Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης

Interpretable Decision Trees

Diploma Thesis

by

Maria-Nefeli Kondyli

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Interpretable Decision Trees

Diploma Thesis

by

Maria-Nefeli Kondyli

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 17η Ιουλίου, 2024.

.....	.....	.....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

.....

Μαρια-Νεφελη Κονδυλη
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η αυξανόμενη ενσωμάτωση της Τεχνητής Νοημοσύνης σε κρίσιμες περιοχές λήψης αποφάσεων, όπως η υγειονομική περίθαλψη, τα οικονομικά και η ποινική δικαιοσύνη, υπογραμμίζει την αναγκαιότητα αυτά τα συστήματα να είναι σαφή και κατανοητά. Πολλές αποφάσεις υψηλού κινδύνου λαμβάνονται επί του παρόντος με τη χρήση μοντέλων "μαύρου κουτιού", και οι προσπάθειες να εξηγηθούν αυτά τα μοντέλα, αντί να αναπτυχθούν μοντέλα που είναι εγγενώς διαφανή και ερμηνεύσιμα, μπορούν να διαιωνίσουν επιβλαβείς πρακτικές και να οδηγήσουν σε σημαντική κοινωνική ζημία. Τα δέντρα αποφάσεων είναι γνωστά στον τομέα της μηχανικής μάθησης για την εγγενή τους διαφάνεια, αποτελώντας παράδειγμα του πώς οι πολύπλοκοι αλγόριθμοι μπορούν να καταστούν κατανοητοί. Ωστόσο, αυτή η εγγενής ερμηνευσιμότητα δεν ισοδυναμεί πάντα με εξηγησιμότητα στον πραγματικό κόσμο, όπου η σαφήνεια των αποφάσεων ενός μοντέλου πρέπει να είναι εφαρμόσιμη και να έχει νόημα για όλους τους χρήστες.

Η παρούσα εργασία διερευνά τις υπάρχουσες προκλήσεις για να καταστούν τα δέντρα αποφάσεων πραγματικά ερμηνεύσιμα, με στόχο να γεφυρώσει το χάσμα μεταξύ της τεχνικής διαφάνειας και της πρακτικής κατανόησης. Εμβαθύνουμε στην κατανόηση της ανθρώπινης λήψης αποφάσεων και στους περιορισμούς της ανθρώπινης αντίληψης. Με γνώμονα αυτά, προτείνουμε τεχνικές αύξησης της ερμηνευσιμότητας στα δέντρα απόφασης, τις οποίες αξιολογούμαι μέσω μιας έρευνας χρηστών.

Λέξεις-κλειδιά — Δένδρο Απόφασης, Εξηγησιμότητα, Ερμηνευσιμότητα, Εξηγήσιμη Τεχνητή Νοημοσύνη, Ανθρώπινη Λήψη Αποφάσεων, Strong Optimal Classification Trees, C4.5

Abstract

The increasing integration of Artificial Intelligence in critical decision-making areas, such as healthcare, finance, and criminal justice, underscores the necessity for these systems to be clear and understandable. Many high-stakes decisions are currently made using "black box" models, and efforts to explain these models, rather than developing models that are inherently transparent and interpretable, can perpetuate harmful practices and lead to significant social harm. Decision trees are well-known in the field of machine learning for their inherent transparency, exemplifying how complex algorithms can be made understandable. However, this inherent interpretability does not always equate to explainability in the real world, where the clarity of a model's decisions must be applicable and meaningful to all users.

This thesis explores the existing challenges in making decision trees truly interpretable, aiming to bridge the gap between technical transparency and practical understanding. We delve into the understanding of human decision-making and the limitations of human perception. Guided by these insights, we propose techniques to enhance the interpretability of decision trees, which we evaluate through a user survey.

Keywords — Decision Trees, Explainability, Interpretability, Explainable AI, Human Decision Making, Strong Optimal Classification Trees, C4.5

Ευχαριστίες

Ευχαριστώ πολύ τον επιβλέποντα μου, κ. Στάμου Γεώργιο, για την πολύτιμη καθοδήγηση του στην εκπόνηση αυτής της εργασίας. Ευχαριστώ επίσης τον Ορφέα Μενή-Μαστρομιχαλάκη για τη στενή συνεργασία και την υποστήριξη καθ' όλη τη διάρκεια εξερεύνησης των καινούριων αυτών αντικειμένων, καθώς και τον Ιάσονα Λιάρτη που μοιράστηκε τις ιδέες του μαζί μου.

Θα ήθελα να ευχαριστήσω την οικογένεια μου που μου προσέφερε τα εφόδεια για να πετύχω τους στόχους μου.

Τέλος θα ήθελα να ευχαριστήσω τους φίλους μου και ιδιαίτερα τις Γρηγορία, Κατερίνα και Πελαγία για την αμέριστη στήριξη τους στην ακαδημαϊκή μου περιπέτεια και τις όμορφες στιγμές που ζήσαμε. Χωρίς αυτούς δεν θα μπορούσα να φτάσω εδώ που βρίσκομαι σήμερα.

Μαρία-Νεφέλη Κονδύλη, Ιούλιος 2024

Contents

List of Figures

Chapter 0

Εκτεταμένη Περίληψη στα Ελληνικά

0.1 Εισαγωγή

Καθώς η τεχνητή νοημοσύνη γίνεται όλο και πιο αναπόσπαστο μέρος της λήψης αποφάσεων σε κρίσιμους τομείς όπως την υγειονομική περίθαλψη, τα οικονομικά και την ποινική δικαιοσύνη, η ανάγκη να είναι τα συστήματα αυτά σαφή και κατανοητά δεν έχει υπάρξει ποτέ άλλοτε ήταν ποτέ πιο ουσιαστική. Πολλές από αυτές τις αποφάσεις υψηλού ρίσκου λαμβάνονται με τη χρήση μοντέλων "μαύρου κουτιού" και την προσπάθεια να εξηγηθούν αυτά τα μοντέλα, αντί να αναπτυχθούν μοντέλα που είναι εγγενώς σαφή και ερμηνεύσιμα, μπορεί να διαιωνίσει επιβλαβείς πρακτικές και θα μπορούσε ακόμη και να οδηγήσει σε σοβαρές ζημιές στην κοινωνία. Τα δέντρα αποφάσεων, φημίζονται ευρέως για την εγγενή διαφάνειά τους στον τομέα της μηχανικής μάθησης, αποτελούν παράδειγμα για το πώς οι πολύπλοκοι αλγόριθμοι μπορούν να γίνουν κατανοητοί- η ιεραρχική δομή τους επιτρέπει συχνά την οπτικοποίηση και την κατανόηση ακόμη και από μη τεχνικά καταρτισμένους ενδιαφερόμενους. Ωστόσο, αυτή η ενσωματωμένη ερμηνευσιμότητα δεν ισοδυναμεί πάντα με την επεξηγηματικότητα στον πραγματικό κόσμο, όπου η σαφήνεια των αποφάσεων του μοντέλου πρέπει να είναι εφαρμόσιμη και να έχει νόημα για όλους τους χρήστες. Η παρούσα εργασία διερευνά τις υφιστάμενες προκλήσεις για να καταστούν τα δέντρα αποφάσεων πραγματικά ερμηνεύσιμα, με στόχο να γεφυρώσει το χάσμα μεταξύ της τεχνικής διαφάνειας και της πρακτικής κατανοητότητας. Βελτιώνοντας τον τρόπο με τον οποίο τα δέντρα αποφάσεων επικοινωνούν τη συλλογιστική τους, η μελέτη επιδιώκει να συμβάλει σημαντικά στην ανάπτυξη συστημάτων τεχνητής νοημοσύνης που δεν είναι μόνο αποτελεσματικά αλλά και αξιόπιστα και δίκαια, διασφαλίζοντας ότι η Τεχνητή Νοημοσύνη υποστηρίζει αντί να υπονομεύει κρίσιμες ανθρώπινες αποφάσεις.

0.1.1 Ερμηνεύσιμα Δένδρα Αποφάσεων

Τα δέντρα αποφάσεων θεωρούνται ερμηνεύσιμα λόγω της απλής, ιεραρχικής δομής τους, η οποία μιμείται στενά την ανθρώπινη λογική λήψης αποφάσεων. Κάθε κόμβος σε ένα δέντρο αποφάσεων αντιπροσωπεύει ένα σημείο απόφασης με βάση ένα συγκεκριμένο χαρακτηριστικό, και οι κλάδοι που πηγάζουν από τους κόμβους απεικονίζουν τα πιθανά αποτελέσματα. Αυτό το μοντέλο που μοιάζει με δέντρο επιτρέπει στους χρήστες να ακολουθήσουν τη διαδρομή από τη ρίζα προς τα φύλλα, παρακολουθώντας οπτικά τη σειρά των αποφάσεων που οδηγούν σε ένα τελικό αποτέλεσμα. Αυτή η σαφήνεια του τρόπου με τον οποίο προκύπτουν οι αποφάσεις διευκολύνει τους χρήστες να κατανοήσουν και να επαληθεύσουν τη διαδικασία συλλογισμού, ενισχύοντας την εμπιστοσύνη και διευκολύνοντας την ευκολότερη επικοινωνία του τρόπου με τον οποίο λαμβάνονται οι αποφάσεις του μοντέλου. Πολλές φορές, όμως, η δομή, η πολυπλοκότητα και η ορολογία που χρησιμοποιείται σε ένα δέντρο αποφάσεων μπορεί να είναι λογική για έναν αλγόριθμο, αλλά μπορεί να είναι αινιγματική ή αντιδιανοητική για τους ανθρώπινους ενδιαφερόμενους. Καθώς το βάθος και το πλάτος των δέντρων αυξάνονται για να φιλοξενήσουν πολύπλοκα σύνολα δεδομένων, μπορεί να γίνουν υπερβολικά περίπλοκα, με πολυάριθμες διακλαδώσεις και συνθήκες που είναι δύσκολο να ακολουθηθούν. Αυτή η πολυπλοκότητα μπορεί να επισκιάσει τη σαφή, γραμμική λογική που καθιστά τα μικρότερα δέντρα προσιτά, μετατρέποντάς τα σε ένα μπερδεμένο ιστό αποφάσεων που δυσκολεύει την κατανόηση. Επιπλέον, η χρήση τεχνικής ορολογίας ή ειδικών όρων για τον τομέα στους κόμβους μπορεί να δυσχεράνει την κατανόηση της υποκείμενης συλλογιστικής από μη ειδικούς χρήστες χωρίς εξειδικευμένες γνώσεις. Έτσι, ενώ τα δέντρα αποφάσεων επιδιώκουν εγγενώς τη διαφάνεια, η πρακτική εφαρ-

μογή τους σε πιο σύνθετα σενάρια μπορεί να οδηγήσει ακούσια σε αδιαφάνεια, περιπλέκοντας την κατανόηση των ενδιαφερομένων και μειώνοντας ενδεχομένως την εμπιστοσύνη που αποδίδεται σε αυτά τα μοντέλα.

0.1.2 Συνεισφορά

Η παρούσα εργασία επικεντρώνεται στα ερμηνεύσιμα δέντρα αποφάσεων, έναν ακρογωνιαίο λίθο της ερμηνεύσιμης μηχανικής μάθησης, που οφείλεται στην επιτακτική ανάγκη να οικοδομηθεί εμπιστοσύνη και υπευθυνότητα στις εφαρμογές τεχνητής νοημοσύνης. Παρά την εγγενή τους διαφάνεια σε σύγκριση με άλλα πολύπλοκα μοντέλα, τα δέντρα αποφάσεων εξακολουθούν να παρουσιάζουν προκλήσεις στην ερμηνευσιμότητα που πρέπει να αντιμετωπιστούν για την πλήρη αξιοποίηση των δυνατοτήτων τους. Για την αντιμετώπιση αυτών των προκλήσεων, η παρούσα έρευνα εισάγει και αξιολογεί νέα μέτρα ερμηνευσιμότητας ειδικά σχεδιασμένα για δέντρα αποφάσεων, ενισχύοντας την αναλυτική αυστηρότητα με την οποία αξιολογείται η ερμηνευσιμότητά τους. Με τη διερεύνηση των παραγόντων που επηρεάζουν την ανθρώπινη αντίληψη της ερμηνευσιμότητας, η παρούσα μελέτη επιδιώκει να συμβάλει στην ανάπτυξη δέντρων αποφάσεων που δεν είναι μόνο ισχυρά ως προς την απόδοση αλλά και ως προς τη σαφήνεια και την εμπιστοσύνη των χρηστών. Η προσέγγιση αυτή υπογραμμίζει τον κρίσιμο ρόλο της ερμηνευσιμότητας στη γεφύρωση του χάσματος μεταξύ των δυνατοτήτων της Τεχνητής Νοημοσύνης και των ανθρωποκεντρικών εφαρμογών, τονίζοντας τον ευρύτερο αντίκτυπο της παρούσας έρευνας στην προώθηση ηθικών και υπεύθυνων πρακτικών Τεχνητής Νοημοσύνης.

0.2 Θεωρητικό Υπόβαθρο

0.2.1 Ερμηνευσιμότητα

Η ερμηνευσιμότητα, η οποία ορίζεται ως η ικανότητα ενός μοντέλου να παρέχει ερμηνεύσιμες εξηγήσεις για τις προβλέψεις του, έχει αναδειχθεί σε κρίσιμη πτυχή της έρευνας και της πρακτικής της μηχανικής μάθησης. Τα επεξηγήσιμα μοντέλα προσφέρουν πληροφορίες σχετικά με τους παράγοντες που οδηγούν τις αποφάσεις τους, δίνοντας στους χρήστες τη δυνατότητα να κατανοήσουν, να επικυρώσουν και ενδεχομένως να αμφισβητήσουν τα αποτελέσματά τους. Στην εξόρυξη δεδομένων και τη μηχανική μάθηση, η ερμηνευσιμότητα ορίζεται ως η ικανότητα εξήγησης ή παροχής του νοήματος με κατανοητούς όρους σε έναν άνθρωπο. Η κατανόηση ενός μοντέλου που δημιουργείται από υπολογιστή αποτελεί συχνά προϋπόθεση για να εμπιστευτούν οι χρήστες τις προβλέψεις του μοντέλου και να ακολουθήσουν τις συστάσεις που σχετίζονται με αυτές τις προβλέψεις.

Δύο κύριες προσεγγίσεις έχουν αναδειχθεί στη βιβλιογραφία για τη διευκόλυνση της κατανόησης των μοντέλων μηχανικής μάθησης: η επεξήγηση του μαύρου κουτιού και ο σχεδιασμός του διαφανούς μοντέλου[4]. Τα μοντέλα μαύρου κουτιού είναι μοντέλα μηχανικής μάθησης ή αλγόριθμοι που είναι ιδιαίτερα πολύπλοκα και δύσκολα ερμηνεύονται ή εξηγούνται ως προς τον τρόπο με τον οποίο καταλήγουν στις προβλέψεις ή τις αποφάσεις τους. Αυτά τα μοντέλα συχνά περιλαμβάνουν περίπλοκες μαθηματικές συναρτήσεις ή αρχιτεκτονικές με μεγάλο αριθμό παραμέτρων, γεγονός που καθιστά δύσκολο για τους ανθρώπους να κατανοήσουν τους υποκείμενους μηχανισμούς ή τη λογική πίσω από τα αποτελέσματά τους. Οι τεχνικές εξήγησης του μαύρου κουτιού (post-hoc εξηγήσεις) αναφέρονται σε μεθόδους που αποσκοπούν στην εξήγηση του τρόπου με τον οποίο τα μοντέλα παράγουν τα αποτελέσματά τους. Είναι ακατάλληλα σε συστήματα λήψης αποφάσεων υψηλού κινδύνου, καθώς μπορούν να χειραγωγηθούν ώστε να αφηγηθούν μια διαφορετική ιστορία από εκείνη του black-box που εξηγούν. Σε αντίθεση με τα παραδοσιακά μοντέλα black-box, τα οποία δίνουν προτεραιότητα στην ακρίβεια πρόβλεψης εις βάρος της ερμηνευσιμότητας, τα επεξηγήσιμα μοντέλα επιτυγχάνουν μια ισορροπία μεταξύ απόδοσης και διαφάνειας, προωθώντας την εμπιστοσύνη και την υπευθυνότητα στα συστήματα ΤΝ [32]. Συνήθως τα ερμηνεύσιμα μοντέλα περιορίζονται στη μορφή του μοντέλου ώστε αυτό είτε να είναι χρήσιμο σε κάποιον, είτε να υπακούει σε βασική γνώση του τομέα, όπως η μονοτονία, η αιτιότητα [63].

Η σημασία των ερμηνεύσιμων μοντέλων

Η σημασία των ερμηνεύσιμων μοντέλων Μηχανικής Μάθησης απορρέει από διάφορα ζητήματα. Αρχικά, η κατανόηση ενός μοντέλου που παράγεται από έναν υπολογιστή είναι συχνά απαραίτητη για να έχουν οι χρήστες εμπιστοσύνη στις προβλέψεις του μοντέλου και να ακολουθούν τις σχετικές συστάσεις. Αυτή η ανάγκη για εμπιστοσύνη στις υπολογιστικές προβλέψεις είναι ιδιαίτερα έντονη σε κρίσιμους τομείς όπως η ιατρική, όπου διακυβεύονται ανθρώπινες ζωές [48], [8], αλλά και σε οικονομικά πλαίσια [21]. Η απαίτηση για διαφανή μοντέλα για την ενίσχυση της εμπιστοσύνης των χρηστών γίνεται ακόμη πιο εμφανής όταν το σύστημα παρουσιάζει

ένα απροσδόκητο μοντέλο στον χρήστη, γεγονός που απαιτεί διεξοδικές εξηγήσεις από το σύστημα για την αποδοχή του μοντέλου. Επιπλέον, σε ορισμένους τομείς εφαρμογών, οι χρήστες απαιτούν επαρκή κατανόηση των συστάσεων του συστήματος για να παρέχουν νομικά ορθές εξηγήσεις για τις ενέργειές τους σε άλλους. Σίγουρα, υπάρχουν περιπτώσεις όπου η ερμηνευσιμότητα του μοντέλου έχει μικρή σημασία και ορισμένοι χρήστες μπορεί να βρουν ικανοποίηση στο να υιοθετήσουν τις προβλέψεις ενός μοντέλου αποκλειστικά και μόνο λόγω της υψηλής προβλεπτικής ακρίβειας, παραβλέποντας την κατανοητότητά του. Η σχετική σημασία της κατανοητότητας έναντι της ακρίβειας πρόβλεψης παραμένει υποκειμενική και εξαρτάται από τα συμφέροντα του χρήστη και το συγκεκριμένο πεδίο εφαρμογής.

Εκτός από την κρίσιμη ανάγκη για την κατανόηση των μοντέλων σε τομείς με υψηλά διακυβεύματα, διάφοροι θεμελιώδεις λόγοι υπογραμμίζουν τη σημασία της ερμηνευσιμότητας των μοντέλων μηχανικής μάθησης [15]:

Εμπιστοσύνη: Η ανάπτυξη ενός μοντέλου πρόβλεψης εξαρτάται καθοριστικά από την εμπιστοσύνη και την αποδοχή. Μόνο με την κατανόηση των δυνατών και αδύνατων σημείων του μοντέλου μπορούν οι χρήστες να αναπτύξουν την απαραίτητη εμπιστοσύνη ώστε να βασιστούν στις προβλέψεις του. Αυτή η εμπιστοσύνη είναι θεμελιώδης για την ευρεία υιοθέτηση και χρήση των μοντέλων μηχανικής μάθησης.

Αιτιότητα: Η ερμηνευσιμότητα, ιδίως μέσω μηχανισμών όπως το feature importance, προσδίδει μια αίσθηση αιτιότητας. Αυτό βοηθά το κοινό-στόχο να κατανοήσει τις υποκείμενες σχέσεις που οδηγούν τα αποτελέσματα του μοντέλου, γεφυρώνοντας το χάσμα μεταξύ πολύπλοκων υπολογισμών και κατανοητών αποτελεσμάτων.

Μεταφερσιμότητα: Για να μπορεί ένας άνθρωπος που λαμβάνει αποφάσεις να χρησιμοποιήσει αποτελεσματικά ένα μοντέλο πρόβλεψης με νέα, αθέατα δεδομένα, το μοντέλο πρέπει να παρέχει μια σαφή κατανόηση της μελλοντικής συμπεριφοράς. Ο χρήστης πρέπει να είναι βέβαιος ότι το μοντέλο γενικεύει καλά ή κατανοεί τα συγκεκριμένα πλαίσια στα οποία αποδίδει αξιόπιστα για να εμπιστευτεί το μοντέλο για τη λήψη αποφάσεων.

Πληροφόρηση: Πέρα από την εκπλήρωση των εκπαιδευτικών του στόχων, ένα μοντέλο πρέπει να ανταποκρίνεται αποτελεσματικά στις ανάγκες του πραγματικού κόσμου. Η κατανόηση του κατά πόσον ένα σύστημα εξυπηρετεί πραγματικά τον επιδιωκόμενο σκοπό του είναι ζωτικής σημασίας για την ανάπτυξή του, διασφαλίζοντας ότι λειτουργεί ως πρακτικό εργαλείο.

Δίκαιη και δεοντολογική λήψη αποφάσεων: Η κατανόηση των λόγων πίσω από μια απόφαση αποτελεί κοινωνική ανάγκη και αναμένεται να γίνει νομικό δικαίωμα για τους πολίτες της ΕΕ [31]. Αυτό το "δικαίωμα στην εξήγηση" υποχρεώνει τους υπεύθυνους λήψης αποφάσεων να παρουσιάζουν τα αποτελέσματά τους με σαφήνεια ώστε να τηρούν τα δεοντολογικά πρότυπα. Οποιοσδήποτε επηρεάζεται από μια αυτοματοποιημένη απόφαση μπορεί να ασκήσει αυτό το δικαίωμα για να λάβει εξηγήσεις.

Ευθύνη: Η ενσωμάτωση της δυνατότητας επεξήγησης στα μοντέλα μηχανικής μάθησης αφορά επίσης την ευθύνη. Ένα μοντέλο που μπορεί να δικαιολογήσει τις αποφάσεις του μπορεί να λογοδοτήσει για τις ενέργειές του. Αυτή η πτυχή είναι ιδιαίτερα σημαντική για την αντιμετώπιση πιθανών μεταβολών στα δεδομένα με την πάροδο του χρόνου, διασφαλίζοντας ότι τα μοντέλα παραμένουν υπεύθυνα και αξιόπιστα.

Προσαρμογή: Η κατανόηση του μοντέλου πρόβλεψης και των υποκείμενων παραγόντων του επιτρέπει στους ειδικούς του τομέα να συγκρίνουν τις προβλέψεις του μοντέλου με την υπάρχουσα γνώση του τομέα. Η ερμηνευσιμότητα είναι απαραίτητη για την προσαρμογή του μοντέλου πρόβλεψης με την ενσωμάτωση γνώσεων που αφορούν συγκεκριμένο τομέα. Σύμφωνα με τους Selvaraju κ.ά. (2016) [64], τα ερμηνεύσιμα μοντέλα πρόβλεψης μπορούν να βοηθήσουν τους ανθρώπους, ιδίως τους ειδικούς του τομέα, να λαμβάνουν καλύτερες αποφάσεις. Από αλγοριθμική άποψη, η ερμηνευσιμότητα επιτρέπει στους σχεδιαστές συστημάτων να βελτιώσουν το μοντέλο πρόβλεψης προσαρμόζοντας τις παραμέτρους. Επιπλέον, η ερμηνευσιμότητα βοηθά τους προγραμματιστές στον εντοπισμό και την αντιμετώπιση των τρόπων αποτυχίας.

Proxy Functionality: Όταν ένα μοντέλο είναι ερμηνεύσιμο, μπορεί να αξιολογηθεί σε μετρικές πέραν αυτών που εκπαιδεύτηκε άμεσα για να βελτιστοποιήσει, όπως η ασφάλεια, η δικαιοσύνη και η ιδιωτικότητα. Αυτή η πτυχή καθιστά την ερμηνευσιμότητα υποκατάστατο για την αξιολόγηση ευρύτερων κοινωνικών και λειτουργικών επιπτώσεων, ενισχύοντας τη συνολική χρησιμότητα και αποδοχή των συστημάτων μηχανικής μάθησης.

Η ενσωμάτωση αυτών των αρχών στην ανάπτυξη μοντέλων μηχανικής μάθησης όχι μόνο ενισχύει την πρακτική χρηστικότητά τους, αλλά και τα ευθυγραμμίζει με ευρύτερες κοινωνικές αξίες και προσδοκίες, καθιστώντας τα πιο ισχυρά, αξιόπιστα και ευθυγραμμισμένα με τις ανθρώπινες ανάγκες.

Προκλήσεις των ερμηνεύσιμων μοντέλων

Αν και η αναγκαιότητα της ερμηνευσιμότητας στη μηχανική μάθηση, όπως περιγράφηκε προηγουμένως, αναγνωρίζεται ευρέως, ιδίως για την ενίσχυση της εμπιστοσύνης και τη διασφάλιση της ηθικής συμμόρφωσης, η επίτευξή της εμπεριέχει προκλήσεις. Οι προκλήσεις αυτές προκύπτουν από μια ποικιλία τεχνικών, πρακτικών και ρυθμιστικών περιπλοκών που επηρεάζουν την ανάπτυξη και την εφαρμογή ερμηνεύσιμων μοντέλων. Παρατηρήθηκε από τον Pazzani [57] ότι λίγες εργασίες στοχεύουν πραγματικά στην εμπειρική αξιολόγηση της ερμηνευσιμότητας πέρα από την απλή χρήση του μεγέθους των αναπαραστάσεων που προκύπτουν. Παρά τα σαφή πλεονεκτήματα των διαφανών συστημάτων τεχνητής νοημοσύνης, η πορεία προς την πλήρως ερμηνεύσιμη τεχνητή νοημοσύνη είναι γεμάτη με εμπόδια που καλύπτουν το φάσμα από την πολυπλοκότητα του μοντέλου έως τις απαιτήσεις των ειδικών χρηστών.

Μοντέλα Μαύρου κουτιού

Τα σύγχρονα μοντέλα μηχανικής μάθησης, ιδίως τα βαθιά νευρωνικά δίκτυα, αποτελούνται από εκατομμύρια παραμέτρους και χαρακτηρίζονται από επίπεδα αφαίρεσης. Αυτά τα στρώματα, αν και είναι ευεργετικά για τη σύλληψη πολύπλοκων προτύπων σε μεγάλα σύνολα δεδομένων, δημιουργούν ένα σενάριο "μαύρου κουτιού" όπου η σχέση εισόδου-εξόδου είναι ουσιαστικά μη κατανοητή. Αυτή η πολυπλοκότητα αποτελεί σημαντικό εμπόδιο όχι μόνο για την κατανόηση του τι κάνει το μοντέλο αλλά και για τη διάγνωση σφαλμάτων ή προκαταλήψεων στις προβλέψεις του μοντέλου.

Τα μοντέλα μαύρου κουτιού συχνά επιτυγχάνουν ανώτερες επιδόσεις σε σύγκριση με απλούστερα, πιο ερμηνεύσιμα μοντέλα. Σε εργασίες που περιλαμβάνουν μεγάλα σύνολα δεδομένων με υψηλή διαστατικότητα και πολυπλοκότητα, όπως η αναγνώριση εικόνας και ομιλίας, τα μοντέλα μαύρου κουτιού μπορούν να εντοπίσουν λεπτά μοτίβα και συσχετίσεις που απλούστερα μοντέλα μπορεί να χάσουν [63]. Φυσικά, όταν εξετάζονται προβλήματα που έχουν δομημένα δεδομένα με σημαντικά χαρακτηριστικά, συχνά δεν υπάρχει σημαντική διαφορά στην απόδοση μεταξύ πιο σύνθετων ταξινομητών και πολύ απλούστερων ταξινομητών μετά από προεπεξεργασία.

Το γεγονός ότι πολλοί επιστήμονες δυσκολεύονται να κατασκευάσουν ερμηνεύσιμα μοντέλα μπορεί να τροφοδοτεί την πεποίθηση ότι τα μαύρα κουτιά έχουν την ικανότητα να αποκαλύπτουν κρυμμένα μοτίβα στα δεδομένα τα οποία ο χρήστης δεν γνώριζε προηγουμένως. Ένα διαφανές μοντέλο μπορεί να είναι σε θέση να αποκαλύψει αυτά τα ίδια μοτίβα.

Λόγω της πολύπλοκης δομής των μοντέλων μαύρου κουτιού έχει προκύψει η ανάγκη για τη δημιουργία μεταγενέστερων εξηγήσεων των μοντέλων. Οι post-hoc μέθοδοι ML παρέχουν εξηγήσεις που δεν είναι πιστές στο αρχικό μοντέλο. Πολλές από τις μεθόδους που ισχυρίζονται ότι παράγουν εξηγήσεις υπολογίζουν αντ' αυτού χρήσιμα συνοπτικά στατιστικά στοιχεία των προβλέψεων που γίνονται από το αρχικό μοντέλο. Αυτές οι post-hoc εξηγήσεις είναι ακατάλληλες σε συστήματα λήψης αποφάσεων με υψηλό διακύβευμα, καθώς μπορούν να χειραγωγηθούν για να αφηγηθούν μια διαφορετική αφήγηση από εκείνη του μαύρου κουτιού που εξηγούν [4]. Επίσης, συχνά δεν βγάζουν νόημα ή δεν παρέχουν αρκετές λεπτομέρειες για να κατανοήσουμε τι κάνει το μαύρο κουτί. Για να αντιμετωπιστεί το πρόβλημα της ερμηνευσιμότητας, πρόσφατες εργασίες μέσω της αξιοποίησης των γράφων γνώσης [51, 49, 50], προσφέρουν μια πιο δομημένη προσέγγιση στις εξηγήσεις των μαύρων κουτιών.

Γενικές Προκλήσεις στην Ερμηνευσιμότητα

Μια από τις σημαντικότερες προκλήσεις της ερμηνευσιμότητας στη σύγχρονη Μηχανική Μάθηση είναι η έλλειψη τυποποιημένων μετρικών [23]. Το πεδίο δεν διαθέτει ενιαίες μετρικές για τη μέτρηση της ερμηνευσιμότητας, γεγονός που περιπλέκει τις προσπάθειες βελτίωσης ή ακόμα και τον ορισμό του τι καθιστά ένα μοντέλο ερμηνεύσιμο.

Οι απαιτήσεις για την ερμηνευσιμότητα ποικίλλουν σε μεγάλο βαθμό ανάλογα με την εμπειρία του χρήστη και την περιοχή εφαρμογής. Σε γενικές γραμμές, η ερμηνευσιμότητα ορίζεται αναγκαστικά με τρόπο που αφορά συγκεκριμένο τομέα. Ένα σημαντικό κριτήριο για την επιλογή ενός μοντέλου από μια σειρά υποψήφιων μοντέλων με παρόμοιες επιδόσεις είναι να συνάδει με την προηγούμενη γνώση του τομέα. Για τους λόγους αυτούς, οι αλγόριθμοι επαγωγής κανόνων, οι οποίοι επιστρέφουν ένα σύνολο κανόνων "if-then", ή οι εκπαιδευτές

δέντρων αποφάσεων είναι συχνά η προτιμώμενη επιλογή, καθώς θα πρέπει να προσφέρουν το απαιτούμενο επίπεδο ερμηνευσιμότητας [40].

Ερμηνεύσιμα μοντέλα

Τα εγγενώς ερμηνεύσιμα μοντέλα ξεχωρίζουν για την ικανότητά τους να παρέχουν σαφείς, διαισθητικές γνώσεις σχετικά με τις διαδικασίες λήψης αποφάσεων, καθιστώντας τα απαραίτητα σε ευαίσθητους και υψηλού ρίσκου τομείς όπως η υγειονομική περίθαλψη, η χρηματοδότηση και η νομική συμμόρφωση. Αυτά τα μοντέλα έχουν σχεδιαστεί όχι μόνο για να εκτελούν εργασίες αλλά και για να εξηγούν τις αποφάσεις τους με τρόπο κατανοητό για τον άνθρωπο, γεφυρώνοντας το χάσμα μεταξύ των προηγμένων υπολογιστικών τεχνικών και των πρακτικών, καθημερινών αναγκών λήψης αποφάσεων.

Linear Models

Τα γραμμικά μοντέλα, όπως η γραμμική παλινδρόμηση και η λογιστική παλινδρόμηση, είναι γνωστά για την απλότητα και τη διαφάνειά τους. Η σχέση μεταξύ των χαρακτηριστικών εισόδου και του προβλεπόμενου αποτελέσματος εκφράζεται μέσω βαρών ή συντελεστών, οι οποίοι είναι άμεσα ερμηνεύσιμοι. Κάθε συντελεστής ποσοτικοποιεί τον αντίκτυπο ενός αντίστοιχου χαρακτηριστικού στην πρόβλεψη, προσφέροντας απλές πληροφορίες για τη συμπεριφορά του μοντέλου.

Rule-lists

Ο πιο συνηθισμένος τύπος κανόνων είναι αναμφίβολα οι προτασιακοί κανόνες if-then. Το τμήμα συνθήκης ενός προτασιακού κανόνα αποτελείται από έναν συνδυασμό συνθηκών στις μεταβλητές εισόδου. Ενώ το τμήμα συνθήκης μπορεί να περιέχει συνδέσεις, διαζεύξεις και αρνήσεις, οι περισσότεροι αλγόριθμοι θα επιστρέψουν κανόνες που περιέχουν μόνο συνδέσεις. Τα συστήματα βασισμένα σε κανόνες δημιουργούν σύνολα κανόνων αν-τότε για τη λήψη αποφάσεων, τα οποία είναι εύκολο να κατανοηθούν και να ελεγχθούν από τον άνθρωπο. Μπορούν να χρησιμοποιηθούν διάφορες μορφές για την αναπαράσταση προτασιακών κανόνων. Η πιο απλή προσέγγιση είναι η απλή καταγραφή των κανόνων, όπως στο ακόλουθο παράδειγμα:

IF (INCOME $>$ 400 AND GOAL=CAR) THEN ACCEPT
IF (INCOME $>$ 900 AND GOAL=HOUSE) THEN ACCEPT
DEFAULT:REJECT

Η εξαγωγή κανόνων μπορεί να πραγματοποιηθεί σε μοντέλα μαύρου κουτιού, όπως τα νευρωνικά δίκτυα και τα SVM, για την εξαγωγή ενός συνόλου κανόνων που προσεγγίζουν όσο το δυνατόν περισσότερο το μαύρο κουτί και ταυτόχρονα παρέχουν μια πιο κατανοητή αναπαράσταση στους χρήστες. Ωστόσο, προηγούμενες έρευνες σχετικά με τις τεχνικές εξαγωγής κανόνων [39] έδειξαν ότι ορισμένοι αλγόριθμοι επιστρέφουν μοντέλα που προσεγγίζουν στενά το υποκείμενο μοντέλο του μαύρου κουτιού, αλλά με κόστος την αυξημένη πολυπλοκότητα.

Οι λίστες κανόνων θεωρούνται επεξηγήσιμες επειδή παρέχουν μια σαφή, διαδοχική ανάλυση των κριτηρίων λήψης αποφάσεων σε μια μορφή που είναι απλή και εύκολα κατανοητή από τους ανθρώπους. Κάθε κανόνας σε μια λίστα αποτελείται από μια δήλωση "if-then" που δηλώνει ρητά μια συνθήκη και το αποτέλεσμα αν η συνθήκη αυτή ικανοποιείται. Αυτή η μορφή μιμείται τις λογικές ανθρώπινες διαδικασίες σκέψης, όπως η βήμα-προς-βήμα αντιμετώπιση προβλημάτων ή οι διαγνωστικές διαδικασίες, καθιστώντας τη διαισθητική παρακολούθηση. Αυτή η διαφάνεια επιτρέπει στους χρήστες όχι μόνο να βλέπουν ποιες συνθήκες οδηγούν σε συγκεκριμένες αποφάσεις, αλλά και να επαληθεύουν και να εμπιστεύονται το σκεπτικό πίσω από κάθε απόφαση.

Παρά την αρχική τους απήχηση, τα μοντέλα που βασίζονται σε κανόνες συχνά υποφέρουν από διάφορα αξιοσημείωτα μειονεκτήματα που μπορούν να μειώσουν τη χρησιμότητά τους όσον αφορά την επεξηγηματικότητα. Πρώτον, ο μεγάλος αριθμός κανόνων που παράγονται για πολύπλοκα μοντέλα μπορεί να είναι συντριπτικός, μειώνοντας τη σαφήνεια και δυσκολεύοντας τους χρήστες να κατανοήσουν την υποκείμενη λογική. Επιπλέον, η ανελαστικότητα των συστημάτων που βασίζονται σε κανόνες μπορεί να οδηγήσει σε εύθραυστη συμπεριφορά σε απρόβλεπτα σενάρια που δεν καλύπτονται από τους υπάρχοντες κανόνες, περιορίζοντας την προσαρμοστικότητα και την αξιοπιστία τους. Έτσι η συντήρηση μεγάλων συνόλων κανόνων καθίσταται ανέφικτη, καθώς η ενημέρωση ενός κανόνα μπορεί να απαιτεί αλυσιδωτές αλλαγές σε όλο το σύστημα.

Πίνακες αποφάσεων

Άλλες πιο γραφικές αναπαραστάσεις που χρησιμοποιούνται συχνά για την απεικόνιση της λογικής υπό όρους είναι οι πίνακες αποφάσεων και τα δέντρα αποφάσεων. Ένας πίνακας αποφάσεων [70] είναι μια αναπαράσταση σε μορφή πίνακα που αποτελείται από τέσσερα τεταρτημόρια που χωρίζονται από οριζόντιες και κάθετες διπλές γραμμές (βλ. Σχήμα 0.2.1). Η οριζόντια γραμμή χωρίζει τον πίνακα σε ένα τμήμα συνθήκης (πάνω) και ένα τμήμα δράσης (κάτω), ενώ η κάθετη γραμμή διαχωρίζει τα θέματα (αριστερά) από τις καταχωρήσεις (δεξιά). Κάθε στήλη στο μέρος της καταχώρησης αντιστοιχεί σε έναν κανόνα, που συνδυάζει καταστάσεις συνθηκών με την κατάλληλη ενέργεια ή ενέργειες που πρέπει να γίνουν. Ένα σύμβολο παύλας (-) στο μέρος της συνθήκης του πίνακα υποδηλώνει ότι η τιμή είναι άσχετη με τη συγκεκριμένη συνθήκη και ένα "X" στο μέρος της δράσης αντιπροσωπεύει το σωστό συμπέρασμα που πρέπει να βγει αν ικανοποιούνται οι συνθήκες που οδηγούν στη συγκεκριμένη στήλη. Το γεγονός ότι κάθε δυνατός συνδυασμός καταστάσεων συνθήκης εμφανίζεται μόνο σε μία ακριβώς στήλη είναι το βασικό πλεονέκτημα των πινάκων ενός χτυπήματος.

(a) Single-hit table

INCOME	< 1000		≥ 1000
AGE	< 25	≥ 25	-
ACCEPT	X		
REJECT		X	X

(b) Multiple-hit table

INCOME	≥ 1000	-	< 1000
AGE	-	< 25	< 25
ACCEPT			X
REJECT	X	X	

Figure 0.2.1: Παραδείγματα πινάκων αποφάσεων

Οι πίνακες αποφάσεων είναι ιδιαίτερα επεξηγήσιμοι λόγω της δομημένης μορφής τους, η οποία οργανώνει και παρουσιάζει τις πληροφορίες με σαφήνεια, παραθέτοντας τις πιθανές συνθήκες και τις αντίστοιχες ενέργειες σε μορφή πίνακα. Κάθε γραμμή σε έναν πίνακα αποφάσεων αντιπροσωπεύει ένα συγκεκριμένο σενάριο με καθορισμένες συνθήκες και κάθε στήλη συσχετίζεται με μια απόφαση ή ένα αποτέλεσμα με βάση αυτές τις συνθήκες. Αυτή η σαφής, οπτική μέθοδος απεικόνισης κανόνων και αποφάσεων όχι μόνο βοηθά στη γρήγορη κατανόηση και επικύρωση της διαδικασίας λήψης αποφάσεων, αλλά και στον εντοπισμό ασυνέπειας ή επικαλύψεων στους κανόνες.

