μεταξύ βασιλιάδων (chebyshev και manhattan) διατηρούνται σκόπιμα και οι δύο, γιατί καμία δεν είναι πλήρης από μόνη της για το KRK. Η Chebyshev πιάνει την πραγματική απόσταση σε κινήσεις βασιλιά, ενώ η Manhattan αποτυπώνει καλύτερα την «ενέργεια» που χρειάζεται για την πλήρη προσέγγιση σε γραμμή και στήλη — κρίσιμη σε ένα φινάλε όπου η ταυτόχρονη προσέγγιση των δύο διαστάσεων έχει διαφορετική σημασία.

Οι αποστάσεις από την άκρη και τη γωνία (bk_distance_to_edge, bk_distance_to_corner) αποτυπώνουν το πόσο έχει «πιεστεί» ο μαύρος βασιλιάς προς τα όρια της σκακιάς — δηλαδή πόσο κοντά είναι στην τελική οδήγηση σε ματ. Η αντίστοιχη απόσταση για τον λευκό βασιλιά (wk_distance_to_edge) αντικατοπτρίζει την τεχνική δυνατότητά του να χρησιμοποιήσει τον πύργο ως γραμμή περιορισμού. Οι δύο αποστάσεις μεταξύ πύργου και μαύρου βασιλιά (rank και file distance ξεχωριστά) πιάνουν αν ο πύργος βρίσκεται σε θέση να δημιουργήσει cutoff line.

Σκακιστικά κωδικοποιημένα χαρακτηριστικά

Η ομάδα των σκακιστικά κωδικοποιημένων features περιλαμβάνει τρία χαρακτηριστικά, καθένα από τα οποία ενσωματώνει διαφορετική πτυχή της θεωρίας του KRK.

Το πρώτο, bk_legal_moves_if_black_to_move, μετρά πόσες νόμιμες κινήσεις θα είχε ο μαύρος βασιλιάς αν έπαιζε ο μαύρος. Είναι ουσιαστικά μια αριθμητική ποσοτικοποίηση του περιορισμού: όσο λιγότερες οι νόμιμες κινήσεις, τόσο πιο περιορισμένος ο μαύρος βασιλιάς και τόσο πιο κοντά στο ματ. Ο υπολογισμός γίνεται με προσωρινή αλλαγή της σειράς κίνησης, χωρίς να τροποποιείται το πραγματικό αντικείμενο board.

Το δεύτερο, bk_in_mating_zone, είναι μια δυαδική ένδειξη που αποτυπώνει αν ο μαύρος βασιλιάς βρίσκεται σε ζώνη ματ. Η ζώνη ματ ορίζεται προγραμματιστικά ως η περιοχή κοντά στην άκρη ή τη γωνία της σκακιάς, όπου η θεωρητική διαδικασία ματ είναι πρακτικά εκτελέσιμη. Ο ορισμός αυτός είναι σκακιστικά νοηματοδοτημένος: αντιστοιχεί στη γενική αρχή ότι το KRK ματ γίνεται στην περιφέρεια, όχι στο κέντρο.

Το τρίτο, rook_cuts_off_black_king, ελέγχει αν ο πύργος δημιουργεί γραμμή περιορισμού που εμποδίζει τον μαύρο βασιλιά να ξεφύγει προς μεγαλύτερο ελεύθερο χώρο. Το rook cutoff είναι μια από τις πιο βασικές τεχνικές του KRK: ο πύργος σχηματίζει «τείχος» σε συγκεκριμένη γραμμή ή στήλη, αναγκάζοντας τον μαύρο βασιλιά να παραμείνει στη μικρότερη πλευρά της σκακιάς. Ο έλεγχος γίνεται λαμβάνοντας υπόψη τη σχετική θέση του λευκού βασιλιά, ώστε το cutoff να θεωρείται ενεργό μόνο όταν πραγματικά μπορεί να επιβληθεί.

Opposition

Η τρίτη ομάδα περιλαμβάνει ένα βασικό χαρακτηριστικό, το kings_in_direct_opposition, το οποίο αποτυπώνει αν οι δύο βασιλιάδες βρίσκονται σε άμεση opposition. Στο KRK, η opposition παραμένει σημαντική σκακιστική έννοια, καθώς επηρεάζει τη συνεργασία λευκού βασιλιά και πύργου στον περιορισμό του μαύρου βασιλιά και στη μετάβαση προς το τελικό mating net. Ωστόσο, η σημασία της δεν εμφανίζεται απαραίτητα ως υψηλή τιμή στα feature importances του δέντρου. Αυτό συμβαίνει επειδή η ίδια πληροφορία συχνά επικαλύπτεται από άλλα χαρακτηριστικά, όπως η απόσταση των βασιλιάδων, η θέση του μαύρου βασιλιά ως προς την άκρη ή τη γωνία της σκακιάς και ο αριθμός των διαθέσιμων κινήσεών του. Επομένως, η χαμηλή σχετική βαρύτητα του feature δεν σημαίνει ότι η opposition

είναι ασήμαντη, αλλά ότι το μοντέλο μπορεί να αντλεί μέρος της ίδιας πληροφορίας μέσω άλλων, συσχετισμένων χαρακτηριστικών.

4.3.3 Εκπαίδευση δέντρων απόφασης

Η εκπαίδευση του μοντέλου για το KRK ακολουθεί την ίδια λογική με το KPK, ώστε τα αποτελέσματα των δύο φινάλε να παραμένουν συγκρίσιμα. Το dataset χωρίζεται σε σύνολο εκπαίδευσης και ελέγχου με αναλογία 80/20 και με stratification ως προς την κλάση στόχο, ώστε οι τέσσερις φάσεις μετατροπής να διατηρούν παρόμοια κατανομή και στα δύο σύνολα.

Το μοντέλο που χρησιμοποιείται είναι και εδώ DecisionTreeClassifier της scikit-learn. Η επιλογή του μέγιστου βάθους και του class weight δεν γίνεται χειροκίνητα, αλλά μέσω της ίδιας διαδικασίας grid search που χρησιμοποιήθηκε και στο KPK. Για κάθε συνδυασμό υπερπαραμέτρων εκτελούνται πολλαπλές εκπαιδεύσεις με διαφορετικά seeds, ώστε να ελεγχθεί η σταθερότητα της απόδοσης και να μην εξαρτάται το αποτέλεσμα από έναν μόνο διαχωρισμό των δεδομένων.

Η βασική μετρική αξιολόγησης είναι η balanced accuracy, επειδή οι τέσσερις κλάσεις δεν έχουν την ίδια συχνότητα. Παράλληλα, χρησιμοποιείται και εδώ adjusted score, το οποίο συνδυάζει την απόδοση του μοντέλου με ποινή πολυπλοκότητας ως προς το πλήθος των φύλλων. Με αυτόν τον τρόπο, το τελικό δέντρο δεν επιλέγεται μόνο επειδή προβλέπει καλά τις φάσεις μετατροπής, αλλά και επειδή παραμένει αρκετά απλό ώστε οι κανόνες του να μπορούν να ερμηνευθούν.

Η βασική διαφορά σε σχέση με το KPK είναι ότι το target δεν είναι δυαδικό Win/Draw, αλλά τετρατάξιο. Οι κλάσεις immediate_mate_or_near, short_conversion, medium_conversion και long_conversion προκύπτουν από ομαδοποίηση των DTZ τιμών. Αυτό κάνει το πρόβλημα δυσκολότερο, επειδή τα σφάλματα δεν εμφανίζονται ως λανθασμένη πρόβλεψη νίκης ή ισοπαλίας, αλλά ως σύγχυση μεταξύ γειτονικών φάσεων μετατροπής.

4.3.4 Σύγκριση επιπέδων αναπαράστασης

Όπως και στο KPK, πραγματοποιήθηκε ξεχωριστή σύγκριση τριών επιπέδων αναπαράστασης της θέσης. Στόχος της σύγκρισης είναι να φανεί αν η απόδοση του μοντέλου προκύπτει κυρίως από τις ωμές συντεταγμένες των κομματιών, από απλές γεωμετρικές σχέσεις ή από τα πλήρη σκακιστικά χαρακτηριστικά που σχεδιάστηκαν ειδικά για το KRK.

Το πρώτο επίπεδο, raw coordinates, περιλαμβάνει μόνο τις θέσεις του λευκού βασιλιά, του λευκού πύργου και του μαύρου βασιλιά στη σκακιέρα. Το δεύτερο επίπεδο, geometric features, χρησιμοποιεί παράγωγα γεωμετρικά μεγέθη, όπως αποστάσεις μεταξύ των κομματιών και αποστάσεις του μαύρου βασιλιά από την άκρη ή τη γωνία. Το τρίτο επίπεδο, full strategic features, προσθέτει χαρακτηριστικά που συνδέονται άμεσα με τη θεωρία του KRK, όπως ο περιορισμός του μαύρου βασιλιά, το rook cutoff, η mating zone και ο αριθμός διαθέσιμων κινήσεων του μαύρου βασιλιά.

Για κάθε feature set χρησιμοποιούνται οι ίδιες παράμετροι εκπαίδευσης και η ίδια πειραματική λογική, ώστε η μόνη ουσιαστική διαφορά να είναι η πληροφορία που δίνεται στο μοντέλο. Με αυτόν τον τρόπο, η σύγκριση δεν απαντά μόνο στο ποιο feature set δίνει μεγαλύτερη ακρίβεια, αλλά και στο ποιο οδηγεί σε πιο συμπαγείς και πιο ερμηνεύσιμους κανόνες.

Στο KRK, η σύγκριση αυτή έχει ιδιαίτερο ενδιαφέρον, επειδή η σχετική γεωμετρία των κομματιών παίζει πιο άμεσο ρόλο από ό,τι στο KPK. Η πρόοδος προς το ματ εξαρτάται από το πόσο περιορισμένος είναι ο μαύρος βασιλιάς, πόσο κοντά βρίσκεται στην άκρη ή στη γωνία της σκακιάς και πόσο αποτελεσματικά συνεργάζονται ο λευκός βασιλιάς και ο πύργος. Για τον λόγο αυτό, τα γεωμετρικά και στρατηγικά χαρακτηριστικά αναμένεται να είναι πιο εκφραστικά από τις απλές συντεταγμένες.

4.3.5 Ομαδοποίηση κανόνων σε motifs

Το βήμα του rule consolidation ακολουθεί την ίδια βασική μεθοδολογία με το KPK: κάθε leaf rule κανονικοποιείται σε signature μέσω binning και οι signatures αντιστοιχίζονται σε ονοματισμένα motifs μέσω ενός semi-handcrafted interpretive layer. Οι κατηγορίες όμως που χρησιμοποιούνται στο KRK είναι διαφορετικές, επειδή αντικατοπτρίζουν τη διαφορετική σκακιστική φύση του συγκεκριμένου φινάλε.

Στο KPK, τα motifs περιστρέφονται γύρω από έννοιες όπως το τετράγωνο του πιονιού, τα key squares, το blockade και η opposition. Στο KRK, αντίθετα, η τεχνική νίκης βασίζεται στη συνεργασία του λευκού βασιλιά και του πύργου για τον περιορισμό του μαύρου βασιλιά. Για τον λόγο αυτό, τα motifs οργανώνονται γύρω από έννοιες όπως η προσέγγιση του λευκού βασιλιά, η μείωση της κινητικότητας του μαύρου, η οδήγησή του προς την άκρη ή τη γωνία της σκακιάς, το rook cutoff και η τελική δημιουργία mating net.

Ένα βασικό χαρακτηριστικό του KRK είναι ότι η ίδια φάση μετατροπής μπορεί να εμφανιστεί μέσα από αρκετές διαφορετικές γεωμετρικές διατάξεις. Για παράδειγμα, μια θέση μπορεί να ανήκει στη long φάση επειδή ο μαύρος βασιλιάς βρίσκεται ακόμα μακριά από την άκρη, ή επειδή ο λευκός βασιλιάς δεν έχει πλησιάσει αρκετά ώστε να υποστηρίξει τον πύργο. Αντίστοιχα, μια medium θέση μπορεί να δείχνει ήδη πρόοδο προς την άκρη, αλλά να μην έχει ακόμη διαμορφωθεί πλήρως το mating net. Επομένως, τα motifs στο KRK δεν περιγράφουν απλώς απόσταση μέχρι το ματ, αλλά στάδια της τεχνικής διαδικασίας.

Στις πιο πρώιμες φάσεις, τα motifs δίνουν έμφαση στην προσέγγιση του λευκού βασιλιά και στον αρχικό περιορισμό του μαύρου. Η έννοια αυτή είναι σημαντική, γιατί ο πύργος μόνος του μπορεί να περιορίσει τον αντίπαλο βασιλιά, αλλά δεν μπορεί να ολοκληρώσει την τεχνική χωρίς τη σωστή υποστήριξη του λευκού βασιλιά. Για τον λόγο αυτό, εμφανίζονται motifs που ξεχωρίζουν θέσεις όπου ο μαύρος βασιλιάς έχει αρχίσει να οδηγείται προς την άκρη, αλλά ο λευκός βασιλιάς παραμένει ακόμα μακριά.

Στις ενδιάμεσες φάσεις, τα motifs εστιάζουν περισσότερο στην πρόοδο του περιορισμού. Ο μαύρος βασιλιάς μπορεί να βρίσκεται ήδη σε περιοχή που ευνοεί τη μετατροπή, αλλά να διατηρεί ακόμη αρκετή κινητικότητα. Σε αυτές τις περιπτώσεις, ο αριθμός των νόμιμων κινήσεων του μαύρου βασιλιά και η απόστασή του από την άκρη ή τη γωνία αποκτούν ιδιαίτερη σημασία. Το δέντρο, μέσω των αντίστοιχων

features, μπορεί να διακρίνει αν η θέση βρίσκεται ακόμη σε στάδιο προετοιμασίας ή αν πλησιάζει στο τελικό mating net.

Στις σύντομες και άμεσες φάσεις, τα motifs συνδέονται περισσότερο με την τελική συνεργασία βασιλιά και πύργου. Εδώ ο μαύρος βασιλιάς έχει συνήθως οδηγηθεί προς την άκρη ή τη γωνία, ενώ ο λευκός βασιλιάς βρίσκεται σε πιο ενεργή υποστηρικτική θέση. Το rook cutoff, η μειωμένη κινητικότητα του μαύρου βασιλιά και η παρουσία του σε mating zone αποτελούν βασικές ενδείξεις ότι η τεχνική βρίσκεται κοντά στην ολοκλήρωσή της.

Όπως και στα άλλα δύο endgames, υπάρχει και στο KRK ένα ποσοστό signatures που δεν αντιστοιχίζονται σε κάποιο ειδικό σκακιστικό motif και εντάσσονται σε data-driven κατηγορίες. Αυτό είναι αναμενόμενο, επειδή το KRK είναι πιο ποικιλόμορφο από το KPK: δεν υπάρχει ένα μόνο στρατηγικό ερώτημα τύπου Win/Draw, αλλά τέσσερις διαδοχικές φάσεις μετατροπής με αρκετές πιθανές γεωμετρικές μορφές. Παρ' όλα αυτά, η διαδικασία του rule consolidation παραμένει χρήσιμη, γιατί μετατρέπει τα τεχνικά leaf rules του δέντρου σε ερμηνεύσιμες ομάδες που αντιστοιχούν σε πραγματικά στάδια της τεχνικής του KRK.

Τα αριθμητικά αποτελέσματα του rule consolidation, δηλαδή το πλήθος των leaf rules, των signatures και των κυρίαρχων motifs ανά φάση μετατροπής, παρουσιάζονται στην ενότητα 4.3.7 μαζί με τα υπόλοιπα αποτελέσματα του μοντέλου.

4.3.6 Οπτικοποίηση και τεκμηρίωση

Όπως και στο KPK, το pipeline του KRK ολοκληρώνεται με δύο βοηθητικά αρχεία που δεν αλλάζουν τη μεθοδολογία του πειράματος, αλλά υποστηρίζουν την ανάλυση και την αναπαραγωγικότητα των αποτελεσμάτων. Το KRKVisual.py παράγει τα βασικά γραφήματα της εκτέλεσης, όπως το confusion matrix, τα feature importances, τα συγκριτικά διαγράμματα των τριών επιπέδων αναπαράστασης και τα διαγράμματα σταθερότητας ως προς το βάθος του δέντρου.

Αντίστοιχα, το KRKReadMe.py δημιουργεί ένα συνοπτικό αρχείο τεκμηρίωσης, στο οποίο συγκεντρώνονται οι βασικές επιλογές της εκτέλεσης, τα παραγόμενα αρχεία, οι επιλεγμένες υπερπαραμέτροι, οι κύριες μετρικές και η ερμηνεία των σημαντικότερων motifs. Η ύπαρξη αυτών των αρχείων βοηθά ώστε κάθε εκτέλεση να μπορεί να ελεγχθεί εκ των υστέρων χωρίς να χρειάζεται να ανατρέξει κανείς απευθείας στον κώδικα ή στα επιμέρους CSV.

4.3.7 Αποτελέσματα για το φινάλε KRK

Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα της εφαρμογής του pipeline στο φινάλε KRK, οργανωμένα σε τέσσερις άξονες: γενική απόδοση του τελικού μοντέλου, ανάλυση της συμβολής κάθε feature, σύγκριση των τριών επιπέδων αναπαράστασης και ανάλυση των παραγόμενων κανόνων.

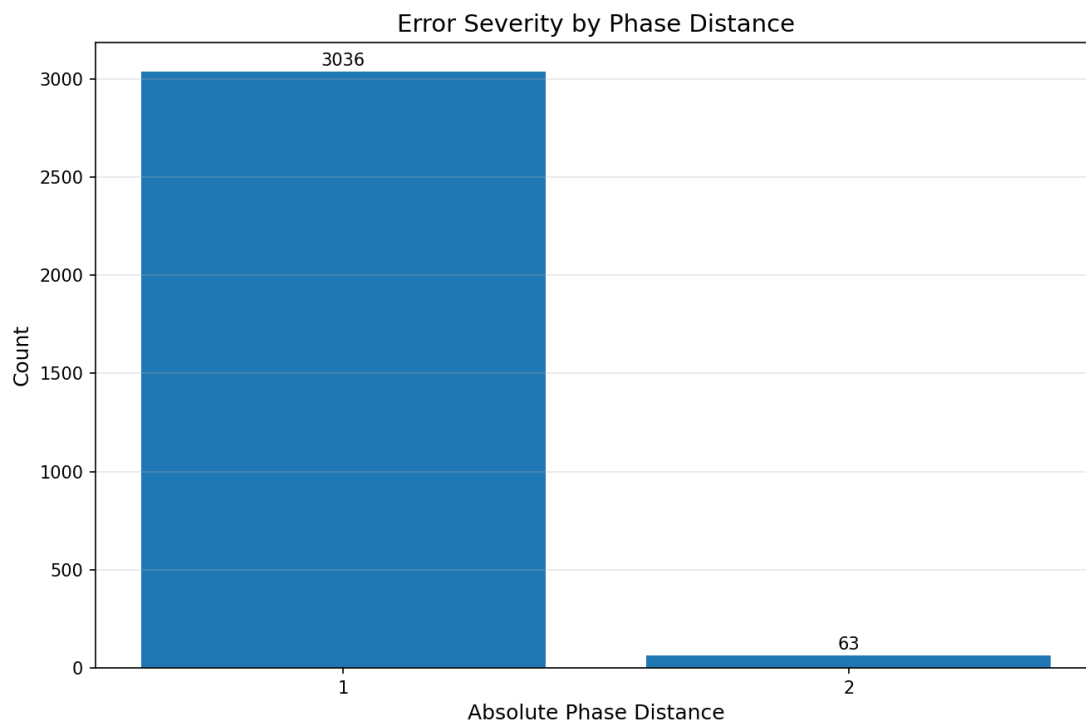
Απόδοση του τελικού μοντέλου

Ο βέλτιστος συνδυασμός υπερπαραμέτρων που επιλέχθηκε ήταν βάθος 12 με class weight balanced. Το αντίστοιχο μοντέλο επιτυγχάνει balanced accuracy 0.9477 στο test set και αντίστοιχη τιμή 0.9485 ± 0.0012 στην 5-fold cross-validation, με 686 φύλλα. Το test accuracy είναι 0.9130 και το macro F1 0.9080. Η πολύ κοντινή τιμή μεταξύ test και cross-validation μετρικών υποδεικνύει ότι η επίδοση του μοντέλου είναι σταθερή ανεξάρτητα από τον τρόπο χωρισμού των δεδομένων, οπότε τα αναφερόμενα ποσοστά αντικατοπτρίζουν πραγματικά τη γενικευτική του ικανότητα.

Από τα συνολικά 3.099 λάθη του μοντέλου στο test set, μια σημαντική παρατήρηση αφορά τη σοβαρότητα αυτών των λαθών. Ταξινομώντας κάθε λάθος με βάση την απόλυτη διαφορά φάσης μεταξύ πραγματικής και προβλεπόμενης κλάσης:

- Λάθη με phase-distance = 1: 3.036 (98.0%)
- Λάθη με phase-distance = 2: 63 (2.0%)
- Λάθη με phase-distance = 3: 0

Δηλαδή, σχεδόν όλα τα λάθη του μοντέλου είναι συγκεντρωμένα σε γειτονικές φάσεις. Το μοντέλο δεν προβλέπει long όταν είναι immediate, ούτε short όταν είναι long. Η μεγάλη πλειονότητα των λαθών βρίσκεται στα σύνορα μεταξύ διαδοχικών φάσεων, όπου η οριοθέτηση είναι εγγενώς ασαφής.

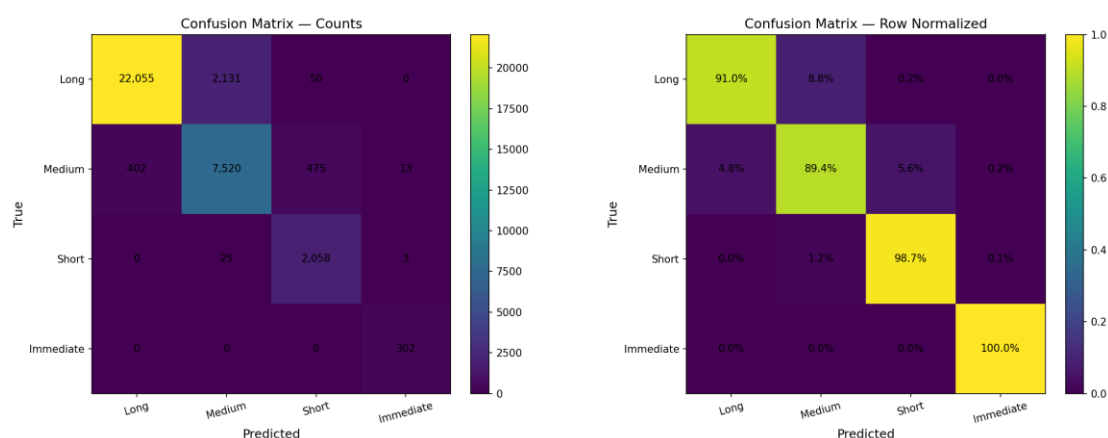


Εικόνα 4.11 Error Severity by Phase Distance

Η ανάλυση με βάση την κατηγορία λάθους αποκαλύπτει ότι η κύρια πηγή λαθών είναι το σύνορο long-medium με 2.533 λάθη — 82% του συνόλου. Ακολουθεί το σύνορο medium-short με 500 λάθη, ενώ όλα τα υπόλοιπα σύνορα συνεισφέρουν λιγότερο από 100 λάθη το καθένα. Το γεγονός αυτό είναι σκακιστικά αναμενόμενο: το όριο

DTZ=16 που διαχωρίζει long από medium είναι το πιο δύσκολο να προβλεφθεί με ακρίβεια, καθώς πολλές θέσεις στο εύρος 14-18 έχουν πολύ παρόμοια σκακιστικά χαρακτηριστικά και η ταξινόμηση εξαρτάται από λεπτομέρειες που το μοντέλο δεν μπορεί πάντα να πιάσει.