Παρά τη δομημένη σαφήνειά τους, οι πίνακες αποφάσεων μπορεί να αντιμετωπίζουν περιορισμούς όσον αφορά την ερμηνευσιμότητα, ιδίως όσο αυξάνεται η πολυπλοκότητα της διαδικασίας λήψης αποφάσεων. Όταν οι πίνακες αποφάσεων περιέχουν μεγάλο αριθμό συνθηκών ή όταν οι συνθήκες αυτές αλληλεπιδρούν με πολύπλοκους τρόπους, ο πίνακας μπορεί να γίνει υπερβολικά μεγάλος και μη διαχειρίσιμος. Επιπλέον, οι πίνακες αποφάσεων μπορεί να μην αποτυπώνουν αποτελεσματικά την πολυπλοκότητα των σεναρίων που περιλαμβάνουν συνεχή δεδομένα ή απαιτούν πιο λεπτομερή λήψη αποφάσεων, καθώς συνήθως λειτουργούν καλύτερα με διακριτές, σαφώς καθορισμένες κατηγορίες.

0.2.2 Δένδρα Αποφάσεων

Το δέντρο απόφασης είναι ένας αλγόριθμος μάθησης με επίβλεψη, ο οποίος χρησιμοποιείται τόσο για εργασίες ταξινόμησης όσο και για εργασίες παλινδρόμησης. Έχει μια ιεραρχική, δενδρική δομή, η οποία αποτελείται από έναν κόμβο ρίζας, κλάδους, εσωτερικούς κόμβους και κόμβους φύλλων. Όπως φαίνεται στο Σχήμα 0.2.2 ένα δέν-

τρο απόφασης ξεκινά με έναν κόμβο ρίζας, ο οποίος δεν έχει εισερχόμενους κλάδους. Οι εξερχόμενοι κλάδοι από τον κόμβο ρίζας τροφοδοτούν στη συνέχεια τους εσωτερικούς κόμβους, γνωστούς και ως κόμβους απόφασης. Με βάση τα διαθέσιμα χαρακτηριστικά, και οι δύο τύποι κόμβων διεξάγουν αξιολογήσεις για να σχηματίσουν ομοιογενή υποσύνολα, τα οποία συμβολίζονται με κόμβους φύλλων ή τερματικούς κόμβους. Οι κόμβοι φύλλων αντιπροσωπεύουν όλα τα πιθανά αποτελέσματα εντός του συνόλου δεδομένων. Τα δέντρα αποφάσεων αποτελούν ακρογωνιαίο λίθο της μηχανικής μάθησης, παρέχοντας τη βάση τόσο για βασικά όσο και για προηγμένα μοντέλα πρόβλεψης. Είναι γνωστά για την απλότητα και την αποτελεσματικότητά τους, γεγονός που τα καθιστά έναν από τους πιο δημοφιλείς αλγορίθμους στην κοινότητα της επιστήμης των δεδομένων. Τα δέντρα αποφάσεων ανακαλύπτουν αυτόματα όρια αποφάσεων από τα δεδομένα. Τα όρια αυτά μπορεί να είναι τόσο απλά όσο μια απλή διάσπαση σε μονοδιάστατα δεδομένα ή τόσο πολύπλοκα όσο πολλαπλά επίπεδα αποφάσεων σε χώρους υψηλών διαστάσεων. Αλγόριθμοι όπως οι ID3, C4.5 και CART διαφέρουν κυρίως στα κριτήρια επιλογής του χαρακτηριστικού και στη συνθήκη διαχωρισμού των δεδομένων σε κάθε κόμβο, γεγονός που επηρεάζει σημαντικά την απόδοσή τους και την τελική δομή του δέντρου. Η διαισθητική διάταξη των δέντρων αποφάσεων όχι μόνο βοηθά στην αποτελεσματική εκτέλεση εργασιών ανάλυσης δεδομένων αλλά και στην οπτική αναπαράσταση της διαδικασίας λήψης αποφάσεων, καθιστώντας έτσι τα αποτελέσματα εύκολα κατανοητά και ερμηνεύσιμα. Αυτή η διαφάνεια είναι ζωτικής σημασίας για εφαρμογές σε τομείς όπως η υγειονομική περίθαλψη και σε κάθε τομέα όπου είναι απαραίτητη η λήψη τεκμηριωμένων και δικαιολογημένων αποφάσεων. Καθώς εμβαθύνουμε σε αυτό το κεφάλαιο, θα διερευνήσουμε τις εγγενείς ιδιότητες που συμβάλλουν στην ερμηνευσιμότητα των δέντρων αποφάσεων, θα συζητήσουμε τις πρακτικές εφαρμογές τους και θα αντιμετωπίσουμε ορισμένες από τις προκλήσεις που αντιμετωπίζουν όσον αφορά την επεκτασιμότητα και την πολυπλοκότητα.

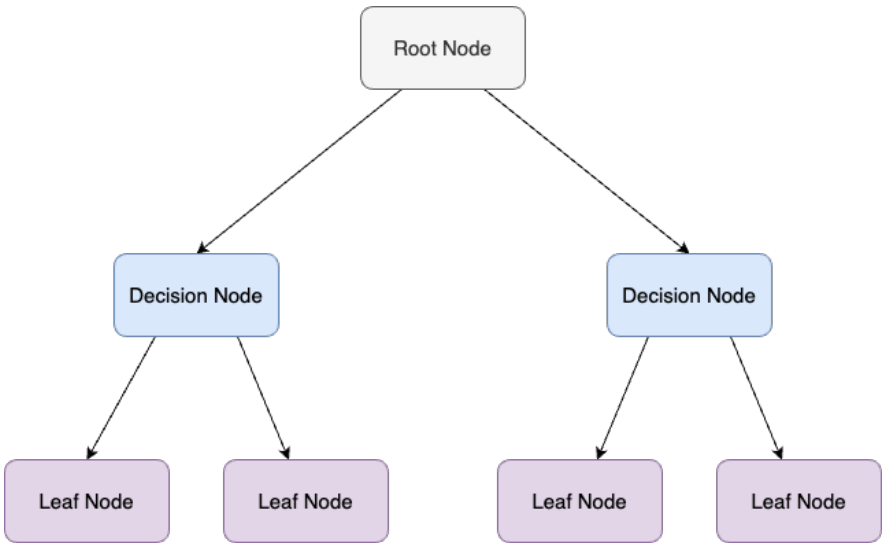

Figure 0.2.2: Δέντρο αποφάσεων

Γιατί τα δέντρα αποφάσεων είναι ερμηνεύσιμα;

Τα δέντρα αποφάσεων είναι εγγενώς ερμηνεύσιμα λόγω της διαδικασίας λήψης αποφάσεων, η οποία αντικατοπτρίζει στενά τα ανθρώπινα πρότυπα συλλογισμού (τα οποία θα συζητηθούν στο κεφάλαιο 2.3). Στο επίκεντρο της ερμηνευσιμότητας ενός δέντρου αποφάσεων βρίσκεται η δομική του αναπαράσταση, όπου οι αποφάσεις λαμβάνονται μέσω μιας σειράς απλών ερωτήσεων και απαντήσεων που διαχωρίζουν τα δεδομένα σταδιακά. Κάθε κόμβος στο δέντρο αντιπροσωπεύει μια συγκεκριμένη ερώτηση ή μια συνθήκη για ένα συγκεκριμένο χαρακτηριστικό, και οι διακλαδώσεις προς τα παιδιά αντιπροσωπεύουν τις πιθανές απαντήσεις σε αυτή την ερώτηση, οδηγώντας τελικά σε κόμβους φύλλων που παρέχουν τα αποτελέσματα ή τις προβλέψεις. Αυτή η δενδρική δομή επιτρέπει σε οποιονδήποτε εξετάζει το μοντέλο να δει ακριβώς πώς οι εισροές μετατρέπονται σε εκροές, ακολουθώντας τις διαδρομές από τη ρίζα στα φύλλα.

Οπτική Διαφάνεια

Μια από τις βασικές πτυχές των δέντρων αποφάσεων που ενισχύει την ερμηνευσιμότητά τους είναι ο οπτικός τους χαρακτήρας. Τα δέντρα αποφάσεων μπορούν να αναπαρασταθούν γραφικά [29], επιτρέποντας στους χρήστες

να ανιχνεύσουν οπτικά τα μονοπάτια αποφάσεων. Αυτή η οπτικοποίηση βοηθά τους χρήστες όχι μόνο να κατανοήσουν τα κριτήρια που χρησιμοποιούνται σε κάθε σημείο απόφασης, αλλά και να αξιολογήσουν τη λογική και τη δικαιοσύνη αυτών των κριτηρίων.

Επιλογή και σημασία χαρακτηριστικών

Ένα δέντρο αποφάσεων συνήθως περιλαμβάνει μόνο ένα υποσύνολο και όχι ολόκληρο το σύνολο των χαρακτηριστικών. Αυτή η επιλεκτική συμπερίληψη συμβάλλει στον εξορθολογισμό της διαδικασίας λήψης αποφάσεων, εστιάζοντας την προσοχή στα πιο σημαντικά χαρακτηριστικά, μειώνοντας έτσι την πολυπλοκότητα και διευκολύνοντας τη σαφέστερη κατανόηση των υποκείμενων μοτίβων των δεδομένων. Επιπλέον, η ιεραρχική διάταξη του δέντρου παρέχει πληροφορίες σχετικά με τη σχετική σημασία των διαφόρων χαρακτηριστικών.

Rule Extraction

Κάθε διαδρομή από τη ρίζα του δέντρου σε ένα φύλλο μπορεί να μεταφραστεί απευθείας σε μια μορφή βασισμένη σε κανόνες: μια δήλωση αν-τότε που εξηγεί σαφώς γιατί λήφθηκε μια συγκεκριμένη απόφαση ή πρόβλεψη. Αυτό το χαρακτηριστικό είναι ιδιαίτερα επωφελές για την παροχή εξηγήσεων για συγκεκριμένες αποφάσεις, κάτι που αποτελεί απαίτηση σε πολλούς ρυθμιζόμενους κλάδους, όπως η χρηματοοικονομική και η υγειονομική περίθαλψη. Οι τοπικές αποφάσεις μπορούν να ερμηνευθούν εύκολα, δεδομένου ότι κάθε φύλλο μεταφράζεται σε μια σαφή σύνδεση χαρακτηριστικών [60].

0.2.3 Σύγκριση ερμηνεύσιμων μοντέλων

Rules - Decision Trees

Τα μοντέλα που βασίζονται σε κανόνες είναι γνωστά για την ερμηνευσιμότητά τους. Οι κανόνες δεν καταγράφουν ασήμαντες ρήτρες, ενώ τα δέντρα αποφάσεων μπορούν επίσης να έχουν ασήμαντες διακλαδώσεις. Αυτό συμβαίνει επειδή οι ταξινομητές που βασίζονται σε κανόνες επιλέγουν γενικά ένα χαρακτηριστικό-τιμή κατά την επέκταση ενός κανόνα, ενώ οι αλγόριθμοι δέντρων απόφασης συνήθως επιλέγουν ένα χαρακτηριστικό κατά την επέκταση του δέντρου [32]. Τα δέντρα αποφάσεων συχνά επαινούνται για την ανώτερη κατανοητότητά τους σε σύγκριση με τις λίστες κανόνων, όπως τονίζεται σε μια μελέτη του Allahyari and Lavesson (2011) [65].

Πίνακες αποφάσεων - Decision Trees

Τα δέντρα αποφάσεων χρησιμοποιούν μια ιεραρχική δομή με κόμβους και διακλαδώσεις για την αναπαράσταση των αποφάσεων και των πιθανών αποτελεσμάτων τους, παρέχοντας μια διαισθητική οπτική αναπαράσταση που αντικατοπτρίζει την ανθρώπινη λογική αποφάσεων. Ωστόσο, τα δέντρα αποφάσεων μπορεί να γίνουν πολύπλοκα και επιρρεπή σε υπερβολική προσαρμογή, ιδίως με μεγάλα σύνολα δεδομένων, και μπορεί να είναι ασταθή με μικρές αλλαγές στα δεδομένα. Αντίθετα, οι πίνακες αποφάσεων προσφέρουν μια μορφή πίνακα που απαριθμεί όλες τις πιθανές συνθήκες και τις αντίστοιχες ενέργειες, εξασφαλίζοντας ολοκληρωμένη κάλυψη και συνέπεια στη λήψη αποφάσεων. Ενώ είναι απλοί και σαφείς για απλά σενάρια, οι πίνακες αποφάσεων μπορεί να γίνουν δυσκίνητοι και δύσκολα ερμηνεύσιμοι καθώς αυξάνεται ο αριθμός των συνθηκών. Δεν διαθέτουν επίσης την οπτική ελκυστικότητα των δέντρων αποφάσεων. Τελικά, η επιλογή μεταξύ των δέντρων αποφάσεων και των πινάκων αποφάσεων εξαρτάται από τη συγκεκριμένη περίπτωση χρήσης, την πολυπλοκότητα της διαδικασίας λήψης αποφάσεων και από το αν προτιμάται η οπτική ή η αναπαράσταση σε πίνακες. Γενικά, οι Subramanian et al. [67] διερεύνησαν περαιτέρω τη διαφορά μεταξύ των δύο γραφικών αναπαραστάσεων (δέντρα αποφάσεων και πίνακες αποφάσεων) και κατέληξαν στο συμπέρασμα ότι τα δέντρα αποφάσεων αποδίδουν σημαντικά καλύτερα από τους πίνακες αποφάσεων.

0.2.4 Μειονεκτήματα των δέντρων αποφάσεων

Overfitting

Τα δέντρα αποφάσεων είναι επιρρεπή σε υπερπροσαρμογή, ιδίως όταν τους επιτρέπεται να αναπτυχθούν σε βάθος χωρίς περιορισμούς. Η υπερπροσαρμογή συμβαίνει όταν ένα δέντρο μοντελοποιεί το θόρυβο στα δεδομένα εκπαίδευσης και όχι τα πραγματικά υποκείμενα μοτίβα. Αυτό έχει ως αποτέλεσμα υπερβολικά πολύπλοκα δέντρα που είναι δύσκολο να ερμηνευθούν και δεν αποδίδουν καλά όταν παρουσιάζονται νέα, αθέατα δεδομένα.

Instability

Τα δέντρα αποφάσεων μπορεί να παρουσιάζουν υψηλό επίπεδο διακύμανσης στη δομή τους με μικρές αλλαγές στα δεδομένα εισόδου. Μια μικρή μεταβολή στο σύνολο δεδομένων, όπως η προσθήκη ή η αφαίρεση μερικών σημείων δεδομένων, μπορεί να οδηγήσει στη δημιουργία ενός εντελώς διαφορετικού δέντρου. Αυτή η αστάθεια μπορεί να προκαλέσει σύγχυση στους χρήστες, καθώς το σκεπτικό των αποφάσεων μπορεί να αλλάξει απρόβλεπτα, υπονομεύοντας την εμπιστοσύνη στην αξιοπιστία και τη συνέπεια του μοντέλου.

Βάθος και πολυπλοκότητα

Καθώς τα δέντρα αποφάσεων αντιμετωπίζουν πιο σύνθετα σύνολα δεδομένων, τείνουν να γίνονται βαθύτερα για να καταγράφουν πιο λεπτομερή πρότυπα. Ωστόσο, τα βαθύτερα δέντρα μπορεί να είναι δύσκολο να ερμηνευθούν. Η αύξηση του αριθμού των αποφάσεων (βάθος) και η πολυπλοκότητα των διακλαδώσεων μπορεί να καταβάλει τους χρήστες, καθιστώντας δυσκολότερη την ανίχνευση της λογικής από τη ρίζα στα φύλλα.

Local-Global Explanations

Τα δέντρα αποφάσεων είναι συχνά καλύτερα στο να παρέχουν τοπικές εξηγήσεις (εξηγώντας μια συγκεκριμένη πρόβλεψη) παρά σφαιρικές εξηγήσεις (κατανοώντας τη συνολική συμπεριφορά του μοντέλου). Αυτό μπορεί να αποτελέσει περιορισμό όταν προσπαθούμε να κατανοήσουμε τη γενική διαδικασία λήψης αποφάσεων του μοντέλου.

Bias Toward Certain Splits

Οι αλγόριθμοι δέντρων αποφάσεων όπως οι ID3 και C4.5 μπορεί να παρουσιάσουν μια προκατάληψη προς τα χαρακτηριστικά με περισσότερες κατηγορίες. Τείνουν να ευνοούν αυτά τα χαρακτηριστικά για διαχωρισμούς στην κορυφή του δέντρου, επειδή μπορούν να οδηγήσουν σε υψηλότερες μετρικές κέρδους πληροφορίας, ακόμη και αν δεν είναι οι πιο προγνωστικές για το αποτέλεσμα. Τα μοντέλα ενός δέντρου, όπως το CART [13], είναι πλήρως ερμηνεύσιμα, καθώς η λογική πρόβλεψής τους μπορεί εύκολα να ακολουθηθεί παρατηρώντας τις διασπάσεις στο τελικό δέντρο απόφασης. Ωστόσο, ο CART χρησιμοποιεί μια άπληστη ευρετική μέθοδο η οποία έχει ορισμένα μειονεκτήματα. Πρώτα απ' όλα, αυτό μπορεί να οδηγήσει σε δέντρα που απέχουν πολύ από το βέλτιστο. Αυτοί οι περιορισμοί προκαλούν ανησυχία κατά την ερμηνεία της μεταβλητής σπουδαιότητας, καθώς τα επιλεγμένα χαρακτηριστικά μπορεί να μεροληπτούν προς εκείνα με μεγαλύτερο αριθμό μοναδικών τιμών, και ο άπληστος αλγόριθμος μπορεί να οδηγήσει στη χρήση λανθασμένων χαρακτηριστικών στις διασπάσεις κοντά στη ρίζα του δέντρου, οι οποίες είναι συνήθως εκείνες που λαμβάνουν τη μεγαλύτερη σπουδαιότητα [24].

Χειρισμός κατηγορηματικών μεταβλητών

Τα δέντρα αποφάσεων χειρίζονται κατηγορηματικές μεταβλητές μετατρέποντάς τες σε μορφή που μπορεί να χρησιμοποιηθεί αποτελεσματικά στη διαδικασία δημιουργίας του δέντρου. Συχνά, οι κατηγορηματικές μεταβλητές μετατρέπονται σε δυαδικές (ψευδομεταβλητές), μια διαδικασία γνωστή ως κωδικοποίηση ενός σημείου (one-hot encoding). Ενώ η μέθοδος αυτή επιτρέπει στα δέντρα αποφάσεων να ενσωματώνουν κατηγορηματικά δεδομένα χωρίς να υποθέτουν μια συγκεκριμένη σχέση, μπορεί να αυξήσει σημαντικά τη διάσταση των δεδομένων, ιδίως με χαρακτηριστικά που έχουν μεγάλο αριθμό κατηγοριών. Αυτή η επέκταση μπορεί να οδηγήσει σε μεγαλύτερα και πιο πολύπλοκα δέντρα. Ο χειρισμός κατηγορηματικών χαρακτηριστικών με αυτόν τον δυαδικό τρόπο μπορεί επομένως να συμβάλει στην πολυπλοκότητα και το βάθος του δέντρου, επηρεάζοντας ενδεχομένως τόσο την απόδοση όσο και την ερμηνευσιμότητα του μοντέλου.

0.2.5 Ανθρώπινες επεξηγήσεις και λήψη αποφάσεων

Η κατανόηση των ανθρώπινων διαδικασιών λήψης αποφάσεων και των γνωστικών περιορισμών είναι ζωτικής σημασίας για το σχεδιασμό και την ανάπτυξη επεξηγήσιμων δομών δέντρων αποφάσεων [52], [55]. Η ανθρώπινη λήψη αποφάσεων, ενώ συχνά γίνεται αντιληπτή ως λογική και ορθολογική, υπόκειται επίσης σε πλήθος γνωστικών προκαταλήψεων, ευρετικών μεθόδων και περιορισμών. Με την εξέταση της βιβλιογραφίας της ψυχολογίας, της νευροεπιστήμης και τη γνωστικής επιστήμης, μπορούμε να αποκαλύψουμε τα όρια της ανθρώπινης κατανόησης, να αποσαφηνίσουμε τις γνωστικές διεργασίες που παίζουν ρόλο και να εντοπίσουμε πιθανές παγίδες που μπορεί να συναντήσουν οι αλγόριθμοι δέντρων αποφάσεων όταν προσομοιώνουν την ανθρώπινη λήψη αποφάσεων. Σε αυτό το κεφάλαιο στοχεύουμε να γεφυρώσουμε το χάσμα μεταξύ της θεωρητικής κατανόησης της ανθρώπινης λήψης αποφάσεων και της πρακτικής εφαρμογής των μοντέλων δέντρων αποφάσεων σε σενάρια του πραγματικού κόσμου. Τελικά, στόχος μας είναι να εξάγουμε πολύτιμες γνώσεις από

την ανθρώπινη νόηση για να καθοδηγήσουμε την ανάπτυξη πιο ερμηνεύσιμων, διαφανών και αποτελεσματικών αλγορίθμων δέντρων αποφάσεων.

Ανθρώπινη λήψη αποφάσεων

Από μια γενική γνωστική προοπτική, η λήψη αποφάσεων είναι η διαδικασία επιλογής μιας επιλογής ή μιας πορείας δράσης από ένα σύνολο εναλλακτικών λύσεων. Τα περισσότερα ψυχολογικά μοντέλα περιγράφουν την ανθρώπινη διαδικασία λήψης αποφάσεων ως τμηματικές σταδιακές διαδικασίες που περιλαμβάνουν βήματα με επίκεντρο τη συλλογή πληροφοριών, την εκτίμηση της πιθανότητας, τη διαβούλευση και την επιλογή απόφασης. Οι δύο θεμελιώδεις γνωστικές διεργασίες που διέπουν την ανθρώπινη συλλογιστική και λήψη αποφάσεων είναι η προσοχή και η μνήμη.

Προσοχή και working memory

Η προσοχή είναι ο τρόπος με τον οποίο ο εγκέφαλος, συχνά συνειδητά αν και μερικές φορές αυτόματα, επιλέγει πληροφορίες για γνωστική επεξεργασία. Η ανθρώπινη μνήμη είναι η ικανότητα κωδικοποίησης, αποθήκευσης και ανάκτησης πληροφοριών. Η προσοχή και η μνήμη λειτουργούν ως σημαντικοί περιορισμοί στην επεξεργασία πληροφοριών, οπότε η κατανόηση του τρόπου με τον οποίο αυτές οι διαδικασίες επηρεάζουν τα συστατικά στοιχεία της συλλογιστικής και της λήψης αποφάσεων είναι ζωτικής σημασίας για την ανάπτυξη τεχνολογιών υποστήριξης αποφάσεων, όπως τα δέντρα αποφάσεων. Η working memory αναφέρεται σε μια ποικιλία διαδικασιών που χρησιμοποιούνται για τη διατήρηση των νοητικών πληροφοριών σε μια ιδιαίτερα προσβάσιμη κατάσταση. Η working memory θα πρέπει να θεωρείται ως μια προσωρινή αποθήκη όπου εκτελούνται συνειδητοί, επίπονοι (που απαιτούν προσοχή) εσωτερικοί υπολογισμοί [6]. Η αποθήκευση στη μνήμη εργασίας παίζει πολύ σημαντικό ρόλο σε αυτές τις εργασίες. Όπως είδαμε από τον Miller [54], οι περισσότεροι άνθρωποι είναι σε θέση να ανακαλέσουν μια λίστα με όχι περισσότερα από 7 τυχαία διατεταγμένα νοηματικά στοιχεία. Το όριο ανάκλησης είναι σημαντικό καθώς μετράει τη μνήμη εργασίας. Σίγουρα η ικανότητα της εργαζόμενης μνήμης ποικίλλει μεταξύ των ανθρώπων και μεταβάλλεται κατά τη διάρκεια της ζωής. [20]. Πιο πρόσφατες μελέτες ([36], [19]) σε πολλούς τύπους υλικών και εργασιών δείχνουν ότι υπάρχει μια κεντρική ικανότητα μνήμης εργασίας που περιορίζεται σε 3-5 κομμάτια πληροφοριών στους ενήλικες. Αυτά τα κομμάτια πληροφοριών μπορεί να είναι μεμονωμένες λέξεις ή ζεύγη λέξεων.

Τα δέντρα αποφάσεων είναι ένας τύπος μοντέλου μηχανικής μάθησης που χρησιμοποιούν οι άνθρωποι για να λαμβάνουν αποφάσεις επεξεργαζόμενοι νοερά μια ακολουθία κανόνων "if-then". Δεδομένου ότι οι άνθρωποι έχουν αυτό το όριο στον αριθμό των πληροφοριών που μπορούν να επεξεργαστούν αποτελεσματικά, τα δέντρα αποφάσεων γίνονται λιγότερο ερμηνεύσιμα καθώς μεγαλώνουν και γίνονται πιο πολύπλοκα, οδηγώντας σε υψηλότερο γνωστικό φορτίο για τους ανθρώπους που προσπαθούν να τα κατανοήσουν.

Human Reasoning Οι διαδικασίες λήψης αποφάσεων διευκολύνονται με τη χρήση μιας ποικιλίας διαφορετικών τεχνικών συλλογισμού. Η αναλογική συλλογιστική, για παράδειγμα, περιλαμβάνει την εξαγωγή συμπερασμάτων για νέες λύσεις μέσω παραλληλισμών με γνωστές. Οι αποφάσεις λαμβάνονται τελικά μέσω συλλογισμού σχετικά με το πρόβλημα και τα πιθανά αποτελέσματά του, συχνά λαμβάνοντας υπόψη παρελθοντικά γεγονότα λήψης αποφάσεων.

Οι συστατικές διαδικασίες του αναλογικού συλλογισμού περιλαμβάνουν τις ακόλουθες σειριακές διαδικασίες [66]:

1. **Encoding**: Μετάφραση ερεθισμάτων σε εσωτερικές (νοητικές) αναπαραστάσεις
2. **Inference**: Προσδιορισμός της σχέσης μεταξύ προβλημάτων
3. **Mapping**: Καθορισμός αντιστοιχιών μεταξύ νέων και παλαιών στοιχείων
4. **Application**: Εκτέλεση της διαδικασίας λήψης αποφάσεων
5. **Response**: Ένδειξη του αποτελέσματος της διαδικασίας συλλογισμού

Δεδομένου ότι τα βήματα σε αυτή τη διαδικασία συλλογισμού εκτυλίσσονται διαδοχικά, η χρονική διάταξη και ο χρονισμός της υποστήριξης της απόφασης είναι κρίσιμα για την ενίσχυση της λήψης αποφάσεων με βάση

τη χρονική ευαισθησία. Περαιτέρω ανάλυση αποκαλύπτει ότι οι χρόνοι αντίδρασης και τα ποσοστά σφάλματος αυξάνονται με πιο σύνθετες κωδικοποιήσεις. Ανεξάρτητα από τα ερεθίσματα, το βήμα της κωδικοποίησης αποτελεί το σημαντικότερο τμήμα της διαδικασίας συλλογισμού, αντιπροσωπεύοντας περίπου το 45 % του συνολικού χρόνου συλλογισμού. Για παράδειγμα, η κωδικοποίηση λέξεων διαρκεί περισσότερο από την κωδικοποίηση σχηματικών εικόνων, γεγονός που υποδηλώνει ότι η μείωση του περιεχομένου κειμένου στις οθόνες θα μπορούσε να επιταχύνει τη λήψη αποφάσεων. Επομένως, θα πρέπει να δίνεται προτεραιότητα στη διευκόλυνση της ταχύτερης κωδικοποίησης, ενδεχομένως μέσω της χρήσης πιο διαισθητικών συμβόλων και εργασιών. Παρόμοια με την ανθρώπινη λήψη αποφάσεων, τα δέντρα αποφάσεων βασίζονται σε μια σειρά απλών κανόνων που αναπαρίστανται από κόμβους, καθιστώντας εύκολο να κατανοηθεί ο τρόπος με τον οποίο λαμβάνονται οι αποφάσεις. Η σαφής και κατανοητή κωδικοποίηση των χαρακτηριστικών ενισχύει την ερμηνευσιμότητα των δέντρων αποφάσεων και διευκολύνει την εξήγηση των προβλέψεων του μοντέλου στους ενδιαφερόμενους.

Ανθρώπινη αντίληψη

Δεδομένου ότι το μέρος της κωδικοποίησης του συλλογισμού είναι το πιο σημαντικό και χρονοβόρο μέρος της λήψης αποφάσεων, είναι ζωτικής σημασίας η βελτιστοποίηση της αναπαράστασης των μοντέλων με βάση την ανθρώπινη αντίληψη. Οι άνθρωποι έχουν μια αξιοσημείωτη ικανότητα να εκτελούν μια μεγάλη ποικιλία φυσικών και νοητικών εργασιών χωρίς μετρήσεις και χωρίς υπολογισμούς. Κατά την εκτέλεση τέτοιων καθηκόντων οι άνθρωποι βασίζουν τις όποιες αποφάσεις πρέπει να λάβουν σε πληροφορίες που, ως επί το πλείστον, είναι αντίληψη και όχι μέτρηση [75]. Μια ουσιαστική διαφορά μεταξύ μετρήσεων και αντιλήψεων είναι ότι γενικά οι μετρήσεις είναι σαφείς, ενώ οι αντιλήψεις είναι ασαφείς (Σχήμα 0.2.3).

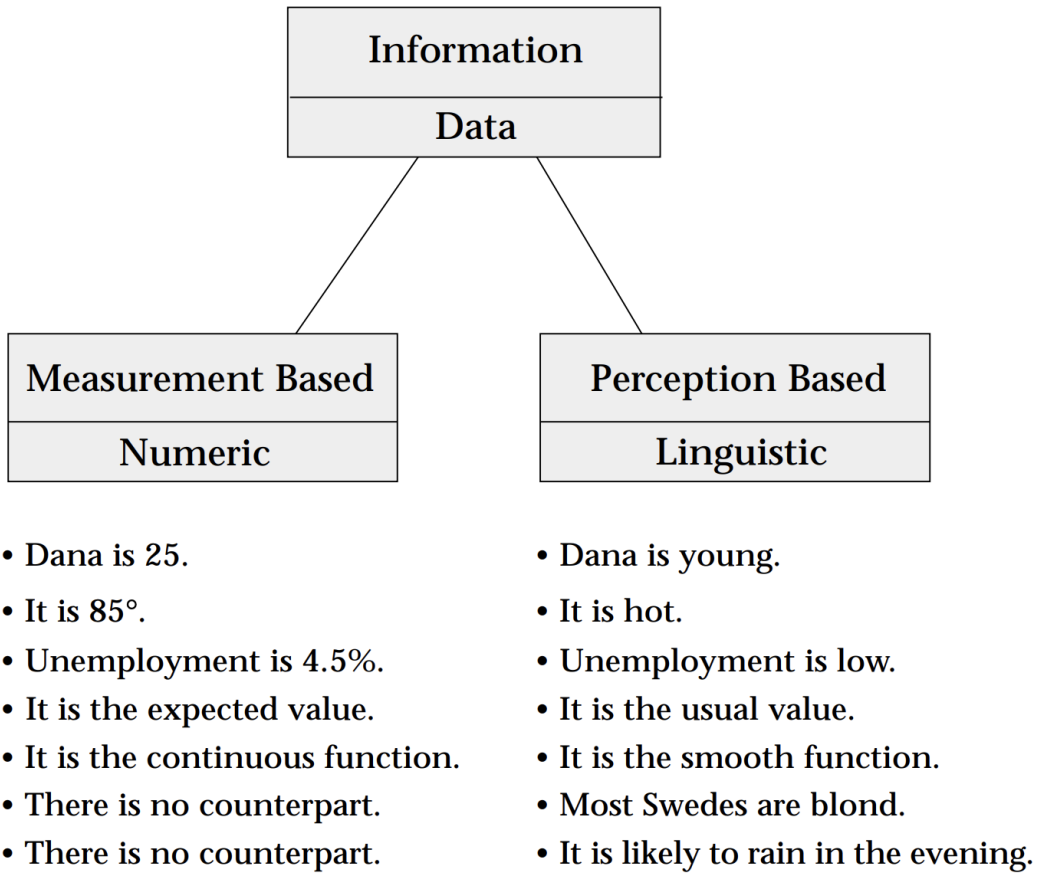

Figure 0.2.3: Πληροφοριακά δεδομένα(Zadeh 2001)

Με βάση τα παραπάνω, μπορούμε να συμπεράνουμε ότι τα κατηγορηματικά δεδομένα, με σαφείς ετικέτες μπορούν να είναι πιο διαισθητικά για τους χρήστες στην ερμηνεία τους σε σύγκριση με τις αριθμητικές τιμές. Ο κατηγορηματικός διαχωρισμός περιλαμβάνει τη διαίρεση μιας συνεχούς μεταβλητής σε διακριτές κατηγορίες ή ομάδες με βάση σημαντικές διακρίσεις. Οι κατηγορίες με νόημα διευκολύνουν τους χρήστες να εντοπίζουν μοτίβα, να κάνουν συγκρίσεις και να εξάγουν συμπεράσματα από τα δεδομένα. Οι κατηγορηματικοί διαχωρισμοί συμβάλλουν στη διαφάνεια και την ερμηνευσιμότητα των μοντέλων καθιστώντας την υποκείμενη λογική πιο σαφή και κατανοητή. Οι χρήστες μπορούν εύκολα να αντιληφθούν γιατί επιλέχθηκαν ορισμένες κατηγορίες και πώς σχετίζονται με το συγκεκριμένο πρόβλημα. Αυτή η διαφάνεια ενισχύει την εμπιστοσύνη στα αποτελέσματα του μοντέλου [45]. Οι άνθρωποι τείνουν να προσεγγίζουν τα αριθμητικά στοιχεία με ασαφείς γλωσσικές ετικέτες οι οποίες συχνά διαφέρουν μεταξύ των ανθρώπων. Παραδόξως, οι άνθρωποι είναι σε θέση να επικοινωνούν με αυτές τις ασαφείς και αόριστες γλωσσικές ετικέτες και δεν ζητούν τις ακριβείς τιμές όταν τις συζητούν. Στην πραγματικότητα, αυτές οι αβέβαιες έννοιες επιτρέπουν στους ανθρώπους να είναι σε θέση να εκτελούν πολύ εξελιγμένες εργασίες [35].

Κατανόηση μοντέλου

Ένας σημαντικός μηχανισμός για τη βελτίωση της απόδοσης των εργασιών επίλυσης προβλημάτων θεωρείται συχνά η κατάλληλη οπτικοποίηση των πληροφοριών [[68], [69], [73]]. Η πληροφορία με τη μορφή εικόνων ή γραφημάτων θεωρείται γενικά ανώτερη από εκείνη σε άλλες αναπαραστάσεις. Έχουν γίνει πολλές έρευνες σε μια προσπάθεια επαλήθευσης αυτής της δήλωσης, αλλά τα αποτελέσματα ήταν αντιφατικά [41].

Προκειμένου να αντιμετωπίσει αυτό το ζήτημα, η Vessey ανέπτυξε μια θεωρία για να περιγράψει τη σχέση μεταξύ των αναπαραστάσεων και των τύπων εργασιών που υποστηρίζουν εργασίες που είναι καλύτερα κατάλληλες για γραφικές ή αναπαραστάσεις πινάκων [72]. Το έργο της Vessey εισήγαγε την έννοια του "cognitive fit". Η έννοια αυτή υποστηρίζει ότι η πολυπλοκότητα στο περιβάλλον της εργασίας μπορεί να μετριαστεί αποτελεσματικά όταν τα βοηθήματα επίλυσης προβλημάτων -όπως τα εργαλεία, οι τεχνικές ή οι αναπαραστάσεις προβλημάτων- ευθυγραμμίζονται με τις στρατηγικές της εργασίας (μέθοδοι ή διαδικασίες) που είναι απαραίτητες για την ολοκλήρωση της εργασίας.

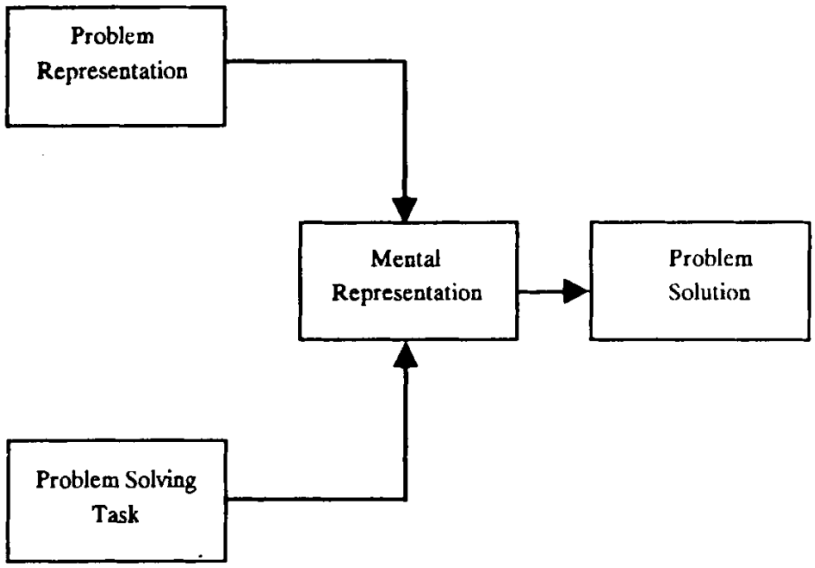

Figure 0.2.4: Γενικό μοντέλο επίλυσης προβλημάτων(Vessey 1991)

Η Vessey προτείνει το μοντέλο που παρουσιάζεται στο Σχήμα 0.2.4 για τη γενική επίλυση προβλημάτων στο οποίο βασίζεται το επιχείρημα του cognitive fit. Το μοντέλο θεωρεί την επίλυση προβλήματος ως αποτέλεσμα της σχέσης μεταξύ της αναπαράστασης του προβλήματος και του έργου επίλυσης προβλήματος. Η νοητική αναπαράσταση είναι ο τρόπος με τον οποίο το πρόβλημα αναπαρίσταται στην ανθρώπινη μνήμη εργασίας. Όταν οι τύποι των πληροφοριών που τονίζονται στα στοιχεία επίλυσης προβλήματος (αναπαράσταση προβλήματος και έργο) ταιριά-

ζουν, ο λύτης του προβλήματος χρησιμοποιεί διαδικασίες (και επομένως διαμορφώνει μια νοητική αναπαράσταση) που δίνουν επίσης έμφαση στον ίδιο τύπο πληροφοριών. Με άλλα λόγια, η αντιστοίχιση της αναπαράστασης με το έργο οδηγεί στη χρήση παρόμοιων, και επομένως συνεπών, διαδικασιών επίλυσης προβλήματος, και συνεπώς στη διαμόρφωση μιας συνεπούς νοητικής αναπαράστασης. Δεν θα χρειαστεί να μετασχηματιστεί η νοητική αναπαράσταση για να προσαρμοστεί στη χρήση διαφορετικών διαδικασιών για την εξαγωγή πληροφοριών από την αναπαράσταση του προβλήματος και την επίλυση του προβλήματος. Ως εκ τούτου, η επίλυση προβλημάτων με γνωστική προσαρμογή οδηγεί σε αποτελεσματική και αποδοτική επίδοση στην επίλυση προβλημάτων.

Τα δέντρα αποφάσεων είναι χωρικές αναπαραστάσεις προβλημάτων, καθώς παρουσιάζουν χωρικά συσχετιζόμενες πληροφορίες. Με βάση το *cognitive fit* τα εξηγήσιμα δέντρα αποφάσεων ευθυγραμμίζονται καλά με τα νοητικά μοντέλα του ανθρώπου για το πρόβλημα ταξινόμησης. Αυτή η ευθυγράμμιση ενισχύει την κατανόηση από τον χρήστη του τρόπου με τον οποίο το δέντρο αποφάσεων κατέληξε στα συμπεράσματά του, οδηγώντας σε πιο αποτελεσματική και σίγουρη λήψη αποφάσεων.

0.2.6 Μέτρα ερμηνευσιμότητας

Παραδοσιακά, η αξιολόγηση των μοντέλων ταξινόμησης έχει επικεντρωθεί κυρίως στην ακρίβεια πρόβλεψης ως πρωταρχικό κριτήριο. Ωστόσο, σε εφαρμογές του πραγματικού κόσμου όπου τα μοντέλα πρέπει όχι μόνο να αποδίδουν αλλά και να είναι κατανοητά στους χρήστες, η ερμηνευσιμότητα καθίσταται εξίσου σημαντική. Έχει σημειωθεί σημαντική πρόοδος στην ανάπτυξη μεθόδων για την ενίσχυση της κατανοητότητας των μοντέλων ταξινόμησης [29]. Η αξιολόγηση των διαφόρων μεθόδων επεξηγησιμότητας συχνά στερείται ενός σαφούς ποσοτικού πλαισίου και αντ' αυτού βασίζεται σε ποιοτικές αξιολογήσεις και μελέτες χρηστών. Αυτή η εξάρτηση από ποιοτικά μέτρα αναδεικνύει την πολυπλοκότητα της ερμηνευσιμότητας, η οποία περιλαμβάνει όχι μόνο τη διαφάνεια του μοντέλου αλλά και τον τρόπο με τον οποίο οι πληροφορίες γίνονται αντιληπτές και κατανοητές από τους ανθρώπους.

Ποσοτική αξιολόγηση της ερμηνευσιμότητας

Η ποσοτική αξιολόγηση της ερμηνευσιμότητας απαιτεί τον ορισμό και την εφαρμογή συγκεκριμένων μετρικών. Αυτές οι μετρικές αποσκοπούν στην καταγραφή διαφόρων πτυχών της ερμηνευσιμότητας, όπως η απλότητα, η διαφάνεια, η πιστότητα και η ανθρώπινη αξιολόγηση.

- Απλότητα
- Διαφάνεια
- Fidelity
- Ανθρώπινη αξιολόγηση

Ανθρώπινη αξιολόγηση της ερμηνευσιμότητας

Στις μετρικές ανθρώπινης αξιολόγησης για την εκτίμηση της ερμηνευσιμότητας σε μοντέλα μηχανικής μάθησης, συνήθως αξιολογούμε πτυχές που σχετίζονται με το πόσο καλά οι άνθρωποι κατανοούν και εμπιστεύονται τις αποφάσεις του μοντέλου. Ακολουθούν ορισμένες συγκεκριμένες πτυχές που αξιολογούμε στις μετρικές ανθρώπινης αξιολόγησης:

- Comprehensibility: Αξιολογούμε τη σαφήνεια και την ευκολία με την οποία οι άνθρωποι μπορούν να κατανοήσουν τη διαδικασία λήψης αποφάσεων του μοντέλου.
- Διαφάνεια: Αξιολογούμε το βαθμό στον οποίο οι εσωτερικές λειτουργίες του μοντέλου είναι διαφανείς και προσβάσιμες στους ανθρώπους.
- Αξιοπιστία: Μετράμε το επίπεδο εμπιστοσύνης που δείχνουν οι άνθρωποι στις προβλέψεις και τις αποφάσεις του μοντέλου [37].
- Ικανοποίηση του χρήστη: Μετράμε τη συνολική ικανοποίηση των συμμετεχόντων από την ερμηνευσιμότητα και τη χρηστικότητα του μοντέλου. Αυτό περιλαμβάνει τη συλλογή ανατροφοδότησης σχετικά με την εμπειρία του χρήστη, συμπεριλαμβανομένης της ευκολίας χρήσης, της σαφήνειας των εξηγήσεων και της αντιλαμβανόμενης αξίας της ερμηνευσιμότητας που παρέχει το μοντέλο [22], [37].

- Αποτελεσματικότητα στη λήψη αποφάσεων: Αξιολογούμε κατά πόσον η ερμηνευσιμότητα που παρέχει το μοντέλο επιτρέπει στους συμμετέχοντες να λαμβάνουν τεκμηριωμένες αποφάσεις ή να αναλαμβάνουν τις κατάλληλες ενέργειες με βάση τις προβλέψεις του μοντέλου. Αυτό περιλαμβάνει την αξιολόγηση της πρακτικής χρησιμότητας των εξηγήσεων του μοντέλου σε πραγματικές εφαρμογές και σενάρια.

Μετρικές ερμηνευσιμότητας δέντρων αποφάσεων

Η ερμηνεία των δέντρων αποφάσεων περιλαμβάνει την αξιολόγηση διαφόρων μετρικών που παρέχουν πληροφορίες σχετικά με την πολυπλοκότητα και την κατανοητότητά τους. Ενώ πολλές αξιολογήσεις απλοποιούν υπερβολικά αυτό το έργο εστιάζοντας αποκλειστικά στο μέγεθος του μοντέλου, η πραγματική ερμηνευσιμότητα εξαρτάται από πολλαπλούς παράγοντες που αποτυπώνουν τόσο συντακτικές όσο και σημασιολογικές πτυχές του δέντρου [26].

1. **Model Size**: Στη συντριπτική πλειονότητα των εργασιών όπου η ερμηνευσιμότητα ενός μοντέλου ταξινόμησης αξιολογείται, η αξιολόγηση αυτή γίνεται σε ένα υπεραπλουστευτικό τρόπο, μετρώντας μόνο το μέγεθος του μοντέλου. Η κατανοητότητα ενός μοντέλου εξαρτάται σε μεγάλο βαθμό από το πραγματικό του περιεχόμενο, όπως τα χαρακτηριστικά σε ένα δέντρο αποφάσεων, οι συνθήκες χαρακτηριστικών-τιμών σε κανόνες ταξινόμησης. Κατά συνέπεια, είναι απολύτως νοητό ένα μεγαλύτερο δέντρο αποφάσεων να είναι πιο κατανοητό για τον χρήστη από ένα μικρότερο, καθώς το μεγαλύτερο δέντρο μπορεί να ενσωματώνει χαρακτηριστικά που είναι πιο σημαντικά ή διαισθητικά για τον χρήστη. Για τη μέτρηση του μεγέθους του δέντρου χρησιμοποιούνται πολλές διαφορετικές μετρικές στη βιβλιογραφία:
   (a) **Βάθος δέντρου**: Το βάθος ενός δέντρου αποφάσεων είναι ένας άμεσος δείκτης της πολυπλοκότητάς του [3].
   (b) **Αριθμός κόμβων**: Ο συνολικός αριθμός των κόμβων (σημεία απόφασης) στο δέντρο.
   (c) **Αριθμός φύλλων**: Ο αριθμός των κόμβων φύλλων ή των τελικών σημείων του δέντρου.
2. **Αριθμός χαρακτηριστικών**: Ο αριθμός των χαρακτηριστικών που χρησιμοποιούνται από το δέντρο.
3. **Sparsity of nodes**: Το ποσοστό των κόμβων που έχουν λίγα σημεία δεδομένων που σχετίζονται με αυτούς σε σύγκριση με το συνολικό αριθμό κόμβων στο δέντρο.
4. **Καθαρότητα φύλλων**: Μετράει πόσο ομοιογενή είναι τα σημεία δεδομένων σε κάθε φύλλο.
5. **Ισορροπία δέντρου**: Ένα καλά ισορροπημένο δέντρο είναι συχνά ευκολότερο στην ερμηνεία.
6. **Μήκος κανόνα**: Μέσο μήκος των κανόνων απόφασης που προκύπτουν από τα μονοπάτια του δέντρου.

0.3 Τεχνικές Προσεγγίσεις

Ενώ στη βιβλιογραφία έχουν προταθεί πολλές προσεγγίσεις για τη βελτίωση της ερμηνευσιμότητας των μοντέλων μηχανικής μάθησης, οι μέθοδοι αυτές μπορούν να κατηγοριοποιηθούν σε τρεις κύριες οικογένειες: τεχνικές προεπεξεργασίας, τεχνικές αλγοριθμικής τροποποίησης και τεχνικές μετα-επεξεργασίας. Σε αυτό το κεφάλαιο, θα διερευνήσουμε αυτές τις προσεγγίσεις ειδικά στο πλαίσιο των δέντρων απόφασης. Οι τεχνικές προεπεξεργασίας αποσκοπούν στην τροποποίηση των χαρακτηριστικών των δεδομένων εισόδου ώστε να διασφαλιστεί ότι κάθε ταξινομητής που εκπαιδεύεται σε αυτά τα δεδομένα επιτυγχάνει υψηλή επεξηγηματικότητα στις προβλέψεις του. Αντίθετα, οι τεχνικές αλγοριθμικής τροποποίησης ενσωματώνουν τους περιορισμούς ερμηνευσιμότητας απευθείας στον αλγόριθμο μάθησης, εξασφαλίζοντας ότι το μοντέλο δέντρου αποφάσεων που προκύπτει είναι εγγενώς πιο επεξηγήσιμο. Τέλος, οι τεχνικές μετα-επεξεργασίας προσαρμόζουν τα αποτελέσματα ενός ήδη εκπαιδευμένου δέντρου αποφάσεων για να βελτιώσουν την ερμηνευσιμότητά του.

0.3.1 Τεχνικές προεπεξεργασίας Δεδομένων

Σε αυτό το κεφάλαιο θα θέλαμε να ελέγξουμε τις τεχνικές προεπεξεργασίας που μπορούν να βοηθήσουν στο να γίνουν τα μοντέλα πιο ερμηνεύσιμα. Οι άνθρωποι έχουν εγγενείς γνωστικούς περιορισμούς που μπορούν να επηρεάσουν την ικανότητά τους να κατανοούν πολύπλοκα μοντέλα. Είδαμε ότι οι χρόνοι αντίδρασης και τα ποσοστά λάθους στη λήψη αποφάσεων αυξάνονται με πιο σύνθετα μοντέλα. Ανεξάρτητα από τα ερεθίσματα, η μετάφραση του μοντέλου σε νοητική αναπαράσταση (κωδικοποίηση) αποτελεί το σημαντικότερο τμήμα της

διαδικασίας συλλογισμού, αντιπροσωπεύοντας περίπου το μισό του συνολικού χρόνου συλλογισμού, γεγονός που επισημαίνει τη σημασία του βήματος της προεπεξεργασίας για τη δημιουργία του μοντέλου.

Feature Engineering

Το feature engineering είναι μια θεμελιώδης διαδικασία στην προετοιμασία των μοντέλων μηχανικής μάθησης, όπου τα ακατέργαστα δεδομένα μετασχηματίζονται και εμπλουτίζονται για να βελτιωθεί η απόδοση και η ερμηνευσιμότητα του μοντέλου. Η διαδικασία αυτή περιλαμβάνει τη δημιουργία νέων χαρακτηριστικών από υπάρχοντα δεδομένα, την επιλογή των πιο σχετικών χαρακτηριστικών και τον μετασχηματισμό των χαρακτηριστικών για τη βελτίωση της χρηστικότητας και της αποτελεσματικότητάς τους για την προγνωστική μοντελοποίηση. Για τα δέντρα αποφάσεων, τα οποία διαχωρίζουν τα δεδομένα με βάση τις τιμές των επιμέρους χαρακτηριστικών, τα καλά σχεδιασμένα χαρακτηριστικά μπορούν να οδηγήσουν σε απλούστερα και πιο ερμηνεύσιμα δέντρα. Επίσης, μπορούν συχνά να ενσωματώσουν πολύπλοκες σχέσεις σε απλούστερες μορφές, μειώνοντας το βάθος και την πολυπλοκότητα του δέντρου απόφασης που απαιτείται για την επίτευξη υψηλής ακρίβειας. Επιπλέον, το feature engineering βοηθά στο χειρισμό ζητημάτων όπως οι ελλείπουσες τιμές, η κωδικοποίηση κατηγορηματικών δεδομένων και η κανονικοποίηση των αριθμητικών περιοχών, διασφαλίζοντας έτσι ότι ο αλγόριθμος του δέντρου αποφάσεων επικεντρώνεται σε γνήσιες συσχετίσεις δεδομένων και όχι σε τεχνουργήματα της αναπαράστασης των δεδομένων.

Επιλογή χαρακτηριστικών

Η επιλογή χαρακτηριστικών είναι ένα κρίσιμο βήμα στον αγωγό μηχανικής μάθησης που περιλαμβάνει την επιλογή ενός υποσυνόλου σχετικών χαρακτηριστικών (μεταβλητών, προβλεπτικών παραγόντων) για χρήση στην κατασκευή μοντέλων. Οι πρωταρχικοί στόχοι της επιλογής χαρακτηριστικών είναι η βελτίωση της απόδοσης του μοντέλου, η μείωση της υπερπροσαρμογής και η ενίσχυση της ερμηνευσιμότητας. Με την εξάλειψη των άσχετων ή περιττών χαρακτηριστικών, η επιλογή χαρακτηριστικών συμβάλλει στον εξορθολογισμό του μοντέλου, καθιστώντας το ταχύτερο και αποτελεσματικότερο. Όπως αναφέρεται στο κεφάλαιο 2.3, υπάρχει μεγάλη ανάγκη περιορισμού του αριθμού των χαρακτηριστικών που εμφανίζονται στην αναπαράσταση του μοντέλου μας, καθώς ένας μεγάλος αριθμός χαρακτηριστικών φαίνεται να προκαλεί μόνο σφάλματα στις ανθρώπινες αποφάσεις και να μην συμβάλλει περαιτέρω στη διαδικασία λήψης αποφάσεων.

Ένας τρόπος για να σκεφτούμε τις μεθόδους επιλογής χαρακτηριστικών είναι με όρους εποπτευόμενων και μη εποπτευόμενων μεθόδων [44]. Η διαφορά έχει να κάνει με το αν τα χαρακτηριστικά επιλέγονται με βάση τη μεταβλητή-στόχο ή όχι. Ένας άλλος τρόπος να εξετάσουμε τον μηχανισμό που χρησιμοποιείται για την επιλογή χαρακτηριστικών που μπορεί να χωριστεί σε μεθόδους wrapper, filter και embedded [74].

Μια δημοφιλής μέθοδος για την επιλογή χαρακτηριστικών περιλαμβάνει την αξιολόγηση των importance scores που αποδίδονται στις μεταβλητές από ένα μοντέλο μηχανικής μάθησης. Δεδομένου ότι η ακριβής επιλογή χαρακτηριστικών είναι ζωτικής σημασίας, είναι σημαντικό οι βαθμολογίες σημαντικότητας να αντιπροσωπεύουν με ακρίβεια την πραγματικότητα. Αυτή η μέθοδος επιχειρεί να ποσοτικοποιήσει τη σχετική σημασία κάθε χαρακτηριστικού για την πρόβλεψη της μεταβλητής-στόχου. Η σημασία της μεταβλητής υπολογίζεται με τη μέτρηση της επαυξητικής βελτίωσης της απόδοσης που αποδίδεται σε κάθε χρήση ενός χαρακτηριστικού εντός του μοντέλου και τη σύνοψη αυτής της πληροφορίας σε ολόκληρο το μοντέλο. Ένα βασικό πρόβλημα που αναφέρεται συχνά στη βιβλιογραφία είναι ότι η μέθοδος επιλογής διάσπασης του CART μεροληπτεί προς την επιλογή χαρακτηριστικών με μεγαλύτερο αριθμό πιθανών σημείων διάσπασης [43], [42]. Για να αντιμετωπιστεί αυτό το ζήτημα έχει προκύψει ένα νέο ρεύμα εργασίας με την κατασκευή δέντρων απόφασης με τεχνικές συνολικής βελτιστοποίησης αντί για άπληστες ευρετικές τεχνικές. Τα βέλτιστα δέντρα [9], [10] χρησιμοποιούν το mixed-integer optimization για την κατασκευή δέντρων απόφασης σε ένα μόνο βήμα που είναι συνολικά βέλτιστα. Οι Jack Dunn, Luca Mingardi, Ying Daisy Zhuo [24] διερεύνησαν την απόδοση της σημασίας των μεταβλητών ως χαρακτηριστικό γνώρισμα επιλογή για τις μεθόδους CART, Optimal Trees, XGBoost και κατέληξαν στο συμπέρασμα ότι η μέθοδος με τις καλύτερες επιδόσεις για την επιλογή χαρακτηριστικών είναι η Optimal Trees.

Τα Random Forests [14], μια ensemble μέθοδος μάθησης, χρησιμοποιούνται ευρέως για την επιλογή χαρακτηριστικών λόγω της ικανότητάς τους να χειρίζονται μεγάλα σύνολα δεδομένων και πολύπλοκες αλληλεπιδράσεις μεταξύ των χαρακτηριστικών. Κατασκευάζοντας ένα πλήθος δέντρων απόφασης κατά τη διάρκεια της εκπαίδευσης και υπολογίζοντας τον μέσο όρο των προβλέψεών τους, τα random forests παρέχουν εγγενώς ένα μέτρο της σημασίας των χαρακτηριστικών. Τα πλεονεκτήματα της χρήσης random forests για την επι-

λογή χαρακτηριστικών περιλαμβάνουν την ανθεκτικότητα στην υπερπροσαρμογή, την ικανότητα χειρισμού τόσο αριθμητικών όσο και κατηγορηματικών δεδομένων και την ικανότητα καταγραφής μη γραμμικών σχέσεων.

Ενσωμάτωση γνώσης του τομέα

Η ενσωμάτωση της γνώσης του τομέα κατά τη φάση της προεπεξεργασίας είναι μια ισχυρή στρατηγική που μπορεί να βελτιώσει σημαντικά την απόδοση και την ερμηνευσιμότητα των μοντέλων μηχανικής μάθησης. Η γνώση τομέα αναφέρεται στην εμπειρογνωμοσύνη και τις γνώσεις που αφορούν ειδικά τον τομέα ή τη βιομηχανία από την οποία προέρχονται τα δεδομένα. Με την εφαρμογή αυτής της εξειδικευμένης κατανόησης, οι επαγγελματίες μπορούν να λαμβάνουν πιο τεκμηριωμένες αποφάσεις σχετικά με την προετοιμασία δεδομένων, τη μηχανική των χαρακτηριστικών και τις διαδικασίες μετασχηματισμού, οδηγώντας τελικά σε πιο ακριβή και συναφή μοντέλα. Υπήρξε τεράστια έρευνα για την ενσωμάτωση της γνώσης του τομέα στα μοντέλα μηχανικής μάθησης παρουσιάζοντας εμπνευσμένα αποτελέσματα [18], [17].

Χρησιμοποιώντας τη γνώση του τομέα έχουμε τη δυνατότητα να δημιουργήσουμε ή να μετασχηματίσουμε χαρακτηριστικά, καθώς οι ειδικοί του τομέα μπορούν να εντοπίσουν και να δημιουργήσουν χαρακτηριστικά που αποτυπώνουν βασικές πτυχές των δεδομένων, οι οποίες μπορεί να μην είναι άμεσα εμφανείς μέσω αυτοματοποιημένων μεθόδων. Επίσης, η εμπειρογνωμοσύνη στον τομέα μπορεί να βοηθήσει στον εντοπισμό φυσικών ομαδοποιήσεων ή τμημάτων μέσα στα δεδομένα. Η γνώση του τομέα μπορεί να βοηθήσει στη μετατροπή αριθμητικών δεδομένων σε κατηγορηματικά δεδομένα, καθιστώντας τα πιο ερμηνεύσιμα για τον άνθρωπο.

Supervised Discretization

Το supervised discretization είναι μια ισχυρή τεχνική προεπεξεργασίας που μετασχηματίζει συνεχή ή αριθμητικά χαρακτηριστικά σε κατηγορηματικά χρησιμοποιώντας πληροφορίες για την ετικέτα κλάσης για να καθοδηγήσει τη διαδικασία διακριτοποίησης. Η μέθοδος αυτή είναι ιδιαίτερα πολύτιμη σε σενάρια όπου η ερμηνευσιμότητα και η κατανοητότητα του μοντέλου είναι υψίστης σημασίας. Διακριτοποιώντας τις συνεχείς μεταβλητές με βάση τα αποτελέσματα-στόχους, η επιβλεπόμενη διακριτοποίηση διασφαλίζει ότι οι προκύπτουσες κατηγορηματικές θέσεις είναι βέλτιστα ευθυγραμμισμένες με τους προγνωστικούς στόχους του μοντέλου. Αυτή η ευθυγράμμιση όχι μόνο ενισχύει τη διαφάνεια του μοντέλου απλοποιώντας τα αριθμητικά δεδομένα σε ερμηνεύσιμες κατηγορίες, αλλά ενδεχομένως αυξάνει και την ακρίβεια της πρόβλεψης με την καταγραφή των μη γραμμικών εξαρτήσεων μεταξύ των χαρακτηριστικών και των ετικετών κατηγορίας με πιο ουσιαστικό τρόπο.

Στην πράξη, η εποπτευόμενη διακριτοποίηση χρησιμοποιεί συνήθως αλγορίθμους όπως τα δέντρα αποφάσεων για τον καθορισμό των βέλτιστων ορίων δυαδικών πεδίων. Αυτοί οι αλγόριθμοι αξιολογούν τις πιθανές διαιρέσεις με βάση το κέρδος πληροφορίας ή παρόμοια κριτήρια που μετρούν την αποτελεσματικότητα μιας διάκρισης όσον αφορά τη διαφοροποίηση των κλάσεων. Διαχωρίζοντας τον συνεχή χώρο εισόδου σε διαστήματα που αντιστοιχούν σε διακριτά προγνωστικά αποτελέσματα, η τεχνική μεταφράζει αποτελεσματικά πολύπλοκα αριθμητικά μοτίβα σε απλή, βασισμένη σε κανόνες γνώση. Ως αποτέλεσμα, τα μοντέλα που έχουν υποστεί προεπεξεργασία με επιβλεπόμενη διακριτοποίηση προσφέρονται καλύτερα για επικύρωση και εμπιστοσύνη από τους τελικούς χρήστες.

0.3.2 Αλγοριθμικές Τεχνικές

Βελτίωση της ερμηνευσιμότητας στα δέντρα αποφάσεων CART, C4.5 και ID3

Οι αλγόριθμοι CART, C4.5 και ID3 είναι άπληστοι αλγόριθμοι που κατασκευάζουν δέντρα απόφασης με top-down τρόπο [13], [62], [61]. Η προσέγγιση αυτή περιλαμβάνει την αναδρομική επιλογή του καλύτερου χαρακτηριστικού για τη διαίρεση των δεδομένων σε κάθε κόμβο, με βάση συγκεκριμένα κριτήρια όπως το Gini impurity, το gain ratio ή το κέρδος πληροφορίας. Ενώ αυτή η μέθοδος είναι αποδοτική και αποτελεσματική στη δημιουργία ακριβών μοντέλων, μπορεί μερικές φορές να οδηγήσει σε πολύπλοκα και λιγότερο ερμηνεύσιμα δέντρα. Για να βελτιωθεί η ερμηνευσιμότητα των δέντρων απόφασης που προκύπτουν, μπορούν να χρησιμοποιηθούν διάφορες στρατηγικές. Αυτές περιλαμβάνουν:

1. Κλάδεμα για την αφαίρεση περιττών κλάδων
2. Περιορισμός του βάθους του δέντρου για την αποφυγή υπερβολικά πολύπλοκων δομών

3. Ορισμός ενός ελάχιστου αριθμού δειγμάτων που απαιτούνται για τη δημιουργία ενός κόμβου φύλλου
4. Προσεκτική επιλογή χαρακτηριστικών για εστίαση στα πιο κατατοπιστικά χαρακτηριστικά
5. Δημιουργία σαφών οπτικών αναπαραστάσεων του δέντρου
6. Περιορισμός του συντελεστή διακλάδωσης

Εφαρμόζοντας αυτές τις τεχνικές, μπορούμε να δημιουργήσουμε μοντέλα δέντρων απόφασης που δεν είναι μόνο ακριβή αλλά και διαφανή και εύκολα κατανοητά.

Strong Optimal Classification Tree

Η πρόκληση της εκμάθησης βέλτιστων δέντρων απόφασης ταξινομείται ως ένα NP-hard πρόβλημα, όπως συζητήθηκε από τους Hyafil και Rivest [47] και Breiman et al. [13]. Αυτό το πρόβλημα μπορεί να γίνει διαισθητικά κατανοητό ως ένα πρόβλημα συνδυαστικής βελτιστοποίησης με εκθετικό αριθμό μεταβλητών απόφασης.

Οι παραδοσιακοί αλγόριθμοι για την εκμάθηση δέντρων απόφασης, όπως οι CART, C4.5 και ID3, χρησιμοποιούν άπληστες τεχνικές για να κάνουν τοπικά βέλτιστες διακλαδώσεις. Οι Bertsimas και Dunn [9] πρότειναν πρόσφατα μια εναλλακτική μέθοδο που χρησιμοποιεί mixed-integer optimization (MIO) για την εκμάθηση βέλτιστων δέντρων ταξινόμησης (OCT). Τα OCTs χρησιμοποιούν μια προσέγγιση σφαιρικής βελτιστοποίησης με χρήση mixed-integer programming για την κατασκευή ολόκληρου του δέντρου σε ένα μόνο βήμα. Το μοντέλο που προκύπτει διατηρεί την ερμηνευσιμότητα ενός ενιαίου δέντρου απόφασης, αλλά έχει αποδειχθεί ότι υπερτερεί έναντι του CART και έχει απόδοση ανταγωνιστική με τα μοντέλα μαύρου κουτιού.

Αναλυτικότερα, τα OCT χρησιμοποιούν μια μαθηματική προσέγγιση βελτιστοποίησης για την κατασκευή του δέντρου επιλύοντας ένα πρόβλημα mixed-integer optimization. Ο στόχος είναι η εύρεση της δομής του δέντρου που ελαχιστοποιεί το σφάλμα ταξινόμησης, λαμβάνοντας επίσης υπόψη την πολυπλοκότητα του δέντρου. Λαμβάνοντας υπόψη τα παραπάνω, τα OCTs μπορούν να εντοπίσουν τις καλύτερες δυνατές διασπάσεις που οι παραδοσιακοί άπληστοι αλγόριθμοι μπορεί να χάσουν. Αυτή η σφαιρική προοπτική εξασφαλίζει ότι το δέντρο είναι τόσο ακριβές όσο και ερμηνεύσιμο. Για να ενισχυθεί η απλότητα και η ερμηνευσιμότητα οι OCTs σχεδιάζονται για να παράγουν απλούστερα μοντέλα ενσωματώνοντας περιορισμούς που περιορίζουν το βάθος και τον αριθμό των κόμβων του δέντρου.

Η χρήση mixed-integer programming για τη δημιουργία ενός βέλτιστου δέντρου ταξινόμησης έχει κερδίσει σημαντική προσοχή στη βιβλιογραφία, όπως αποδεικνύεται από τις μεταγενέστερες εργασίες των Günlük et al. [33], Aghaei et al. [1], και Verwer και Zhang [71], οι οποίοι διερεύνησαν τη χρήση του MIO για την εκμάθηση δέντρων απόφασης. Αυτό δεν είναι τυχαίο. Πρώτον, μια ποικιλία έτοιμων επιλυτών και αλγορίθμων MIO, όπως οι CPLEX (2009) και Gurobi (2015), έχουν αναπτυχθεί κατά τη διάρκεια δεκαετιών έρευνας και είναι αποτελεσματικοί στη μείωση του χώρου αναζήτησης για προβλήματα MIO. Δεύτερον, η MIO παρέχει μια ιδιαίτερα εκφραστική γλώσσα που επιτρέπει την προσαρμογή της αντικειμενικής συνάρτησης και την προσθήκη πρακτικών περιορισμών στο πρόβλημα μάθησης.

Ένας βασικός παράγοντας για την αποτελεσματική επίλυση των MIOs είναι η δημιουργία αποτελεσματικών διατυπώσεων, η οποία είναι ένα δύσκολο έργο. Η τυπική μέθοδος για την επίλυση προβλημάτων MIO είναι η branch-and-bound, η οποία διαιρεί αναδρομικά τον χώρο αναζήτησης και επιλύει χαλαρώσεις γραμμικής βελτιστοποίησης (LO) για κάθε διαίρεση για τη δημιουργία ορίων που μπορούν να εξαλείψουν τμήματα του χώρου αναζήτησης. Οι Aghaei et al. [2] στην εργασία τους παρουσιάζουν μια διατύπωση MIO με βάση τη ροή για την εκμάθηση βέλτιστων δέντρων ταξινόμησης με δυαδικά χαρακτηριστικά. Σε αυτό το μοντέλο, τα σωστά ταξινομημένα σημεία δεδομένων θεωρούνται ότι ρέουν από τη ρίζα σε ένα κατάλληλο φύλλο, ενώ τα λανθασμένα ταξινομημένα σημεία δεδομένων περιορίζονται από τη ροή μέσα στο δέντρο. Εκμεταλλεύονται τη δομή max-flow των υποπροβλημάτων για την αποτελεσματική επίλυσή τους με τη χρήση μιας προσαρμοσμένης διαδικασίας min-cut.

Ένα κρίσιμο βήμα σε αυτή τη διατύπωση MIO με βάση τη ροή περιλαμβάνει τη μετατροπή του δέντρου αποφάσεων σταθερού βάθους σε κατευθυνόμενο ακυκλικό γράφημα, όπου όλα τα τόξα ρέουν από τη ρίζα του δέντρου προς τα φύλλα του. Το Σχήμα 0.3.1 (αριστερά) απεικονίζει ένα imbalanced δέντρο απόφασης. Η κεντρική ιδέα αυτού του μοντέλου είναι να μετατρέψουμε αυτό το imbalanced δέντρο αποφάσεων σε ένα κατευθυνόμενο ακυκλικό

γράφημα προσθέτοντας έναν μοναδικό κόμβο πηγής s που συνδέεται με τον κόμβο ρίζας του δέντρου (κόμβος 1) και έναν μοναδικό κόμβο καταβόθρας t που συνδέεται με όλους τους κόμβους φύλλων του δέντρου. Ονομάζουμε αυτό το γράφημα γράφημα ροής του δέντρου αποφάσεων. Το Σχήμα 0.3.1 παρέχει μια απεικόνιση αυτών των ιδεών που εφαρμόζονται σε ένα δέντρο απόφασης βάθους d=2.

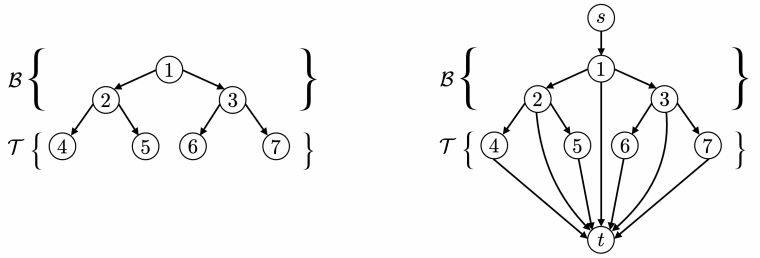

Figure 0.3.1: Ένα δέντρο απόφασης βάθους 2 (αριστερά) και ο σχετικός γράφος ροής (δεξιά).

Για την εργασία μας, χρησιμοποιήσαμε το μοντέλο FlowOCT, το οποίο είναι η διατύπωση του προβλήματος με βάση τη ροή για ένα imbalanced δέντρο απόφασης.

Στην έρευνά μας, επιλέξαμε να χρησιμοποιήσουμε τα βέλτιστα δέντρα ταξινόμησης με βάση τη ροή (FlowOCT) έναντι των κλασικών βέλτιστων δέντρων ταξινόμησης (OCT) λόγω των σημαντικών πλεονεκτημάτων απόδοσης και της βελτιωμένης ερμηνευσιμότητάς τους. Έχει αποδειχθεί ότι τα μοντέλα FlowOCT είναι έως και 29 φορές ταχύτερα από τα αντίστοιχα κλασικά μοντέλα, γεγονός που τα καθιστά ιδιαίτερα αποδοτικά για τους σκοπούς μας [2]. Αυτή η αποδοτικότητα, σε συνδυασμό με την ικανότητά τους να διατηρούν μια ισορροπημένη και αραιή δενδροειδή δομή, οδηγεί σε συντομότερες και πιο συνεπείς διαδρομές λήψης αποφάσεων, ενισχύοντας έτσι την ερμηνευσιμότητα του μοντέλου. Ωστόσο, το κύριο μειονέκτημα του μοντέλου FlowOCT είναι η εξάρτησή του από δυαδικά δεδομένα, γεγονός που μπορεί να περιορίσει τη δυνατότητα εφαρμογής του σε σενάρια που απαιτούν την ανάλυση συνεχών ή πολλαπλών κατηγοριών κατηγορηματικών δεδομένων.

C4.5 Modifications

Ο C4.5, που αναπτύχθηκε από τον Ross Quinlan, είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος για τη δημιουργία δέντρων απόφασης από ένα σύνολο δεδομένων. Επεκτείνει τον προηγούμενο αλγόριθμο ID3 και χρησιμοποιείται κυρίως για εργασίες ταξινόμησης. Ο C4.5 κατασκευάζει ένα δέντρο με αναδρομική κατάτμηση των δεδομένων με βάση το χαρακτηριστικό που παρέχει τον υψηλότερο λόγο κέρδους πληροφορίας σε κάθε κόμβο. Αυτός ο αλγόριθμος μπορεί να χειριστεί τόσο κατηγορηματικά όσο και συνεχή χαρακτηριστικά, τα οποία μπορεί να μετατρέψει σε διακριτά διαστήματα. Ενσωματώνει επίσης μηχανισμούς για το χειρισμό των ελλιπών τιμών και το κλάδεμα των δέντρων μετά τη δημιουργία τους για τη βελτίωση της γενίκευσης και την αποφυγή της υπερπροσαρμογής. Αυτά τα χαρακτηριστικά καθιστούν τον C4.5 ένα στιβαρό και ευέλικτο εργαλείο για διάφορα προβλήματα ταξινόμησης.

Χειρισμός κατηγορηματικών δεδομένων στον C4.5

Ο αλγόριθμος C4.5 χειρίζεται κατηγορηματικά δεδομένα αξιολογώντας το information gain ratio κάθε κατηγορηματικού χαρακτηριστικού για να καθορίσει τον καλύτερο διαχωρισμό σε κάθε κόμβο. Για ένα κατηγορηματικό χαρακτηριστικό με πολλές διαφορετικές τιμές, ο αλγόριθμος C4.5 θεωρεί κάθε πιθανή τιμή ως έναν πιθανό κλάδο στο δέντρο αποφάσεων. Ο αλγόριθμος υπολογίζει το κέρδος πληροφορίας για κάθε χαρακτηριστικό και στη συνέχεια κανονικοποιεί αυτή την τιμή χρησιμοποιώντας τις πληροφορίες διαχωρισμού για να λάβει τον λόγο κέρδους. Το χαρακτηριστικό με τον υψηλότερο λόγο κέρδους επιλέγεται για τη διάσπαση.

Με τη χρήση του λόγου κέρδους, ο C4.5 ευνοεί τα χαρακτηριστικά που οδηγούν σε σημαντικές διασπάσεις, αποφεύγοντας προκαταλήψεις προς χαρακτηριστικά με πολλές διαφορετικές τιμές. Μια άλλη πρόκληση στην ερμηνεία του δέντρου αποφάσεων είναι ότι ορισμένα υποδέντρα του κατασκευασμένου δέντρου μπορεί να περιέχουν άσχετα χαρακτηριστικά, ακόμη και όταν τα δεδομένα δεν είναι θορυβώδη. Το ζήτημα αυτό προκύπτει επειδή η δομή του δέντρου αποφάσεων απαιτεί αυστηρά ότι μετά από την επιλογή ενός χαρακτηριστικού για την επισήμανση ενός κόμβου, κάθε τιμή αυτού του χαρακτηριστικού πρέπει να συμπεριληφθεί στο δέντρο. Κατά συνέπεια, ορισμένοι κλάδοι μπορεί να προστεθούν αποκλειστικά για τη διατήρηση της δομής του δέντρου, ακόμη και αν οι

κλάδοι αυτοί συνδέονται με άσχετες τιμές χαρακτηριστικών [12], [16], [28], [30]. Αυτές οι άσχετες τιμές μπορούν να παραπλανήσουν την ερμηνεία του δέντρου από τον χρήστη και μπορεί να οδηγήσουν σε υπερπροσαρμογή.

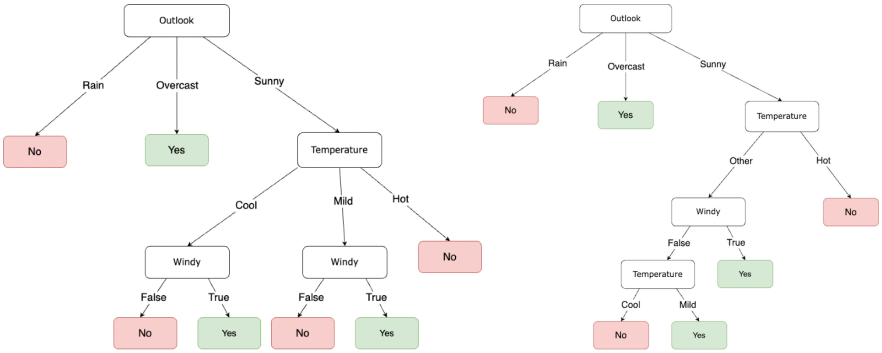

Figure 0.3.2: Ένα δέντρο αποφάσεων χωρίς ομαδοποίηση (αριστερά) και ένα δέντρο αποφάσεων με ομαδοποίηση (δεξιά).

Για την αντιμετώπιση αυτού του ζητήματος, προτείνουμε την ομαδοποίηση των μη σημαντικών τιμών των χαρακτηριστικών σε μια ενιαία τιμή με την ετικέτα "Other" (βλ. Σχήμα 0.3.2). Αυτό περιλαμβάνει τον υπολογισμό του κέρδους πληροφορίας για όλους τους πιθανούς συνδυασμούς διαχωρισμού και ομαδοποίησης και την επιλογή του διαχωρισμού που μεγιστοποιεί το κέρδος πληροφορίας. Με την ενοποίηση λιγότερο σημαντικών χαρακτηριστικών, μπορούμε να απλοποιήσουμε τη δομή του δέντρου, να μειώσουμε τον παράγοντα διακλάδωσης, να βελτιώσουμε την ερμηνευσιμότητα και να μειώσουμε τον κίνδυνο υπερπροσαρμογής.

Χειρισμός συνεχών δεδομένων στο C4.5

Ο αλγόριθμος C4.5 επεκτείνει τον προκάτοχό του, ID3, παρέχοντας μια ισχυρή μέθοδο για το χειρισμό αριθμητικών (συνεχών) χαρακτηριστικών. Σε αντίθεση με τα κατηγορηματικά χαρακτηριστικά, τα οποία έχουν ένα σταθερό σύνολο διακριτών τιμών, τα αριθμητικά χαρακτηριστικά μπορούν να λάβουν ένα δυνητικά άπειρο εύρος τιμών. Το C4.5 αντιμετωπίζει αυτή την πρόκληση μετατρέποντας τα συνεχή χαρακτηριστικά σε διακριτά διαστήματα μέσω μιας διαδικασίας που ονομάζεται thresholding.