Αυτή η δομή λαθών είναι, στην πράξη, το πιο ενθαρρυντικό εύρημα του KRK μοντέλου. Ένα μοντέλο που κάνει «κοντινά» λάθη — προβλέποντας long αντί για medium, ή medium αντί για short — είναι πρακτικά χρήσιμο: ένας σκακιστής που το συμβουλευέται θα λάβει σωστή γενική εικόνα της φάσης, με πιθανή απόκλιση μιας κατηγορίας. Ένα μοντέλο που έκανε πηδήματα δύο ή τριών κατηγοριών θα ήταν ουσιαστικά αναξιόπιστο. Το KRK μοντέλο, με 98% των λαθών σε phase-distance = 1, βρίσκεται καθαρά στην πρώτη κατηγορία.



Εικόνα 4.12 KRK Confusion Matrix

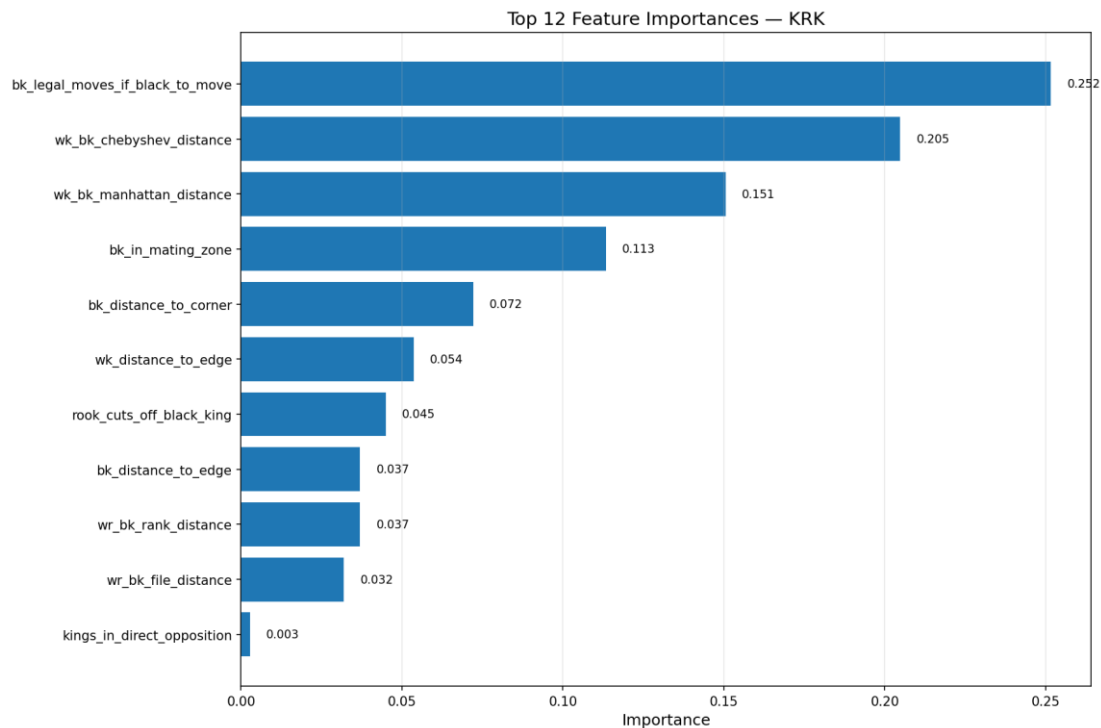
Feature importances

Η ανάλυση της συμβολής κάθε feature στις αποφάσεις του δέντρου αποκαλύπτει διαφορετική ιεραρχία από το KPK. Το πιο σημαντικό feature είναι το `bk_legal_moves_if_black_to_move` με importance 0.252, ακολουθούμενο από τις δύο αποστάσεις μεταξύ βασιλιάδων: `wk_bk_chebyshev_distance` με 0.205 και `wk_bk_manhattan_distance` με 0.151. Στη συνέχεια έρχονται το `bk_in_mating_zone` (0.113) και το `bk_distance_to_corner` (0.072). Σε μεγαλύτερη απόσταση ακολουθούν το `wk_distance_to_edge` (0.054), το `rook_cuts_off_black_king` (0.045), οι δύο αποστάσεις πύργου-βασιλιά (0.037 και 0.032 αντίστοιχα), ενώ το `kings_in_direct_opposition` έχει importance σχεδόν μηδέν (0.003).

Σκακιστικά τα αποτελέσματα βγάζουν πλήρες νόημα. Το κυρίαρχο feature είναι ο βαθμός περιορισμού του μαύρου βασιλιά — πόσες νόμιμες κινήσεις έχει διαθέσιμες — που αποτυπώνει άμεσα το πόσο κοντά είναι η τελική οδήγηση σε ματ. Οι δύο επόμενες θέσεις καταλαμβάνονται από τις δύο αποστάσεις μεταξύ των βασιλιάδων, υπογραμμίζοντας την κεντρικότητα της προσέγγισης του λευκού βασιλιά: τεχνικά, ο πύργος δεν αρκεί από μόνος του, χρειάζεται η ταυτόχρονη συνεργασία του βασιλιά. Το `bk_in_mating_zone` και το `bk_distance_to_corner` επιβεβαιώνουν ότι η οδήγηση προς την άκρη ή τη γωνία είναι η δομική αρχή του KRK ματ.

Αξιοσημείωτο είναι ότι το `rook_cuts_off_black_king`, παρότι σκακιστικά κεντρικό στην τεχνική του KRK, καταλαμβάνει μόλις την έβδομη θέση με importance 0.045. Αυτό

δεν σημαίνει ότι το rook cutoff είναι σκακιστικά ασήμαντο — σημαίνει ότι η πληροφορία που μεταφέρει είναι σε μεγάλο βαθμό ήδη ενσωματωμένη στα άλλα features, κυρίως στο `bk_legal_moves_if_black_to_move`, καθώς ένας αποτελεσματικός cutoff μειώνει άμεσα τις νόμιμες κινήσεις του μαύρου βασιλιά. Το γεγονός αυτό αντικατοπτρίζει μια χαρακτηριστική αλληλεπικάλυψη που εμφανίζεται συχνά σε feature engineering: πολλά features που είναι εννοιολογικά διακριτά καταλήγουν να μοιράζονται σημαντική στατιστική πληροφορία στα δεδομένα.



Εικόνα 4.13 KRK Feature importances

Σύγκριση των τριών επιπέδων αναπαράστασης

Η σύγκριση των τριών επιπέδων features δείχνει ενδιαφέροντα ευρήματα, με ορισμένες σημαντικές διαφορές από το KPK:

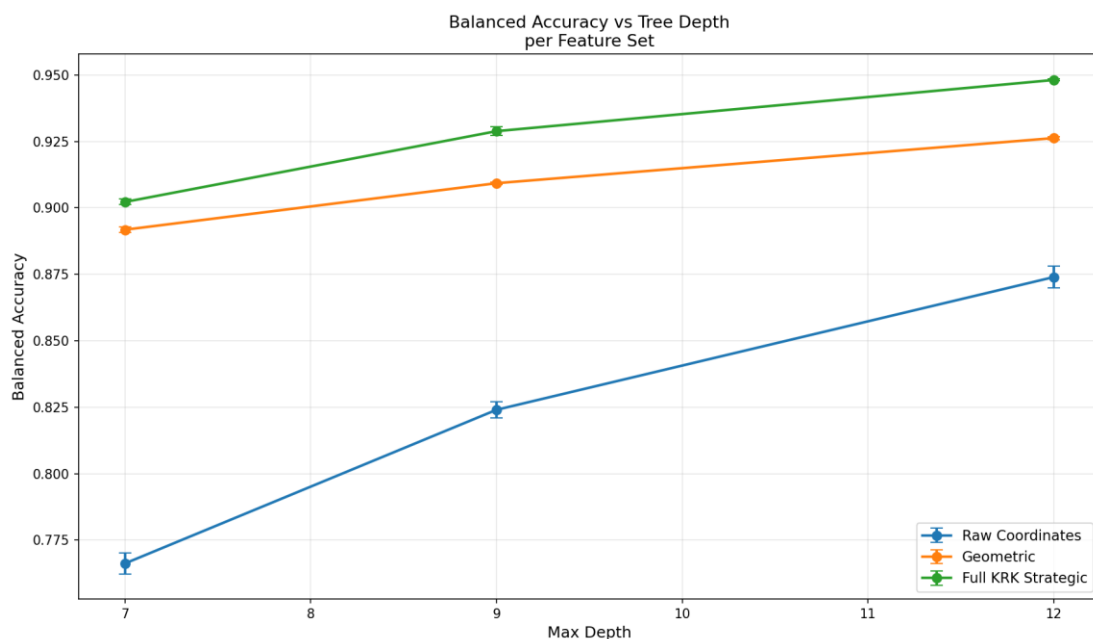
Επίπεδο	Balanced Accuracy	Πλήθος φύλλων	Λάθη στο test
Raw coordinates	0.8740	1.055	4.597
Geometric Features	0.9262	748	4.133
Full Strategic	0.9481	686	3.096

Πίνακας 4.3 Σύγκριση των τριών επιπέδων αναπαράστασης στο KRK

Δύο βασικές διαφορές σε σχέση με το αντίστοιχο συμπέρασμα του KPK. Πρώτον, εδώ το geometric επίπεδο τα πάει σαφώς καλύτερα από το raw, όχι χειρότερα. Αυτό είναι σκακιστικά αναμενόμενο: σε αντίθεση με το KPK, όπου η απόλυτη θέση των κομματιών (κέντρο vs άκρη) είναι κρίσιμη λόγω του πιονιού με προσανατολισμό, στο KRK η δυναμική είναι πλήρως συμμετρική και οι σχετικές αποστάσεις αποτυπώνουν τη γεωμετρία της θέσης με ελάχιστη απώλεια πληροφορίας. Η βελτίωση από raw σε geometric είναι της τάξης των 5 ποσοστιαίων μονάδων balanced accuracy, μια σημαντική διαφορά.

Δεύτερον, η βελτίωση από geometric σε full strategic είναι μεν πιο συγκρατημένη σε accuracy ($0.9262 \rightarrow 0.9481$) αλλά μειώνει τα λάθη κατά 25% (από 4.133 σε 3.096) και το πλήθος φύλλων κατά 8% (από 748 σε 686). Η σκακιστική γνώση εδώ δεν είναι απαραίτητη για βασική ταξινόμηση, αλλά προσθέτει σημαντική βελτίωση στα οριακά σημεία.

Το συγκεντρωτικό συμπέρασμα από τη σύγκριση παραμένει το ίδιο με το KPK: η σκακιστικά κωδικοποιημένη γνώση δεν προσφέρει δραματική αύξηση ακρίβειας, αλλά προσφέρει σημαντική μείωση της πολυπλοκότητας και βελτίωση της ερμηνευσιμότητας. Η ίδια ακρίβεια επιτυγχάνεται με λιγότερα και πιο ερμηνεύσιμα features, με κανόνες που αντιστοιχούν σε γνωστές σκακιστικές έννοιες.



Εικόνα 4.14 Συγκριτικό γράφημα balanced accuracy

Η ανάλυση των παραγόμενων κανόνων, που παρουσιάστηκε στην ενότητα 4.3.5, δείχνει ότι το KRK είναι λιγότερο συμπαγές ερμηνευτικά από το KPK. Αυτό είναι αναμενόμενο, καθώς εδώ το πρόβλημα δεν είναι δυαδικό Win/Draw αλλά ταξινόμηση τεσσάρων φάσεων μετατροπής. Παρ' όλα αυτά, τα κυρίαρχα motifs συνδέονται με βασικές έννοιες της τεχνικής του KRK, όπως η προσέγγιση του λευκού βασιλιά, ο περιορισμός του μαύρου βασιλιά, η οδήγηση προς την άκρη και η τελική δημιουργία mating net.