Στην εργασία μας για τη βελτίωση του αλγορίθμου C4.5, εισαγάγαμε μια προσέγγιση για τον χειρισμό αριθμητικών χαρακτηριστικών επιτρέποντας multiway διαχωρισμούς. Ο παραδοσιακός C4.5 χρησιμοποιεί δυαδικές διαιρέσεις για αριθμητικά χαρακτηριστικά, χωρίζοντας τα δεδομένα σε δύο ομάδες με βάση ένα μόνο threshold. Αν και αποτελεσματική, αυτή η μέθοδος μπορεί μερικές φορές να οδηγήσει σε υπερβολικά πολύπλοκα δέντρα με πολλά επίπεδα, καθώς τα εμπλεκόμενα χαρακτηριστικά θα πρέπει να μπορούν να εμφανίζονται πολλές φορές στα μονοπάτια από τη ρίζα του δέντρου προς τα φύλλα του [27], μειώνοντας την ερμηνευσιμότητα. Για την αντιμετώπιση αυτού του προβλήματος, υλοποιήσαμε multiway splits για αριθμητικά χαρακτηριστικά, τα οποία μπορούν να δημιουργήσουν πολλαπλά διαστήματα και να απλοποιήσουν τη δομή του δέντρου. Η εργασία μας βασίστηκε στις εργασίες των Fayyad και Irani [25] και Dougherty et al.

Λεπτομέρειες Υλοποίησης

1. **Εισαγωγή παραμέτρων**: Εισάγαμε μια παράμετρο που ονομάζεται **max_splits** για τον έλεγχο του μέγιστου αριθμού σημείων διαχωρισμού για κάθε αριθμητικό χαρακτηριστικό.

2. **Προσδιορισμός πιθανών σημείων διαχωρισμού**: Για κάθε αριθμητικό χαρακτηριστικό, το σύνολο δεδομένων ταξινομείται με βάση τις τιμές του χαρακτηριστικού. Ενδεχόμενα σημεία διαχωρισμού εντοπίζονται εκεί όπου υπάρχει αλλαγή στην κατηγορία μεταξύ δύο διαδοχικών σημείων δεδομένων. Με τον τρόπο αυτό διασφαλίζεται ότι η διάσπαση εξετάζεται μόνο σε σημαντικές μεταβάσεις στην κατανομή των δεδομένων.

3. **Calculating Information Gain**: Για κάθε αριθμητικό χαρακτηριστικό, αξιολογούμε διαχωρισμούς που κυμαίνονται από 2 έως *max_splits* πιθανά σημεία. Το κέρδος πληροφορίας υπολογίζεται για κάθε συνδυασμό διαχωρισμού για να προσδιοριστεί το βέλτιστο σύνολο σημείων διαχωρισμού.

Algorithm 1: C4.5 categorical attributes grouping

Input: *dataset*, *attributes*
Output: *tree* ▷ Decision Tree

```
Function BuildTree(dataset, attributes):
    if all instances in dataset have the same class then
        return a leaf node with that class
    end
    if attributes is empty then
        return a leaf node with the majority class of dataset
    end
    best_attribute ← None
    best_split ← None
    best_gain ← −∞
    best_grouping ← −∞
    foreach attribute ∈ attributes do
        foreach grouping ∈ possibleGroupings(attribute) do
            split ← calculateSplit(dataset, attribute, grouping)
            gain ← calculateInformationGain(dataset, split)
            if gain > best_grouping_gain then
                best_grouping ← grouping
                best_grouping_gain ← gain
            end
        end
        if best_grouping_gain > best_gain then
            best_attribute ← attribute
            best_split ← best_grouping
            best_gain ← best_grouping_gain
        end
    end
    tree ← DecisionNode(best_attribute)
    attributes ← remove(attributes, best_attribute)
    foreach value ∈ best_split do
        subset ← splitDataset(dataset, best_attribute, value)
        if subset is empty then
            child_node ← LeafNodemajorityClass(dataset)
        end
        else
            child_node ← BuildTree(subset, attributes)
        end
        tree.addChild(value, child_node)
    end
    return tree
End Function
```

4. **Επιλογή των καλύτερων σημείων διαχωρισμού**: Ο αλγόριθμος επιλέγει το συνδυασμό των σημείων διαχωρισμού που μεγιστοποιεί το κέρδος πληροφορίας. Αυτά τα σημεία διάσπασης χρησιμοποιούνται στη συνέχεια για τη δημιουργία πολλαπλών διακλαδώσεων στο δέντρο απόφασης.

Βελτίωση της ερμηνευσιμότητας

Η χρήση των multiway splits για τα αριθμητικά χαρακτηριστικά ενισχύει την ερμηνευσιμότητα με διάφορους τρόπους, όπως φαίνεται και στο Σχήμα 0.3.3:

1. **Απλουστευμένη δενδρική δομή**
2. **Καθαρότερα όρια αποφάσεων**
3. **Μειωμένη υπερπροσαρμογή**
4. **Βελτιωμένη διορατικότητα**

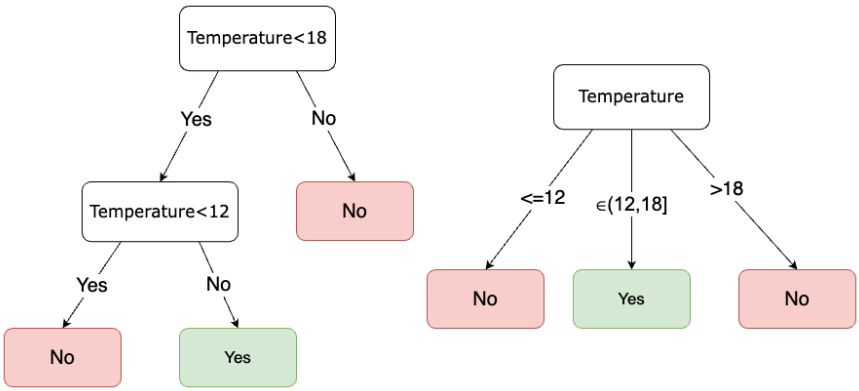

Figure 0.3.3: Binary split για αριθμητικά χαρακτηριστικά ενός δέντρου αποφάσεων (αριστερά) και Multiway split για αριθμητικά χαρακτηριστικά (δεξιά).

Με τα multiway splits για τα αριθμητικά χαρακτηριστικά, ο βελτιωμένος αλγόριθμος C4.5 παράγει πιο ερμηνεύσιμα και αποτελεσματικά δέντρα αποφάσεων. Αυτή η προσέγγιση αξιοποιεί την ευελιξία των multiway splits για τη δημιουργία απλούστερων, σαφέστερων και πιο κατατοπιστικών μοντέλων.

Αν και η εισαγωγή των multiway splits για αριθμητικά χαρακτηριστικά στον τροποποιημένο αλγόριθμό μας C4.5 βελτιώνει την ερμηνευσιμότητα και την απλότητα του μοντέλου, εισάγει επίσης μια σημαντική υπολογιστική πρόκληση.

0.3.3 Τεχνικές μετεπεξεργασίας

Για να βελτιωθεί η ερμηνευσιμότητα των δέντρων απόφασης, μπορούν να χρησιμοποιηθούν διάφορες τεχνικές μετα-επεξεργασίας. Μια αποτελεσματική μέθοδος είναι το post-pruning, το οποίο περιλαμβάνει την αφαίρεση κλάδων που δεν συμβάλλουν σημαντικά στην ακρίβεια του μοντέλου αφού το δέντρο έχει αναπτυχθεί πλήρως. Η οπτικοποίηση του δέντρου είναι μια άλλη κρίσιμη τεχνική για τη βελτίωση της ερμηνευσιμότητας. Οι βελτιωμένες οπτικοποιήσεις χρησιμοποιούν σαφείς και συνοπτικές αναπαραστάσεις για την απεικόνιση της δομής του δέντρου αποφάσεων. Τεχνικές όπως η χρωματική κωδικοποίηση των κόμβων με βάση τις πιθανότητες των κλάσεων,οι διαδραστικές απεικονίσεις, η επισήμανση σημαντικών διαδρομών και η προσθήκη tooltips με πρόσθετες πληροφορίες μπορούν να κάνουν το δέντρο πιο προσιτό και διαισθητικό. Η απλούστευση των κόμβων και η εξισορρόπηση των δέντρων είναι επίσης απαραίτητες για τη δημιουργία πιο ερμηνεύσιμων δέντρων αποφάσεων. Η απλούστευση κόμβων περιλαμβάνει τη συγχώνευση παρόμοιων κόμβων που έχουν περιττά κριτήρια απόφασης ή οδηγούν στην ίδια ταξινόμηση, μειώνοντας έτσι τον πλεονασμό και εξορθολογίζοντας το δέντρο. Επιπλέον, μπορούν να αφαιρεθούν ασήμαντοι κόμβοι που συμβάλλουν ελάχιστα στη συνολική ακρίβεια, με αποτέλεσμα ένα καθαρότερο μοντέλο. Η εξισορρόπηση του δέντρου αποσκοπεί στη διατήρηση ενός ομοιόμορφου βάθους σε διάφορους κλάδους, αποτρέποντας τον σχηματισμό βαθιών και πολύπλοκων κλάδων που είναι δύσκολο να ερμηνευθούν.

0.4 Έρευνα χρηστών

Στόχος της παρούσας μελέτης ήταν να μετρήσει και να συγκρίνει την ερμηνευσιμότητα των δέντρων αποφάσεων που παράγονται από διάφορους αλγορίθμους και σύνολα δεδομένων. Για να το επιτύχουμε αυτό, συγκεντρώσαμε 52 συμμετέχοντες και τους χωρίσαμε σε δύο ομάδες των 29 και 23 συμμετεχόντων η καθεμία. Οι συμμετέχοντες κλήθηκαν να συμπληρώσουν έντυπα που περιείχαν πολλαπλά μοντέλα δέντρων απόφασης και να ταξινομήσουν δείγματα δεδομένων χρησιμοποιώντας αυτά τα μοντέλα.

0.4.1 Γενική περιγραφή του ερωτηματολογίου

Ανάλυση ομάδων

Για να κατανοήσουμε το υπόβαθρο και την εξοικείωση των συμμετεχόντων με την τεχνητή νοημοσύνη και τα δέντρα αποφάσεων, ζητήσαμε από κάθε συμμετέχοντα να απαντήσει σε τρεις προκαταρκτικές ερωτήσεις σε κλίμακα από το 1 έως το 5:

1. **AI familiarity**: Πόσο εξοικειωμένοι είστε με την τεχνητή νοημοσύνη;
2. **XAI Familiarity**: Πόσο εξοικειωμένοι είστε με την εξηγήσιμη τεχνητή νοημοσύνη (XAI);
3. **Decision Trees Familiarity**: Πόσο εξοικειωμένοι είστε με τα δέντρα αποφάσεων;

Η εξοικείωση των χρηστών εμφανίζεται στο Σχήμα 0.4.1. Τα επίπεδα εξοικείωσης των χρηστών ανά έκδοση εμφανίζονται στο Σχήμα 0.4.2.

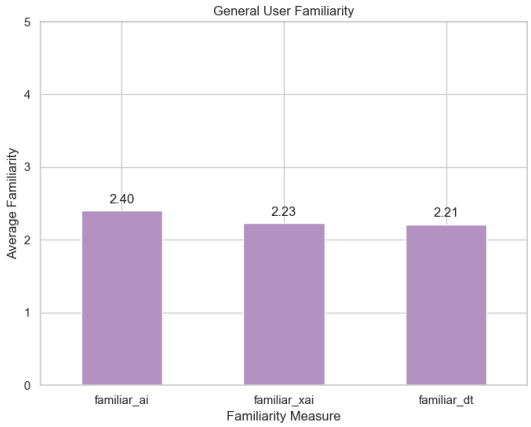

Figure 0.4.1: Επίπεδο εξοικείωσης του χρήστη με την TN

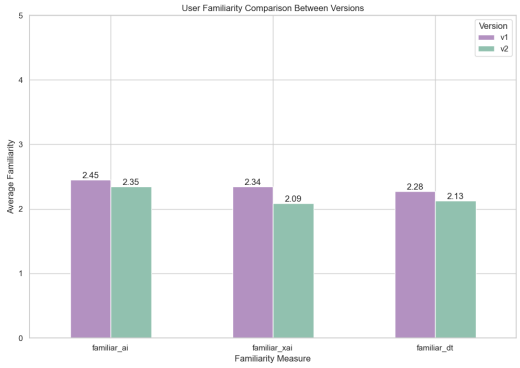

Figure 0.4.2: Επίπεδο εξοικείωσης του χρήστη με την TN ανά έκδοση

0.4.2 Ερωτηματολόγια

Κάθε συμμετέχων συμπλήρωσε ένα ερωτηματολόγιο που περιλάμβανε τέσσερα διαφορετικά μοντέλα δέντρων αποφάσεων που προέκυψαν από τέσσερα διαφορετικά σύνολα δεδομένων. Για κάθε μοντέλο, στους συμμετέχοντες δόθηκαν πέντε διαφορετικά δείγματα δεδομένων τα οποία έπρεπε να ταξινομήσουν χρησιμοποιώντας το δέντρο απόφασης.

Μετά την ταξινόμηση κάθε δείγματος, οι συμμετέχοντες απάντησαν σε τρεις ερωτήσεις για να αξιολογήσουν την εμπειρία και την αυτοπεποίθησή τους:

1. **Answer Confidence**: (Κλίμακα: 1 έως 5)
2. **Clarity of Decision Path** (σαφήνεια της πορείας λήψης απόφασης): Ήμουν σε θέση να ακολουθήσω με σαφήνεια το μονοπάτι απόφασης από τη ρίζα στους κόμβους φύλλων. (Κλίμακα: 1 = Διαφωνώ απόλυτα έως 5 = Συμφωνώ απόλυτα)
3. **Κατανοητότητα της δομής**: Η συνολική δομή του δέντρου αποφάσεων ήταν εύκολα κατανοητή. (Κλίμακα: 1 έως 5)
4. Επιπλέον, ζητήθηκε από τους συμμετέχοντες να περιγράψουν το μοντέλο με δικά τους λόγια. Αυτή η ανοιχτή ερώτηση παρείχε ποιοτικές πληροφορίες σχετικά με την κατανόηση και την ερμηνεία του δέντρου αποφάσεων.

0.4.3 Υλοποίηση του ερωτηματολογίου

Στη μελέτη μας για τους χρήστες, η ακριβής μέτρηση του χρόνου απόκρισης για κάθε εργασία ταξινόμησης ήταν ζωτικής σημασίας. Επιπλέον, λόγω του μεγάλου μεγέθους των δέντρων απόφασης που προέκυψαν, ήταν σημαντικό για τους χρήστες να έχουν τη δυνατότητα να κάνουν ζουμ στις εικόνες των δέντρων απόφασης. Δεδομένου ότι κανένας γνωστός πάροχος ερωτηματολογίων δεν προσέφερε και τα δύο αυτά χαρακτηριστικά, αποφασίσαμε να υλοποιήσουμε μια προσαρμοσμένη λύση.

Το front-end της εφαρμογής είναι μια δυναμική φόρμα που κατασκευάστηκε με τη χρήση της βιβλιοθήκης React. Στόχος της είναι να παρέχει μια φιλική προς το χρήστη και αποτελεσματική πλατφόρμα για τη συλλογή δεδομένων από τους χρήστες, ενσωματώνοντας προηγμένα χαρακτηριστικά για βελτιωμένη εμπειρία χρήσης. Η εφαρμογή καταγράφει το χρόνο που χρειάζεται ο χρήστης για να απαντήσει σε κάθε ερώτηση της φόρμας. Η φόρμα υποστηρίζει επίσης την προβολή εικόνων με δυνατότητα ζουμ, επιτρέποντας στους χρήστες να εξετάζουν λεπτομέρειες των εικόνων. Για το backend, χρησιμοποιήσαμε μια εφαρμογή Flask και για την αποθήκευση των δεδομένων που συλλέχθηκαν, χρησιμοποιήσαμε τη MongoDB, μια NoSQL βάση δεδομένων γνωστή για την ευελιξία και την επεκτασιμότητά της.

0.4.4 Διαμόρφωση Ερωτηματολογίου

Σύνολα Δεδομένων

Για την έρευνά μας χρησιμοποιήσαμε τέσσερα σύνολα δεδομένων από διάφορους τομείς, όπως η πιστωτική βαθμολόγηση, η υγειονομική περίθαλψη και η ποινική δικαιοσύνη, για να καταδείξουμε την καθολική αξία και τη δυνατότητα εφαρμογής της ερμηνευσιμότητας στην προγνωστική μοντελοποίηση.

1. COMPAS [46]
2. German Credit [38]
3. Framingham Heart Study [11]
4. Adult Income [7]

COMPAS

Το σύνολο δεδομένων διαθέτει ποινικά μητρώα και δημογραφικά χαρακτηριστικά για 7.214 κατηγορούμενους που αποφυλακίστηκαν με εγγύηση στα πολιτειακά δικαστήρια των ΗΠΑ κατά την περίοδο 1990-2009 και χρησιμοποιείται για την αξιολόγηση της πιθανότητας υποτροπής ενός καταδικασθέντος εγκληματία.

Στόχος

Ο πρωταρχικός στόχος με αυτό το σύνολο δεδομένων είναι η ταξινόμηση των κατηγορουμένων σε δύο κατηγορίες:

- **Recid**: Κατηγορούμενοι που είναι πιθανό να υποτροπιάσουν.
- **No Recid**: Κατηγορούμενοι που δεν είναι πιθανό να υποτροπιάσουν

Προεπεξεργασία

Το αρχικό σύνολο δεδομένων περιείχε 7214 περιπτώσεις κατηγορουμένων με 53 χαρακτηριστικά που περιέγραφαν κάθε κατηγορούμενο. Ωστόσο, για τους σκοπούς της παρούσας μελέτης, αποκλείσαμε τα χαρακτηριστικά που σχετίζονται με τη διαδικασία λήψης αποφάσεων του COMPAS και κρατήσαμε μόνο τις δημογραφικές πληροφορίες και το ποινικό ιστορικό. Αυτό είχε ως αποτέλεσμα ένα τελικό σύνολο 8 χαρακτηριστικών:

- **Sex**: Το φύλο του κατηγορουμένου
- **Age**: Η ηλικία του κατηγορουμένου κατά τον χρόνο τέλεσης του εγκλήματος.
- **Race**: Η φυλή του κατηγορουμένου
- **juv_fel_count**: Ο αριθμός των κακουργηματικών κατηγοριών ανηλίκων που έχει ο κατηγορούμενος.
- **juv_misd_count**: Ο αριθμός των κατηγοριών για πλημμελήματα ανηλίκων που έχει ο κατηγορούμενος.
- **juv_other_count**: Ο αριθμός των άλλων κατηγοριών ανηλίκων που έχει ο κατηγορούμενος.
- **priors_count**: Ο αριθμός των προηγούμενων ποινικών κατηγοριών που έχει ο κατηγορούμενος.
- **c_charge_degree**: Ο βαθμός της τρέχουσας κατηγορίας

Μετά την αφαίρεση των περιττών στηλών, τα βήματα προεπεξεργασίας για αυτό το σύνολο δεδομένων περιλαμβάνουν:

- **Handling Missing Values**: Απορρίψαμε μεταβλητές με μηδενικές τιμές για να διασφαλίσουμε την ακεραιότητα των δεδομένων.
- **Feature importance**: Για να αξιολογήσουμε τη σημασία των διαφόρων χαρακτηριστικών στο σύνολο δεδομένων μας, χρησιμοποιήσαμε τον αλγόριθμο Random Forest, μια ισχυρή μέθοδο μάθησης συνόλου. Συγκεκριμένα, χρησιμοποιήσαμε έναν ταξινομητή Random Forest με 100 εκτιμητές (δέντρα). Ο ταξινομητής Random Forest υλοποιήθηκε με τη χρήση του RandomForestClassifier από τη βιβλιοθήκη scikit-learn [58].

Decision Trees Created

Για το σύνολο δεδομένων COMPAS, δημιουργήσαμε δύο μοντέλα δέντρων απόφασης χρησιμοποιώντας διαφορετικούς αλγορίθμους για να συγκρίνουμε την ερμηνευσιμότητά τους:

Μοντέλο 1: Δημιουργήθηκε με τη χρήση του DecisionTreeClassifier από την python βιβλιοθήκη scikit-learn [58] που χρησιμοποιεί μια βελτιστοποιημένη έκδοση του αλγορίθμου CART. Επιλέχθηκε το μοντέλο με την υψηλότερη ακρίβεια. Η προκύπτουσα δενδρική δομή περιελάμβανε δυαδικές διασπάσεις τυπικές για τον αλγόριθμο CART, και βάθος $d = 5$

Μοντέλο 2: Για αυτό το μοντέλο χρειάστηκε ένα επιπλέον βήμα προεπεξεργασίας των δεδομένων, η διακριτοποίηση με επίβλεψη. Το FlowOCT έχει το όριο της χρήσης μόνο δυαδικών μεταβλητών για ταξινόμηση. Για συνεχή χαρακτηριστικά έχουμε 2 επιλογές. Είτε να χρησιμοποιήσουμε την ενσωματωμένη συνάρτηση *binarize* στο πακέτο odtlearn, την υλοποίηση της οποίας μπορείτε να βρείτε στο διαδίκτυο στη διεύθυνση https://github.com/D3M-Research-Group/StrongTree, είτε να χρησιμοποιήσουμε τη διακριτοποίηση με επίβλεψη.

Για αριθμητικά χαρακτηριστικά με μεγάλο αριθμό μοναδικών τιμών, χρησιμοποιήσαμε διακριτοποίηση με επίβλεψη. Για να το υλοποιήσουμε αυτό χρησιμοποιήσαμε τον *DecisionTreeClassifier* που παρέχεται από τη βιβλιοθήκη scikit-learn της python[58] για το αριθμητικό χαρακτηριστικό Age.

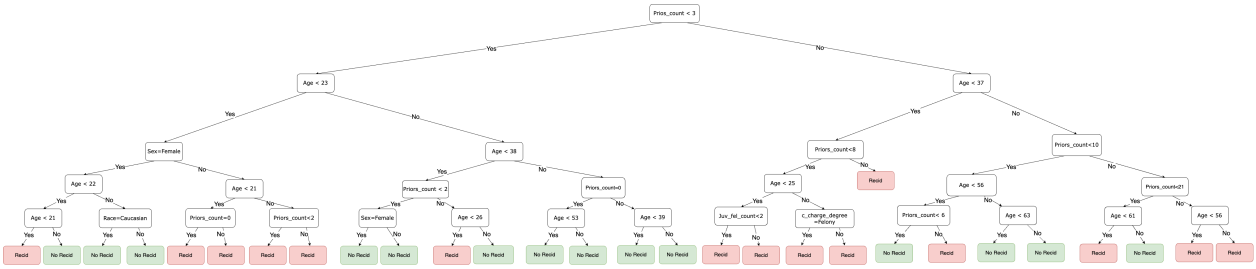

Figure 0.4.3: COMPAS CART model

Για αριθμητικά χαρακτηριστικά με σχετικά μικρό αριθμό μοναδικών τιμών, χρησιμοποιήσαμε την ενσωματωμένη συνάρτηση *binarize* από το πακέτο odtlearn. Η δυαδικοποίηση μετατρέπει τις τιμές των χαρακτηριστικών σε δυαδικές τιμές (0 ή 1) με βάση τις μοναδικές τιμές. Το μοντέλο FlowOCT δημιουργήθηκε με τη χρήση του επιλυτή Gurobi.

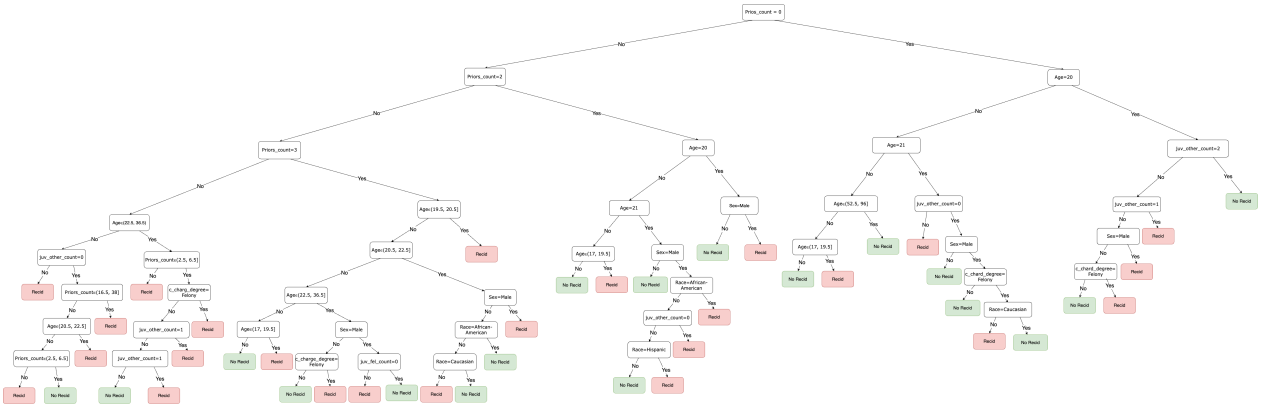

Figure 0.4.4: COMPAS FlowOCT model

Διαφορές μεταξύ των δέντρων

Όπως είναι προφανές από τα σχήματα των δέντρων, το πρώτο μοντέλο είναι ένα ισορροπημένο δέντρο απόφασης που χρησιμοποιεί διαχωρισμούς με βάση ένα μόνο κατώφλι. Το δεύτερο μοντέλο είναι imbalanced και χρησιμοποιεί διαιρεμένα χαρακτηριστικά για την ταξινόμηση. Θέλουμε να αξιολογήσουμε πώς η ισορροπία του δέντρου ενισχύει την ερμηνευσιμότητα και αν τα διαιρεμένα αριθμητικά χαρακτηριστικά είναι αρκετά περιεκτικά για τους χρήστες.

German Credit

Το σύνολο δεδομένων German Credit [38] περιλαμβάνει δημογραφικά (ηλικία, φύλο), προσωπικά (οικογενειακή κατάσταση) και οικονομικά (πιστωτικό ποσό, ποσό ελέγχου) χαρακτηριστικά από 1.000 αιτούντες πίστωσης, όπου κατηγοριοποιούνται σε καλούς vs. κακούς πελάτες ανάλογα με τον πιστωτικό τους κίνδυνο.

Στόχος

Ο πρωταρχικός στόχος της ανάλυσης του συνόλου δεδομένων German Credit είναι η ταξινόμηση των ατόμων σε δύο κατηγορίες:

- **Good Credit**: Άτομα που θεωρούνται φερέγγυα και ενέχουν χαμηλότερο κίνδυνο για τα χρηματοπιστωτικά ιδρύματα.
- **Bad Credit**: Άτομα που δεν θεωρούνται φερέγγυα και ενέχουν υψηλότερο κίνδυνο για τα χρηματοπιστωτικά ιδρύματα.

Προεπεξεργασία

Το αρχικό σύνολο δεδομένων περιείχε 1000 αιτούντες πιστώσεων με 9 χαρακτηριστικά που περιέγραφαν κάθε αιτούντα.

- **Age**: Η ηλικία του ατόμου σε έτη
- **Sex**: Το φύλο του ατόμου
- **Job**: Το είδος της εργασίας του ατόμου
- **Housing**: Το είδος της κατοικίας στην οποία διαμένει το άτομο
- **Saving accounts**: Το ποσό των αποταμιεύσεων που διαθέτει το άτομο
- **Checking account**: Το ποσό στον τρεχούμενο λογαριασμό του ατόμου
- **Credit amount**: Το ποσό της πίστωσης που έχει ζητήσει το άτομο
- **Duration**: Η διάρκεια της πίστωσης σε μήνες
- **Purpose**: Ο σκοπός για τον οποίο ζητείται η πίστωση

Μετά την αφαίρεση των περιττών στηλών, τα βήματα προεπεξεργασίας για αυτό το σύνολο δεδομένων περιελάμβαναν:

- **Handling Missing Values**: Απορρίψαμε μεταβλητές με μηδενικές τιμές για να διασφαλίσουμε την ακεραιότητα των δεδομένων.
- **Feature importance**: Για να αξιολογήσουμε τη σημασία των διαφόρων χαρακτηριστικών στο σύνολο δεδομένων μας, χρησιμοποιήσαμε τον αλγόριθμο Random Forest, μια ισχυρή μέθοδο μάθησης συνόλου. Συγκεκριμένα, χρησιμοποιήσαμε έναν ταξινομητή Random Forest με 100 εκτιμητές (δέντρα). Ο ταξινομητής Random Forest υλοποιήθηκε με τη χρήση του RandomForestClassifier από τη βιβλιοθήκη scikit-learn [58].

Δέντρα αποφάσεων που δημιουργήθηκαν

Για το σύνολο δεδομένων German Credit, δημιουργήσαμε δύο μοντέλα δέντρων απόφασης χρησιμοποιώντας διαφορετικούς αλγορίθμους για να συγκρίνουμε την ερμηνευσιμότητά τους:

Μοντέλο 1: Για την υλοποίηση του ταξινομητή δέντρων απόφασης, χρησιμοποιήσαμε τη βιβλιοθήκη *c45-decision-tree*, η οποία είναι διαθέσιμη στο PyPI [59]. Αυτή η βιβλιοθήκη χειρίζεται μόνο κατηγορηματικά δεδομένα και υλοποιεί multiway splits. Το δέντρο που προκύπτει παρουσιάζεται στο Σχήμα 0.4.5.

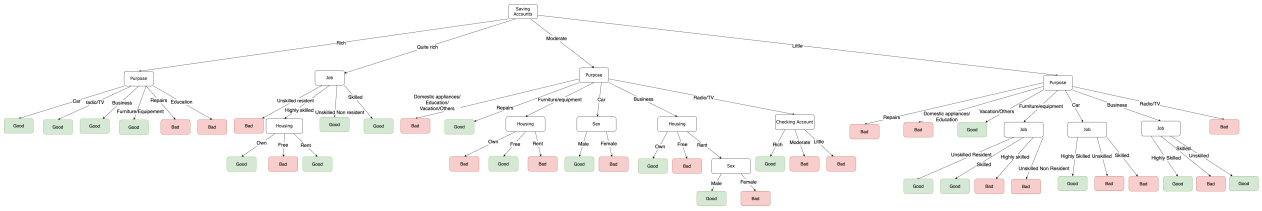

Figure 0.4.5: Μοντέλο German Credit C4.5

Μοντέλο 2: Για αυτό το μοντέλο χρησιμοποιήσαμε την προηγμένη έκδοση του C4.5 την οποία υλοποιήσαμε με ομαδοποίηση των μη σημαντικών τιμών χαρακτηριστικών σε κλάδους με ετικέτα *Other*. Το δέντρο που προέκυψε παρουσιάζεται στην εικόνα 0.4.6.

Διαφορές μεταξύ των δέντρων

Όπως είναι προφανές από τα σχήματα των δέντρων, το πρώτο μοντέλο είναι ένα ευρύ δέντρο απόφασης που χρησιμοποιεί multiway splits με βάση κατηγορηματικά χαρακτηριστικά. Το δεύτερο μοντέλο είναι ένα βαθύ δέντρο απόφασης και χρησιμοποιεί ομαδοποίηση για τις τιμές των χαρακτηριστικών με την ετικέτα *Other*. Θέλουμε να

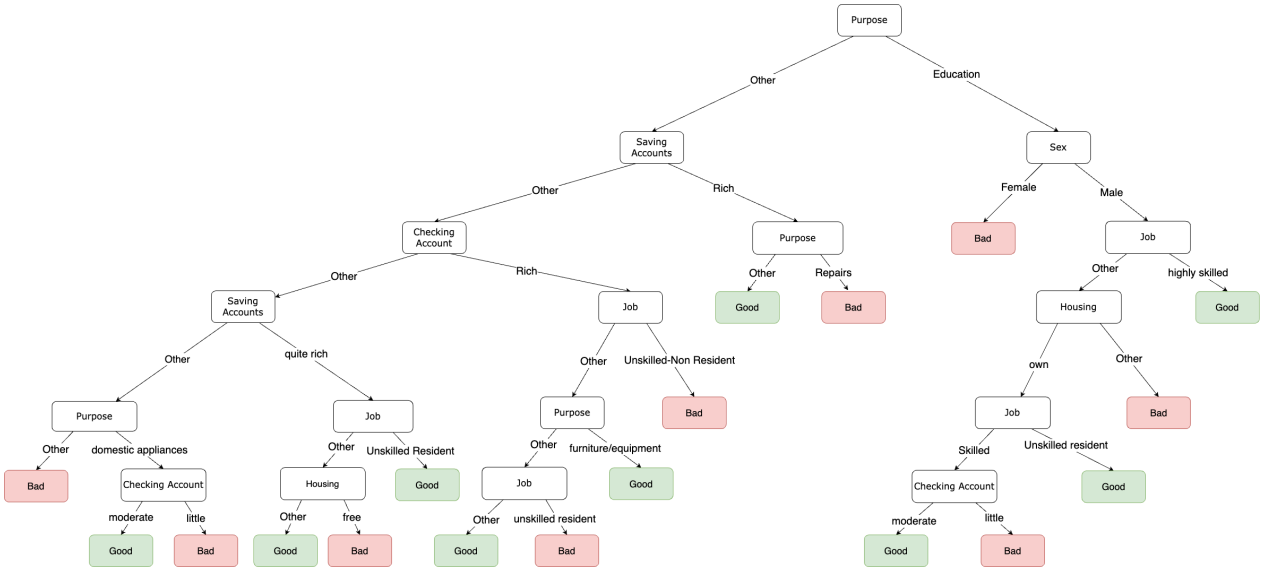

Figure 0.4.6: German Credit C4.5 Advanced model

αξιολογήσουμε αν η πολυδρομική διάσπαση ενισχύει την ερμηνευσιμότητα του δέντρου και αν οι ομαδοποιημένες τιμές προκαλούν σύγχυση στους χρήστες.

Framingham Heart Study

Το σύνολο δεδομένων Framingham Heart Study [11] προέρχεται από μια πρωτοποριακή διαχρονική μελέτη που ξεκίνησε το 1948 στο Framingham της Μασαχουσέτης, με πρωταρχικό στόχο τον εντοπισμό κοινών παραγόντων που συμβάλλουν στην καρδιαγγειακή νόσο. Η μελέτη, στην οποία αρχικά συμμετείχαν περισσότεροι από 5.000 συμμετέχοντες, επεκτάθηκε σημαντικά ώστε να συμπεριλάβει διαδοχικές γενιές, επιτρέποντας στους ερευνητές να διερευνήσουν γενετικούς, περιβαλλοντικούς και παράγοντες του τρόπου ζωής που επηρεάζουν την υγεία της καρδιάς.

Στόχος Ο πρωταρχικός στόχος της ανάλυσης του συνόλου δεδομένων της Framingham Heart Study είναι η ταξινόμηση των ατόμων σε δύο κατηγορίες με βάση τον 10ετή κίνδυνο εμφάνισης στεφανιαίας νόσου (CHD):

- **Yes**: Άτομα που διατρέχουν κίνδυνο να αναπτύξουν CHD εντός 10 ετών.
- **No**: Άτομα που δεν κινδυνεύουν να αναπτύξουν CHD εντός 10 ετών.

Προεπεξεργασία

Το αρχικό σύνολο δεδομένων περιείχε 4240 ασθενείς με 14 χαρακτηριστικά που περιέγραφαν κάθε ασθενή.

- **Male**: Φύλο του συμμετέχοντα (1 για άνδρα, 0 για γυναίκα).
- **Age**: Ηλικία του συμμετέχοντα σε έτη.
- **Education**: Επίπεδο εκπαίδευσης του συμμετέχοντα. Συνήθως κωδικοποιείται αριθμητικά.
- **CurrentSmoker**: Δείκτης του κατά πόσον ο συμμετέχων είναι σημερινός καπνιστής
- **cigsPerDay**: Αριθμός τσιγάρων που καπνίζει ανά ημέρα ο συμμετέχων.
- **BPMeds**: Δείκτης του κατά πόσον ο συμμετέχων λαμβάνει φάρμακα για την αρτηριακή πίεση
- **prevalentStroke**: Δείκτης του κατά πόσον ο συμμετέχων έχει υποστεί εγκεφαλικό επεισόδιο
- **prevalentHyp**: Δείκτης του κατά πόσον ο συμμετέχων έχει υπέρταση
- **diabetes**: Δείκτης του κατά πόσον ο συμμετέχων έχει διαβήτη

- **totChol**: Επίπεδο ολικής χοληστερόλης σε mg/dL.
- **sysBP**: Μέτρηση της συστολικής αρτηριακής πίεσης σε mmHg.
- **diaBP**: Μέτρηση διαστολικής αρτηριακής πίεσης σε mmHg.
- **BMI**: Δείκτης μάζας σώματος που υπολογίζεται από το ύψος και το βάρος (kg/m^2).
- **heartRate**: Καρδιακός ρυθμός σε παλμούς ανά λεπτό.
- **glucose**: Επίπεδο γλυκόζης στο αίμα.

Τα βήματα προεπεξεργασίας για αυτό το σύνολο δεδομένων περιλαμβάνουν:

- **Handling Missing Values**: Απορρίψαμε μεταβλητές με μηδενικές τιμές για να διασφαλίσουμε την ακεραιότητα των δεδομένων.
- **Feature Importance**: Για να αξιολογήσουμε τη σημασία των διαφόρων χαρακτηριστικών στο σύνολο δεδομένων μας, χρησιμοποιήσαμε τον αλγόριθμο Random Forest, μια ισχυρή μέθοδο μάθησης συνόλου. Κρατήσαμε μόνο τα 10 κορυφαία χαρακτηριστικά με βάση τη σημαντικότητά τους για να απλοποιήσουμε τα δέντρα που προέκυψαν.

Δέντρα αποφάσεων που δημιουργήθηκαν

Για το σύνολο δεδομένων Framingham Heart Study, δημιουργήσαμε δύο μοντέλα δέντρων απόφασης χρησιμοποιώντας διαφορετικούς αλγορίθμους για να συγκρίνουμε την ερμηνευσιμότητά τους:

Μοντέλο 1: Δημιουργήθηκε με τη χρήση του DecisionTreeClassifier από τη βιβλιοθήκη scikit-learn python [58] που χρησιμοποιεί μια βελτιστοποιημένη έκδοση του αλγορίθμου CART. Επιλέχθηκε το μοντέλο με την υψηλότερη ακρίβεια. Η προκύπτουσα δενδρική δομή περιελάμβανε δυαδικές διασπάσεις τυπικές για τον αλγόριθμο CART και βάθος $d = 9$. Το δέντρο που προέκυψε παρουσιάζεται στην εικόνα 0.4.7.

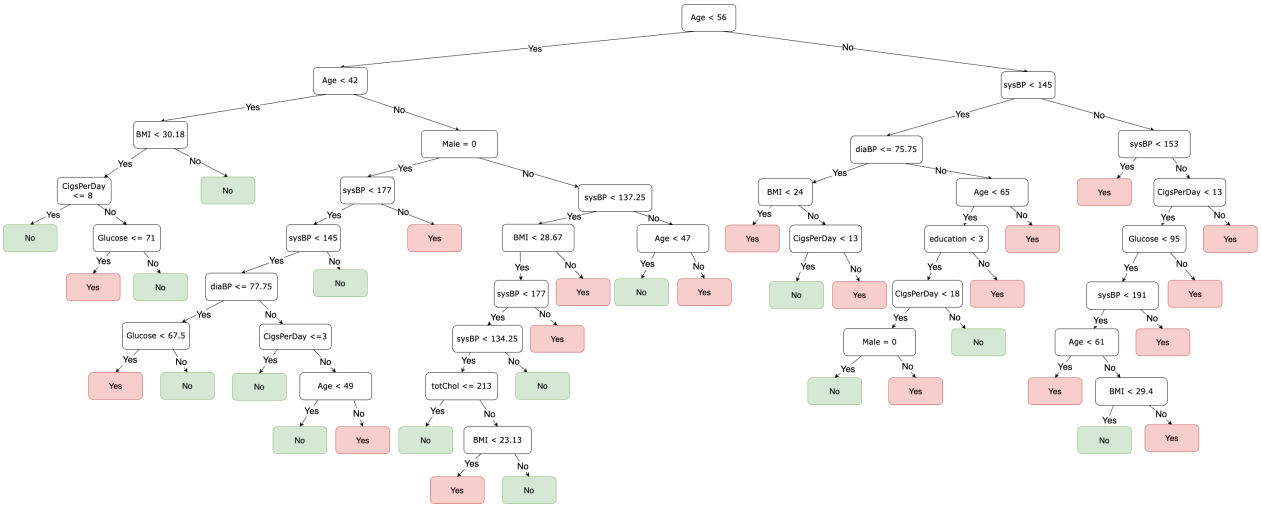

Figure 0.4.7: Μοντέλο Framingham CART

Μοντέλο 2: Για αυτό το μοντέλο χρειαζόταν ένα επιπλέον βήμα προεπεξεργασίας των δεδομένων, η επαγωγή γνώσης τομέα. Η χρήση γνώσης τομέα μπορεί να βοηθήσει στη μετατροπή αριθμητικών μεταβλητών σε κατηγορηματικές κάνοντας το δέντρο που προκύπτει πιο ερμηνεύσιμο από τον άνθρωπο.

Μετά την ενσωμάτωση της γνώσης τομέα στο σύνολο δεδομένων χρησιμοποιήσαμε το FlowOCT για να εξάγουμε το δέντρο απόφασης που προέκυψε. Για να χρησιμοποιήσουμε το FlowOCT έπρεπε πρώτα να δυαδικοποιήσουμε τα δεδομένα χρησιμοποιώντας την ενσωματωμένη συνάρτηση binarize του πακέτου odtlearn. Το δέντρο που προέκυψε παρουσιάζεται παρακάτω στο Σχήμα 0.4.8.

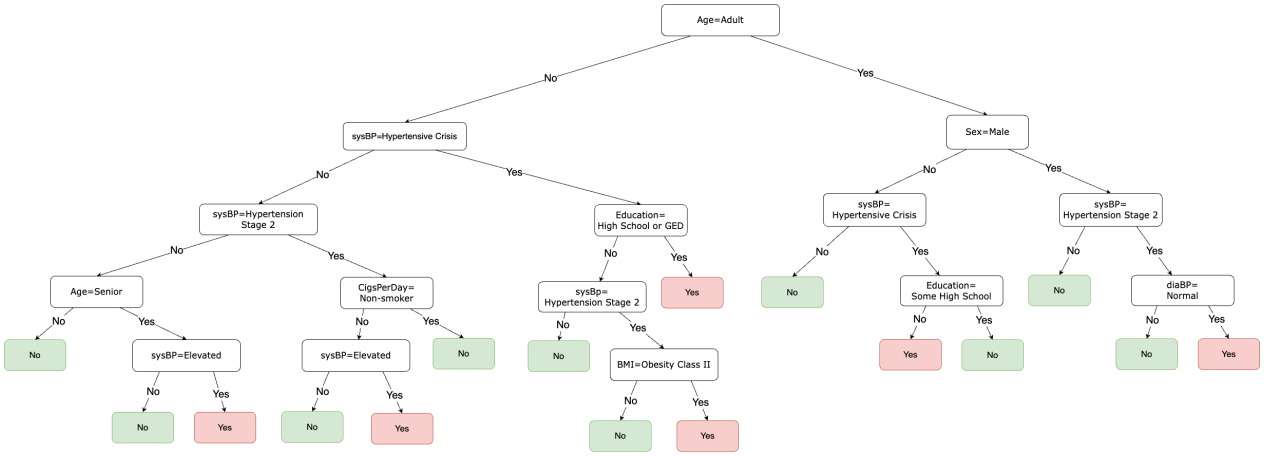

Figure 0.4.8: Framingham Categorical model

Διαφορές μεταξύ των δέντρων

Όπως είναι φανερό από τα σχήματα των δέντρων, το πρώτο μοντέλο είναι ένα βαθύ δέντρο απόφασης το οποίο χρησιμοποιεί συνεχή χαρακτηριστικά για την ταξινόμηση. Το δεύτερο μοντέλο είναι ένα μικρότερο δέντρο απόφασης με κατηγορηματικά χαρακτηριστικά Θέλουμε να αξιολογήσουμε αν τα κατηγορηματικά χαρακτηριστικά ενισχύουν την ερμηνευσιμότητα του δέντρου.

Adult Income

Το σύνολο δεδομένων Adult Income [7] περιέχει δημογραφικά (π.χ. ηλικία, φυλή και φύλο), εκπαιδευτικά (πτυχίο),εργασιακά (επάγγελμα, ώρες εβδομαδιαίως), προσωπικά (οικογενειακή κατάσταση, σχέση) και οικονομικά (κεφαλαιακά κέρδη/απώλειες) χαρακτηριστικά για 45.222 άτομα. Ο στόχος είναι να προβλεφθεί αν το εισόδημα ενός ατόμου υπερβαίνει τα 50 χιλ. δολάρια ετησίως ή όχι. Περιέχει 14 χαρακτηριστικά.

Στόχος

Ο πρωταρχικός στόχος της ανάλυσης του συνόλου δεδομένων για το εισόδημα των ενηλίκων είναι η ταξινόμηση των ατόμων σε δύο κατηγορίες με βάση το ετήσιο εισόδημά τους:

- $<=$**50K**: Άτομα των οποίων το ετήσιο εισόδημα είναι 50.000 δολάρια ή λιγότερο.
- $>$**50K**: Άτομα των οποίων το ετήσιο εισόδημα υπερβαίνει τα 50.000 δολάρια.

Προεπεξεργασία

Το αρχικό σύνολο δεδομένων περιείχε 47621 εγγραφές ατόμων με 14 χαρακτηριστικά που περιέγραφαν κάθε άτομο.

- **Age**: Η ηλικία του ατόμου.
- **Workclass**: Το είδος της απασχόλησης.
- **Education**: Το υψηλότερο επίπεδο εκπαίδευσης που έχει αποκτηθεί.
- **Education-num**: Ο αριθμός των ετών εκπαίδευσης.
- **Marital-status**: Η οικογενειακή κατάσταση του ατόμου.
- **Occupation**: Το είδος της εργασίας.
- **Relationship**: Σχέση με το νοικοκυριό.
- **Race**: Η φυλή του ατόμου.
- **Sex**: Το φύλο του ατόμου.

- **Capital-gain**: Κεφαλαιακά κέρδη.
- **Capital-loss**: Απώλειες κεφαλαίου.
- **Hours-per-week**: Ώρες εργασίας ανά εβδομάδα.
- **Native-country**: Χώρα καταγωγής.

Τα βήματα προεπεξεργασίας για αυτό το σύνολο δεδομένων περιλάμβαναν:

- **Handling Missing Values**: Απορρίψαμε μεταβλητές με μηδενικές τιμές για να διασφαλίσουμε την ακεραιότητα των δεδομένων.
- **Balance Dataset**: Το σύνολο δεδομένων παρουσιάζει συχνά ανισορροπία, με σημαντικά περισσότερες περιπτώσεις ατόμων που κερδίζουν λιγότερα από 50.000 δολάρια σε σύγκριση με εκείνα που κερδίζουν περισσότερα. Η εξισορρόπηση του συνόλου δεδομένων, με υποδειγματοληψία της πλειοψηφικής κλάσης, βελτιώνει την ακρίβεια, την ευαισθησία και την ειδικότητα του μοντέλου, διασφαλίζει ότι το μοντέλο μαθαίνει να αναγνωρίζει μοτίβα και στις δύο κλάσεις, οδηγώντας σε δικαιότερες προβλέψεις και καλύτερη γενίκευση σε νέα δεδομένα.
- **Feature Importance**: Για να αξιολογήσουμε τη σημασία των διαφόρων χαρακτηριστικών στο σύνολο δεδομένων μας, χρησιμοποιήσαμε τον αλγόριθμο Random Forest, μια ισχυρή μέθοδο μάθησης συνόλου.
- **Encoding Categorical Features**: Χρησιμοποιήσαμε one-hot encoding για τα κατηγορηματικά χαρακτηριστικά για να τα μετατρέψουμε σε μορφή κατάλληλη για αλγορίθμους μηχανικής μάθησης.

Δέντρα αποφάσεων που δημιουργήθηκαν

Για το σύνολο δεδομένων Adult Income, δημιουργήσαμε δύο μοντέλα δέντρων απόφασης χρησιμοποιώντας διαφορετικούς αλγορίθμους για να συγκρίνουμε την ερμηνευσιμότητά τους:

Μοντέλο 1: Δημιουργήθηκε με τη χρήση του DecisionTreeClassifier από τη βιβλιοθήκη scikit-learn python [58] που χρησιμοποιεί μια βελτιστοποιημένη έκδοση του αλγορίθμου CART. Επιλέχθηκε το μοντέλο με την υψηλότερη ακρίβεια. Η προκύπτουσα δενδρική δομή περιελάμβανε δυαδικές διασπάσεις τυπικές για τον αλγόριθμο CART, και βάθος $d = 6$

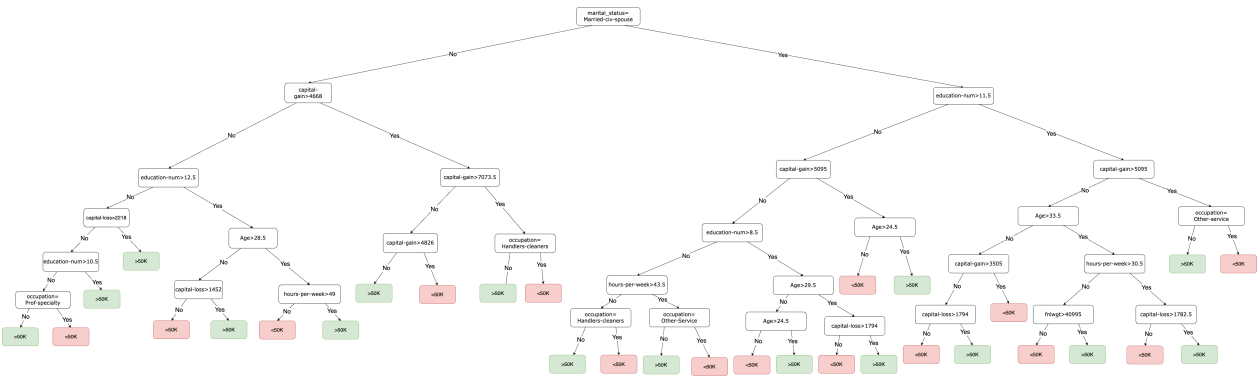

Figure 0.4.9: Adult Income CART wide model

Μοντέλο 2: Για αυτό το μοντέλο χρησιμοποιήσαμε επίσης το CART αλλά επιλέξαμε ένα πιο βαθύ δέντρο με βάθος $d = 8$.

Διαφορές μεταξύ των δέντρων

Όπως είναι προφανές από τα σχήματα των δέντρων, το πρώτο μοντέλο είναι ένα ευρύ δέντρο απόφασης και το δεύτερο μοντέλο είναι βαθύτερο. Θέλουμε να αξιολογήσουμε αν τα πλατιά δέντρα είναι πιο ερμηνεύσιμα από τα βαθιά δέντρα.

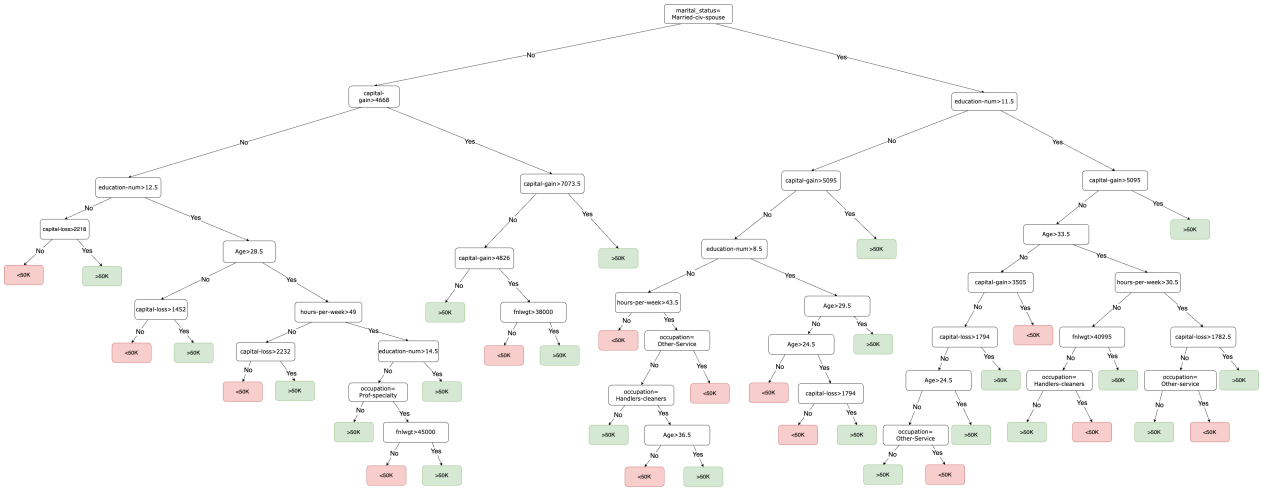

Figure 0.4.10: Βαθύ μοντέλο CART Adult Income

0.4.5 Αποτελέσματα

Γενικά Αποτελέσματα

Όπως βλέπουμε στα σχήματα 0.4.11, 0.4.12, 0.4.13, 0.4.14 έχουμε υψηλή ακρίβεια όπως περιμέναμε. Το σύνολο δεδομένων COMPAS έχει τη χαμηλότερη ακρίβεια καθώς ήταν το πρώτο σύνολο δεδομένων που οι χρήστες ταξινόμησαν.

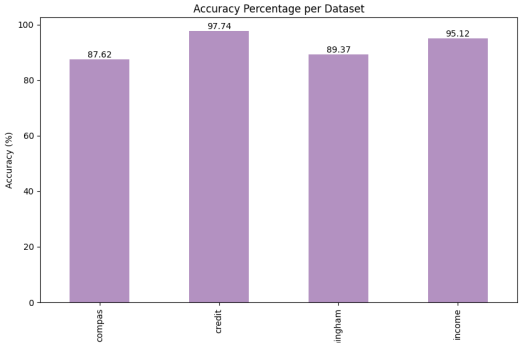

Figure 0.4.11: Ακρίβεια ανά σύνολο δεδομένων

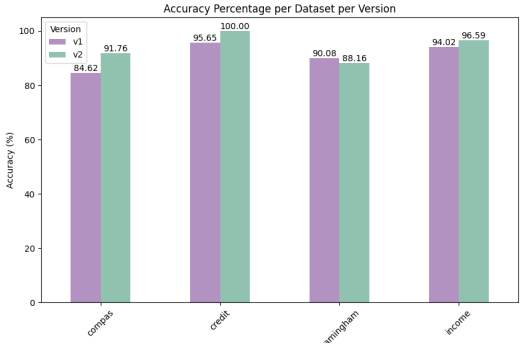

Figure 0.4.12: Ακρίβεια ανά σύνολο δεδομένων ανά έκδοση

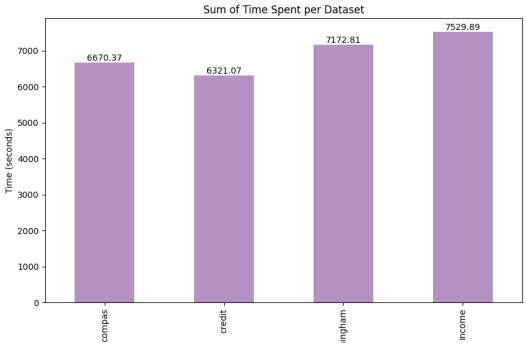

Figure 0.4.13: Χρόνος που απαιτείται ανά σύνολο δεδομένων

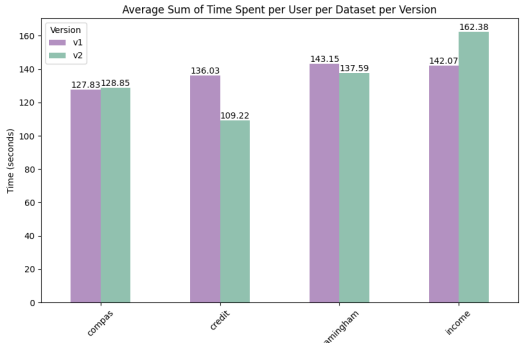

Figure 0.4.14: Χρόνος που απαιτείται ανά σύνολο δεδομένων ανά έκδοση

Στα σχήματα 0.4.15, 0.4.16, 0.4.17 απεικονίζουμε την αξιοπιστία των απαντήσεων, τη σαφήνεια των μονοπατιών και την απλότητα της δενδρικής δομής.

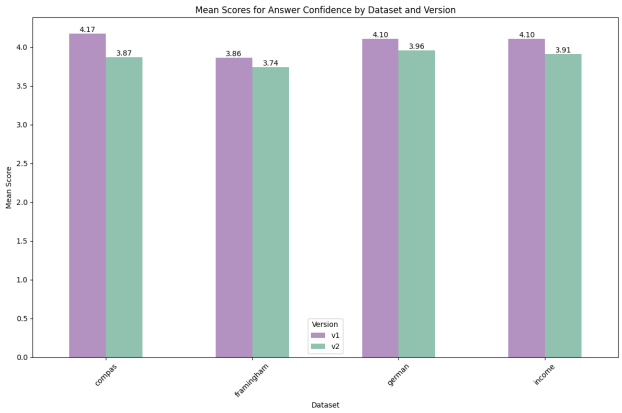

Figure 0.4.15: Εμπιστοσύνη απαντήσεων ανά σύνολο δεδομένων ανά έκδοση

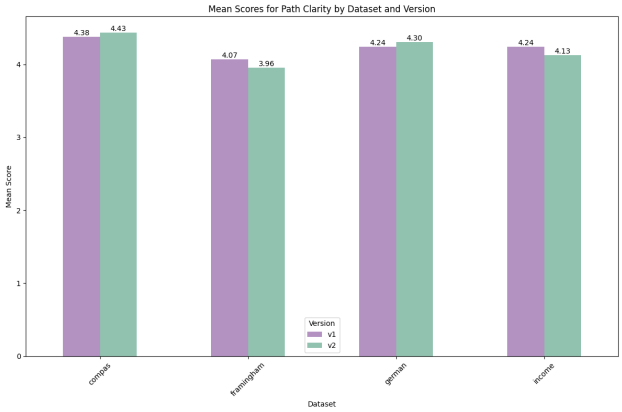

Figure 0.4.16: Σαφήνεια Δεδομένων ανά σύνολο δεδομένων ανά έκδοση

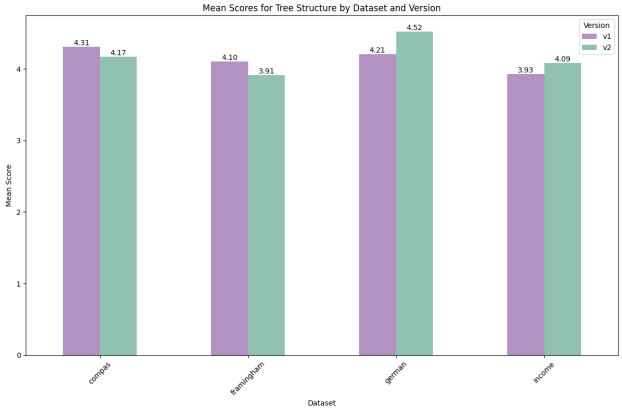

Figure 0.4.17: Απλότητα της δενδρικής δομής ανά σύνολο δεδομένων ανά έκδοση

Θα αναλύσουμε τα αποτελέσματα ανά σύνολο δεδομένων και θα εξάγουμε συμπεράσματα συγκρίνοντας τα δέντρα απόφασης και τα παρακάτω αποτελέσματα.

COMPAS

Με βάση τα αποτελέσματα που παρουσιάζονται στα Σχήματα 0.4.18, 0.4.19 και 0.4.20, παρατηρούμε ότι το δέντρο απόφασης FlowOCT οδηγεί σε αύξηση της ακρίβειας, αν και με μικρή αύξηση του χρόνου που απαιτείται για την ταξινόμηση.Επιπλέον, τα δεδομένα αποκαλύπτουν ότι οι χρήστες παρουσιάζουν χαμηλότερη εμπιστοσύνη στις απαντήσεις στο δεύτερο μοντέλο, παρά το γεγονός ότι οι απαντήσεις τους είναι σωστές. Η σαφήνεια του μονοπατιού είναι καλύτερη στο δεύτερο μοντέλο λόγω των συντομότερων διαδρομών- ωστόσο, η δομή του δέντρου είναι πιο πολύπλοκη επειδή τα διαιρεμένα χαρακτηριστικά δεν είναι διαισθητικά για τους χρήστες. Από τις περιγραφές των δέντρων από τους χρήστες, είναι προφανές ότι οι χρήστες επικεντρώθηκαν στην προφανή προκατάληψη κατά των Αφροαμερικανών στο σύνολο δεδομένων. Το μοντέλο με τα binned χαρακτηριστικά γίνεται αντιληπτό ως πιο περίπλοκο από τους χρήστες, ενώ το αριθμητικό μοντέλο υποφέρει από το μειονέκτημα της επανάληψης των χαρακτηριστικών.

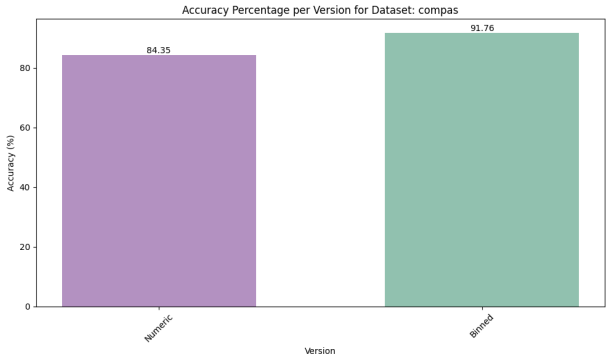

Figure 0.4.18: Ακρίβεια COMPAS

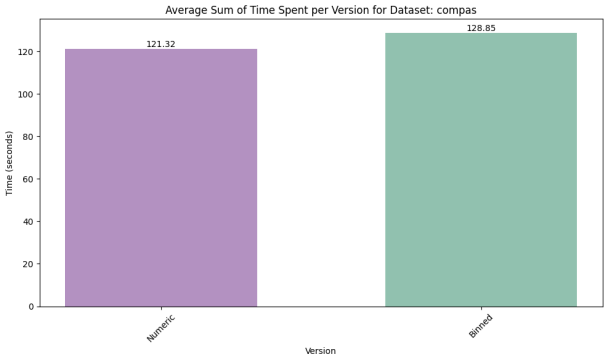

Figure 0.4.19: COMPAS Time needed

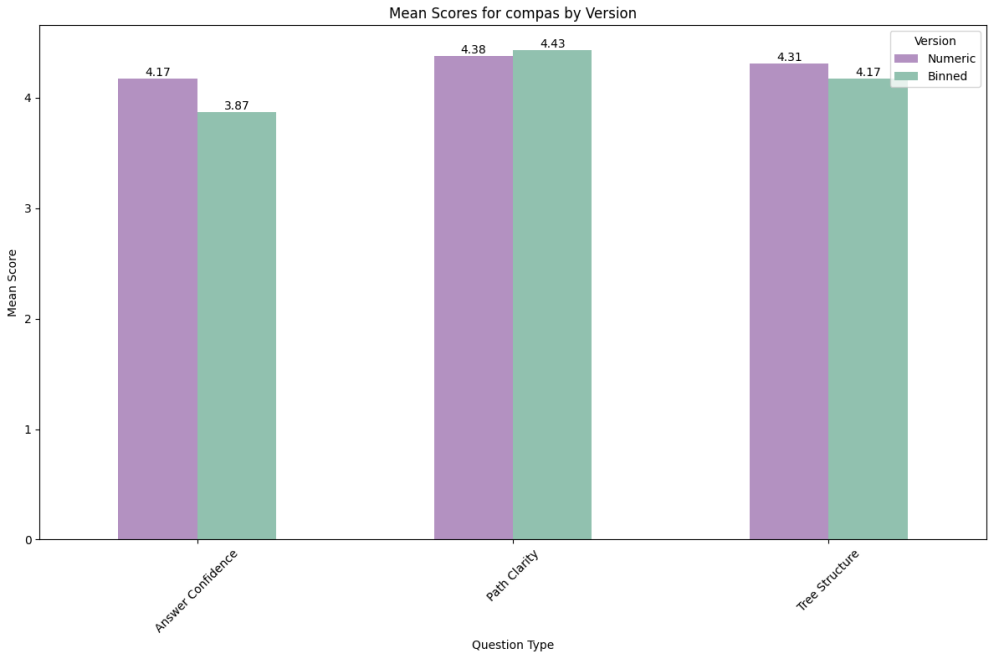

Figure 0.4.20: Μετρήσεις αξιολόγησης COMPAS

German Credit

Με βάση τα αποτελέσματα που παρουσιάζονται στα Σχήματα 0.4.21, 0.4.22 και 0.4.23, παρατηρούμε ότι το multiway δέντρο απόφασης οδηγεί σε τέλειο σκορ ακρίβειας με σημαντική μείωση του χρόνου ταξινόμησης.

Επιπλέον, τα δεδομένα αποκαλύπτουν ότι οι χρήστες παρουσιάζουν χαμηλότερη εμπιστοσύνη στις απαντήσεις τους στο δεύτερο μοντέλο, παρά την υψηλή ακρίβεια των απαντήσεών τους. Αυτό συμβαίνει λόγω του μεγάλου συντελεστή διακλάδωσης (3,8 αντί 2 για το δυαδικό μοντέλο) κάθε κόμβου. Η σαφήνεια των διαδρομών και η απλότητα είναι καλύτερη στο δεύτερο μοντέλο λόγω των μικρότερων διαδρομών και της ευρύτερης δομής του δέντρου. Από τις περιγραφές των χρηστών για τα δέντρα, είναι προφανές ότι οι χρήστες επικεντρώθηκαν στην προφανή προκατάληψη έναντι των γυναικών στο σύνολο δεδομένων. Το δυαδικό δέντρο, χρησιμοποιώντας τον όρο "Other", επέτρεψε στους χρήστες να βγάλουν κάποια συμπεράσματα σχετικά με τη διαδικασία λήψης αποφάσεων. Ενώ ο πολυδιάστατος διαχωρισμός θεωρείται πιο διαισθητικός σε σύγκριση με τα binary χαρακτηριστικά στο σύνολο δεδομένων COMPAS, θεωρείται επίσης πιο πολύπλοκος λόγω του μεγάλου συντελεστή διακλάδωσης.

Figure 0.4.21: German Credit Ακρίβεια

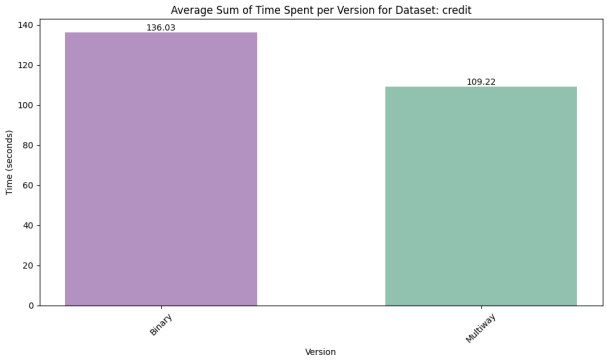

Figure 0.4.22: German Credit time needed

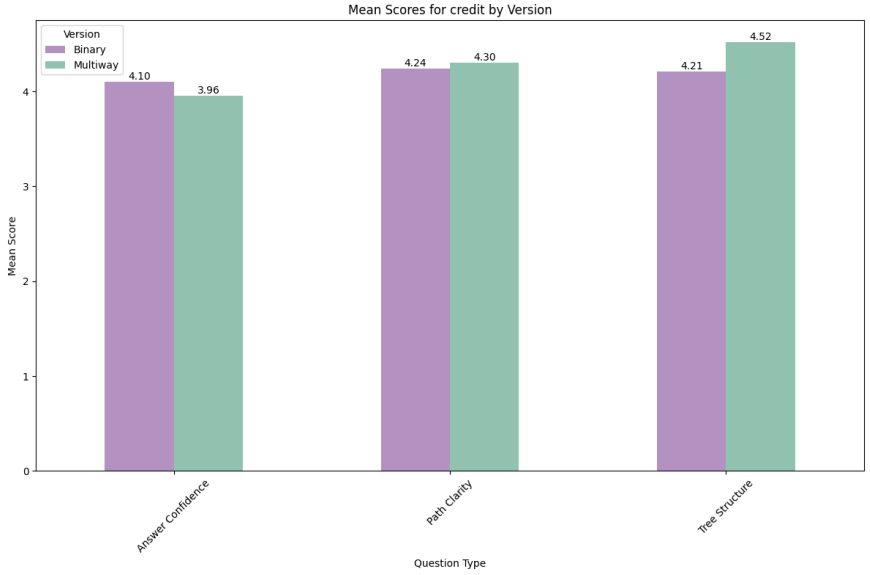

Figure 0.4.23: Μετρήσεις αξιολόγησης German Credit

Framingham Heart Study

Σε αυτή την ενότητα, συγκρίνουμε τα δέντρα απόφασης που δημιουργήθηκαν Με βάση τα αποτελέσματα που παρουσιάζονται στα Σχήματα 0.4.24, 0.4.25 και 0.4.26, παρατηρούμε ότι το κατηγορηματικό δέντρο απόφασης επιτυγχάνει υψηλή βαθμολογία ακρίβειας με μικρή αύξηση του χρόνου ταξινόμησης.

Επιπλέον, τα δεδομένα αποκαλύπτουν ότι τα κατηγορημκά χαρακτηριστικά οδηγούν σε υψηλότερες βαθμολογίες σε όλες τις μετρικές: εμπιστοσύνη στην απάντηση, σαφήνεια μονοπατιών και απλότητα της δομής του δέντρου.

Από τις περιγραφές των δέντρων από τους χρήστες, είναι προφανές ότι οι χρήστες δυσκολεύονταν να κατανοήσουν τους ιατρικούς όρους, καθώς δεν είναι ειδικοί στον τομέα. Το αριθμητικό δέντρο γίνεται αντιληπτό ως πιο πολύπλοκο λόγω του βάθους και της επαναληπτικότητάς του. Ενώ και τα δύο δέντρα θεωρούνται δύσκολα στην κατανόηση, το κατηγορηματικό δέντρο φαίνεται να είναι πιο σαφές.

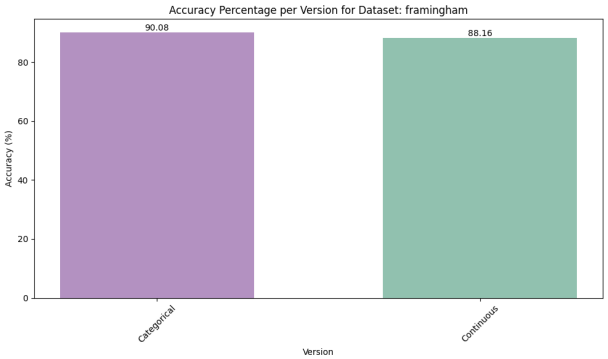

Figure 0.4.24: Framingham Heart Study ακρίβεια

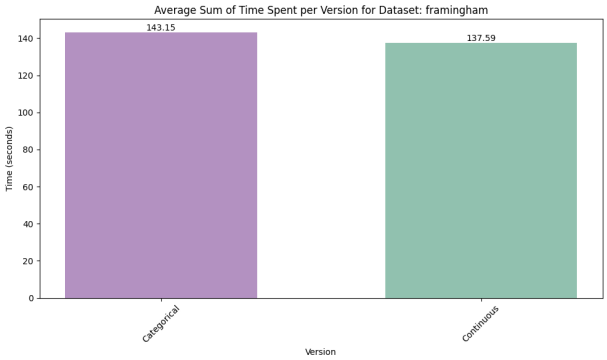

Figure 0.4.25: Framingham Heart Study Time needed

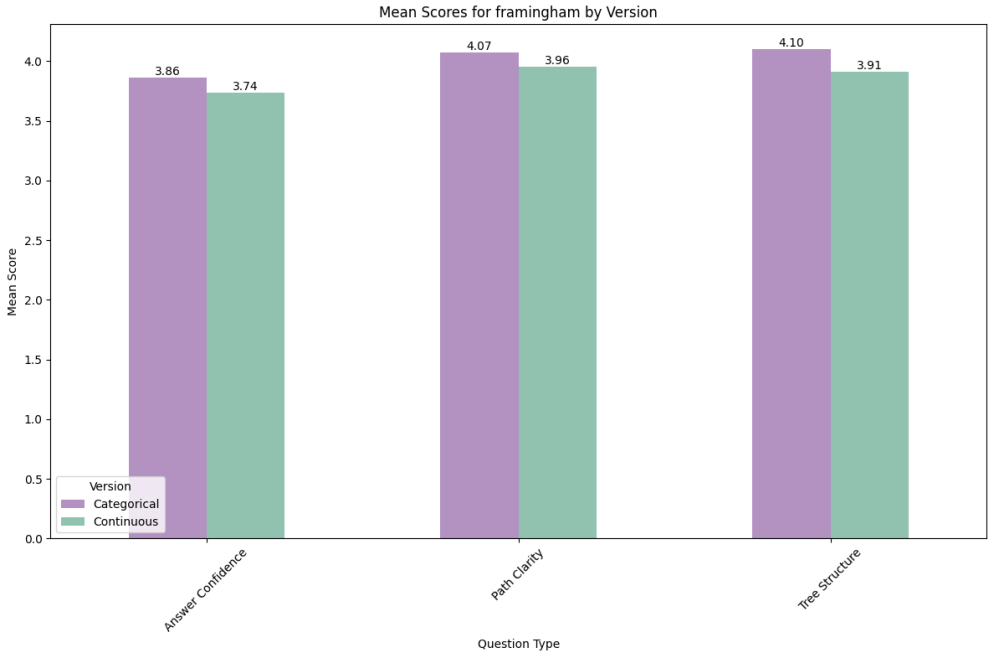

Figure 0.4.26: Μετρήσεις αξιολόγησης Framingham

Adult Income

Με βάση τα αποτελέσματα που παρουσιάζονται στα Σχήματα 0.4.27, 0.4.28 και 0.4.29, παρατηρούμε ότι το ευρύτερο δέντρο απόφασης οδηγεί σε αύξηση της ακρίβειας, αν και με μικρή αύξηση του χρόνου που απαιτείται για την ταξινόμηση. Επιπλέον, τα δεδομένα αποκαλύπτουν ότι οι χρήστες παρουσιάζουν χαμηλότερη εμπιστοσύνη στις απαντήσεις τους στο ευρύ μοντέλο, παρά την υψηλή ακρίβεια των απαντήσεών τους. Οι διαφορές στη σαφήνεια της διαδρομής και στην απλότητα της δομής είναι ελάχιστες και θα μπορούσαν να αποδοθούν σε στατιστικό σφάλμα. Από τις περιγραφές των χρηστών για τα δέντρα, είναι προφανές ότι το μέγεθος του δέντρου σημειώνεται από τους χρήστες. Πολλοί χρήστες που χρησιμοποίησαν το βαθύ δέντρο σχολίασαν το μέγεθός του και πώς αυτό δυσκόλευε την κατανόηση του δέντρου.

Figure 0.4.27: Adult Income ακρίβεια

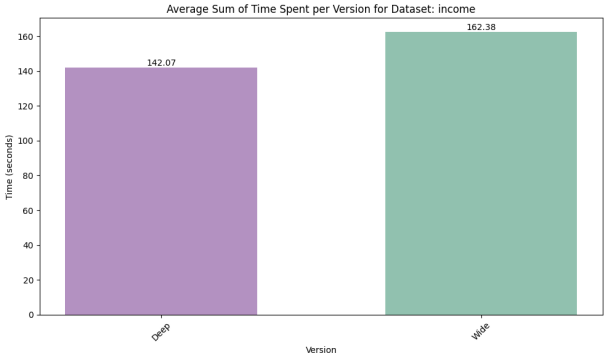

Figure 0.4.28: Adult Income Time needed

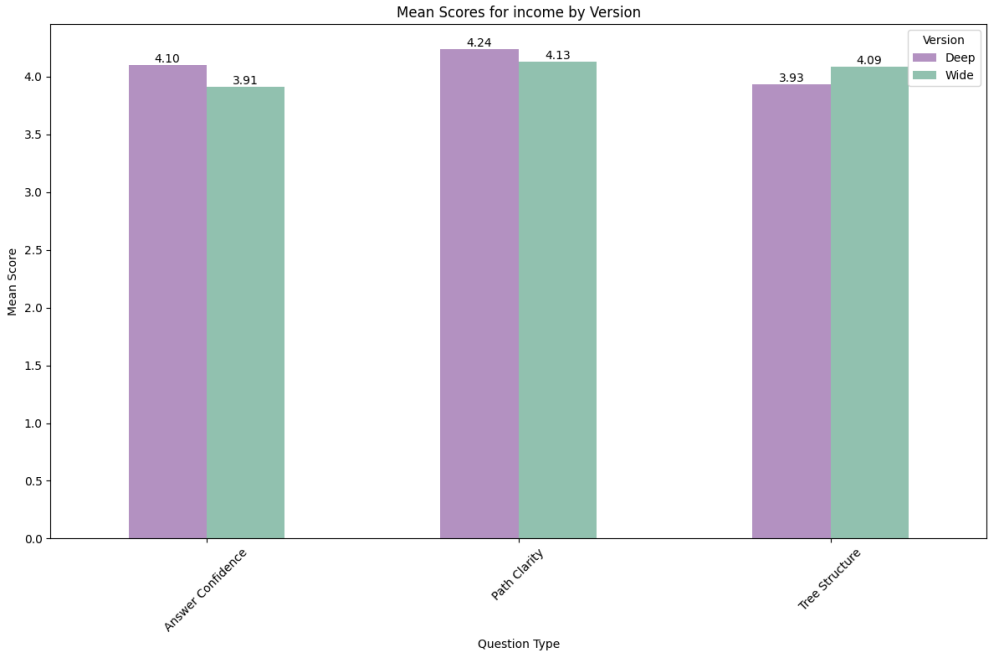

Figure 0.4.29: Μετρήσεις αξιολόγησης Adult Income

0.4.6 Συμπεράσματα

Αξιολόγηση της δομής του δέντρου

Με βάση τα αποτελέσματα της μελέτης μας για τους χρήστες, παρατηρήσαμε αξιοσημείωτες διαφορές στην απόδοση και την ερμηνευσιμότητα των δέντρων απόφασης σε διάφορα σύνολα δεδομένων και δομές δέντρων. Η μελέτη αποκάλυψε ότι τα δέντρα αποφάσεων που χρησιμοποιούν κατηγορηματικές μεταβλητές πέτυχαν γενικά υψηλότερη ακρίβεια και καλύτερες μετρικές ερμηνευσιμότητας, όπως η εμπιστοσύνη στην απάντηση, η σαφήνεια της διαδρομής και η απλότητα της δομής του δέντρου. Αυτό ήταν εμφανές στο σύνολο δεδομένων Framingham Heart Study, όπου το κατηγορηματικό δέντρο απόφασης υπερείχε του αριθμητικού στις αξιολογήσεις των χρηστών, παρά τη μικρή αύξηση του χρόνου ταξινόμησης.

Αντίθετα, τα δέντρα αποφάσεων με αριθμητική διάσπαση, ιδίως αυτά που είναι βαθύτερα, συχνά οδηγούσαν σε επανάληψη χαρακτηριστικών και αυξημένη πολυπλοκότητα, καθιστώντας τα πιο δυσνόητα για τους χρήστες. Αυτό αναδείχθηκε στο σύνολο δεδομένων Adult Income, όπου το βαθύτερο δέντρο αναδείχθηκε για το μέγεθος και την πολυπλοκότητά του, επηρεάζοντας την εμπιστοσύνη των χρηστών παρά την ακρίβειά του. Γενικά παρατηρήθηκε ότι ο περιορισμός των ατόμων στο βάθος των 7 κόμβων αντικατοπτρίζεται στην πραγματικότητα.