Συνολικά, το KRK επιβεβαιώνει ότι η μεθοδολογία της εργασίας μπορεί να εφαρμοστεί και σε πιο σύνθετα φινάλε από το KPK. Παρότι οι κανόνες είναι λιγότερο συμπυκνωμένοι και οι φάσεις μετατροπής παρουσιάζουν μεγαλύτερη γεωμετρική ποικιλία, το μοντέλο πετυχαίνει υψηλή απόδοση και ταυτόχρονα παράγει ερμηνεύσιμα σκακιστικά μοτίβα. Επομένως, το KRK λειτουργεί ως ενδιάμεσο και πιο απαιτητικό παράδειγμα ανάμεσα στο απλούστερο KPK και στο επόμενο φινάλε KQK.

Ανάλυση των παραγόμενων κανόνων

Από τις 686 leaf rules του τελικού δέντρου, το rule consolidation παράγει 645 διακριτές signatures. Η μείωση αυτή είναι αρκετά περιορισμένη σε σχέση με το KPK, όπου οι κανόνες συμπύκνωσης πολύ πιο έντονα. Αυτό είναι αναμενόμενο, επειδή το KRK δεν είναι δυαδικό πρόβλημα Win/Draw, αλλά ταξινόμηση τεσσάρων φάσεων μετατροπής. Επιπλέον, η ίδια φάση μπορεί να εμφανιστεί μέσα από πολλές διαφορετικές γεωμετρικές διατάξεις του λευκού βασιλιά, του πύργου και του μαύρου βασιλιά.

Οι signatures αντιστοιχίζονται τελικά σε 15 ονοματισμένα motifs, οργανωμένα ανά φάση μετατροπής: 5 για τη long, 5 για τη medium, 4 για τη short και 2 για την immediate_mate_or_near. Η κατανομή αυτή δείχνει ότι οι πρώιμες και ενδιάμεσες φάσεις του KRK χρειάζονται περισσότερες ερμηνευτικές κατηγορίες, επειδή η διαδικασία περιορισμού του μαύρου βασιλιά μπορεί να εμφανιστεί με αρκετές διαφορετικές μορφές.

Για τη long φάση, τα κυρίαρχα motifs περιγράφουν κυρίως θέσεις όπου η τεχνική βρίσκεται ακόμη σε πρώιμο στάδιο. Το long_king_approach_signature αντιστοιχεί σε θέσεις όπου ο λευκός βασιλιάς χρειάζεται ακόμη να πλησιάσει, ενώ το long_zone_but_white_king_far_signature περιγράφει περιπτώσεις όπου ο μαύρος βασιλιάς έχει αρχίσει να περιορίζεται, αλλά ο λευκός βασιλιάς δεν έχει φτάσει ακόμη σε ενεργή υποστηρικτική θέση. Με άλλα λόγια, η long φάση δεν σημαίνει απλώς ότι το ματ απέχει αριθμητικά περισσότερο, αλλά ότι η συνεργασία βασιλιά και πύργου δεν έχει ακόμη ολοκληρωθεί.

Για τη medium φάση, το βασικό motif είναι το medium_zone_white_king_far_signature. Η φάση αυτή αντιστοιχεί σε ενδιάμεσες θέσεις, όπου ο μαύρος βασιλιάς βρίσκεται ήδη σε ζώνη που ευνοεί τη μετατροπή, αλλά ο λευκός βασιλιάς παραμένει σχετικά μακριά. Ένα δεύτερο σημαντικό motif είναι το medium_edge_progress_signature, όπου ο μαύρος βασιλιάς έχει αρχίσει να οδηγείται προς την άκρη, χωρίς όμως να έχει δημιουργηθεί πλήρως το τελικό mating net.

Για τη short φάση, το κυρίαρχο motif είναι το short_edge_king_support_signature. Εδώ ο μαύρος βασιλιάς έχει ήδη οδηγηθεί προς την άκρη ή τη γωνία και ο λευκός βασιλιάς βρίσκεται σε υποστηρικτική απόσταση. Πρόκειται για θέσεις όπου ο βασικός περιορισμός έχει επιτευχθεί και απομένει η ακριβής ολοκλήρωση της τεχνικής.

Για την immediate_mate_or_near φάση, το βασικό ερμηνεύσιμο motif είναι το immediate_rook_cutoff_king_support_signature, το οποίο συνδυάζει rook cutoff, υποστήριξη από τον λευκό βασιλιά και άμεση απειλή ματ. Ωστόσο, μεγάλο μέρος αυτής της φάσης καλύπτεται από data-driven signatures χωρίς ειδικότερο σκακιστικό όνομα. Αυτό δείχνει ότι οι θέσεις πολύ κοντά στο ματ είναι αριθμητικά αναγνωρίσιμες από το δέντρο, αλλά δεν συμπυκνώνονται όλες σε ένα μοναδικό απλό motif με βάση το συγκεκριμένο interpretation layer.

Συνολικά, η ανάλυση των κανόνων δείχνει ότι το KRK είναι λιγότερο συμπαγές ερμηνευτικά από το KPK, αλλά αυτό δεν αποτελεί αδυναμία του μοντέλου. Αντίθετα, αντανakλά τη μεγαλύτερη γεωμετρική ποικιλία του προβλήματος και τη δυσκολία διάκρισης τεσσάρων γειτονικών φάσεων μετατροπής. Παρ' όλα αυτά, τα βασικά motifs συνδέονται καθαρά με γνωστές έννοιες της τεχνικής του KRK: προσέγγιση του λευκού βασιλιά, περιορισμός του μαύρου βασιλιά, οδήγηση προς την άκρη και τελική δημιουργία mating net.

4.4 Φινάλε KQK (King and Queen vs King)

Το φινάλε KQK ολοκληρώνει τη σειρά των τριών endgames που μελετώνται. Η αρχιτεκτονική της εφαρμογής παραμένει η ίδια — έξι αρχεία Python με την ίδια ροή επεξεργασίας — και οι μεθοδολογικές επιλογές (grid υπερπαραμέτρων, adjusted score, class weight, cross-validation) είναι κοινές με τα δύο προηγούμενα φινάλε. Οι διαφορές που εντοπίζονται εδώ αφορούν κυρίως τη σκακιστική φύση του KQK, η οποία, παρά την επιφανειακή του απλότητα, ενσωματώνει ορισμένες πραγματικά ιδιαίτερες πτυχές — με σημαντικότερη το ενδεχόμενο του πατ. Στην παρουσίαση που ακολουθεί, η έμφαση δίνεται σε αυτές τις πτυχές, ενώ όσα είναι κοινά με τα προηγούμενα φινάλε αναφέρονται συνοπτικά.

4.4.1 Παραγωγή του dataset

Το αρχείο KQKGenerator.py υλοποιεί την παραγωγή του dataset ακολουθώντας την ίδια λογική με το KRK: εξαντλητική απαρίθμηση, φιλτράρισμα μόνο σε winning θέσεις, και αξιολόγηση κάθε θέσης μέσω probe_dtz. Η μετάβαση από το KRK στο KQK δεν αλλάζει τη δομή της διαδικασίας, αλλά επηρεάζει δύο κρίσιμα σημεία: την κλίμακα των τιμών DTZ και τα όρια των φάσεων μετατροπής.

Ταχύτερες νίκες, πιο σφιχτά όρια φάσεων

Η ουσιαστικότερη διαφορά του KQK από το KRK είναι ότι η διαδικασία μετατροπής είναι αισθητά ταχύτερη. Στο KRK ο πύργος χρειάζεται συχνά περισσότερα βήματα για να περιορίσει σταδιακά τον μαύρο βασιλιά και να συνεργαστεί με τον λευκό βασιλιά για τη δημιουργία mating net. Αντίθετα, στο KQK η βασίλισσα διαθέτει πολύ μεγαλύτερη κινητικότητα και μπορεί να ελέγχει ταυτόχρονα γραμμές, στήλες και διαγώνιες. Για τον λόγο αυτό, οι περισσότερες κερδισμένες θέσεις βρίσκονται πιο κοντά στην τελική μετατροπή.

Στον κώδικα της εργασίας, οι τιμές που επιστρέφει η Syzygy μέσω DTZ δεν χρησιμοποιούνται απευθείας ως όρια φάσεων. Πρώτα μετατρέπονται σε μια πιο φυσική κλίμακα κινήσεων μέσω της ποσότητας `mate_moves = |DTZ| / 2`. Η μετατροπή αυτή γίνεται επειδή η DTZ τιμή μετρίεται σε ημικινήσεις, ενώ για την ερμηνεία των φάσεων είναι πιο κατανοητό να χρησιμοποιείται μια προσέγγιση σε πλήρεις κινήσεις. Έτσι, οι φάσεις `immediate`, `short`, `medium` και `long` δεν ορίζονται πάνω στην ωμή DTZ τιμή, αλλά πάνω στην παράγωγη ποσότητα `mate_moves`.

Με βάση αυτή τη λογική, επιλέχθηκαν πιο σφιχτά όρια σε σχέση με το KRK, ώστε να αντικατοπτρίζεται η ταχύτερη φύση του KQK. Αν χρησιμοποιούνταν τα ίδια όρια με το KRK, η ομαδοποίηση θα ήταν πολύ άνιση και οι περισσότερες θέσεις θα συγκεντρώνονταν στις αρχικές κατηγορίες. Τα νέα όρια οδηγούν σε πιο κατάλληλη κατανομή των φάσεων για το συγκεκριμένο φινάλε:

Φάση	Όριο mate_moves	Πλήθος θέσεων	Ποσοστό
Immediate mate	≤ 1	2.448	1.7%
Short conversion	≤ 3	14.076	9.7%
Medium conversion	≤ 6	79.192	54.1%
Long conversion	> 6	49.792	34.5%

Πίνακας 4.4 Ομαδοποίηση των KQK θέσεων σε φάσεις μετατροπής με βάση την ποσότητα $\text{mate_moves} = |\text{DTZ}| / 2$

4.4.2 Σχεδίαση των χαρακτηριστικών

Για το KQK σχεδιάστηκαν έντεκα χαρακτηριστικά, οργανωμένα ξανά σε τρεις ομάδες: γεωμετρικά, σκακιστικά κωδικοποιημένα, και μία ομάδα ειδική για τη βασίλισσα. Ο συνολικός αριθμός ταυτίζεται με του KRK, αλλά η σύνθεση διαφέρει σημαντικά: αρκετά features του KRK που αφορούν τον πύργο αντικαθίστανται από αντίστοιχα για τη βασίλισσα, και προστίθενται δύο features που δεν είχαν αντίστοιχία στα προηγούμενα φινάλε — το `queen_box_area` και το `stalemate_risk_level`.

Γεωμετρικά χαρακτηριστικά

Η ομάδα των γεωμετρικών περιλαμβάνει έξι χαρακτηριστικά: `wk_bk_chebyshev_distance`, `wk_bk_manhattan_distance`, `wk_distance_to_edge`, `bk_distance_to_edge`, `bk_distance_to_corner` και `wk_wq_distance`. Τα πέντε πρώτα αντιστοιχούν εννοιολογικά σε αυτά του KRK και υπολογίζονται με τον ίδιο τρόπο. Η κύρια διαφορά είναι η προσθήκη του `wk_wq_distance` — της απόστασης μεταξύ του λευκού βασιλιά και της βασίλισσας του — που αντικαθιστά τις δύο αποστάσεις πύργου-βασιλιά του KRK. Η μονή αυτή απόσταση αρκεί για τη βασίλισσα καθώς, σε αντίθεση με τον πύργο, η βασίλισσα δεν εξαρτάται από ευθυγράμμιση σε γραμμή ή στήλη για να εφαρμόσει περιορισμό. Το κρίσιμο ερώτημα είναι απλώς αν βρίσκεται σε συνεργασία με τον βασιλιά της, ανεξάρτητα από γεωμετρικές λεπτομέρειες.

Σκακιστικά κωδικοποιημένα χαρακτηριστικά

Η ομάδα των σκακιστικά κωδικοποιημένων features περιλαμβάνει τρία χαρακτηριστικά, με ενδιαφέρουσες διαφοροποιήσεις από το KRK.

Το πρώτο, `bk_legal_moves_if_black_to_move`, είναι πανομοιότυπο με του KRK: αριθμητική ποσοτικοποίηση του βαθμού περιορισμού του μαύρου βασιλιά. Ο υπολογισμός γίνεται με προσωρινή αλλαγή της σειράς κίνησης.

Το δεύτερο, `queen_controls_bk_zone_count`, είναι ένα KQK-specific feature που μετρά πόσα από τα οκτώ γειτονικά τετράγωνα του μαύρου βασιλιά ελέγχονται από τη

βασίλισσα. Είναι σκακιστικά η φυσική εξέλιξη του `rook_cuts_off_black_king` του KRK, προσαρμοσμένη στη δυνατότητα της βασίλισσας να ελέγχει ταυτόχρονα διαγώνιες, γραμμές και στήλες. Ένα υψηλό `value` σημαίνει ότι η βασίλισσα έχει σχεδόν ολοκληρώσει τη «στένωση του κουτιού» — τη βασική τεχνική του KQK.

Το τρίτο, `stalemate_risk_level`, είναι η πιο ουσιώδης καινοτομία του KQK feature set και αντιστοιχεί στη σκακιστικά σημαντικότερη ιδιαιτερότητα του φινάλε: τον κίνδυνο πατ. Στο KQK, σε αντίθεση με τα άλλα δύο φινάλε, το πατ είναι πραγματικός κίνδυνος. Μια βασίλισσα που πλησιάζει υπερβολικά τον μαύρο βασιλιά χωρίς σωστή υποστήριξη μπορεί κατά λάθος να αφαιρέσει όλες τις νόμιμες κινήσεις του, μετατρέποντας μια κερδισμένη θέση σε ισοπαλία. Το feature αυτό δεν είναι δυαδικό αλλά κατηγορικό με τρεις τιμές: 0 (κανένας κίνδυνος), 1 (μέτριος κίνδυνος), 2 (υψηλός κίνδυνος). Η κατανομή στο dataset δείχνει ότι το 85.7% των θέσεων δεν έχει καθόλου stalemate risk, το 13.7% έχει μέτριο, και μόλις το 0.6% έχει υψηλό — ποσοστά που είναι ρεαλιστικά για ένα φινάλε όπου το πατ είναι κίνδυνος αλλά όχι κυρίαρχο ζήτημα.

Ομάδα βασίλισσας

Η τρίτη ομάδα περιλαμβάνει δύο features που είναι μοναδικά για το KQK και δεν έχουν αντίστοιχο στα προηγούμενα φινάλε.

Το `queen_box_area` μετρά την επιφάνεια του «κουτιού» της βασίλισσας — δηλαδή του ορθογωνίου παραλληλογράμμου που ορίζεται από τη βασίλισσα και τον μαύρο βασιλιά, και εντός του οποίου ο μαύρος βασιλιάς είναι περιορισμένος να κινείται. Είναι η αριθμητική έκφραση της βασικότερης έννοιας του KQK: η σταδιακή στένωση του κουτιού μέχρι ο μαύρος βασιλιάς να εξωθηθεί σε άκρη ή γωνία. Ένα μικρό `queen_box` σημαίνει ότι η θέση είναι κοντά στο τέλος, ένα μεγάλο ότι απομένει σημαντική τεχνική πρόοδος.

Το `wq_bk_chebyshev_distance` καταγράφει την απόσταση μεταξύ της βασίλισσας και του μαύρου βασιλιά. Σε αντίθεση με το KRK, όπου ο πύργος πρέπει να διατηρεί απόσταση για ασφάλεια, στο KQK η βασίλισσα μπορεί να πλησιάσει πολύ κοντά — αρκεί να υπάρχει υποστήριξη από τον λευκό βασιλιά. Η απόσταση αυτή είναι επομένως δείκτης της φάσης της τεχνικής: αρχική στένωση από μακριά, ολοκλήρωση από κοντά.

4.4.3 Εκπαίδευση δέντρων απόφασης

Η εκπαίδευση του δέντρου για το KQK ακολουθεί την ίδια διαδικασία που περιγράφηκε στις 4.2.3 και 4.3.3: 80/20 stratified split, grid υπερπαραμέτρων με τρία βάθη (7, 9, 12) και δύο επιλογές `class_weight`, τρία `seeds`, `adjusted score` με `penalty weight` 0.01. Οι μετρικές (`balanced accuracy`, `macro F1`) υπολογίζονται εκ νέου πάνω σε multi-class classification με τέσσερις φάσεις.

Η επιλογή του `class_weight = balanced` παραμένει σημαντική, καθώς παρότι η κατανομή των φάσεων είναι πιο ισορροπημένη από το KRK, η `immediate` φάση αντιπροσωπεύει μόλις 1.7% των θέσεων. Χωρίς στάθμιση, η φάση αυτή θα υπεκπροσωπεύτο στην εκπαίδευση, με συνέπεια το μοντέλο να δυσκολεύεται να αναγνωρίσει τις πραγματικά επικείμενες θέσεις ματ.

Παρά τη διαφορετική σκακιστική φύση του προβλήματος, η μεθοδολογία εκπαίδευσης παραμένει πλήρως ενοποιημένη με τα άλλα endgames. Αυτή η ομοιομορφία είναι σχεδιαστική επιλογή: αντί να προσαρμοστεί η διαδικασία σε κάθε φινάλε ξεχωριστά, εφαρμόζεται η ίδια ρουτίνα και στα τρία, εξασφαλίζοντας συγκρισιμότητα των αποτελεσμάτων και αποφυγή τυχαίων μεθοδολογικών αποκλίσεων.

4.4.4 Σύγκριση επιπέδων αναπαράστασης

Το βήμα της σύγκρισης τριών επιπέδων αναπαράστασης — raw coordinates, geometric features, full strategic features — υλοποιείται στο KQKComp.py με ακριβώς την ίδια μεθοδολογία που περιγράφηκε στα δύο προηγούμενα endgame: ίδιο train/test split, ίδιο βάθος και class weight (αυτά που επιλέχθηκαν ως βέλτιστα για το πλήρες σύνολο), ίδιο πλήθος seeds. Η μόνη διαφορά μεταξύ των τριών εκπαιδεύσεων είναι το σύνολο των features που τροφοδοτεί το μοντέλο.

4.4.5 Ομαδοποίηση κανόνων σε motifs

Το βήμα του rule consolidation ακολουθεί την ίδια μεθοδολογία με τα δύο προηγούμενα endgames: κάθε leaf rule κανονικοποιείται σε signature μέσω binning, και οι signatures αντιστοιχίζονται σε ονοματισμένα motifs μέσω semi-handcrafted interpretive layer, οργανωμένα ανά φάση μετατροπής.

Οι κύριες σκακιστικές έννοιες που κωδικοποιούνται στα motifs είναι σαφώς διαφορετικές από αυτές του KRK, αντικατοπτρίζοντας τη διαφορετική τεχνική νίκης. Στο KRK κυριαρχούσαν έννοιες όπως το rook cutoff, η ζώνη ματ και η προσέγγιση του βασιλιά. Στο KQK, τα motifs οργανώνονται γύρω από τρεις άξονες: το μέγεθος του queen box (πόσο έχει στενέψει), η προσέγγιση του λευκού βασιλιά (αν έχει ξεκινήσει τη συνεργασία), και η αναγνώριση κινδύνου πατ (ένας εντελώς νέος άξονας που δεν εμφανίζεται στα άλλα φινάλε).

Ένα ιδιαίτερα ενδιαφέρον σχεδιαστικό στοιχείο είναι η ύπαρξη ενός motif ειδικά για τον κίνδυνο πατ στην immediate φάση: το immediate_stalemate_precision_signature, που περιγράφει θέσεις όπου η μία λάθος κίνηση θα μπορούσε να μετατρέψει την επικείμενη νίκη σε πατ. Η ύπαρξη αυτού του motif αναδεικνύει πρακτικά την σημασία που δίνει το μοντέλο στο stalemate_risk_level feature, και ταυτόχρονα αποτυπώνει σκακιστικά μια αληθινή διάκριση: οι θέσεις σε πραγματικά επικείμενο ματ μπορούν να χωριστούν σε «καθαρές» (χωρίς κίνδυνο) και «λεπτές» (με ανάγκη ακρίβειας).

Όπως και στο KRK, υπάρχει μια κατηγορία medium_other_data_signature για signatures που ανήκουν σαφώς σε μια φάση αλλά δεν εντάσσονται σε ένα από τα κύρια σκακιστικά μοτίβα. Αυτές δεν συνιστούν αδυναμία του μοντέλου — είναι θέσεις που ταξινομούνται σωστά, αλλά με μεγαλύτερη γεωμετρική ποικιλομορφία που δεν επιτρέπει αυτόνομη σκακιστική ονομασία.

4.4.6 Οπτικοποίηση και τεκμηρίωση

Όπως και στα άλλα δύο endgames, το αρχείο KQKVisual.py παράγει τα γραφήματα που συνοδεύουν την ανάλυση, ενώ το KQKReadMe.py δημιουργεί αυτοτελές αρχείο README με πλήρη τεκμηρίωση κάθε εκτέλεσης.

4.4.7 Αποτελέσματα για το φινάλε KQK

Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα της εφαρμογής του pipeline στο φινάλε KQK, οργανωμένα ξανά σε τέσσερις άξονες: γενική απόδοση του τελικού μοντέλου, ανάλυση της συμβολής κάθε feature, σύγκριση των τριών επιπέδων αναπαράστασης, και ανάλυση των παραγόμενων κανόνων.

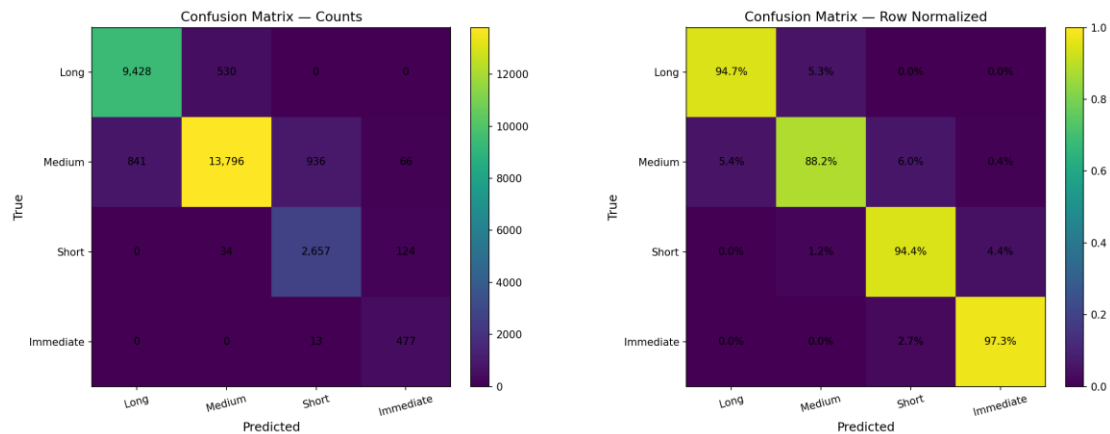
Απόδοση του τελικού μοντέλου

Ο βέλτιστος συνδυασμός υπερπαραμέτρων που επιλέχθηκε ήταν βάθος 12 με class weight balanced. Το αντίστοιχο μοντέλο επιτυγχάνει balanced accuracy 0.9366 στο test set και αντίστοιχη τιμή 0.9381 ± 0.0039 στην 5-fold cross-validation, με 689 φύλλα. Το test accuracy είναι 0.9120 και το macro F1 0.8760.

Από τα συνολικά 2.544 λάθη του μοντέλου στο test set, η ανάλυση σοβαρότητας δίνει και εδώ ιδιαίτερα ενθαρρυντικά αποτελέσματα. Τα λάθη με phase-distance 1 είναι 2.478 (97.4%), ενώ τα λάθη με phase-distance 2 είναι μόλις 66 (2.6%) και δεν υπάρχει κανένα λάθος με phase-distance 3. Δηλαδή, όπως και στο KRK, η συντριπτική πλειονότητα των λαθών του μοντέλου είναι συγκεντρωμένη σε γειτονικές φάσεις — το μοντέλο δεν κάνει «άλματα» μεταξύ πολύ διαφορετικών κλάσεων.