Επιπλέον, το σύνολο δεδομένων COMPAS κατέδειξε τις προκλήσεις που αντιμετωπίζουν οι χρήστες με τα διαιρεμένα χαρακτηριστικά, τα οποία, αν και μειώνουν το μέγεθος του δέντρου, μπορούν να περιπλέξουν τη διαδικασία λήψης αποφάσεων.

Συνολικά, η μελέτη υπογραμμίζει τη σημασία της συνεκτίμησης τόσο της ακρίβειας όσο και της ερμηνευσιμότητας κατά το σχεδιασμό δέντρων αποφάσεων. Ενώ η κατηγορηματική διάσπαση μπορεί να ενισχύσει την κατανόηση και την εμπιστοσύνη των χρηστών, πρέπει να δοθεί ιδιαίτερη προσοχή στους συμβιβασμούς μεταξύ της πολυπλοκότητας του δέντρου και της κατανοητότητας, ώστε να διασφαλιστούν αποτελεσματικά και φιλικά προς τον χρήστη εργαλεία λήψης αποφάσεων.

Αξιολόγηση των μετρικών ερμηνευσιμότητας

Με βάση τα αποτελέσματα της μελέτης μας, αξιολογήσαμε την ερμηνευσιμότητα των δέντρων απόφασης χρησιμοποιώντας διάφορες βασικές μετρικές: μέγεθος μοντέλου, αριθμός χρησιμοποιούμενων χαρακτηριστικών, αραιότητα κόμβων, καθαρότητα φύλλων, ισορροπία δέντρου και μήκος κανόνα. Τα ευρήματά μας δείχνουν ότι ενώ το μικρότερο μέγεθος μοντέλου συνέβαλε γενικά στην καλύτερη κατανόηση από τον χρήστη, αυτό δεν συνέβαινε πάντα. Για παράδειγμα, τα δέντρα απόφασης του συνόλου δεδομένων COMPAS, τα οποία χρησιμοποιούσαν διαιρεμένα χαρακτηριστικά, είχαν ως αποτέλεσμα λιγότερους κόμβους αλλά μεγαλύτερη πολυπλοκότητα λόγω μη διαισθητικών διαχωρισμών των χαρακτηριστικών, τονίζοντας ότι το μέγεθος του δέντρου από μόνο του δεν καθορίζει την κατανοητότητα. Όσον αφορά τον αριθμό των χρησιμοποιούμενων χαρακτηριστικών, τα μοντέλα που ενσωμάτωναν λιγότερα και πιο συναφή χαρακτηριστικά ήταν ευκολότερα κατανοητά από τους χρήστες, όπως φάνηκε στο σύνολο δεδομένων Framingham Heart Study. Η αραιότητα των κόμβων, ιδίως σε δέντρα με λιγότερα σημεία δεδομένων που σχετίζονται με πολλούς κόμβους, επηρέασε επίσης την ερμηνευσιμότητα, καθώς τα αραιότερα δέντρα έτειναν να μπερδεύουν τους χρήστες. Η καθαρότητα των φύλλων ήταν ένας άλλος κρίσιμος παράγοντας- τα δέντρα με μεγαλύτερη ομοιογένεια στους κόμβους των φύλλων τους παρείχαν σαφέστερες διαδρομές λήψης αποφάσεων, βελτιώνοντας την εμπιστοσύνη των χρηστών και τη σαφήνεια της διαδρομής. Η ισορροπία των δέντρων διαδραμάτισε σημαντικό ρόλο στην ερμηνευσιμότητα, με τα καλά ισορροπημένα δέντρα από το σύνολο δεδομένων German Credit να είναι ευκολότερο για τους χρήστες να πλοηγηθούν και να κατανοήσουν σε σύγκριση με τα πιο imbalanced αντίστοιχα. Τέλος, το μικρότερο μήκος κανόνων, όπως παρατηρήθηκε στις κατηγορηματικές διαχωριστικές γραμμές του συνόλου δεδομένων Framingham Heart Study, διευκόλυνε την ευκολότερη κατανόηση, ενισχύοντας τη σημασία των συνοπτικών και απλών κανόνων απόφασης. Συνολικά, η μελέτη μας υπογραμμίζει ότι ο συνδυασμός αυτών των μετρικών, και όχι κάποιο μεμονωμένο μέτρο, είναι ουσιώδης για την ενίσχυση της ερμηνευσιμότητας των μοντέλων δέντρων απόφασης.

Chapter 1

Introduction

As artificial intelligence (AI) becomes increasingly integral to making decisions in critical areas such as healthcare, finance, and criminal justice, the need for these systems to be clear and understandable has never been more essential. Many of these high-stakes decisions are made using black box models and attempting to explain these opaque models, instead of developing ones that are inherently clear and interpretable, may perpetuate harmful practices and could even lead to severe damage in society. Decision trees, widely celebrated for their inherent transparency in the field of machine learning, exemplify how complex algorithms can be made understandable; their hierarchical structure often allows for visualization and comprehension even by non-technical stakeholders. However, this built-in interpretability does not always equate to real-world explainability, where the clarity of the model's decisions must be actionable and meaningful to all users. This thesis investigates the existing challenges in making decision trees truly interpretable, aiming to bridge the gap between technical transparency and practical understandability. By enhancing how decision trees communicate their reasoning, the study seeks to contribute significantly to the development of AI systems that are not only effective but are also trusted and fair, ensuring that AI supports rather than undermines critical human decisions.

1.1 Interpretable Decision Trees

Decision trees are considered interpretable due to their straightforward, hierarchical structure, which closely mimics human decision-making logic. Each node in a decision tree represents a decision point based on a specific attribute, and branches stemming from nodes depict the possible outcomes. This tree-like model allows users to follow the path from the root to the leaves, visually tracing the series of decisions that lead to a final outcome. This clarity in how decisions are derived makes it easy for users to understand and verify the reasoning process, enhancing trust and facilitating easier communication of how model decisions are made. Many times though the structure, complexity, and terminology used in a decision tree might make sense to an algorithm, but can still be cryptic or counter-intuitive to human stakeholders. As trees grow in depth and breadth to accommodate complex datasets, they can become overwhelmingly intricate, with numerous branches and conditions that are hard to follow. This complexity may obscure the clear, linear logic that makes smaller trees accessible, turning them into a tangled web of decisions that challenge comprehension. Additionally, using technical jargon or domain-specific terms in the nodes can make it hard for non-expert users to understand the underlying reasoning without specialized knowledge. Thus, while decision trees inherently strive for transparency, their practical implementation in more complex scenarios can inadvertently lead to opacity, complicating stakeholder understanding and potentially diminishing the trust placed in these models.

1.2 Contribution

This thesis focuses on interpretable decision trees, a cornerstone of interpretable machine learning, driven by the imperative to build trust and accountability in AI applications. Despite their inherent transparency

compared to other complex models, decision trees still present challenges in explainability that must be addressed to fully harness their potential. To address these challenges, this research introduces and evaluates novel explainability measures specifically designed for decision trees, enhancing the analytical rigor with which their interpretability is assessed. The research aims to enhance the interpretability of decision trees, ensuring that users can easily understand and trust the logic behind their decisions. By investigating the factors that influence human perception of interpretability, this study seeks to contribute to the development of decision trees that are not only powerful in performance but also in clarity and user confidence. This approach underlines the critical role of explainability in bridging the gap between AI capabilities and human-centric applications, emphasizing the broader impact of this research in promoting ethical and responsible AI practices.

1.3 Structure

Chapter 2 lays the theoretical foundation by exploring key concepts of interpretability, decision trees, and human decision-making. Chapter 3 discusses practical methodologies, including preprocessing, algorithmic modifications, and post-processing techniques to enhance interpretability. Chapter 4 presents a user study to evaluate the practical implications of these methodologies. Finally, Chapter 5 concludes with a summary of findings, discussions on implications, and suggestions for future research.

Chapter 2

Theoretical background

In recent years, machine learning models have demonstrated remarkable success in various domains, ranging from healthcare and finance to manufacturing and transportation. However, alongside their predictive power, these models often operate as black boxes, making it challenging for stakeholders to understand the reasoning behind their decisions. This lack of transparency raises concerns about accountability, fairness, and trust in AI systems, particularly in high-stakes applications where decisions impact individuals' lives or livelihoods.

2.1 Interpretability

Interpretability, defined as the ability of a model to provide interpretable explanations for its predictions, has emerged as a crucial aspect of machine learning research and practice. Explainable models offer insights into the factors driving their decisions, empowering users to understand, validate, and potentially challenge their outcomes. In data mining and machine learning, interpretability is defined as the ability to explain or to provide the meaning in understandable terms to a human. Understanding a computer-induced model is often a prerequisite for users to trust the model's predictions and follow the recommendations associated with those predictions.

Two main approaches have emerged in the literature to facilitate the understanding of machine learning models: black-box explanation and transparent box design[4]. Black box models are machine learning models or algorithms that are highly complex and difficult to interpret or explain in terms of how they arrive at their predictions or decisions. These models often involve intricate mathematical functions or architectures with a large number of parameters, making it challenging for humans to understand the underlying mechanisms or reasoning behind their outputs. Black-box explanation techniques (post-hoc explanations) refer to methods designed to explain how black-box ML models produce their outcomes. They are inappropriate in high-stake decision systems as they can be manipulated to tell a different story than that of the black-box they are explaining. Unlike traditional black-box models, which prioritize predictive accuracy at the expense of interpretability, explainable models strike a balance between performance and transparency, fostering trust and accountability in AI systems [32]. Usually interpretable models are constrained in model form that it is either useful to someone, or obeys structural knowledge of the domain, such as monotonicity, causality, structural (generative) constraints, additivity, or physical constraints that come from domain knowledge [63].

2.1.1 Importance of Interpretable models

The importance of interpretable Machine Learning models stems from several issues. Initially, comprehending a model generated by a computer is often essential for users to have confidence in the model's forecasts and follow the associated recommendations. This necessity for trust in computational predictions is particularly pronounced in critical domains such as medicine, where human lives are at stake [48], [8]. Moreover, the comprehensibility of a model is crucial for its acceptance by users in financial contexts [21] and in applications like customer churn prediction. The requirement for transparent models to enhance user trust becomes even more apparent when the system presents an unexpected model to the user, necessitating thorough explanations from the system for model acceptance. Furthermore, in certain domains of application, users require a sufficient understanding of the system's recommendations to provide legally sound explanations for their actions to others. For instance, in the medical field, if a physician makes a decision—such as recommending surgery—based on a classification model's prediction, resulting in significant harm to the patient, the physician must comprehend the rationale behind the model's predictions to justify their decisions in court in the event of a medical negligence lawsuit. Similarly, legal obligations frequently arise in credit scoring applications, where a bank is often mandated to elucidate the reasons behind denying credit to a customer. Certainly, there are instances where model comprehensibility holds little importance, and some users may find satisfaction in embracing a model's predictions solely due to its high predictive accuracy, overlooking its comprehensibility. The relative significance of comprehensibility versus predictive accuracy remains subjective, contingent upon the user's interests and the particular application domain.

In addition to the crucial need for model comprehensibility in high-stakes domains, several fundamental reasons underscore the importance of interpretability in machine learning models [15]:

Trust: The deployment of a prediction model is critically dependent on trust and acceptance. Only by understanding the model's strengths and weaknesses can users develop the necessary confidence to rely on its predictions. This trust is foundational for the widespread adoption and utilization of machine learning models.

Causality: Interpretability, particularly through mechanisms like attribute importance, imparts a sense of causality. This helps the target audience understand the underlying relationships driving the model's outputs, bridging the gap between complex computations and comprehensible results.

Transferability: For a human decision-maker to effectively use a prediction model with new, unseen data, the model must provide a clear understanding of future behavior. The decision-maker needs to be confident that the model generalizes well or understands the specific contexts in which it performs reliably. Only then will the decision-maker trust the model to make decisions.

Informativeness: Beyond fulfilling its training objectives, a model must address real-world needs effectively. Understanding whether a system genuinely serves its intended purpose is crucial for its deployment, ensuring that it operates as a practical tool.

Fair and Ethical Decision Making: Understanding the reasons behind a decision is a societal necessity and is expected to become a legal right for EU citizens [31]. This "right to explanation" obliges decision-makers to present their results clearly to adhere to ethical standards. Anyone impacted by an automated decision can exercise this right to receive an explanation.

Accountability: Incorporating explainability into machine learning models is also about accountability. A model that can justify its decisions can be held accountable for its actions. This aspect is particularly relevant in addressing potential shifts in data over time, ensuring that models remain responsible and trustworthy.

Making Adjustments: Understanding the prediction model and its underlying factors allows domain experts to compare the model's predictions with existing domain knowledge. Interpretability is essential for adjusting the prediction model by incorporating domain-specific insights. According to Selvaraju et al. (2016) [64], explainable prediction models can help humans, particularly domain experts, make better decisions. From an algorithmic perspective, interpretability allows system designers to refine the prediction model by adjusting parameters, for example. Additionally, interpretability aids developers in identifying and addressing failure modes.

Proxy Functionality: When a model is interpretable, it can be assessed on metrics beyond those it was directly trained to optimize, such as safety, fairness, and privacy. This aspect makes interpretability a proxy for evaluating broader societal and operational impacts, enhancing the overall utility and acceptability of machine learning systems.

Integrating these principles into the development of machine learning models not only enhances their practical usability but also aligns them with broader societal values and expectations, making them more robust, trustworthy, and aligned with human needs.

2.1.2 Challenges of Interpretable models

Although the necessity for interpretability in machine learning, as outlined previously, is widely recognized, particularly for fostering trust, ensuring ethical compliance, and meeting regulatory demands, achieving it is not without its challenges. These challenges arise from a variety of technical, practical, and regulatory complexities that impact the development and deployment of interpretable models. It was observed by Pazzani [57] that few papers actually aim to empirically assess comprehensibility beyond simply reporting the size of the resulting representations. Despite the clear advantages of transparent AI systems, the path to fully interpretable AI is filled with obstacles that span the spectrum from model complexity to user-specific requirements.

Black-box models

Modern machine learning models, especially deep neural networks, consist of millions of parameters and are characterized by layers of abstraction. These layers, while beneficial for capturing complex patterns in large datasets, create a "black box" scenario where the input-output relationship is virtually indecipherable. This complexity is a major barrier not only to understanding what the model is doing but also to diagnosing errors or biases in the model's predictions. There are many reasons that black-box models are preferable than interpretable ones.

Black box models often achieve superior performance compared to simpler, more interpretable models. In tasks involving large datasets with high dimensionality and complexity, such as image and speech recognition, black box models can identify subtle patterns and correlations that simpler models might miss [63]. Of course,

when considering problems that have structured data with meaningful features, there is often no significant difference in performance between more complex classifiers (deep neural networks, boosted decision trees, random forests) and much simpler classifiers (logistic regression, decision lists) after preprocessing.

The fact that many scientists have difficulty constructing interpretable models may be fueling the belief that black boxes have the ability to uncover subtle hidden patterns in the data that the user was not previously aware of. A transparent model may be able to uncover these same patterns. If the pattern in the data was important enough that a black box model could leverage it to obtain better predictions, an interpretable model might also locate the same pattern and use it.

Because of the complex structure of black box models a need for creating post-hoc explanations of the models has arised. Explainable ML methods provide explanations that are not faithful to what the original model computes, as they cannot have perfect fidelity with respect to the original model. Many of the methods that claim to produce explanations instead compute useful summary statistics of predictions made by the original model. These post-hoc explanations are inappropriate in high-stake decision systems as they can be manipulated to tell a different narrative than that of the black-box they are explaining [4]. Another issue with the post-hoc extracted explanations is that often they do not make sense, or do not provide enough detail to understand what the black box is doing. The explanation may leave out so much information that it makes no sense like in image processing (saliency maps). To address the problem of interpretability, recent works utilise Knowledge Graphs and semantic descriptions [51, 49, 50, 53], offering a more structured approach to black-box explanations.

General Challenges in Interpretability

One of the most important challenges of interpretability in modern ML is the lack of standard metrics [23]. The field lacks uniform metrics to measure interpretability, which complicates efforts to improve or even define what makes a model interpretable. For instance, one researcher might consider a model interpretable if its decisions can be summarized in a short rule list, while another might prioritize visual explanations that are understandable to non-experts. This discrepancy makes it difficult to develop a standardized approach to enhancing or assessing interpretability across different contexts.

The requirements for interpretability vary widely depending on the user's expertise and the application area. For example, a model used by data scientists for fraud detection might require detailed statistical explanations, whereas a model used by bank tellers for the same task needs to provide simple, actionable insights without overwhelming the user with technical details.

In general interpretability is necessarily defined in a domain specific way, the most important problem areas for the future may be tied to specific important domains. An important criterion for selecting a model from a series of candidate models with similar performance is that it is in line with previous domain knowledge. For these reasons, rule induction algorithms, which return a set of 'if–then' rules, or decision tree learners are often the preferred choice as they should offer the required level of interpretability [40].

2.1.3 Interpretable Models

Inherently interpretable models stand out for their ability to provide clear, intuitive insights into their decision-making processes, making them indispensable in sensitive and high-stakes domains such as healthcare, finance, and legal compliance. These models are designed not just to perform tasks but to explain their decisions in a way that is understandable to humans, bridging the gap between advanced computational techniques and practical, everyday decision-making needs.

Linear Models

Linear models, such as linear regression and logistic regression, are prized for their simplicity and transparency. The relationship between input features and the predicted outcome is expressed through weights or coefficients, which are directly interpretable. Each coefficient quantifies the impact of a corresponding feature on the prediction, offering straightforward insights into the model's behavior. These models are widely used in financial risk assessment and medical outcome prediction, where understanding the influence of variables is crucial.

Rule lists

The most common type of rules is without any doubt propositional if–then rules. The condition part of a propositional rule consists of a combination of conditions on the input variables. While the condition part can contain conjunctions, disjunctions, and negations, most algorithms will return rules that only contain conjunctions. Rule-based systems generate sets of if-then rules to make decisions, which are easy for humans to understand and audit. Various formats can be used to represent propositional rules. The most straightforward approach is to simply write the rules down, as in the following example:

IF (INCOME $>$ 400 AND GOAL=CAR) THEN ACCEPT
IF (INCOME $>$ 900 AND GOAL=HOUSE) THEN ACCEPT
DEFAULT:REJECT

This example shows the credit policy of a financial institution which may be used to decide on loan applications. Based on this policy, the credit manager would accept all applications where the applicant has an income above 900 and the goal is the purchase of a house or where the income is above 400 and the goal is the purchase of a car. If these conditions are not satisfied, the default rule specifies that applications are to be rejected.

Rule extraction can be performed on black box models, such as neural networks and support vector machines, to extract a set of rules that approximate the black box as closely as possible and at the same time provide a more understandable representation to the users. However, previous research concerning rule extraction techniques [39] indicated that some algorithms return models that closely approximate the underlying black box model, but at the cost of being very complex.

Rule lists are considered explainable because they provide a clear, sequential breakdown of decision-making criteria in a format that is straightforward and easy for humans to understand. Each rule in a list consists of an if-then statement that explicitly states a condition and the outcome if that condition is met. This format mimics logical human thought processes, such as step-by-step troubleshooting or diagnostic procedures, making it intuitive to follow. For example, in a medical diagnosis application, a rule might specify that "if the patient's temperature is above 38.0°C and they have a sore throat, then diagnose as strep throat." This transparency allows users not only to see which conditions lead to particular decisions but also to verify and trust the reasoning behind each decision. Because of their simplicity and clarity, rule lists enable users to quickly understand and evaluate the decision-making process, contributing significantly to the overall transparency and user trust in the system.

Despite their initial appeal, these rule-based models often suffer from several notable drawbacks that can diminish their utility in terms of explainability. Firstly, the large number of rules generated for complex models can be overwhelming, reducing the clarity and making it difficult for users to grasp the underlying logic. This complexity can obscure the insights that rules are supposed to provide, making them less intuitive than intended. Moreover, the inflexibility of rule-based systems can lead to fragile behavior in unforeseen scenarios not covered by the existing rules, limiting their adaptability and reliability. Additionally, the maintenance of large rule sets becomes impractical as updating one rule might require cascading changes throughout the system. Consequently, although rule-based models aim to clarify the predictions of black-box models, they can unintentionally add a new layer of complexity, making the goal of true explainability more difficult to achieve.

Decision Tables

Other more graphical-oriented representations that are frequently used to depict conditional logic are decision tables and decision trees. A decision table [70] is a tabular representation that consists of four quadrants separated by horizontal and vertical double lines (see Figure 2.1.1). The horizontal line divides the table into a condition part (top) and an action part (bottom), whereas the vertical line separates subjects (left) from entries (right). Every column in the entry part corresponds to a rule, combining condition states with the appropriate action(s) to take. A dash symbol (-) in the condition part of the table indicates that the value is irrelevant in that condition and an "X" in the action part represents the correct conclusion to make if the conditions leading to that column are satisfied. The fact that each possible combination of condition states

occurs only in exactly one column is the key advantage of single hit tables.

(a) Single-hit table

INCOME	< 1000		≥ 1000
AGE	< 25	≥ 25	-
ACCEPT	X		
REJECT		X	X

(b) Multiple-hit table

INCOME	≥ 1000	-	< 1000
AGE	-	< 25	< 25
ACCEPT			X
REJECT	X	X	

Figure 2.1.1: Example decision tables

Decision tables are highly explainable due to their structured format, which organizes and presents information clearly by listing possible conditions and corresponding actions in a tabular form. Each row in a decision table represents a specific scenario with defined conditions, and each column correlates to a decision or outcome based on those conditions. This arrangement allows users to easily trace how inputs are transformed into outputs, making it straightforward to understand the logic behind each decision. For instance, in a loan approval system, a decision table can show at a glance under what combinations of age, income, and credit score a loan would be approved or denied. This clear, visual method of depicting rules and decisions not only aids in quick comprehension and validation of the decision process but also helps in identifying inconsistencies or overlaps in rules. Consequently, decision tables facilitate transparency and auditability, enhancing trust in automated systems by providing a comprehensible and accessible explanation for each decision made.

Despite their structured clarity, decision tables can face limitations in terms of interpretability, particularly as the complexity of the decision-making process increases. When decision tables contain a large number of conditions or when these conditions interact in complex ways, the table can become overly large and unmanageable. This growth not only makes it difficult for users to quickly understand the rationale behind decisions but also can lead to errors in interpreting the table correctly. Furthermore, decision tables may not effectively capture the complexity of scenarios that involve continuous data or require more granular decision-making, as they typically operate best with discrete, well-defined categories. As a result, while decision tables excel in clarity for straightforward decision processes, their utility diminishes with the increasing complexity of the scenarios they are meant to model, potentially leading to oversimplification of important details and reducing their overall effectiveness in conveying decision logic.

Decision Trees

Decision trees are praised for their explainability, which stems from their intuitive structure that mimics human decision-making logic. Each decision node in a tree represents a clear, simple question or condition regarding one of the input features, and the branches from these nodes denote the possible answers, leading straightforwardly to subsequent conditions or final outcomes. This tree-like architecture allows each path from root to leaf to be easily interpreted as a series of logical steps, providing a transparent explanation of how a decision was reached. This inherent transparency makes decision trees an ideal model for situations where understanding the rationale behind predictions is as important as the accuracy of the predictions themselves. More details on their application and characteristics will be explored in the following section.

2.2 Decision Trees

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes. As shown in Figure 2.2.1 a decision tree starts with a root node, which does not have any incoming branches. The outgoing branches from the root node then feed into the internal nodes, also known as decision nodes. Based on the available features, both node types conduct evaluations to form homogenous subsets, which are denoted by leaf nodes, or terminal nodes. The leaf nodes represent all the possible outcomes within the dataset. Decision trees are a cornerstone of machine learning, providing a foundation for both basic and advanced predictive models. They are particularly praised for their simplicity and efficacy, making them one of the most popular algorithms in the data science community. Commonly used in classification and regression tasks, decision trees automatically discover decision boundaries from the data. These boundaries can be as simple as a single split in one-dimensional data or as complex as multiple layers of decisions in high-dimensional spaces. Algorithms like ID3, C4.5, and CART differ primarily in their criteria for choosing the feature and the condition for splitting the data at each node, which significantly impacts their performance and the tree's final structure. The intuitive layout of decision trees not only helps in performing data analysis tasks efficiently but also aids in visually representing the decision-making process, thereby making the outcomes easy to understand and interpret. This transparency is crucial for applications in fields such as finance, healthcare, and any domain where making informed and justifiable decisions is necessary. As we delve deeper into this chapter, we will explore the intrinsic properties that contribute to the interpretability of decision trees, discuss their practical applications, and address some of the challenges they face in terms of scalability and complexity.

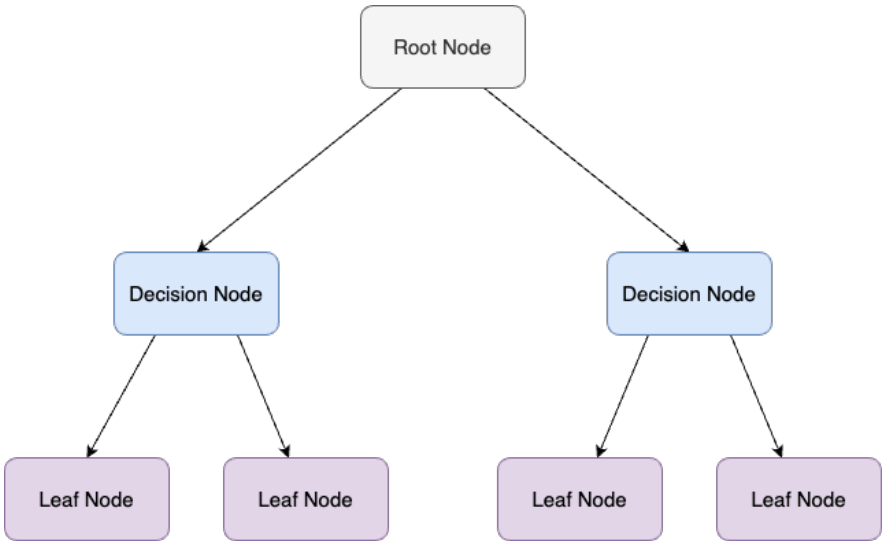

Figure 2.2.1: Decision Tree

2.2.1 Why are decision trees interpretable?

Decision trees are inherently interpretable due to their decision-making process, which closely mirrors human reasoning patterns(which will be discussed in chapter 2.3). At the heart of a decision tree's interpretability is its structural representation, where decisions are made through a series of straightforward questions and answers that split the data incrementally. Each node in the tree represents a specific question or a condition on a particular feature, and the branches to the children represent the possible answers to this question, leading finally to leaf nodes which provide the outcomes or predictions. This tree structure allows anyone examining the model to see exactly how inputs are turned into outputs, following the paths from the root to the leaves. For instance, in a medical diagnosis application, a decision tree might use conditions based on symptoms and test results to determine potential diagnoses, making it clear which symptoms lead to which diagnosis.

Visual Transparency

One of the key aspects of decision trees that enhances their interpretability is their visual nature. Decision trees can be graphically represented [29], allowing users to visually trace through the decision paths, which is much less abstract than coefficients in regression models or weights in neural networks. This visualization helps users not only understand the criteria being used at each decision point but also evaluate the logic and fairness of these criteria.

Feature selection and importance

A decision tree typically encompasses only a subset, rather than the entire set of attributes. This selective inclusion helps streamline the decision-making process by focusing attention on the most relevant features, thereby reducing complexity and facilitating clearer insights into the underlying patterns of the data. Moreover, the hierarchical arrangement of the tree provides insights into the relative importance of various attributes. Generally, attributes positioned closer to the root—indicated by their smaller depth—bear greater significance for classification. It's important to note that an attribute may appear multiple times along the same path from the root to a leaf, particularly in cases involving multiple binary partitions on nodes with numerical attributes. In such instances, we define the attribute's depth as its shallowest occurrence within that path. When an attribute recurs in paths with non-overlapping sets of edges, computing an aggregated measure of depths, such as the mean depth, becomes necessary. The idea of prioritizing attributes with shallower depths in decision trees poses a challenge. Even if Attribute A has a shallower depth than Attribute B, B might be more crucial in classifying more instances. To address this, we can consider a simpler criterion: the count of instances classified by each attribute. This approach sums up instances assigned to leaf nodes where the attribute is involved in classification, making it easier for users to interpret attribute significance.

Rule Extraction

Each path from the root of the tree to a leaf can be directly translated into a rule-based format: an if-then statement that clearly explains why a particular decision or prediction was made. This feature is particularly beneficial for providing explanations for specific decisions, which is a requirement in many regulated industries like finance and healthcare. Local decisions can be easily interpreted since each leaf is translated into a clear conjunction of attributes [60].

2.2.2 Interpretable models Comparison

In the pursuit of enhancing interpretability in machine learning models, decision trees, decision rules, and decision tables emerge as prominent contenders, each offering unique strengths and considerations.

Rules - Decision Trees

As mentioned in chapter 2.1.3, rule-based models are known for their intepretability. Rules do not capture insignificant clauses, while decision trees can also have insignificant branches. This happens because rule based classifiers generally select one attribute-value while expanding a rule, whereas decision tree algorithms usually select one attribute while expanding the tree [32]. Decision trees are often praised for their superior comprehensibility compared to rule lists, as highlighted in a study by Allahyari and Lavesson (2011) that evaluates the understandability of classifiers derived from both decision trees and rule-learning algorithms [5]. The study, conducted through subjective assessments by 100 Computer Science students, concludes decisively in favor of decision trees, affirming their greater clarity and ease of comprehension (Souza et al., 2022) [65].

Decision Tables - Decision Trees

Decision trees and decision tables are both tools used for decision-making but have distinct characteristics. Decision trees utilize a hierarchical structure with nodes and branches to represent decisions and their possible outcomes, providing an intuitive visual representation that mirrors human decision logic. This makes them highly interpretable and transparent, allowing users to easily understand the decision-making process and the importance of different features. However, decision trees can become complex and prone to overfitting, especially with large datasets, and can be unstable with small changes in the data. In contrast, decision tables offer a tabular format that enumerates all possible conditions and corresponding actions, ensuring comprehensive coverage and consistency in decision-making. While they are simple and clear for straightforward scenarios, decision tables can become unwieldy and difficult to interpret as the number of

conditions grows. They also lack the visual appeal of decision trees, which can make them less intuitive for some users. Ultimately, the choice between decision trees and decision tables depends on the specific use case, the complexity of the decision-making process, and whether a visual or tabular representation is preferred. Generally, Subramanian et al. [67] further investigated the difference between both graphical representations (decision trees and decision tables) from an effectiveness perspective, measuring the number of correct decisions without taking decision time into account. It was concluded that decision trees perform significantly better than decision tables.

2.2.3 Disadvantages of Decision Trees

While decision trees are renowned for their straightforward and visual interpretability, they come with certain limitations that can complicate their use, especially in complex scenarios.

Overfitting

Decision trees are susceptible to overfitting, particularly when they are allowed to grow deep without constraints. Overfitting occurs when a tree models the noise in the training data rather than the actual underlying patterns. This results in overly complex trees that are hard to interpret and do not perform well when presented with new, unseen data. Such trees can have an extensive number of branches and leaves, each tailored to specific outliers or anomalies in the data, which makes the model's decisions overly specific and difficult to generalize from.

Instability

Decision trees can exhibit a high level of variance in their structure with small changes in the input data. A slight variation in the dataset, such as the addition or removal of a few data points, can lead to a completely different tree being generated. This instability can be confusing for users, as the rationale for decisions may change unpredictably, undermining trust in the model's reliability and consistency.

Depth and Complexity

As decision trees deal with more complex datasets, they tend to grow deeper to capture more detailed patterns. However, deeper trees can become difficult to interpret. The increase in the number of decisions (depth) and the branching complexity can overwhelm users, making it harder to trace the logic from the root to the leaves. Each additional layer in the tree adds another level of decision-making that a user must understand and validate, which can obscure the clear, simple interpretability that makes trees so appealing in less complex applications.

Local-Global Explanations

Decision trees are often better at providing local explanations (explaining a specific prediction) rather than global explanations (understanding the overall behavior of the model). This can be a limitation when trying to understand the model's general decision-making process.

Bias Toward Certain Splits

Decision tree algorithms such as ID3 and C4.5 can exhibit a bias toward attributes with more categories. They tend to favor these attributes for splits at the top of the tree because they can result in higher information gain metrics, even if they are not the most predictive of the outcome. This bias can lead to misleading representations of the importance of features, where the decision-making process appears to prioritize certain features that may not actually be critical in real-world decision-making scenarios. Single-tree models such as CART [13] are fully interpretable, as their prediction logic can be easily followed by observing the splits in the final decision tree. However, CART is trained using a greedy heuristic that forms the tree one split at a time, which has a number of downsides. First and foremost, this can result in trees that are far from globally optimal, as the best split at any given point in the greedy heuristic may not prove to be the best when viewed in the context of the future growth of the tree. These limitations are cause for concern when interpreting the variable importance, as the selected features may be biased towards those with a greater number of unique values, and the greedy algorithm may lead to incorrect features being used in the splits near the root of the tree, which are usually those that receive most importance [24].

Handling of Continuous Variables

Decision trees handle categorical variables by transforming them into a format that can be effectively used in the tree-building process. Often, categorical variables are converted into binary (dummy) variables, a process known as one-hot encoding. This transformation involves creating a new binary variable for each category of the original feature, where each variable represents the presence (1) or absence (0) of a specific category. While this method allows decision trees to incorporate categorical data without assuming an ordinal relationship, it can significantly increase the dimensionality of the data, especially with features having a large number of categories. This expansion can lead to larger and more complex trees, as each binary variable may be considered for splitting at every node in the tree. Handling categorical features in this binary manner can thus contribute to the complexity and depth of the tree, potentially impacting both the performance and the interpretability of the model.

2.3 Human Explanations and Decision Making

Understanding human decision-making processes and cognitive limitations is crucial for the design and development of explainable decision trees structures [52], [55]. Human decision-making, while often perceived as logical and rational, is also subject to a multitude of cognitive biases, heuristics, and limitations. Exploring the extensive research in psychology, neuroscience and cognitive science provides valuable insights into the underlying mechanisms that govern human decision-making. By examining this body of literature, we can uncover the boundaries of human comprehension, clarify the cognitive processes at play, and identify potential pitfalls that decision tree algorithms may encounter when simulating human decision-making. In this chapter, we embark on a comprehensive exploration of the literature on human decision-making and cognitive science, aiming to shed light on how humans make decisions, the cognitive biases that influence their choices, and the implications of these findings for the design and implementation of decision tree structures and algorithms. Through this interdisciplinary approach, we aim to bridge the gap between the theoretical understanding of human decision-making and the practical application of decision tree models in real-world scenarios. Ultimately, our objective is to extract valuable insights from human cognition to guide the development of more interpretable, transparent, and effective decision tree algorithms.

2.3.1 Human decision-making

From a general cognitive perspective, decision making is the process of selecting a choice or course of action from a set of alternatives. Most psychological models describe the human process of decision making as serial staged processes that include steps centered on information gathering, likelihood estimation, deliberation, and decision selection. The two fundamental cognitive processes that underlie human reasoning and decision making are attention and memory.

Attention and Working Memory

Attention is how the brain, often consciously though sometimes automatically, selects information for cognitive processing. Human memory is the capacity to encode, store, and retrieve information. Attention and memory (working memory, in particular) serve as important bottlenecks in human information processing, so understanding how these processes affect the components of reasoning and decision making are vital to developing decision support technologies such as decision trees. Working memory refers to a variety of processes used to maintain mental information in a highly accessible state. Working memory should be thought of as a temporary store where conscious, effortful (requiring attention) internal computations are performed [6]. Working memory storage plays a very important role in those tasks. As we have seen in Miller [54], most humans are able to recall a list of no more than 7 randomly ordered meaningful items. The recall limit is important as it measures the working memory. For sure working memory capacity varies among people and changes across the life span. [20]. More recent studies ([36], [19]) on many types of materials and tasks indicate that there is a central working memory faculty limited to 3-5 chunks of information in adults. These chunks of information might be single words or pairs of words.

Decision trees are a type of machine learning model that humans use to make decisions by mentally processing a sequence of if-then rules. Since humans have this limit to the number of chunks of information they can process effectively, decision trees become less interpretable as they grow larger and more complex, leading to higher cognitive load for humans trying to understand them.

Human Reasoning

Decision making processes are facilitated using a variety of different reasoning techniques. Analogical reasoning, for instance, involves inferring new solutions by drawing parallels with known ones. Decisions are ultimately reached through reasoning about the problem and its potential outcomes, often by considering past decision-making events.

Component processes of analogical reasoning include the following serial procedures [66]:

1. **Encoding**: Translating stimuli to internal (mental) representations
2. **Inference**: Determining the relationship between problems
3. **Mapping**: Determining correspondences between new and old items
4. **Application**: Execution of the decision process
5. **Response**: Indicating the outcome of the reasoning process

Given that the steps in this reasoning process unfold sequentially, the temporal arrangement and timing of decision support are crucial for enhancing time-sensitive decision-making. Further analysis reveals that reaction times and error rates increase with more complex encodings. Regardless of the stimuli, the encoding step constitutes the most significant portion of the reasoning process, accounting for approximately 45% of the overall reasoning time. For instance, encoding words takes longer than encoding schematic pictures, suggesting that reducing textual content in displays could expedite decision-making. Therefore, displays for time-critical decision-making should prioritize facilitating faster encoding, possibly through the use of more intuitive symbols and tasks. Decision trees are inherently explainable due to their operational simplicity and transparent decision-making process. Similar to human decision-making, decision trees rely on a series of straightforward rules represented by nodes, making it easy to understand how decisions are reached. Clear and comprehensible encoding of features enhances the interpretability of decision trees and facilitates the explanation of model predictions to stakeholders.

2.3.2 Human Perception

Since the encoding part of reasoning is the most important and time-consuming part of decision making is crucial to optimise the representation of the models based on human perception. Humans have a remarkable capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are parking a car, driving in city traffic, cooking a meal, and summarizing a story. In performing such tasks, for example, driving in city traffic, humans base whatever decisions have to be made on information that, for the most part, is perception, rather than measurement, based [75]. An essential difference between measurements and perceptions is that in general, measurements are crisp, whereas perceptions are fuzzy (Figure 2.3.1).