Η ανάλυση με βάση την κατηγορία λάθους αποκαλύπτει διαφορετική εικόνα από το KRK. Η κύρια πηγή λαθών εδώ είναι το σύνορο long-medium με 1.371 λάθη (53.9% του συνόλου), αλλά σε αντίθεση με το KRK, το σύνορο medium-short συνεισφέρει επίσης σημαντικά με 970 λάθη (38.1%). Το σύνορο short-immediate συνεισφέρει 137 λάθη (5.4%), ενώ 66 λάθη είναι medium-immediate. Η μεγαλύτερη ισορροπία στην κατανομή λαθών σε σχέση με το KRK αντικατοπτρίζει την πιο ισορροπημένη κατανομή φάσεων του KQK. Οι κλάσεις είναι πιο ισορροπημένες μεταξύ τους. Επομένως, τα λάθη δεν συγκεντρώνονται σε ένα μόνο σημείο μετάβασης, αλλά μπορούν να εμφανιστούν σε περισσότερα σύνορα μεταξύ γειτονικών φάσεων.

Παρά τη μεγαλύτερη διασπορά των λαθών, η γενική αρχή παραμένει η ίδια: το μοντέλο κάνει «κοντινά» λάθη που είναι πρακτικά χρήσιμα. Ένας σκακιστής που το συμβουλευτείται θα λάβει σωστή γενική εκτίμηση της φάσης της θέσης, με πιθανή απόκλιση μίας κατηγορίας — και ποτέ πηδημάτα δύο ή τριών.



Εικόνα 4.15 KQK Confusion Matrix

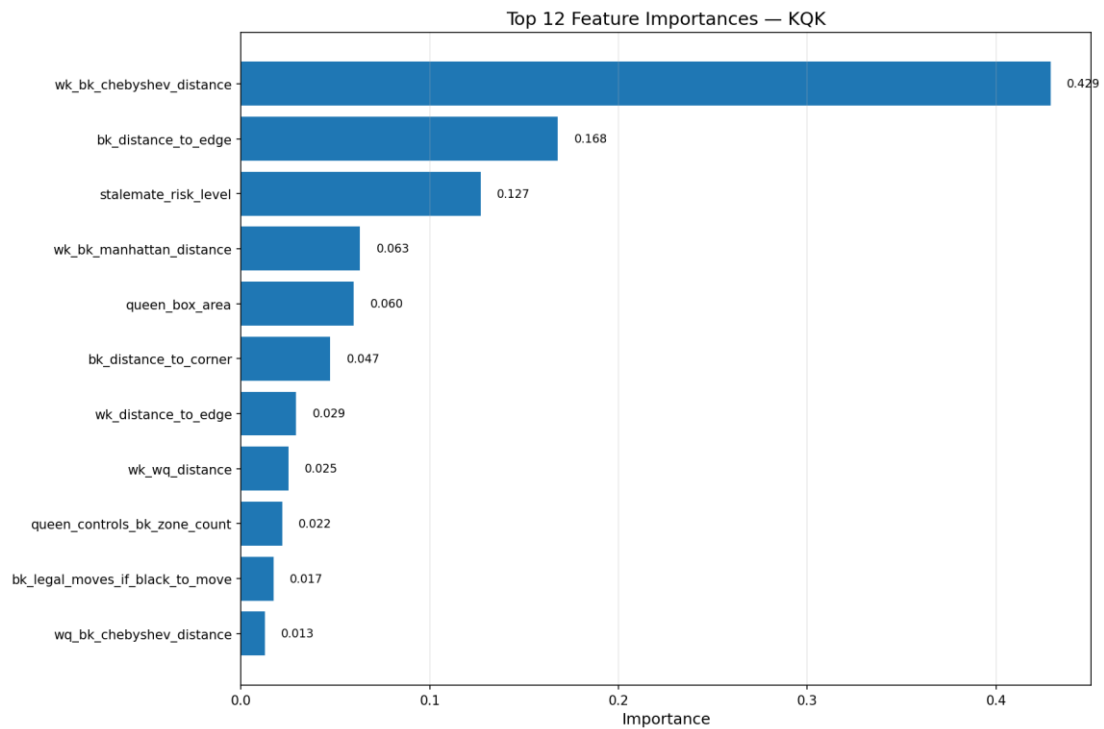
Feature importances

Η ανάλυση της συμβολής κάθε feature αποκαλύπτει μια ιεραρχία με έντονη διαφοροποίηση από τα προηγούμενα endgames. Το κυρίαρχο feature με μεγάλη απόσταση είναι το `wk_bk_chebyshev_distance` με importance 0.429, μια τιμή σαφώς υψηλότερη από οποιοδήποτε feature στα άλλα φινάλε. Ακολουθούν το `bk_distance_to_edge` (0.168) και το `stalemate_risk_level` (0.127), που από κοινού διαμορφώνουν την τριάδα των τριών κρίσιμων features. Σε μεγαλύτερη απόσταση έρχονται το `wk_bk_manhattan_distance` (0.063), το `queen_box_area` (0.060), και το `bk_distance_to_corner` (0.047). Οι υπόλοιπες μεταβλητές συνεισφέρουν λιγότερο από 3% το καθένα.

Σκακιστικά τα αποτελέσματα είναι διαφωτιστικά. Το κυρίαρχο feature — η Chebyshev απόσταση μεταξύ των βασιλιάδων — αντικατοπτρίζει τη βασική τεχνική του KQK ματ: η βασίλισσα από μόνη της δεν ματάρει, χρειάζεται τη συνεργασία του λευκού βασιλιά. Η απόσταση αυτή δεν είναι απλά γεωμετρική· είναι ο κύριος δείκτης φάσης του KQK: όσο μεγαλύτερη, τόσο πιο μακρινή η νίκη, γιατί λείπει η συνεργασία που κάνει το ματ πραγματοποιήσιμο. Είναι χαρακτηριστικό ότι η ίδια απόσταση εμφανίζεται και στο KRK ως δεύτερο πιο σημαντικό feature (importance 0.205), αλλά στο KQK η importance της υπερδιπλασιάζεται (0.429).

Η υψηλή σημασία του `stalemate_risk_level` (0.127, τρίτο σε σειρά) είναι από τα πιο ουσιαστικά ευρήματα του KQK μοντέλου. Δείχνει ότι το δέντρο, όταν του δίνεται η δυνατότητα να αναγνωρίσει τον κίνδυνο πατ, το κάνει με ενεργό τρόπο. Δεν είναι απλά ένα feature που «υπάρχει και δεν χρησιμοποιείται» — το μοντέλο πραγματικά διακρίνει τις stalemate-sensitive θέσεις από τις υπόλοιπες. Αυτό είναι σκακιστικά σωστό: μια θέση με stalemate risk συχνά ταξινομείται διαφορετικά από μια γεωμετρικά παρόμοια χωρίς stalemate risk, επειδή η ανάγκη ακριβούς χειρισμού μεταθέτει τη νίκη σε μεγαλύτερο βάθος.

Αξιοσημείωτο επίσης είναι ότι το `queen_box_area`, παρότι σκακιστικά κεντρικό ως έννοια, καταλαμβάνει την πέμπτη θέση με 0.060. Όπως και στο KRK με το `rook_cuts_off_black_king`, η ερμηνεία δεν είναι ότι το queen box είναι σκακιστικά ασήμαντο, αλλά ότι η πληροφορία που μεταφέρει είναι σε μεγάλο βαθμό ήδη ενσωματωμένη στα άλλα features — κυρίως στις αποστάσεις και στο `bk_distance_to_edge`. Πρόκειται για μια χαρακτηριστική περίπτωση αλληλεπικάλυψης, όπου εννοιολογικά διακριτά features μοιράζονται σημαντική στατιστική πληροφορία στα δεδομένα.



Εικόνα 4.16 KQK Feature importances

Σύγκριση των τριών επιπέδων αναπαράστασης

Η σύγκριση των τριών επιπέδων features δείχνει ξανά ενδιαφέροντα ευρήματα:

Επίπεδο	Balanced Accuracy	Πλήθος φύλλων	Λάθη στο test
Raw coordinates	0.8465	959	5.266
Geometric Features	0.9299	688	2.744
Full Strategic	0.9402	679	2.566

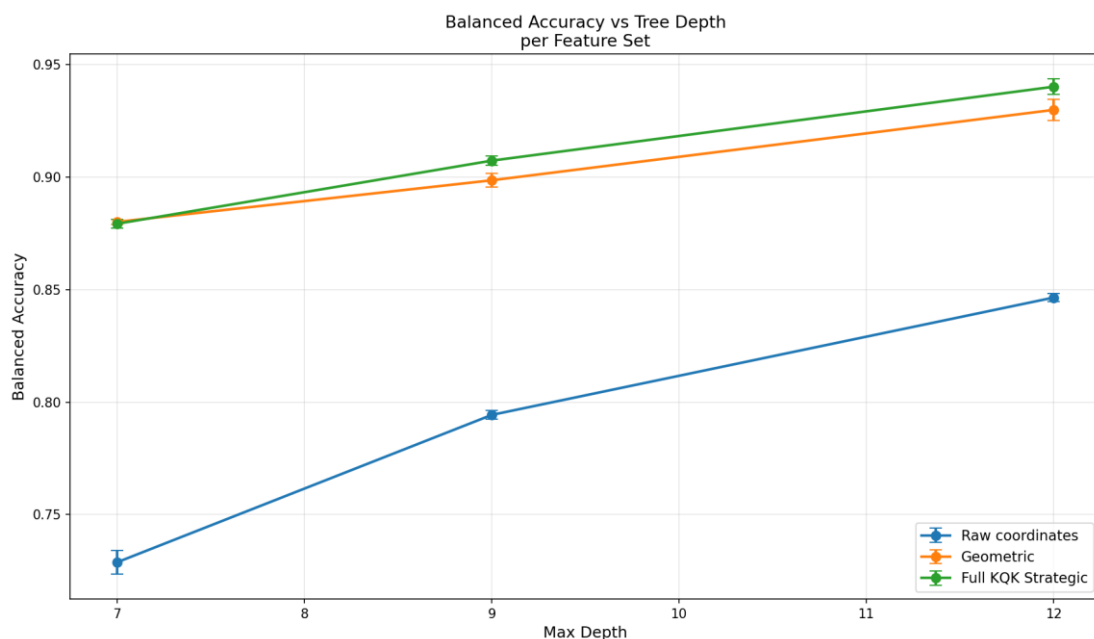
Πίνακας 4.5 Σύγκριση των τριών επιπέδων αναπαράστασης στο KQK

Το μοτίβο επαναλαμβάνει την εικόνα του KRK, με μερικές λεπτές διαφορές. Ξανά, το geometric επίπεδο τα πάει καλύτερα από το raw — και μάλιστα με ακόμα μεγαλύτερη διαφορά (0.847 $\rightarrow$ 0.930, δηλαδή +8.3 μονάδες, ενώ στο KRK ήταν +5.2). Αυτό αντικατοπτρίζει το γεγονός ότι, όπως και στο KRK, η δυναμική του KQK είναι πλήρως συμμετρική και οι σχετικές αποστάσεις αποτυπώνουν καλύτερα τη γεωμετρία από τις απόλυτες συντεταγμένες.

Η μετάβαση από geometric σε full strategic είναι εδώ πιο συγκρατημένη από τα άλλα endgames. Η balanced accuracy αυξάνεται από 0.9299 σε 0.9402 (+1 μονάδα), τα λάθη μειώνονται από 2.744 σε 2.566 (μείωση 6.5%), και το πλήθος φύλλων μειώνεται μόλις κατά 1% (από 688 σε 679). Δηλαδή, η προσθήκη των τριών σκακιστικά κωδικοποιημένων features (bk_legal_moves, queen_controls_bk_zone_count, stalemate_risk_level) και του queen_box_area δεν παράγει τη δραματική βελτίωση που είδαμε στο KPK ή στο KRK.

Δύο πιθανές ερμηνείες για αυτή τη συγκρατημένη βελτίωση. Πρώτον, η γεωμετρική πληροφορία μόνη της είναι ήδη εξαιρετικά ισχυρή στο KQK, καθώς η κυρίαρχη αρχή είναι η προσέγγιση του λευκού βασιλιά — μια αμιγώς γεωμετρική έννοια. Δεύτερον, τα σκακιστικά features του KQK είναι σαφώς πιο αλληλεπικαλυπτόμενα με τα γεωμετρικά από ό,τι στα άλλα φινάλε: το queen_box_area είναι γεωμετρική έννοια στην ουσία του, και τα δύο σκακιστικά features που μένουν (queen_controls_bk_zone_count, stalemate_risk_level) συνεισφέρουν λιγότερο σε accuracy από τη Chebyshev απόσταση από μόνη της.

Παρά τη συγκρατημένη βελτίωση, το γενικό συμπέρασμα παραμένει έγκυρο και εδώ: η προσθήκη σκακιστικής γνώσης, ακόμα και με περιορισμένη ποσοτική βελτίωση, οδηγεί σε μοντέλα που είναι πιο ερμηνεύσιμα. Το γεγονός ότι το τελικό μοντέλο αναγνωρίζει το stalemate_risk_level ως τρίτο πιο σημαντικό feature δείχνει ότι η κωδικοποίηση αυτή της σκακιστικής γνώσης παρέχει στο δέντρο τη δυνατότητα να διακρίνει σκακιστικά διακριτές θέσεις — κάτι που θα ήταν αδύνατο με μόνο γεωμετρικά features.



Εικόνα 4.17 Συγκριτικό γράφημα balanced accuracy

Ανάλυση των παραγόμενων κανόνων

Από τις 689 leaf rules του τελικού δέντρου, το rule consolidation παράγει 648 διακριτές signatures — μείωση της ίδιας τάξης με το KRK. Οι signatures αντιστοιχίζονται σε 14 ονοματισμένα motifs, οργανωμένα ανά φάση: 3 για τη long, 4 για τη medium, 5 για τη short, και 3 για την immediate.

Για τη long φάση, το κυρίαρχο motif με μεγάλη απόσταση (81.8%) είναι το long_large_box_signature — «το queen box παραμένει μεγάλο και ο μαύρος βασιλιάς έχει ακόμη αρκετό χώρο». Το γεγονός ότι ένα μόνο motif καλύπτει τόσο μεγάλο μερίδιο της φάσης αντικατοπτρίζει τη σκακιστική ομοιομορφία της long φάσης στο KQK: όλες αυτές οι θέσεις έχουν το ίδιο ουσιαστικό χαρακτηριστικό — μη επαρκή στένωση του κουτιού και ελλιπή προσέγγιση του λευκού βασιλιά. Ακολουθεί το long_white_king_far_signature (17.2%), που εστιάζει ειδικά στην απόσταση του λευκού βασιλιά.

Για τη medium φάση, το κυρίαρχο motif (73.8%) είναι το medium_queen_box_restriction_signature — «η βασίλισσα έχει περιορίσει τον χώρο, αλλά η θέση δεν έχει φτάσει ακόμη στο τελικό mating net». Ακολουθεί το medium_edge_progress_signature (21.4%), που εστιάζει στην οδήγηση προς άκρη ή γωνία.

Για τη short φάση, το κυρίαρχο motif είναι το short_corner_edge_pressure_signature (57.5%) — «ο μαύρος βασιλιάς έχει οδηγηθεί σε άκρη ή κοντά σε γωνία, άρα η τελική πίεση έχει ήδη διαμορφωθεί». Ακολουθεί το short_compact_queen_box_signature (26.5%), όπου το queen box έχει μικρύνει σημαντικά. Αξιοσημείωτο είναι ότι υπάρχει και ένα ξεχωριστό motif short_stalemate_caution_signature (7.1%) που εντοπίζει θέσεις με stalemate risk και περιορισμένη κινητικότητα — αναγνώριση που εμφανίζεται μόνο στο KQK και όχι στα άλλα φινάλε.

Για την immediate φάση, το κυρίαρχο motif είναι το immediate_stalemate_precision_signature (78.2%) — θέσεις που απαιτούν ακρίβεια λόγω πιθανού stalemate motif ή περιορισμένης κινητικότητας. Ακολουθούν δύο motifs «καθαρού» ματ χωρίς stalemate risk: immediate_quiet_queen_far_finish_signature (18.5%) και immediate_quiet_queen_support_finish_signature (3.4%). Η οργάνωση αυτή είναι σκακιστικά διαφωτιστική — η πλειονότητα των πραγματικά επικείμενων θέσεων ματ στο KQK απαιτεί προσοχή για τον κίνδυνο πατ, ενώ οι «απλές» θέσεις ματ είναι μιοσηφία.

Το catch-all ποσοστό στο KQK κυμαίνεται περίπου στο 4.9% συνολικά, ανάμεσα στο <1% του KPK και στο 7.4% του KRK. Η ενδιάμεση αυτή τιμή είναι εύλογη: το KQK είναι πιο σύνθετο από το δυαδικό KPK, αλλά σκακιστικά πιο «στερεότυπο» από το KRK, καθώς η τεχνική νίκης είναι πιο ομοιόμορφη και τα βασικά μοτίβα (queen box, stalemate, βασιλιάς σε γωνία) καλύπτουν την πλειονότητα των περιπτώσεων.

Το πιο σημαντικό εύρημα από την ανάλυση των motifs του KQK είναι ότι το μοντέλο αναγνώρισε αυτόνομα τη σημασία του stalemate — τόσο ως feature (τρίτη θέση σε importance), όσο και ως δομικό στοιχείο των παραγόμενων μοτίβων (δύο εντελώς ξεχωριστά stalemate-related motifs σε δύο διαφορετικές φάσεις). Το γεγονός αυτό αποδεικνύει ότι, όταν η σκακιστική γνώση κωδικοποιηθεί σωστά στα features, το δέντρο δεν την αγνοεί αλλά την αξιοποιεί για να παράγει κανόνες που αντιστοιχούν σε πραγματικά ιδιαίτερες σκακιστικές περιπτώσεις.

Συνολικά, το KQK επιβεβαιώνει την ευελιξία της προτεινόμενης μεθοδολογίας. Παρότι η τεχνική νίκης είναι ταχύτερη και πιο γεωμετρικά συμπαγής από το KRK, το pipeline εξακολουθεί να παράγει υψηλή απόδοση και ερμηνεύσιμα motifs. Η ιδιαίτερη συμβολή του KQK είναι ότι αναδεικνύει τον ρόλο της βασίλισσας στη στένωση του χώρου, αλλά και την ανάγκη αποφυγής πατ, δείχνοντας ότι τα decision trees μπορούν να αξιοποιήσουν όχι μόνο γενικές γεωμετρικές σχέσεις, αλλά και λεπτές σκακιστικές προειδοποιήσεις όταν αυτές κωδικοποιηθούν κατάλληλα στα features.

5. Συμπεράσματα & Μελλοντικές Βελτιώσεις

5.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία εξέτασε τη δυνατότητα εξαγωγής ερμηνεύσιμων κανόνων από endgame tablebases με χρήση δέντρων απόφασης, εφαρμόζοντας την ίδια μεθοδολογία σε τρία διαφορετικά φινάλε: KPK, KRK και KQK. Η μελέτη αυτή είχε ως στόχο όχι τη μεγιστοποίηση της ακρίβειας ταξινόμησης — που είναι ήδη 100% από τις ίδιες τις Syzygy — αλλά τη μετάφραση της τέλει αλλά αδιαφανούς πληροφορίας των tablebases σε σκακιστικούς κανόνες. Τα αποτελέσματα δείχνουν ότι ο στόχος αυτός είναι επιτεύξιμος, με αρκετές αξιοσημείωτες παρατηρήσεις.

Το κεντρικό εύρημα αφορά τη σχέση ανάμεσα στην αναπαράσταση των δεδομένων και στην ποιότητα των κανόνων. Η συστηματική σύγκριση τριών επιπέδων features — raw coordinates, geometric, και full strategic — δείχνει ότι η προσθήκη σκακιστικά κωδικοποιημένης γνώσης δεν προσφέρει πάντοτε δραματική αύξηση της ακρίβειας. Στο KPK η βελτίωση από raw σε full strategic ήταν σαφής (0.9542 → 0.9733 balanced accuracy), στο KRK μέτρια (0.874 → 0.948), και στο KQK ουσιαστικά αμελητέα σε accuracy. Το πραγματικά σημαντικό αποτέλεσμα όμως ήταν διαφορετικό: σε όλα τα endgames, η σκακιστική γνώση οδήγησε σε σαφώς πιο συμπαγή και ερμηνεύσιμα δέντρα. Το KPK χρειάστηκε σχεδόν τους μισούς κόμβους με τα πλήρη features σε σχέση με τις ωμές συντεταγμένες, το KRK περιόρισε τα λάθη κατά 25%, ενώ στο KQK η σκακιστική γνώση, ακόμα και όταν δεν βελτίωσε την ακρίβεια, ενσωματώθηκε ουσιαστικά στους παραγόμενους κανόνες — με χαρακτηριστικότερο παράδειγμα την ενεργή αξιοποίηση του stalemate_risk_level. Το συμπέρασμα, με άλλα λόγια, είναι ότι η αξία του feature engineering για ένα ερμηνεύσιμο μοντέλο δεν μετρίεται μόνο σε ποσοστά accuracy — μετρίεται και στη συμπαγή δομή των κανόνων που παράγει.

Ένα δεύτερο σημαντικό εύρημα αφορά την ανακάλυψη γνωστών σκακιστικών εννοιών. Τα δέντρα που εκπαιδεύτηκαν, χωρίς καμία ρητή σκακιστική καθοδήγηση πέρα από τα features, παρήγαγαν κανόνες που αντιστοιχούν άμεσα σε γνωστές έννοιες της θεωρίας φινάλε. Στο KPK εμφανίστηκαν τα κλασικά μοτίβα του key square race και του promotion blockade. Στο KRK κυριάρχησαν η οδήγηση προς τη ζώνη ματ, ο περιορισμός του βασιλιά, και η συνεργασία βασιλιά-πύργου. Στο KQK το μοντέλο ανέδειξε αυτόνομα τη σημασία του κινδύνου πατ, δημιουργώντας ξεχωριστά motifs για stalemate-sensitive θέσεις. Το αποτέλεσμα αυτό δείχνει ότι, όταν σχεδιάσουν σωστά τα features, ένα δέντρο απόφασης εκπαιδευμένο σε tablebases δεν είναι απλά «τυχαίος» ταξινομητής — παράγει κανόνες που θα αναγνώριζε ένας σκακιστής.

Ένα τρίτο εύρημα αφορά τη δομή των λαθών στα multi-class προβλήματα των KRK και KQK. Στα δύο αυτά endgames, όπου η ταξινόμηση σε φάσεις μετατροπής μπορεί εγγενώς να είναι ασαφής σε γειτονικά όρια, η συντριπτική πλειονότητα των λαθών του μοντέλου (>97% και στα δύο) βρίσκεται σε phase-distance = 1. Το μοντέλο δηλαδή δεν κάνει «άλλατα» μεταξύ πολύ διαφορετικών κλάσεων — προβλέπει σχεδόν πάντα μια γειτονική φάση. Πρακτικά αυτό σημαίνει ότι το μοντέλο, ακόμα και όταν κάνει λάθος, δίνει μια πρακτικά αξιοποιήσιμη εκτίμηση της φάσης της θέσης.

Τέλος, αξίζει να αναφερθεί μια συνολική παρατήρηση για την ενιαία αρχιτεκτονική που σχεδιάστηκε. Το γεγονός ότι το ίδιο pipeline εφαρμόστηκε σε τρία διαφορετικά endgames, με τις προσαρμογές να περιορίζονται στα features και στα thresholds, επιτρέπει τη συγκρισιμότητα των αποτελεσμάτων και αποδεικνύει ότι η προσέγγιση δεν είναι εξαρτημένη από τη σκακιστική φύση κάθε συγκεκριμένου φινάλε. Αυτή η μεθοδολογική γενικευσιμότητα είναι ίσως το πιο μεταφέσιμο συμπέρασμα της εργασίας — και ανοίγει τον δρόμο για τις μελλοντικές βελτιώσεις που ακολουθούν.

5.2 Μελλοντικές βελτιώσεις

Παρά την επιτυχή εφαρμογή της μεθοδολογίας στα τρία endgames, υπάρχουν αρκετές κατευθύνσεις στις οποίες η εργασία μπορεί να επεκταθεί και να ωφεληθεί.