Based on the above, we can conclude that categorical data, with clear labels can be more intuitive for users to interpret compared to numerical values. Categorical splitting involves dividing a continuous variable into discrete categories or groups based on meaningful distinctions. For example, instead of simply dividing heights into "tall" and "short" categories based on arbitrary thresholds, a user-centric approach would consider what these categories mean to users. Are users more likely to understand and relate to categories like "average height," "above average height," and "below average height"? These categories are more intuitive because they align with common language and perceptions about height. Meaningful categories make it easier for users to identify patterns, make comparisons, and draw conclusions from the data. Categorical splits contribute to the transparency and interpretability of models by making the underlying logic more explicit and understandable. Users can easily grasp why certain categories were chosen and how they relate to the problem at hand. This transparency builds trust in the model's outputs [45]. Humans tend to approximate the numerical elements with imprecise linguistic labels. For sure the numerical meanings of these fuzzy linguistic labels such as "short" or "tall" will differ among humans. Amazingly, humans are nevertheless able to communicate with these ill-defined and vague linguistic labels and do not query the

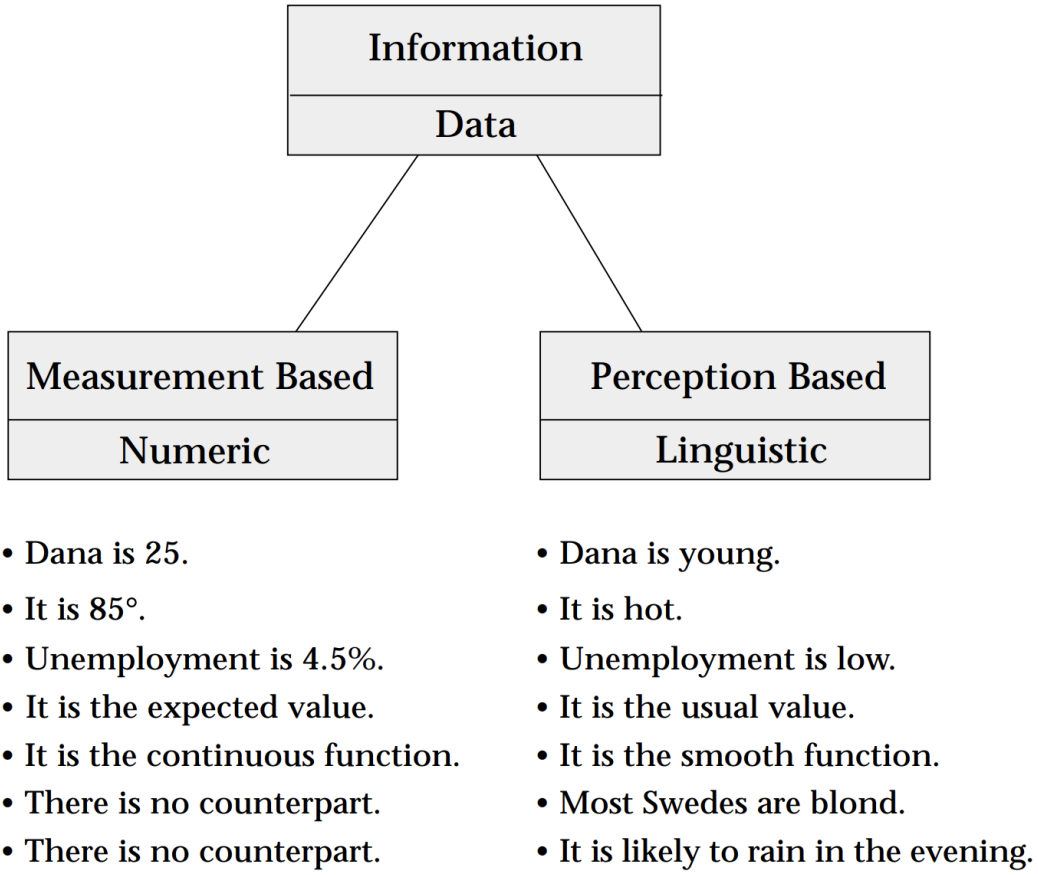

Figure 2.3.1: Information Data(Zadeh 2001)

exact values when they discuss them. In fact, these uncertain concepts allow humans to be able to perform very sophisticated tasks such as driving cars or underwriting financial applications [35].

2.3.3 Model Comprehension

An important mechanism to improve problem-solving task performance is often considered the proper visualization of information [[68], [69], [73]]. Information in the form of pictures or graphs is generally regarded as superior to that in other representations. There have been many researches in an effort of verifying this statement but the results have been inconsistent. Some studies have shown that decision trees outperform decision tables and others have shown that tables are superior than graphs. In order to explain these contradictory results Jarvenpaa, Dickson, and DeSanctis stated: "Future research efforts will keep producing contradictory results unless researchers develop some type of taxonomy of tasks and start interpreting the results within the taxonomy" [41].

In order to address this matter Vessey developed a theory to describe the relationship between graphical and tabular representations and the types of tasks they support tasks that are better suited for graphical or tabular representations [72]. Vessey's work introduced the concept of "cognitive fit". This concept posits that complexity within the task environment can be mitigated effectively when problem-solving aids—such as tools, techniques, or problem representations—align with the task strategies (methods or processes) necessary for task completion.

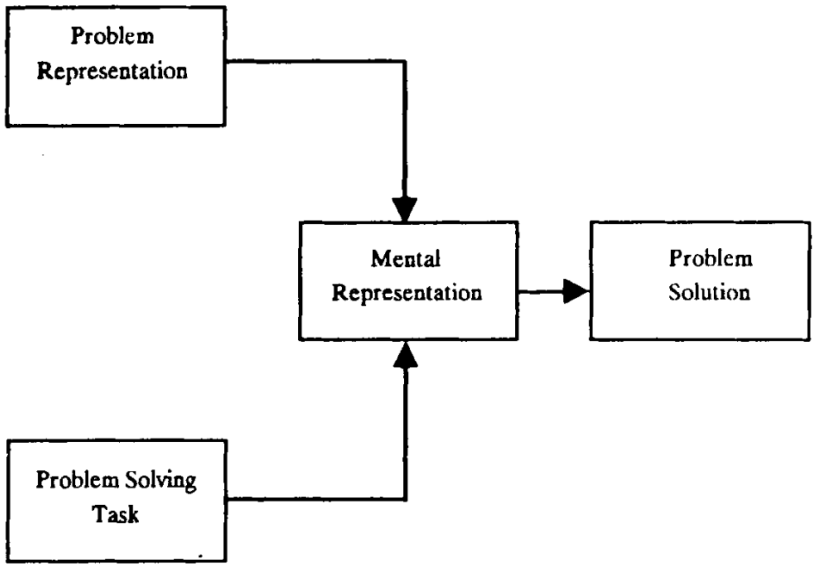

Figure 2.3.2: General problem solving model(Vessey 1991)

Vessey suggest the model shown in Figure 2.3.2 for general problem-solving on which the cognitive fit argument is based. The model views problem solving as an outcome of the relationship between problem representation and problem-solving task. The mental representation is the way the problem is represented in human working memory. When the types of information emphasized in the problem-solving elements (problem representation and task) match, the problem solver uses processes (and therefore formulates a mental representation) that also emphasize the same type of information. In other words, matching representation to task leads to the use of similar, and therefore consistent, problem-solving processes, and hence to the formulation of a consistent mental representation. There will be no need to transform the mental representation to accommodate the use of different processes to extract information from the problem representation and to solve the problem. Hence, problem solving with cognitive fit leads to effective and efficient problem-solving performance.

Decision trees are spatial problem representations since they present spatially related information. Based on

cognitive fit explainable decision trees align well with human's mental models of the classification problem. This alignment enhances user's understanding of how the decision tree arrived at its conclusions, leading to more effective and confident decision-making.

2.4 Interpretability Measures

While the importance of interpretability in machine learning has been well established, quantifying it remains a formidable challenge. Traditionally, the evaluation of classification models has predominantly focused on predictive accuracy as the primary criterion. However, in real-world applications where models must not only perform but also be comprehensible to users, interpretability becomes equally important. Despite the historical emphasis on accuracy, there has been significant progress in developing methods to enhance the comprehensibility of classification models [29]. The evaluation of various explainability methods often lacks a clear quantitative framework and relies instead on qualitative assessments and user studies. This reliance on qualitative measures highlights the complexity of interpretability, which encompasses not only the model's transparency but also how information is perceived and understood by humans. As we delve deeper into interpretability measures, we explore both the advances and the ongoing challenges in creating metrics that can effectively quantify how interpretable a model is to its end-users.

2.4.1 Quantitative evaluation of intepretability

Quantitatively assessing interpretability requires the definition and application of specific metrics. These metrics aim to capture various aspects of interpretability, including simplicity, transparency, fidelity, and human evaluation.

Simplicity

Simplicity metrics evaluate the complexity of a model's representation and decision-making process. Common simplicity metrics include the number of features and model size.

Transparency

An immediate goal for an XAI system—in comparison to an inexplicable intelligent system—is to help end-users understand how the intelligent system works. Machine learning explanations improve users' mental model of the underlying intelligent algorithms by providing comprehensible transparency for the complex intelligent algorithms [56]. Transparency metrics assess how easily a model's decision-making process can be understood by humans. Examples include rule length and interpretability of coefficients.

Fidelity

Fidelity metrics evaluate the extent to which an interpretable model accurately represents the behavior of the underlying complex model. Examples include prediction consistency and local accuracy. Fidelity is a measure of interpretability for models extracted out of complex models (black-box) and not for genuinely interpetable models.

Human evaluation

Human evaluation metrics assess aspects related to how well humans understand and trust the model's decisions. This involves evaluating the comprehensibility, transparency, trustworthiness, user satisfaction, and effectiveness in decision-making of the model.

2.4.2 Human Evaluation of Interpretability

In human evaluation metrics for assessing interpretability in machine learning models, we typically evaluate aspects related to how well humans understand and trust the model's decisions. This involves gathering feedback from human participants through various methods such as user studies, surveys, and cognitive walkthroughs. Here are some specific aspects that we evaluate in human evaluation metrics:

Comprehensibility

We assess the clarity and ease with which humans can understand the model's decision-making process. This includes evaluating whether participants can grasp the logic behind the model's predictions, comprehend the significance of input features, and follow the decision path from inputs to outputs.

Transparency

We evaluate the degree to which the model's inner workings are transparent and accessible to humans. This involves examining whether participants can interpret and explain how the model arrives at its predictions, understand the role and importance of individual features, and trace the decision-making process.

Trustworthiness

We measure the level of trust that humans place in the model's predictions and decisions. This includes assessing participants' confidence in the model's accuracy, reliability, and fairness, as well as their willingness to rely on the model's recommendations in real-world scenarios. Minimally, a trust scale asks two basic questions: *Do you trust the machine's outputs?* (trust) and *Would you follow the machine's advice?* (reliance) [37].

User Satisfaction

We measure participants' overall satisfaction with the model's interpretability and usability. This involves gathering feedback on the user experience, including ease of use, clarity of explanations, and perceived value of the interpretability provided by the model. It must be noted that a person may say that they feel satisfied with an explanation when in fact their understanding is piecemeal or flawed [22]. Key attributes of user satisfaction are understandability, feeling of satisfaction, sufficiency of detail, completeness, usefulness, accuracy, and trustworthiness [37].

Effectiveness in Decision-Making

We assess whether the interpretability provided by the model enables participants to make informed decisions or take appropriate actions based on the model's predictions. This involves evaluating the practical utility of the model's explanations in real-world applications and scenarios.

2.4.3 Decision Tree Interpretability Metrics

Interpreting decision trees involves assessing various metrics that provide insights into their complexity and comprehensibility. While many evaluations overly simplify this task by focusing solely on model size, true interpretability depends on multiple factors that capture both syntactic and semantic aspects of the tree. With the multitude of available methods, the evaluation of XAI approaches is crucial and remains an active field of research to ensure their effectiveness and reliability in various contexts [26].

1. **Model Size**: In the vast majority of papers where the comprehensibility of a classification model is evaluated, that evaluation is done in an over-simplistic way, by measuring only the size of the model. The assumption is that the smaller the model is, the more comprehensible it would be to the user. The size of a model primarily pertains to its syntax and doesn't inherently convey any semantic information. The understandability of a model is largely contingent upon its actual content, such as the attributes in a decision tree, the attribute-value conditions in classification rules. Consequently, it's entirely conceivable for a larger decision tree to be more comprehensible to the user than a shorter one, as the larger tree may incorporate attributes that are more meaningful or intuitive to the user. To measure the tree size many different metrics are used in literature:

   (a) **Tree Depth**: The depth of a decision tree is a direct indicator of its complexity. Shallow trees are generally easier to understand than deeper trees. [3]

   (b) **Number of Nodes**: The total number of nodes (decision points) in the tree. Fewer nodes usually mean the model is simpler and easier to interpret.

   (c) **Number of Leaves**: The number of leaf nodes or end points of the tree. A smaller number of leaves often correlates with higher interpretability.

2. **Number of features used**: The number of features used by the tree. Using fewer, more relevant features can enhance a model's interpretability. To achieve this feature importance techniques are used and pruning.
3. **Sparsity of nodes**: The proportion of nodes that have few data points associated with them compared to the total number of nodes in the tree.
4. **Leaf Purity**: Measures how homogenous the data points in each leaf are. Higher purity generally means that the model's decisions are more definitive and potentially more interpretable.
5. **Tree Balance**: Evaluates whether the branches of the tree are balanced in terms of depth and the number of nodes. A well-balanced tree is often easier to interpret.
6. **Rule Length**: Average length of the decision rules derived from the tree paths. Shorter rules are typically easier to understand.

Chapter 3

Technical Approaches

While many approaches have been proposed in the literature to enhance the interpretability of machine learning models, these methods can be categorized into three main families: preprocessing techniques, algorithmic modification techniques, and post-processing techniques. In this chapter, we will explore these approaches specifically in the context of decision trees. Preprocessing techniques aim to modify the characteristics of the input data to ensure that any classifier trained on this data achieves high explainability in its predictions. In contrast, algorithmic modification techniques incorporate interpretability constraints directly into the learning algorithm, ensuring that the resulting decision tree model is inherently more explainable. Finally, post-processing techniques adjust the outcomes of an already trained decision tree to enhance its interpretability. Each of these techniques offers unique advantages and challenges, which we will discuss in detail.

3.1 Preprocessing Techniques

In this chapter we would like to extend on preprocessing techniques that can help with making the models more intepretable. As we discussed on chapter 2.3, humans have inherent cognitive limitations that can affect their ability to comprehend complex models. We saw that reaction times and error rates of decision making increase with more complex models. Regardless of the stimuli, the translation of the model to a mental representation (encoding) constitutes the most significant portion of the reasoning process, accounting for approximately half the overall reasoning time which points out the importance of the preprocessing step for the model creation.

3.1.1 Feature Engineering

Feature engineering is a fundamental process in the preparation of machine learning models, where raw data is transformed and enriched to improve model performance and interpretability. This process involves creating new features from existing data, selecting the most relevant features, and transforming features to enhance their usability and effectiveness for predictive modeling. Effective feature engineering can significantly influence the success of a model by exposing the underlying patterns to the learning algorithm more clearly and directly. For decision trees, which split data based on the values of individual features, well-engineered features can lead to simpler and more interpretable trees. This is because better features can often encapsulate complex relationships in simpler forms, reducing the depth and complexity of the decision tree needed to achieve high accuracy. Additionally, feature engineering helps in handling issues like missing values, encoding categorical data, and normalizing numerical ranges, thereby ensuring that the decision tree algorithm focuses on genuine data correlations rather than artifacts of the data representation. By carefully crafting features before building the model, researchers can enhance both the performance and the interpretability of decision trees, making the outcomes more intuitive and trustworthy for end-users.

3.1.2 Feature Selection

Feature selection is a crucial step in the machine learning pipeline that involves selecting a subset of relevant features (variables, predictors) for use in model construction. The primary goals of feature selection are to improve the model's performance, reduce overfitting, and enhance interpretability. By eliminating irrelevant or redundant features, feature selection helps streamline the model, making it faster and more efficient. As discussed in chapter 2.3, there is a great need of limiting the number of features displayed in our model representation as a large number of features seems to only cause errors in human decisions and not contribute further in the process of decision making.

One way to think about feature selection methods are in terms of supervised and unsupervised methods [44]. The difference has to do with whether features are selected based on the target variable or not. Unsupervised feature selection techniques ignores the target variable, such as methods that remove redundant variables using correlation. Supervised feature selection techniques use the target variable, such as methods that remove irrelevant variables. Another way to consider the mechanism used to select features which may be divided into wrapper and filter and embedded methods [74]. These methods are almost always supervised and are evaluated based on the performance of a resulting model on a hold out dataset. Both embedded and wrapper methods perform feature selection in the context of learning machines. In embedded methods, feature selection is part of the learning algorithms and is usually specific to giving learning machines. Wrapper methods wrap around a particular learning algorithm that is used to assess the selected feature subsets in terms of estimated classification errors and to build the final classifier. Recursive feature elimination (RFE) is a good example of a wrapper feature selection method [34]. Filter methods, on the other hand, select subsets of features in terms of criterion functions that are independent of the final classifier used for classification. Usually filter methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to choose (filter) those input variables that will be used in the model. Embedded methods are some machine learning algorithms that perform feature selection automatically as part of learning the model and they algorithms such as penalized regression models like Lasso and decision trees, including ensembles of decision trees like random forest.

One popular method for feature selection involves assessing the importance scores assigned to variables by a machine learning model. This helps identify which features are most relevant for making predictions. Since accurate feature selection is vital, it's essential that the importance scores accurately represent reality. Overestimating the importance of irrelevant features can lead to false discoveries, while underestimating the importance of relevant features may cause us to overlook crucial aspects, ultimately leading to less effective model performance. [24]. This method attempts to quantify the relative importance of each feature for predicting the target variable. The variable importance is calculated by measuring the incremental improvement in performance attributed to each use of a feature inside the model, and summarizing this information across the entire model. Single-tree models such as CART although they are broadly used in Machine Learning are trained using a greedy heuristic that forms the tree one split at a time, which has a number of downsides. First and foremost, this can result in trees that are far from globally optimal, as the best split at any given point in the greedy heuristic may not prove to be the best when viewed in the context of the future growth of the tree. Another key problem often cited in the literature is that the split selection method of CART is biased towards selecting features with a greater number of possible split points [43], [42]. To address this issue a new stream of work has emerged by constructing decision trees with global optimization techniques rather than greedy heuristics. Optimal Classification Trees [9], [10] utilizes mixed-integer optimization to construct decision trees in a single step that are globally optimal. The resulting model maintains the interpretability of a single decision tree, but has been shown to outperform CART and has performance competitive with black-box models. Jack Dunn, Luca Mingardi, Ying Daisy Zhuo [24] investigated the performance of variable importance as a feature selection method for CART, Optimal Trees, XGBoost and they concluded that the strongest performing method for feature selection is Optimal Trees. The main problem with Optimal Decision Trees is that the optimization process used is computationally demanding which can be time-consuming and resource-intensive, especially for large datasets.

Random forests [14], an ensemble learning method, are widely used for feature selection due to their ability to handle large datasets and complex interactions among features. By constructing a multitude of decision trees during training and averaging their predictions, random forests inherently provide a measure of feature importance. Each tree in the forest is built on a different random subset of the data, and features that con-

sistently contribute to reducing impurity (such as Gini impurity or entropy) across these trees are considered important. This process yields a ranked list of features based on their importance scores, allowing for the identification and elimination of irrelevant or redundant features. The advantages of using random forests for feature selection include robustness to overfitting, the ability to handle both numerical and categorical data, and the capability to capture nonlinear relationships. However, one must consider that random forests can be computationally intensive, especially with large datasets and a high number of trees. Additionally, the interpretability of the model may decrease as it becomes more complex. Despite these challenges, random forests remain a powerful and effective tool for feature selection, providing valuable insights into the data's underlying structure and improving the performance of subsequent predictive models.

3.1.3 Incorporating Domain Knowledge

Incorporating domain knowledge during the preprocessing phase is a powerful strategy that can significantly enhance the performance and interpretability of machine learning models. Domain knowledge refers to the expertise and insights specific to the field or industry from which the data originates. By applying this specialized understanding, practitioners can make more informed decisions about data preparation, feature engineering, and transformation processes, ultimately leading to more accurate and relevant models. There has been vast research to incorporate domain knowledge in machine learning models showcasing inspiring results [18], [17].

Using domain knowledge we have the ability to create or transform features as domain experts can identify and create features that capture essential aspects of the data, which may not be immediately apparent through automated methods. For example, in the healthcare domain, combining individual patient metrics like blood pressure, cholesterol levels, and BMI into a single composite health risk score can provide a more meaningful predictor for patient outcomes. Also domain expertise can help in identifying natural groupings or segments within the data. Domain knowledge can assist in transforming numerical data into categorical data, making it more interpretable for humans. For instance, continuous variables such as blood sugar levels can be categorized into 'normal', 'pre-diabetic', and 'diabetic' ranges based on medical standards. This transformation simplifies the interpretation and communication of model outputs to healthcare providers and patients.

Leveraging domain knowledge in the preprocessing phase offers significant benefits, including improved model performance, enhanced interpretability, and increased efficiency. By ensuring that the most relevant and informative features are included, domain expertise can enhance the predictive power of the model. Additionally, features engineered and selected based on medical expertise are often more intuitive and meaningful to healthcare providers, facilitating better understanding and trust in the model. Preprocessing guided by domain knowledge can also streamline the data preparation process, saving time and computational resources. For instance, effectively handling missing values based on medical insights can prevent unnecessary data loss and improve model training efficiency. These benefits collectively make domain knowledge an invaluable asset in building robust and reliable machine learning models.

3.1.4 Supervised Discretization

Supervised discretization is a powerful preprocessing technique that transforms continuous or numerical features into categorical ones by using class label information to guide the discretization process. This method is particularly valuable in scenarios where interpretability and comprehensibility of the model are paramount. By discretizing continuous variables based on the target outcomes, supervised discretization ensures that the resultant categorical bins are optimally aligned with the predictive goals of the model. This alignment not only enhances the transparency of the model by simplifying numerical data into interpretable categories but also potentially increases the predictive accuracy by capturing non-linear dependencies between features and the class labels in a more meaningful way.

In practice, supervised discretization typically employs algorithms such as decision trees to determine optimal bin thresholds. These algorithms evaluate potential splits based on information gain or similar criteria that measure the effectiveness of a split in terms of class differentiation. By partitioning the continuous input space into intervals that correspond to distinct predictive outcomes, the technique effectively translates complex numerical patterns into straightforward, rule-based knowledge. For instance, a decision tree might be used

to bin age data into categories directly correlating with risk levels in a medical diagnosis context, making the model's decisions easier to understand and justify. As a result, models preprocessed with supervised discretization lend themselves better to validation and trust by end-users, meeting the critical requirements of domains such as finance, healthcare, and legal, where explaining a model's decision to a lay audience is often necessary.

3.2 Algorithmic Modification Techniques

In the realm of machine learning, decision trees are celebrated for their inherent interpretability and ease of use. However, as the complexity of the data increases, even decision trees can become intricate and challenging to interpret. Algorithmic modifications present a robust approach to refining decision trees, making them more comprehensive and user-friendly. These modifications involve integrating interpretability constraints directly into the learning algorithms, thereby ensuring that the resulting models remain both accurate and understandable. Techniques such as pruning, constraint-based splitting, and the use of optimal classification trees (OCTs) are pivotal in this endeavor. By simplifying tree structures, reducing overfitting, and emphasizing the most significant features, these algorithmic adjustments enhance the clarity and utility of decision trees. This chapter delves into various algorithmic modifications, illustrating how they contribute to the development of more transparent and interpretable decision tree models.

3.2.1 Enhancing Interpretability in CART, C4.5, and ID3 Decision Trees

Classification and Regression Trees (CART) is a versatile algorithm used for both classification and regression tasks. Developed by Breiman et al. in 1984 [13], CART constructs binary trees by recursively partitioning the data based on features that provide the most significant information gain. For classification, CART uses the Gini impurity as the splitting criterion, while for regression, it uses mean squared error (MSE). The resulting binary tree structure is intuitive and easy to follow, making it a popular choice for many applications.

ID3, developed by Ross Quinlan in 1986 [62], is a foundational algorithm for generating decision trees used primarily for classification tasks. It recursively partitions the dataset based on the attribute that maximizes information gain, a measure derived from entropy. Although ID3 is known for its simplicity and effectiveness in handling categorical data, it does not handle continuous data directly and lacks pruning mechanisms, which can lead to overfitting.

C4.5, developed by Ross Quinlan [61], is an extension of the ID3 algorithm designed to handle both categorical and continuous data. It improves upon ID3 by incorporating features such as handling missing values, pruning trees after they are created, and using gain ratio as a splitting criterion. The gain ratio normalizes information gain to account for the number of splits, making the algorithm more robust. C4.5 is primarily used for classification tasks and is known for generating trees that are easier to interpret.

The three algorithms discussed above—CART, C4.5, and ID3—are greedy algorithms that construct decision trees in a top-down manner. This approach involves recursively selecting the best attribute to split the data at each node, based on specific criteria such as Gini impurity, gain ratio, or information gain. While this method is efficient and effective in generating accurate models, it can sometimes result in complex and less interpretable trees. To enhance the interpretability of the resulting decision trees, several strategies can be employed. These include:

1. Pruning to remove unnecessary branches
2. Limiting tree depth to avoid overly complex structures
3. Set a minimum number of samples required to create a leaf node
4. Careful feature selection to focus on the most informative attributes
5. Generating clear visual representations of the tree
6. Limit the branching factor (see Section 3.2.3)

By applying these techniques, we can create decision tree models that are not only accurate but also transparent and easy to understand.

3.2.2 Strong Optimal Classification Tree

The challenge of learning optimal decision trees is classified as an NP-hard problem, as discussed by Hyafil and Rivest [47] and Breiman et al. [13]. This problem can be intuitively understood as a combinatorial optimization problem with an exponential number of decision variables. At each branching node of the tree, a choice must be made regarding which feature to branch on (and possibly the threshold for that feature), directing each data point to either the left or right branch based on logical constraints.

Traditional algorithms for learning decision trees such as CART, C4.5, and ID3, use greedy techniques to make locally optimal splits. Bertsimas and Dunn [9] recently suggested an alternative method using mixed-integer optimization (MIO) to learn optimal classification trees (OCTs). OCTs employ a global optimization approach using mixed-integer programming to construct the entire tree in a single step. The resulting model maintains the interpretability of a single decision tree, but has been shown to outperform CART and has performance competitive with black-box models. Since the method considers optimizing all splits in the tree simultaneously rather than one-by-one greedily, we might expect that the split selection is less susceptible to the same bias issues as CART [24].

In more detail, OCTs use a mathematical optimization approach to construct the tree by solving a mixed-integer optimization problem. The objective is to find the tree structure that minimizes the classification error while also considering the complexity of the tree. Considering the above, OCTs can identify the best possible splits that traditional greedy algorithms might miss. This global perspective ensures that the tree is both accurate and interpretable. To enhance simplicity and interpretability OCTs are designed to produce simpler models by incorporating constraints that limit the depth and the number of nodes in the tree.

The use of mixed-integer programming for creating an optimal classification tree has gained significant attention in the literature, as demonstrated by the subsequent works of Günlük et al. [33], Aghaei et al. [1], and Verwer and Zhang [71], who have all explored the use of MIO for learning decision trees. This is no coincidence. Firstly, a variety of off-the-shelf MIO solvers and algorithms, such as CPLEX (2009) and Gurobi (2015), have been developed over decades of research and are effective at reducing the search space for MIO problems. Secondly, MIO provides a highly expressive language that allows for customization of the objective function and the addition of practical constraints to the learning problem. For instance, Aghaei et al. [1] utilized MIO to develop fair and interpretable classification and regression trees by incorporating additional constraints, and to create decision trees with complex structures like linear branching and leafing rules.

A key factor in efficiently solving MIOs is creating effective formulations, which is a challenging task. The typical method for solving MIO problems is branch-and-bound, which recursively divides the search space and solves Linear Optimization (LO) relaxations for each division to generate bounds that can eliminate parts of the search space. Consequently, since solving an MIO involves tackling a large number of LO problems, having small and compact formulations is advantageous as it allows the LO relaxations to be solved more quickly. Aghaei et al. [2] at their work introduce a flow-based MIO formulation for learning optimal classification trees with binary features. In this model, correctly classified data points are viewed as flowing from the root to an suitable leaf, while incorrectly classified data points are restricted from flowing through the tree. Their formulation can be easily enhanced with constraints (such as fairness), regularization penalties, and can be adapted to address imbalanced datasets. They exploit the max-flow structure of the subproblems to solve them efficiently using a customized min-cut procedure.

A crucial step in this flow-based MIO formulation involves transforming the decision tree of fixed depth into a directed acyclic graph, where all arcs flow from the tree's root to its leaves. Figure 3.2.1 (left) illustrates a imbalanced decision tree. The core concept of this model is to convert this imbalanced decision tree into a directed acyclic graph by adding a single source node s connected to the tree's root node (node 1) and a single sink node t connected to all the tree's leaf nodes. We call this graph the flow graph of the decision tree. Figure 3.2.1 provides an illustration of these ideas applied to a decision tree of depth d=2.

For our work, we used FlowOCT model which is the flow-based formulation of the problem for an imbalanced decision tree. In more details, given an imbalanced decision tree of depth d, they define its associated directed

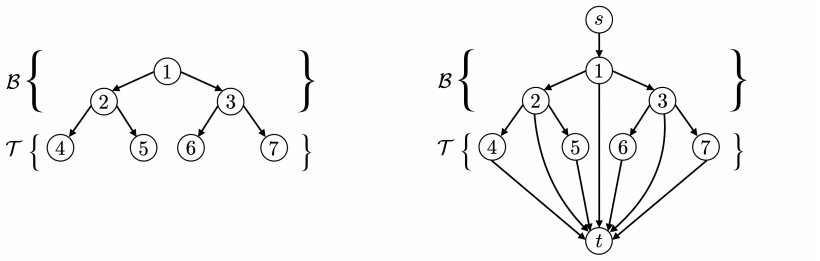

Figure 3.2.1: A decision tree of depth 2 (left) and its associated flow graph (right).

flow graph $G = (\mathcal{V}, \mathcal{A})$ as follows. Let $\mathcal{V} := \{s, t\} \cup \mathcal{B} \cup \mathcal{T}$ be the vertices of the flow graph. Given $n \in \mathcal{B}$, let $\ell(n) := 2n$ be the left descendant of n, $r(n) := 2n + 1$ be the right descendant of n, and

$$\mathcal{A} := \{(n, \ell(n)) : n \in \mathcal{B}\} \cup \{(n, r(n)) : n \in \mathcal{B}\} \cup \{(s, 1)\} \cup \{(n, t) : n \in \mathcal{B} \cup \mathcal{T}\}$$

be the arcs of the graph. Also, given $n \in \mathcal{B} \cup \mathcal{T}$, let $a(n)$ be the parent of n, defined through $a(1) := s$ and $a(n) := \lfloor n/2 \rfloor$ if $n \neq 1$.

The formulation of the problem that allows the design of imbalanced classification trees is described below. The classification tree is described through the branching variables $\mathbf{b}$ and the prediction variables $\mathbf{w}$. In particular, the variables $b_{nf} \in \{0, 1\}$, $f \in \mathcal{F}$, $n \in \mathcal{B}$ are used to indicate if the tree branches on feature f at branching node n (i.e., it equals 1 if and only if the binary test performed at n asks "is $x^i_f = 0$"?). Accordingly, the variables $w^n_k \in \{0, 1\}$, $n \in \mathcal{T}$, $k \in \mathcal{K}$ are used to indicate that at leaf node n the tree predicts class k. They use the auxiliary routing/flow variables $\mathbf{z}$ to decide on the flow of data through the flow graph associated with the decision tree. Specifically, for each node $n \in \mathcal{B} \cup \mathcal{T}$ and for each datapoint $i \in \mathcal{I}$, they introduce a decision variable $z^i_{a(n),n} \in \{0, 1\}$ which equals 1 if and only if the ith datapoint is correctly classified and its flow traverses the arc $(a(n), n)$ on its way to the sink t. Variable $z^i_{n,t}$ is defined accordingly for each arc between node $n \in \mathcal{T}$ and sink t. Datapoint $i \in \mathcal{I}$ is correctly classified if and only if its corresponding flow passes through some leaf node $n \in \mathcal{T}$ such that $w^n_{y_i} = 1$, i.e., where the class predicted coincides with the class of the datapoint. If the flow of a datapoint i arrives at such a leaf node n and the datapoint is correctly classified, its corresponding flow is directed to the sink, i.e., $z^i_{n,t} = 1$; otherwise, the corresponding flow is not initiated from the source at all. With these variables, the flow-based formulation reads. In addition to the decision variables described above, for every node $n \in \mathcal{B} \cup \mathcal{T}$, the binary decision variable p_n is introduced which has a value of one if and only if node n is a leaf node of the tree, i.e., if a prediction is made at node n. The auxiliary routing/flow variables $\mathbf{z}$ now account for all arcs in the flow graph introduced in Definition 5. The problem of learning optimal imbalanced classification trees is then expressible as

$$\text{maximize} \quad (1 - \lambda) \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{B} \cup \mathcal{T}} z^i_{n,t} - \lambda \sum_{n \in \mathcal{B}} \sum_{f \in \mathcal{F}} b_{nf} \tag{1a}$$

$$\text{subject to} \sum_{f \in \mathcal{F}} b_{nf} + p_n + \sum_{m \in \mathcal{P}(n)} p_m = 1 \quad \forall n \in \mathcal{B} \tag{1b}$$

$$p_n + \sum_{m \in \mathcal{P}(n)} p_m = 1 \quad \forall n \in \mathcal{T} \tag{1c}$$

$$z^i_{a(n),n} = z^i_{n,\ell(n)} + z^i_{n,r(n)} + z^i_{n,t} \quad \forall n \in \mathcal{B}, i \in \mathcal{I} \tag{1d}$$

$$z^i_{a(n),n} = z^i_{n,t} \quad \forall n \in \mathcal{T}, i \in \mathcal{I} \tag{1e}$$

$$z^i_{s,1} \leq 1 \quad \forall i \in \mathcal{I} \tag{1f}$$

$$z^i_{n,\ell(n)} \leq \sum_{f \in \mathcal{F}: x^i_f = 0} b_{nf} \quad \forall n \in \mathcal{B}, i \in \mathcal{I} \tag{1g}$$

$$z^i_{n,r(n)} \leq \sum_{f \in \mathcal{F}: x^i_f = 1} b_{nf} \quad \forall n \in \mathcal{B}, i \in \mathcal{I} \tag{1h}$$

$$z^i_{n,t} \leq w^n_{y_i} \quad \forall n \in \mathcal{B} \cup \mathcal{T}, i \in \mathcal{I} \tag{1i}$$

$$\sum_{k \in \mathcal{K}} w^n_k = p_n \quad \forall n \in \mathcal{B} \cup \mathcal{T} \tag{1j}$$

$$w^n_k \in \{0,1\} \quad \forall n \in \mathcal{B} \cup \mathcal{T}, k \in \mathcal{K} \tag{1k}$$

$$b_{nf} \in \{0,1\} \quad \forall n \in \mathcal{B}, f \in \mathcal{F} \tag{1l}$$

$$p_n \in \{0,1\} \quad \forall n \in \mathcal{B} \cup \mathcal{T} \tag{1m}$$

$$z^i_{a(n),n}, z^i_{n,t} \in \{0,1\} \quad \forall n \in \mathcal{B} \cup \mathcal{T}, i \in \mathcal{I} \tag{1n}$$

where $\mathcal{P}(n)$ is the set of all ancestors of node $n \in \mathcal{B} \cup \mathcal{T}$ and $\lambda \in [0,1]$ is a regularization parameter.

A short explanation of the constraints follows:

- **1b:** imply that at any node $n \in \mathcal{B}$ we either branch on a feature f (if $\sum_{f \in \mathcal{F}} b_{nf} = 1$), predict a label (if $p_n = 1$), or get pruned if a prediction is made at one of the node ancestors (i.e., if $\sum_{m \in \mathcal{P}(n)} p_m = 1$).
- **1c:** ensure that any node $n \in \mathcal{T}$ is either a leaf node of the tree or is pruned.
- **1d:** are flow conservation constraints for each datapoint i and node $n \in \mathcal{B}$: they ensure that if a datapoint arrives at a node, then it must also leave the node through one of its descendants.
- **1e:** enforce flow conservation for each node $n \in \mathcal{T}$
- **1f:** The inequality constraints (1f) imply that at most one unit of flow can enter the graph through the source for each datapoint.
- **1g-1h:** ensure that if the flow of a datapoint is routed to the left (resp. right) at node n, then one of the features such that $x^i_f = 0$ (resp. $x^i_f = 1$) must have been selected for branching at the node.
- **1i:** guarantee that datapoints whose flow is routed to the sink node t are correctly classified.
- **1j:** imply that if a node n gets pruned we do not predict any class at the node, i.e., $w^n_k = 0$ for all $k \in \mathcal{K}$.

A penalty term is added to (1a), to encourage sparser trees with fewer branching decisions. Note that while it is feasible to design a decision tree with the same branching decisions in multiple nodes on a single path from root to sink, such solutions are never optimal for (1) if $\lambda > 0$, as a simpler tree would result in the same misclassification.

In our research, we opted to use Flow-based Optimal Classification Trees (FlowOCT) over classic Optimal Classification Trees (OCT) due to their significant performance advantages and enhanced interpretability. FlowOCT models have been demonstrated to be up to 29 times faster than their classic counterparts, making them highly efficient for our purposes [2]. This efficiency, coupled with their ability to maintain a balanced and sparse tree structure, leads to shorter and more consistent decision paths, thereby enhancing the model's interpretability. However, the primary drawback of the FlowOCT model is its reliance on binary data, which may limit its applicability in scenarios requiring the analysis of continuous or multi-class categorical data.

3.2.3 C4.5 Modifications

C4.5, developed by Ross Quinlan, is a widely used algorithm for generating decision trees from a dataset. It extends the earlier ID3 algorithm and is primarily used for classification tasks. C4.5 constructs a tree by recursively partitioning the data based on the attribute that provides the highest information gain ratio at each node. This algorithm is capable of handling both categorical and continuous attributes, which it can convert into discrete intervals. It also incorporates mechanisms for handling missing values and pruning trees after their creation to improve generalization and avoid overfitting. These features make C4.5 a robust and versatile tool for various classification problems.

Handling Categorical Data in C4.5

The C4.5 algorithm handles categorical data by evaluating the information gain ratio of each categorical attribute to determine the best split at each node. For a categorical attribute with multiple distinct values, C4.5 considers each possible value as a potential branch in the decision tree. The algorithm computes the information gain for each attribute and then normalizes this value using the split information to obtain the gain ratio. The attribute with the highest gain ratio is selected for the split.

The information gain for an attribute A is calculated as follows:

$$\text{Information Gain}(A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v)$$

where: - Entropy(S) is the entropy of the original dataset S, - Values(A) is the set of all distinct values for attribute A, - S_v is the subset of S where attribute A has value v, - $\frac{|S_v|}{|S|}$ is the proportion of instances in S with attribute A equal to v.

Entropy is defined as:

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

where c is the number of classes and p_i is the proportion of instances in class i.

To account for the number of distinct values in the attribute, C4.5 calculates the split information as:

$$\text{Split Information}(A) = -\sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \log_2\left(\frac{|S_v|}{|S|}\right)$$

The gain ratio is then given by:

$$\text{Gain Ratio}(A) = \frac{\text{Information Gain}(A)}{\text{Split Information}(A)}$$

By using the gain ratio, C4.5 favors attributes that result in meaningful splits, avoiding biases towards attributes with many distinct values. This approach ensures that the chosen splits provide substantial information about the classification, thereby improving the accuracy and interpretability of the decision tree.

Another challenge in decision tree interpretation is that some subtrees of the constructed tree can contain irrelevant attributes, even when the data is not noisy. This issue arises because the decision tree structure rigidly requires that once an attribute is selected to label a node, each value of that attribute must be included in the tree. Consequently, some branches may be added solely to maintain the tree structure, even if those branches are associated with irrelevant attribute values [12], [16], [28], [30]. These irrelevant values can mislead the user's interpretation of the tree and may lead to overfitting.

For example, consider a decision tree where the attribute "Temperature" is used to split data into "Cold", "Mild" and "Hot". If attribute value "Cold" is irrelevant in the context of the decision, including it in the tree structure can confuse the interpretation and decrease the model's effectiveness. Or maybe if the child trees for 2 different attribute values result to the same child trees there is duplicated information in the tree.

To address this issue, we propose grouping the non-important attribute values into a single value labeled 'Other' (see Figure 3.2.2). This involves calculating the information gain for all possible combinations of splits and groupings, and selecting the split that maximizes information gain. By consolidating less significant attributes, we can simplify the tree structure, reduce the branching factor, enhance interpretability, and reduce the risk of overfitting. The pseudocode for the algorithm is displayed in Algorithm 2.