Πλήρης αξιοποίηση των συμμετριών σε όλα τα endgames

Η αξιοποίηση της οριζόντιας συμμετρίας στο KPK επέτρεψε τη μείωση του χώρου αναζήτησης κατά το ήμισυ και βοήθησε το δέντρο να επικεντρωθεί στα ουσιαστικά σκακιστικά μοτίβα χωρίς να «σπαταλά» πληροφορία σε κατοπτρικά ισοδύναμες θέσεις. Αντίθετα, στα KRK και KQK δεν εφαρμόστηκε κανονικοποίηση συμμετρίας, με αποτέλεσμα ο χώρος αναζήτησης να παραμείνει πλήρης. Δεδομένου ότι το KRK και το KQK, ως φινάλε χωρίς πιόνια, έχουν πλήρη 8-fold συμμετρία, μια μελλοντική επέκταση της εργασίας θα μπορούσε να ενσωματώσει την κανονικοποίηση αυτή στους αντίστοιχους Generators. Το αναμενόμενο όφελος είναι διπλό: μείωση του μεγέθους του dataset σε λιγότερο από το ένα όγδοο των τρεχουσών τιμών (κάτι που επιταχύνει την εκπαίδευση), και ενδεχόμενη βελτίωση της γενικευτικής ικανότητας του μοντέλου, καθώς οι κανόνες θα εκφράζονται πάντα σε μια κανονική μορφή αντί σε πολλαπλές κατοπτρικές παραλλαγές.

Αυτόματη εξαγωγή σκακιστικών μοτίβων

Ένας από τους μεθοδολογικούς περιορισμούς της τρέχουσας υλοποίησης, που αναγνωρίστηκε τίμια στην ενότητα του Rule Consolidation, είναι η semi-handcrafted φύση του interpretation layer: η αντιστοίχιση signatures σε ονοματισμένα motifs έγινε χειροκίνητα, με βάση σκακιστική παρατήρηση των κυρίαρχων συνδυασμών. Αυτή η προσέγγιση δούλεψε πολύ καλά για την τρέχουσα εργασία (με catch-all κάτω από 1% στο KPK και της τάξης του 5-7% στα άλλα δύο), αλλά έχει δύο εγγενείς περιορισμούς: εξαρτάται από τη σκακιστική εμπειρία του σχεδιαστή, και δεν κλιμακώνει αυτόματα σε νέα endgames.

Μια πιο αυστηρή μελλοντική εξέλιξη θα αφαιρούσε πλήρως το χειροκίνητο στοιχείο και θα βασιζόταν σε αυτόματες τεχνικές ομαδοποίησης. Τρεις κατευθύνσεις είναι ιδιαίτερα υποσχόμενες: πρώτον, η αυτόματη εξαγωγή των top-N πιο συχνών signatures ανά κλάση και η εκ των υστέρων ονομασία τους από ειδικό, χωρίς προκαθορισμένα πρότυπα. Δεύτερον, η εφαρμογή clustering αλγορίθμων πάνω στα διανύσματα των features κάθε φύλλου, ώστε να ανιχνευθούν οι πραγματικές δομές που το δέντρο έχει «ανακαλύψει». Τρίτον, η χρήση τεχνικών frequent itemset mining (όπως FP-Growth) για την εντόπιση συχνών συνδυασμών χαρακτηριστικών που ορίζουν αναγνωρίσιμα μοτίβα. Οποιαδήποτε από αυτές τις τεχνικές θα αύξανε την επιστημονική αυστηρότητα του βήματος και θα μείωνε την εξάρτηση από ανθρώπινη κρίση, μετατρέποντας το rule consolidation από έναν συνδυασμό αυτόματης και χειροκίνητης εργασίας σε πλήρως αυτοματοποιημένη διαδικασία.

Επέκταση σε περισσότερα endgames

Η παρούσα εργασία περιορίστηκε σε τρία φινάλε τριών κομματιών, τα οποία έχουν σαφή σκακιστική δομή και πλήρη κάλυψη από τις 3-4-5 Syzygy tablebases. Η επόμενη λογική κατεύθυνση είναι η επέκταση σε πιο σύνθετα φινάλε, όπως το KBBK (βασιλιάς και δύο αξιωματικοί εναντίον βασιλιά), το KBNK (βασιλιάς, αξιωματικός και ίππος εναντίον βασιλιά — ένα φινάλε γνωστό για την τεχνική δυσκολία του), και σταδιακά σε φινάλε τεσσάρων ή πέντε κομματιών όπως τα KRKP, KQKR, ή KRPKR.

Η επέκταση αυτή έχει ιδιαίτερο ενδιαφέρον για δύο λόγους. Αφενός, θα δοκίμαζε τα όρια της τρέχουσας μεθοδολογίας: τα φινάλε με περισσότερα κομμάτια έχουν εκθετικά πιο σύνθετο χώρο θέσεων και πιο ποικίλη σκακιστική δυναμική, οπότε ο σχεδιασμός features θα απαιτούσε βαθύτερη σκακιστική ανάλυση. Αφετέρου, θα προσέφερε συγκεκριμένη εκπαιδευτική αξία: το KBNK, για παράδειγμα, θεωρείται από τα πιο δύσκολα προς κατανόηση φινάλε από τους μεσαίου επιπέδου σκακιστές, και ένα σύνολο ερμηνεύσιμων κανόνων ML-generated για αυτό το φινάλε θα είχε πραγματική χρησιμότητα.

Η κλιμάκωση σε 4-piece και 5-piece φινάλε θα απαιτούσε επίσης εγκατάσταση των μεγαλύτερων Syzygy αρχείων (σχεδόν 150 GB για 6-piece κάλυψη), αλλά αυτό είναι πρακτικά διαχειρίσιμο με σύγχρονα SSD. Η μεγαλύτερη πρόκληση θα ήταν σκακιστική και όχι υπολογιστική: η ανάγκη σχεδιασμού σκακιστικά νοηματοδοτημένων features για κάθε νέο τύπο υλικού, κάτι που για τα σύνθετα φινάλε δεν είναι τετριμμένο.

Επιπρόσθετες πιθανές κατευθύνσεις

Πέρα από τις τρεις κύριες κατευθύνσεις που περιγράφηκαν, υπάρχουν και ορισμένες πιο τεχνικές βελτιώσεις που θα αύξαναν την αυστηρότητα της μεθοδολογίας. Πρώτον, η υιοθέτηση τριμερούς διαχωρισμού των δεδομένων σε train/validation/test, με την επιλογή υπερπαραμέτρων να γίνεται αποκλειστικά στο validation set. Αυτό θα εξάλειφε πλήρως την αλληλεπίδραση μεταξύ επιλογής μοντέλου και τελικής αξιολόγησης που συζητήθηκε στην 4.2.3. Δεύτερον, η δοκιμή εναλλακτικών ερμηνεύσιμων μοντέλων πέρα από τα δέντρα απόφασης, όπως rule sets (π.χ. RIPPER) ή sparse decision trees με explicit ερμηνευσιμότητα, θα προσέφερε ενδιαφέρουσα σύγκριση των διαφορετικών προσεγγίσεων στην ίδια εργασία. Τρίτον, η ενσωμάτωση τεχνικών explainable AI (SHAP values, LIME) θα προσέφερε συμπληρωματική ματιά στη λειτουργία των δέντρων, ιδίως στα σημεία όπου η ερμηνεία τους δεν είναι άμεσα προφανής.

Καμία από αυτές τις κατευθύνσεις δεν αναιρεί τα ευρήματα της παρούσας εργασίας. Αντίθετα, όλες τους προϋποθέτουν την υπάρχουσα μεθοδολογία ως βάση και προτείνουν επεκτάσεις που θα την καταστήσουν πιο πλήρη, πιο αυστηρή και πιο ευρέως εφαρμόσιμη.

6. Βιβλιογραφία

- [1]: Διπλωματική Εργασία Θάνου Δ. Αναστάσιου.
<https://nemertes.library.upatras.gr/server/api/core/bitstreams/345af2d8-63e1-4676-bde1-dc17a6901589/content>
- [2]: FIDE. Laws of Chess. https://rcc.fide.com/fide-laws-of-chess_fulltexthtml/
- [3]: Encyclopedia Britannica. *Chess — History*.
<https://www.britannica.com/topic/chess/History>
- [4]: Dvoretsky, M. Dvoretsky's Endgame Manual, 6th Edition, revised by Karsten Müller and Alex Fishbein, foreword by Magnus Carlsen. Russell Enterprises.
- [5]: Chessprogramming Wiki. *Endgame Tablebases*.
https://www.chessprogramming.org/Endgame_Tablebases
- [6]: Chessprogramming Wiki. *Syzygy Bases*.
https://www.chessprogramming.org/Syzygy_Bases
- [7]: ChessBase. *Syzygy Tablebases*.
https://help.chessbase.com/CBase/15/Eng/index.html?syzygy_tablebases.htm
- [8]: python-chess Documentation. Syzygy Endgame Tablebase Probing.
<https://python-chess.readthedocs.io/en/latest/syzygy.html>
- [9] Machine Learning, Tom Mitchell, McGraw Hill, 1997.
<https://www.cs.cmu.edu/~tom/mlbook.html>
- [10]: Κάλλιπος. *Τεχνητή Νοημοσύνη / Μηχανική Μάθηση*, ηλεκτρονικό εκπαιδευτικό υλικό.

http://repfiles.kallipos.gr/html_books/93/04a-main.html
- [11]: Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

https://www.cs.uoi.gr/~arly/courses/ml/tmp/Bishop_book.pdf
- [12]: IBM. *What is Feature Engineering?*

<https://www.ibm.com/think/topics/feature-engineering>
- [13]: Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and Regression Trees*. Wadsworth International Group, 1984. (Έντυπο Βιβλίο)
- [14]: scikit-learn Documentation. *DecisionTreeClassifier*.

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
- [15]: Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής. *Δημιουργία Βάσεων Κανόνων από Δεδομένα — Εξαγωγή Κανόνων*, σημειώσεις μαθήματος.

https://eclass.upatras.gr/modules/document/file.php/CEID1095/%CE%94%CE%99%CE%91%CE%9B%CE%95%CE%9E%CE%95%CE%99%CE%A3%202022-23%20%CE%92%CE%84%CE%9C%CE%95%CE%A1%CE%9F%CE%A3%08_rules_8-rule-extraction_rev.pdf

[16]: scikit-learn Documentation. *Decision Trees*.

<https://scikit-learn.org/stable/modules/tree.html>

[17]: Ken Thompson, "Retrograde Analysis of Certain Endgames", ICCA Journal, Vol. 9, No. 3, 1986.

[18]: Wikipedia. Endgame tablebase.

https://en.wikipedia.org/wiki/Endgame_tablebase

[19]: Nalimov, E. V., Haworth, G. M. and Heinz, E. A. (2000) Space-efficient Indexing of Chess Endgame Tables. ICGA Journal, 23

[20]: Miguel A. Ballicora, Gaviota Chess Endgame Tablebases — Probing Code.

<https://github.com/michiguel/Gaviota-Tablebases>

[21]: Chessprogramming Wiki. Gaviota Tablebases.

https://www.chessprogramming.org/Gaviota_Tablebases

[22]: Chessprogramming Wiki, Lomonosov Tablebases.

https://www.chessprogramming.org/Lomonosov_Tablebases

[23]: Ronald de Man, Syzygy Tablebases — Documentation and Source Code.

<https://github.com/syzygy1/tb> (Η πρωτογενής πηγή για τη δομή των Syzygy.)

[24]: Chessprogramming Wiki. Scorpio Bitbases.

https://www.chessprogramming.org/Scorpio_Bitbases

[25]: Lichess Tablebase API. <https://lichess.org/api#tag/Tablebase>

[26]: Jon Dart, Fathom — Stand-alone Syzygy probing code.

<https://github.com/jdart1/Fathom>

[27]: Massimiliano Goi, Chess Endgame Database — Tablebases Download.

<https://chess.massimilianogoi.com/download/tablebases/>

[28]: King and pawn versus king endgame

https://en.wikipedia.org/wiki/King_and_pawn_versus_king_endgame#