Algorithm 2: C4.5 categorical attributes grouping

Input: *dataset*, *attributes*
Output: *tree* ▷ Decision Tree

```
1  Function BuildTree(dataset, attributes):
2      if all instances in dataset have the same class then
3          return a leaf node with that class
4      end
5      if attributes is empty then
6          return a leaf node with the majority class of dataset
7      end
8      best_attribute ← None
9      best_split ← None
10     best_gain ← −∞
11     best_grouping ← −∞
12     foreach attribute ∈ attributes do
13         foreach grouping ∈ possibleGroupings(attribute) do
14             split ← calculateSplit(dataset, attribute, grouping)
15             gain ← calculateInformationGain(dataset, split)
16             if gain > best_grouping_gain then
17                 best_grouping ← grouping
18                 best_grouping_gain ← gain
19             end
20         end
21         if best_grouping_gain > best_gain then
22             best_attribute ← attribute
23             best_split ← best_grouping
24             best_gain ← best_grouping_gain
25         end
26     end
27     tree ← DecisionNode(best_attribute)
28     attributes ← remove(attributes, best_attribute)
29     foreach value ∈ best_split do
30         subset ← splitDataset(dataset, best_attribute, value)
31         if subset is empty then
32             child_node ← LeafNodemajorityClass(dataset)
33         end
34         else
35             child_node ← BuildTree(subset, attributes)
36         end
37         tree.addChild(value, child_node)
38     end
39     return tree
40 End Function
```

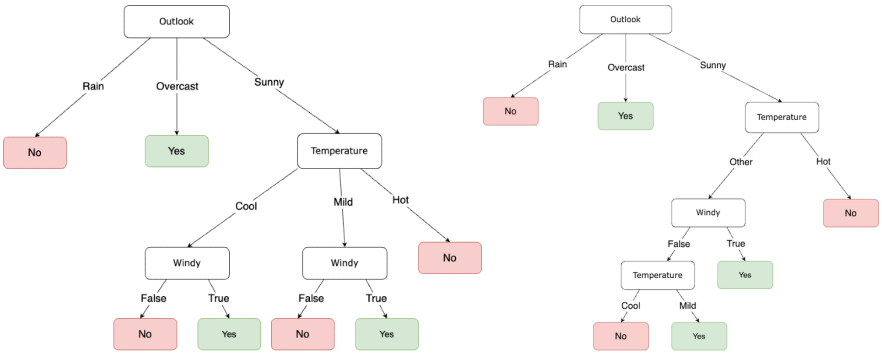

Figure 3.2.2: A decision tree without grouping (left) and a decision tree with grouping (right).

Handling Continuous Data in C4.5

The C4.5 algorithm extends its predecessor, ID3, by providing a robust method for handling numeric (continuous) features. Unlike categorical features, which have a fixed set of discrete values, numeric features can take on a potentially infinite range of values. C4.5 addresses this challenge by converting continuous attributes into discrete intervals through a process called thresholding.

Thresholding Process

1. **Sorting the Data**: For each numeric attribute, the algorithm begins by sorting the dataset based on the values of that attribute. This sorting helps in identifying potential split points that can divide the data into meaningful intervals.

2. **Identifying Potential Split Points**: Potential split points are identified between each pair of consecutive values in the sorted list. For a numeric attribute A, with sorted values $v_1, v_2, \ldots, v_n$, potential split points are calculated as the midpoints between these consecutive values:

$$\text{split\_point}_i = \frac{v_i + v_{i+1}}{2}$$

3. **Calculating Information Gain for Each Split**: For each potential split point, the algorithm calculates the information gain. This involves splitting the dataset at the threshold and computing the entropy for the resulting subsets. The information gain for a split at threshold T is given by:

$$\text{Information Gain}(A, T) = \text{Entropy}(S) - \left(\frac{|S_{\leq T}|}{|S|} \times \text{Entropy}(S_{\leq T}) + \frac{|S_{>T}|}{|S|} \times \text{Entropy}(S_{>T}) \right)$$

where: - $\text{Entropy}(S)$ is the entropy of the original dataset S, - $S_{\leq T}$ is the subset of S where the attribute A has values less than or equal to T, - $S_{>T}$ is the subset of S where the attribute A has values greater than T.

4. **Selecting the Best Split Point**: The algorithm evaluates all potential split points and selects the one that maximizes the information gain. This optimal split point is then used to create a binary split in the decision tree:

$$\text{If } A \leq \text{split\_point then go to left subtree else go to right subtree}$$

5. **Handling Multiple Numeric Features**: This process is repeated for each numeric attribute in the dataset. The attribute with the highest gain ratio (information gain normalized by the split information) is chosen to split the node.

Consider a numeric attribute **Age** with the following values: 22, 25, 30, 35, 40. The potential split points would be 23.5, 27.5, 32.5, and 37.5. The algorithm calculates the information gain for each split point and selects the one with the highest gain. By discretizing numeric features into binary splits based on optimal threshold values, C4.5 effectively incorporates continuous data into the decision tree.

In our work on enhancing the C4.5 algorithm, we introduced an approach to handle numeric attributes by allowing multiway splits. Traditional C4.5 uses binary splits for numeric attributes, dividing the data into two groups based on a single threshold. While effective, this method can sometimes lead to overly complex trees with many levels as the attributes involved should be able to appear several times in the paths from the root of the tree to its leaves [27], reducing interpretability. To address this, we implemented multiway splits for numeric attributes, which can create multiple intervals and simplify the tree structure. Our work was based on the works of Fayyad and Irani [25] and Dougherty et al.

Implementation Details

1. **Parameter Introduction**: We introduced a parameter called **max_splits** to control the maximum number of split points for each numeric attribute. This parameter allows the user to specify the desired level of granularity for splitting numeric attributes.
2. **Identifying Potential Split Points**: For each numeric attribute, the dataset is sorted based on the attribute values.Potential split points are identified where there is a change in class between two consecutive data points. This ensures that splits are only considered at meaningful transitions in the data distribution.
3. **Calculating Information Gain**: For each numeric attribute, we evaluate splits ranging from 1 to *max_splits* possible points. Information gain is calculated for each combination of splits to determine the optimal set of split points. The information gain for multiway splits is given by:

$$\text{Information Gain}(A, T_1, T_2, \ldots, T_k) = \text{Entropy}(S) - \sum_{i=1}^{k+1} \left(\frac{|S_i|}{|S|} \times \text{Entropy}(S_i) \right)$$

   where $T_1, T_2, \ldots, T_k$ are the selected split points, and S_i are the subsets created by these splits.
4. **Selecting the Best Split Points**: The algorithm selects the combination of split points that maximizes the information gain. These split points are then used to create multiway branches in the decision tree.

Enhancing Interpretability

The use of multiway splits for numeric attributes enhances interpretability in several ways as it is shown also in Figure 3.2.3:

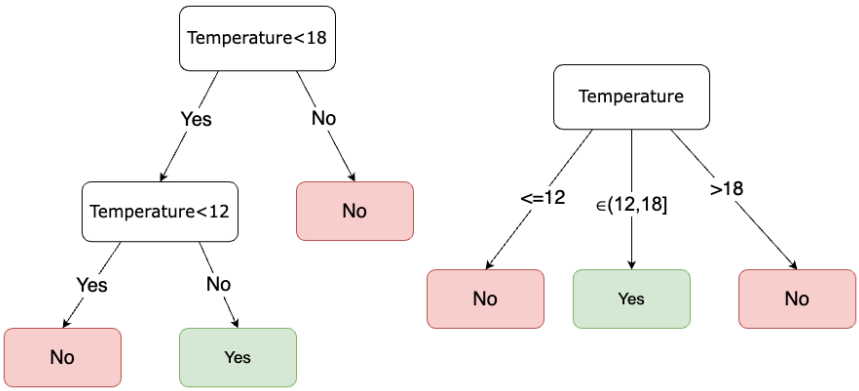

Figure 3.2.3: Binary split for numeric attributes of a decision tree (left) and multiway splits for numericattributes (right).

1. **Simplified Tree Structure**: By creating multiple intervals in a single split, the tree can represent complex decision boundaries more clearly. This reduces the depth of the tree and avoids long chains of binary splits, making the tree easier to understand.
2. **Clearer Decision Boundaries**: Multiway splits create clear and distinct decision boundaries, which are easier for users to follow. This is particularly beneficial when numeric attributes exhibit significant variability and multiple meaningful thresholds exist.

3. **Reduced Overfitting**: Allowing multiway splits helps in capturing the true data distribution more effectively, which can reduce overfitting. A tree with well-defined multiway splits is less likely to include irrelevant or spurious splits that complicate interpretation.

4. **Improved Insight**: - Presenting numeric data in multiple intervals provides more granular insights into how different ranges of values impact the decision process. Users can better understand the influence of numeric attributes on the outcome by examining these intervals.

By implementing multiway splits for numeric attributes, our enhanced C4.5 algorithm produces more interpretable and effective decision trees. This approach leverages the flexibility of multiway splits to create simpler, clearer, and more informative models.

While the introduction of multiway splits for numeric attributes in our modified C4.5 algorithm enhances interpretability and model simplicity, it also introduces a significant computational challenge. The primary downside of this approach is its increased computational complexity compared to the traditional binary splits used in the existing C4.5 algorithm.

Existing C4.5 Algorithm Complexity

In the traditional C4.5 algorithm, handling numeric attributes involves the following steps:

1. **Sorting the Data**: The dataset is sorted based on the numeric attribute. This operation has a complexity of $O(n \log n)$, where n is the number of instances in the dataset.

2. **Identifying Potential Split Points**: Potential split points are identified between each pair of consecutive values, resulting in up to $n - 1$ potential splits.

3. **Calculating Information Gain**: For each potential split, the information gain is calculated, which involves evaluating the entropy of the resulting subsets. This step has a complexity of $O(n)$ for each split point.

Overall, the complexity for handling numeric attributes in the traditional C4.5 algorithm is:

$$O(n \log n) + O(n \times (n-1)) = O(n^2)$$

Enhanced C4.5 Algorithm Complexity

In our modified C4.5 algorithm, the process for handling numeric attributes with multiway splits is more computationally intensive due to the following steps:

1. **Sorting the Data**: Similar to the existing algorithm, the dataset is sorted based on the numeric attribute, with a complexity of $O(n \log n)$.

2. **Identifying Potential Split Points**: Potential split points are identified where there is a change in class between consecutive data points, resulting in up to $n - 1$ potential splits.

3. **Evaluating Multiple Splits**: For each numeric attribute, we evaluate splits ranging from 1 to 'max_splits' possible points. The number of possible combinations of k splits from $n - 1$ potential split points can be calculated using binomial coefficients, leading to a complexity of $O\left(\binom{n-1}{k}\right)$.

4. **Calculating Information Gain**: For each combination of splits, the information gain is calculated, which involves evaluating the entropy of the resulting subsets. This step has a complexity of $O(n)$ for each combination of splits.

Given that evaluating all possible combinations of splits significantly increases the computational burden, the overall complexity for handling numeric attributes with multiway splits is:

$$O(n \log n) + O\left(\sum_{k=1}^{\text{max\_splits}} \binom{n-1}{k} \times n\right)$$

This complexity can be prohibitive, especially for large datasets or when 'max_splits' is high, as the number of combinations grows exponentially with n.

3.3 Post-Processing techniques

To enhance the interpretability of decision trees, several post-processing techniques can be employed. One effective method is post-pruning, which involves removing branches that do not contribute significantly to the model's accuracy after the tree has been fully grown. By using a validation set to evaluate the impact of each branch, post-pruning helps to eliminate unnecessary complexity and overfitting, resulting in a simpler and more generalizable tree. This technique ensures that the tree remains focused on the most relevant splits, making it easier for users to follow and understand the decision-making process.

Tree visualization is another crucial technique for improving interpretability. Enhanced visualizations utilize clear and concise representations to depict the decision tree structure. Techniques such as color-coding nodes based on class probabilities, highlighting important paths, and adding tooltips with additional information can make the tree more accessible and intuitive. Interactive visualizations further enhance user experience by allowing users to dynamically explore the tree, expanding and collapsing branches, viewing detailed split information, and tracing specific decision paths. These visual aids help users to quickly grasp the decision logic and identify key features influencing the outcomes.

Node simplification and tree balancing are also essential for creating more interpretable decision trees. Node simplification involves merging similar nodes that have redundant decision criteria or lead to the same classification, thereby reducing redundancy and streamlining the tree. Additionally, insignificant nodes that contribute little to the overall accuracy can be removed, resulting in a cleaner model. Tree balancing aims to maintain a uniform depth across different branches, preventing the formation of deep and complex branches that are hard to interpret. Techniques such as depth-based pruning or reordering splits ensure that the tree remains balanced, with similar depths for different paths, facilitating easier comprehension and analysis. Together, these techniques contribute to the creation of decision trees that are not only accurate but also transparent and user-friendly.

Chapter 4

User Study

The objective of this study was to measure and compare the interpretability of decision trees generated from various algorithms and datasets. To achieve this, we recruited 52 participants and divided them into two groups of 29 and 23 participants each. The participants were asked to complete forms containing multiple decision tree models and classify samples of data using these models.

4.1 General description of the experiment

4.1.1 Group Analysis

To understand the background and familiarity of the participants with artificial intelligence and decision trees, we asked each participant to answer three preliminary questions on a scale from 1 to 5:

1. **AI Familiarity**: How familiar are you with artificial intelligence?
2. **XAI Familiarity**: How familiar are you with explainable artificial intelligence (XAI)?
3. **Decision Trees Familiarity**: How familiar are you with decision trees? These questions helped us gauge the participants' prior knowledge and allowed us to analyze any potential influence of familiarity on the interpretability of the decision trees.

The general user familiarity is displayed in Figure 4.1.1. The familiarity levels of the users by version are displayed in Figure 4.1.2.

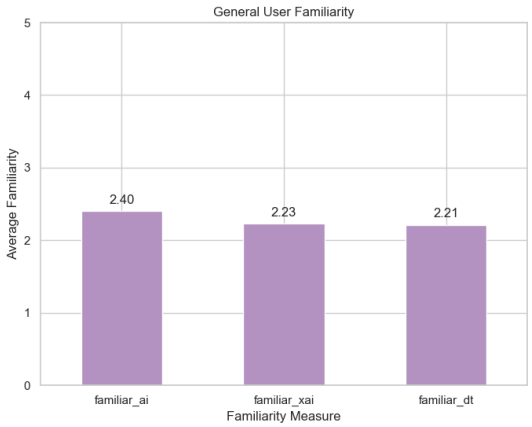

Figure 4.1.1: User's AI familiarity level

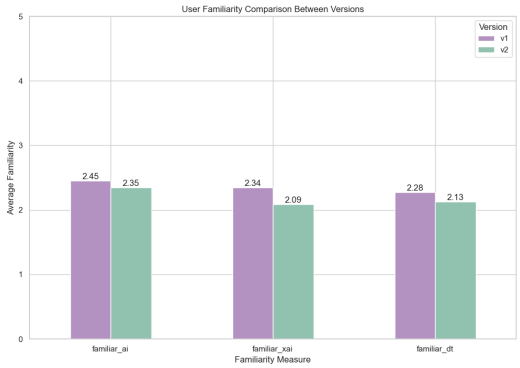

Figure 4.1.2: User's AI familiarity level by version

4.1.2 User Forms

Each participant completed a form that included four different decision tree models derived from four distinct datasets. For each model, the participants were provided with five different samples of data that they were required to classify using the decision tree.

After classifying each sample, participants answered three questions to assess their experience and confidence:

1. **Answer Confidence**: How confident are you in your answer? (Scale: 1 to 5)
2. **Clarity of Decision Path**: I was able to clearly follow the decision path from the root to the leaf nodes. (Scale: 1 = Strongly Disagree to 5 = Strongly Agree)
3. **Comprehensibility of Structure**: The overall structure of the decision tree was easy to comprehend. (Scale: 1 to 5)
4. Additionally, participants were asked to describe the model in their own words. This open-ended question provided qualitative insights into their understanding and interpretation of the decision tree.

4.1.3 Form implementation

In our user study, accurately measuring response time for each classification task was crucial. Additionally, due to the large size of the resulting decision trees, it was important for users to have the capability to zoom into the decision tree images. Since no known form provider offered both of these features, we decided to implement a custom solution.

The front-end of the application is a dynamic form built using the React library. Its goal is to provide a user-friendly and efficient platform for collecting data from users, incorporating advanced features for an enhanced user experience. The application records the time it takes for the user to respond to each question on the form. By using hooks like **useEffect** and **useState**, the application starts a timer when the user navigates to a question and stops the timer when the answer is submitted. The response time data is stored and can be analyzed to optimize the form and understand user behavior.

The form also supports viewing images with zoom capability, allowing users to examine details of the images. This feature is implemented using libraries like *react-zoom-pan-pinch*. Users can click on an image to enlarge it and inspect it in greater detail, thereby improving the accuracy of their responses when images are part of the questions. We hosted the frontend on Vercel, a cloud platform for static sites and serverless functions that enables easy deployment and scaling of web applications.

For the backend, we used a Flask application. Flask is a lightweight WSGI web application framework in Python, designed to make getting started quick and easy, with the ability to scale up to complex applications. It provided the necessary endpoints for handling the form submissions and processing the response times.

To store the collected data, we used MongoDB, a NoSQL database known for its flexibility and scalability. MongoDB stores data in JSON-like documents, making it easy to store and query the diverse data generated from the user study.

4.2 Experiment Set Up

4.2.1 Datasets

For our research we used four datasets from a variety of domains, including credit scoring, healthcare, and criminal justice to demonstrate the universal value and applicability of interpretability in predictive modeling. These fields were specifically chosen due to their significant societal impact and the critical need for transparency and accountability in the decision-making processes that affect individual lives. By applying explainable artificial intelligence (XAI) techniques across these diverse sectors, we highlight how crucial it is to provide clear, understandable models that stakeholders can trust and scrutinize. This approach not only enhances the acceptance and effectiveness of predictive models in high-stakes settings but also underscores the ethical imperative of explainability in AI across various disciplines.

1. COMPAS [46]
2. German Credit [38]
3. Framingham Heart Study [11]
4. Adult Income [7]

4.2.2 COMPAS

The dataset has criminal records and demographics features for 7,214 defendants released on bail at U.S state courts during 1990-2009. The COMPAS dataset consists of the results of a commercial algorithm called COMPAS (Correctional Offender Management Profiling for Alternative Sanctions), used to assess a convicted criminal's likelihood of reoffending. COMPAS has been used by judges and parole officers and is widely known for its bias against African-Americans.

Objective

The primary task with this dataset is to classify defendants into two categories:

- **Recid**: Defendants who are likely to re-offend
- **No Recid**: Defendants who are not likely to re-offend

Pre-processing

The original dataset contained 7214 cases of defendants with 53 features describing each defendant. However, for the purpose of this study, we excluded features related to the COMPAS decision-making process and kept only the demographic information and criminal history. This resulted in a final set of 8 features:

- **Sex**: The sex of the defendant (e.g., Male, Female).
- **Age**: The age of the defendant at the time of the crime.
- **Race**: The race of the defendant (e.g., Caucasian, African-American).
- **juv_fel_count**: The number of juvenile felony charges the defendant has.
- **juv_misd_count**: The number of juvenile misdemeanor charges the defendant has.
- **juv_other_count**: The number of other juvenile charges the defendant has.
- **priors_count**: The number of prior criminal charges the defendant has.
- **c_charge_degree**: The degree of the current charge (e.g., Misdemeanor, Felony).

After dropping the unnecessary columns, pre-processing steps for this dataset involved:

- **Handling Missing Values**: We dropped variables with null values to ensure data integrity.
- **Feature Importance Measurement**: To assess the significance of different features within our dataset, we utilized the Random Forest algorithm, a robust ensemble learning method. Specifically, we employed a Random Forest classifier with 100 estimators (trees). The choice of Random Forest for this task was motivated by its ability to handle high-dimensional data and provide insights into feature importance through its construction of multiple decision trees. The Random Forest classifier was implemented using the RandomForestClassifier from the scikit-learn library [58]. Feature importances were derived from the trained model, which offers an aggregated view of feature significance based on the reduction of impurity (Gini impurity or entropy) across all trees in the ensemble. The resulting importance scores were visualized in Figure 4.2.1.
- **Encoding Categorical Features**: We used one-hot encoding for categorical features such as race, sex, and charge degree to convert them into a format suitable for machine learning algorithms.

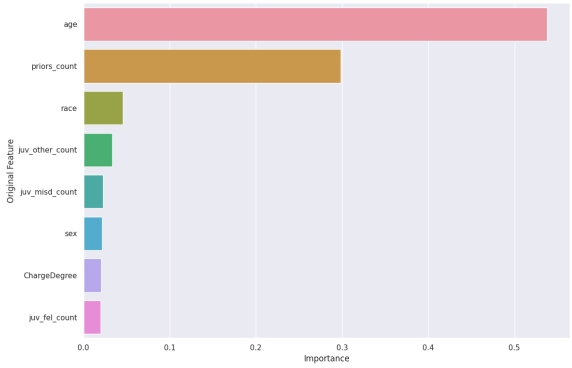

Figure 4.2.1: COMPAS Features Importance

Decision Trees Created

For the COMPAS dataset, we created two decision tree models using different algorithms to compare their interpretability:

Model 1: Created using the DecisionTreeClassifier from scikit-learn python library [58] which uses an optimized version of the CART algorithm. The model with the highest accuracy was selected. The resulting tree structure involved binary splits typical of the CART algorithm, and depth $d = 5$

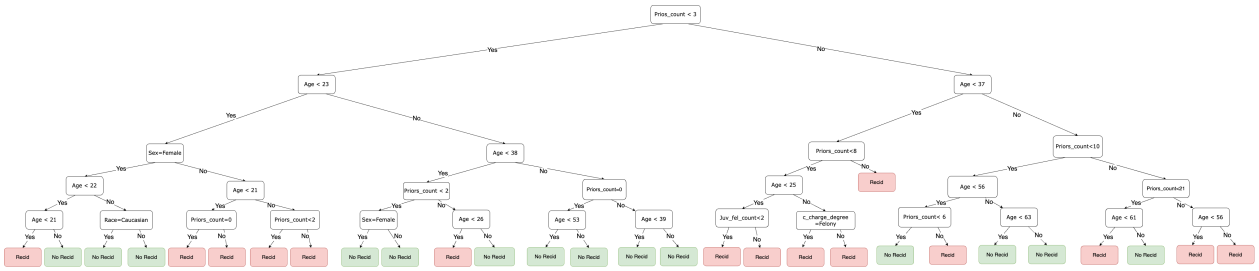

Figure 4.2.2: COMPAS CART model

Model 2: For this model an extra step of data pre-processing was needed, supervised discretization. As we discussed in chapter 3.2, FlowOCT has a limit of using only binary variables for classification. For continuous attributes we have 2 options. Either use the embedded function *binarize* to the odtlearn package the implementation of which can be found online at https://github.com/D3M-Research-Group/StrongTree, or use supervised dicretization.

For numeric features with a high number of unique values, we employed supervised discretization. This method involves using target variable information to create bins, ensuring that the binning process is informed by the underlying relationship between the feature and the target. To implement this we used the *DecisionTreeClassifier* provided by the scikit-learn python library [58] for the numeric features Age and priors_count. Supervised discretization helps in preserving the predictive power of the feature while reducing its complexity and improving model interpretability.

For numeric features with a relatively small number of unique values, we utilized the embedded *binarize* function from the odtlearn package. Binarization transforms the feature values into binary values (0 or 1) based on the unique values. The FlowOCT model was generated using the Gurobi solver.

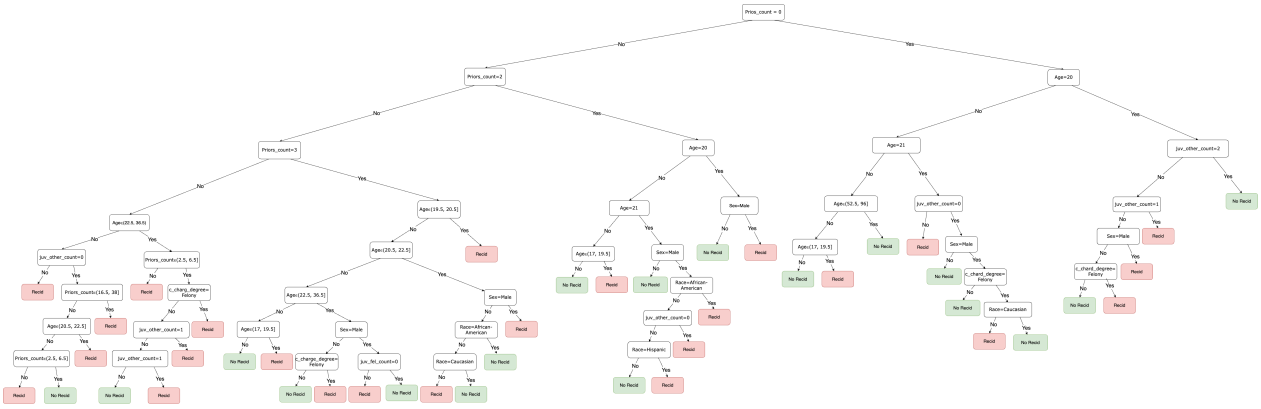

Figure 4.2.3: COMPAS FlowOCT model

Differences Between the Trees

As it is obvious from the figures of the trees, the first model is a balanced decision tree using splits based on a single threshold. The second model is imbalanced and uses binned attributes for the classification. We want to evaluate how the balance of the tree enhances the intepretability and if binned numeric features are comprehensive enough for the users.

4.2.3 German Credit

German Credit dataset [38] comprises of demographic (age, gender), personal (marital status), and financial (Credit Amount, Checking Amount) features from 1,000 credit applicants, where they are categorized into good vs. bad customer depending on their credit risk. Financial institutions use this information to evaluate the risk associated with lending to these individuals. There are 2 datasets provided one containing categorical attributes and one containing numeric. For our study we used the categorical dataset.

Objective

The primary objective of analyzing the German Credit dataset is to classify individuals into two categories:

- **Good Credit**: Individuals who are considered creditworthy and pose a lower risk to financial institutions.
- **Bad Credit**: Individuals who are considered not creditworthy and pose a higher risk to financial institutions.

This classification helps in assessing the risk associated with lending to these individuals, aiding financial institutions in their decision-making processes.

Pre-processing

The original dataset contained 1000 credit applicants with 9 features describing each applicant.

- **Age**: The age of the individual in years (numeric).

- **Sex**: The sex of the individual (e.g., male, female).
- **Job**: The type of job the individual has (unskilled and non-resident, unskilled and resident, skilled, highly skilled).
- **Housing**: The type of housing the individual resides in (e.g., own, rent, or free).
- **Saving accounts**: The amount of savings the individual has (e.g., little, moderate, quite rich, rich).
- **Checking account**: The amount in the individual's checking account (numeric, in Deutsch Mark).
- **Credit amount**: The amount of credit the individual has applied for (numeric, in Deutsch Mark).
- **Duration**: The duration of the credit in months (numeric).
- **Purpose**: The purpose for which the credit is requested (e.g., car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others).

After dropping the unnecessary columns, pre-processing steps for this dataset involved:

- **Handling Missing Values**: We dropped variables with null values to ensure data integrity.
- **Feature Importance Measurement**: To assess the significance of different features within our dataset, we utilized the Random Forest algorithm, a robust ensemble learning method. Specifically, we employed a Random Forest classifier with 100 estimators (trees). The Random Forest classifier was implemented using the RandomForestClassifier from the scikit-learn library [58]. Feature importances were derived from the trained model, which offers an aggregated view of feature significance based on the reduction of impurity (Gini impurity or entropy) across all trees in the ensemble. The resulting importance scores were visualized in Figure 4.2.4.

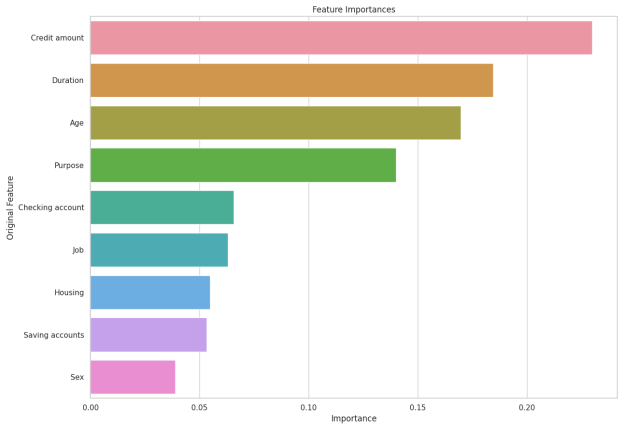

Figure 4.2.4: German Credit Features Importance

Decision Trees Created

For the German Credit dataset, we created two decision tree models using different algorithms to compare their interpretability:

Model 1: To implement the decision tree classifier, we used the *c45-decision-tree* library, which is available on PyPI [59]. This library handles only categorical data and implements multiway splits. The resulting tree is shown in Figure 4.2.5.

Model 2: For this model we used the advanced version of C4.5 which we implemented with grouping unimportant feature values to branches with label *Other*. The resulting tree is shown in Figure 4.2.6.

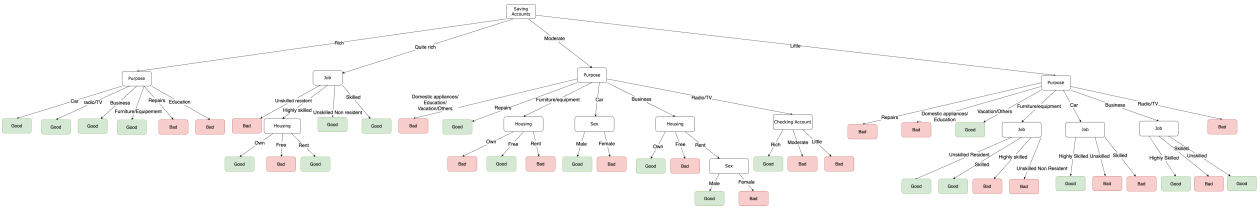

Figure 4.2.5: German Credit C4.5 model

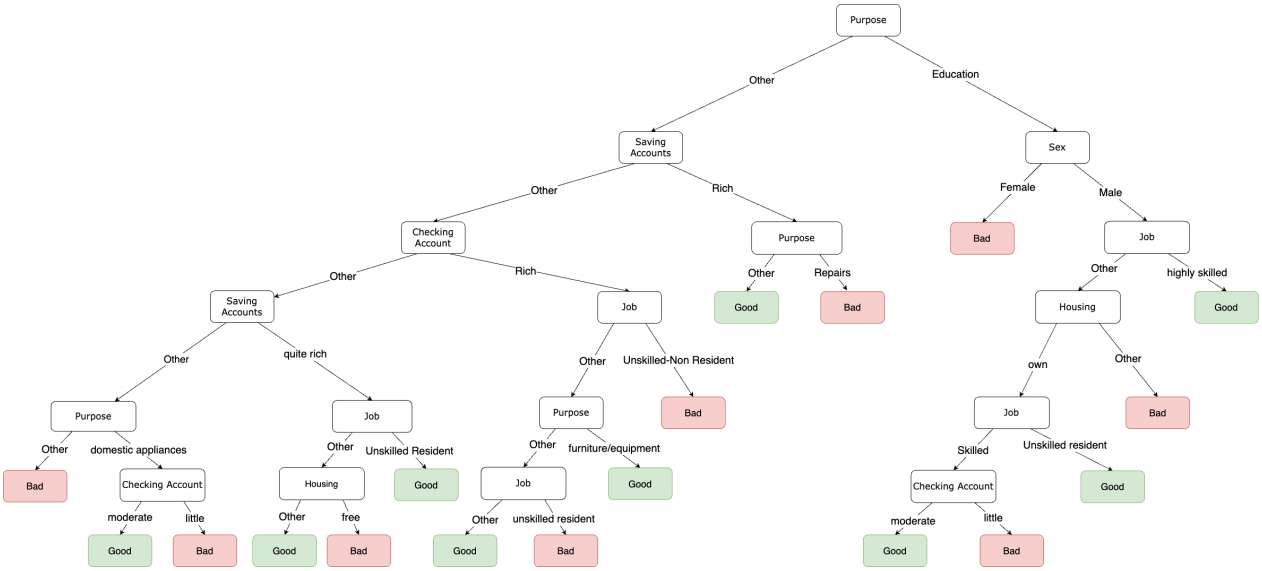

Figure 4.2.6: German Credit C4.5 Advanced model

Differences Between the Trees

As it is obvious from the figures of the trees, the first model is a wide decision tree using multiway splits based on categorical features. The second model is a deep decision tree and uses grouping for attribute values with the label *Other*. We want to evaluate if multiway splits enhances the intepretability of the tree and if the grouped values are confusing for the users.

4.2.4 Framingham Heart Study

The Framingham Heart Study dataset [11] derives from a pioneering longitudinal study initiated in 1948 in Framingham, Massachusetts, with the primary goal of identifying common factors that contribute to cardiovascular disease. Originally enrolling over 5,000 participants, the study has significantly expanded to include successive generations, allowing researchers to explore genetic, environmental, and lifestyle factors influencing heart health. The dataset includes comprehensive variables such as blood pressure, cholesterol levels, smoking habits, and body mass index.

Objective The primary objective of analyzing the Framingham Heart Study dataset is to classify individuals into two categories based on their 10-year risk of developing coronary heart disease (CHD):

- **Yes**: Individuals who are at risk of developing CHD within 10 years.
- **No**: Individuals who are not at risk of developing CHD within 10 years.

This prediction helps in assessing the risk and taking preventive measures to reduce the likelihood of developing CHD.

Pre-processing

The original dataset contained 4240 patients with 14 features describing each patient.

- **Male**: Gender of the participant (1 for male, 0 for female).
- **Age**: Age of the participant in years.
- **Education**: Educational attainment level of the participant. Typically coded numerically (e.g., 1 = Some High School, 2 = High School Graduate, 3 = Some College, 4 = College Graduate).
- **CurrentSmoker**: Indicator of whether the participant is a current smoker (1 for yes, 0 for no).
- **cigsPerDay**: Number of cigarettes smoked per day by the participant.
- **BPMeds**: Indicator of whether the participant is on blood pressure medication (1 for yes, 0 for no).
- **prevalentStroke**: Indicator of whether the participant has had a stroke (1 for yes, 0 for no).
- **prevalentHyp**: Indicator of whether the participant has hypertension (1 for yes, 0 for no).
- **diabetes**: Indicator of whether the participant has diabetes (1 for yes, 0 for no).
- **totChol**: Total cholesterol level in mg/dL.
- **sysBP**: Systolic blood pressure measurement in mmHg.
- **diaBP**: Diastolic blood pressure measurement in mmHg.
- **BMI**: Body Mass Index calculated from height and weight (kg/m^2).
- **heartRate**: Heart rate in beats per minute.
- **glucose**: Blood glucose level.

After dropping the unnecessary columns, pre-processing steps for this dataset involved:

- **Handling Missing Values**: We dropped variables with null values to ensure data integrity.
- **Feature Importance Measurement**: To assess the significance of different features within our dataset, we utilized the Random Forest algorithm, a robust ensemble learning method. Specifically, we employed a Random Forest classifier with 100 estimators (trees). The Random Forest classifier was implemented using the RandomForestClassifier from the scikit-learn library [58]. Feature importances were derived from the trained model, which offers an aggregated view of feature significance based on the reduction of impurity (Gini impurity or entropy) across all trees in the ensemble. The resulting importance scores were visualized in Figure 4.2.7. We kept only the top 10 features based on their importance to simplify the resulting trees.

Decision Trees Created

For the Framingham Heart Study dataset, we created two decision tree models using different algorithms to compare their interpretability:

Model 1: Created using the DecisionTreeClassifier from scikit-learn python library [58] which uses an optimized version of the CART algorithm. The model with the highest accuracy was selected. The resulting tree structure involved binary splits typical of the CART algorithm, and depth $d = 9$. The resulting tree is shown in Figure 4.2.8.

Model 2: For this model an extra step of data pre-processing was needed, domain knowledge induction. As we discussed in chapter 3.1.3, using domain knowledge can help in transforming numeric variables to categorical making the resulting tree more interpretable to humans. The categorizations are displayed in tables 4.1-4.8.

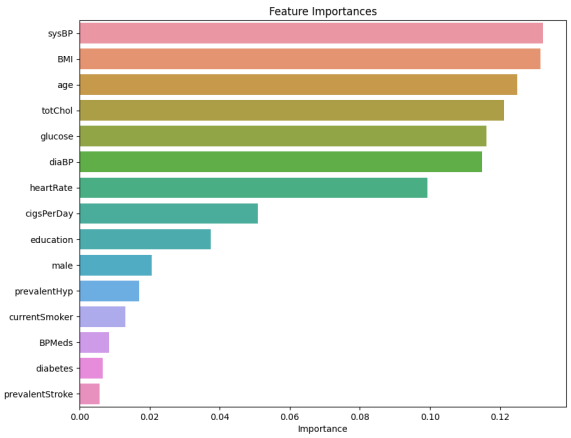

Figure 4.2.7: Framingham Heart Study Features Importance

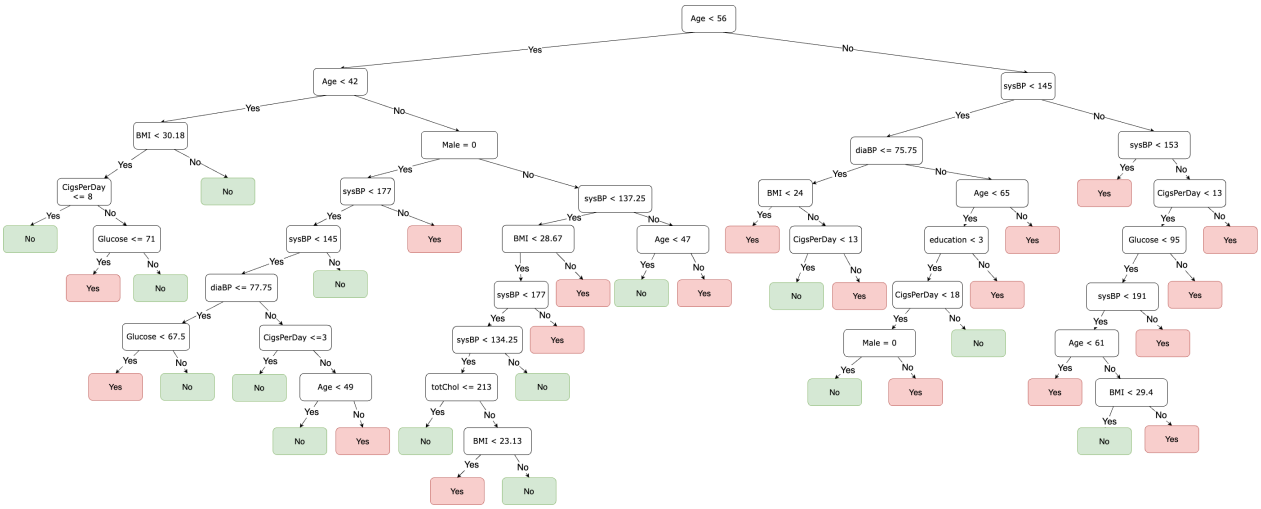

Figure 4.2.8: Framingham CART model

Range	Category
0 - 119	Normal
120 - 129	Elevated
130 - 139	Hypertension Stage 1
140 - 180	Hypertension Stage 2
> 180	Hypertensive Crisis

Table 4.1: Systolic Blood Pressure Categorization

Range	Category
0 - 18.4	Underweight
18.5 - 24.9	Normal weight
25 - 29.9	Overweight
30 - 34.9	Obesity Class I
35 - 39.9	Obesity Class II
> 40	Obesity Class III

Table 4.2: Body Mass Index (BMI) Categorization

Range	Category
0 - 79	Normal
80 - 89	Elevated
90 - 99	Hypertension Stage 1
100 - 120	Hypertension Stage 2
> 120	Hypertensive Crisis

Table 4.3: Diastolic Blood Pressure Categorization

Range	Category
0 - 199	Desirable
200 - 239	Borderline High
> 240	High

Table 4.4: Total Cholesterol Categorization

Range	Category
0 - 139	Normal
140 - 199	Prediabetes
> 200	Diabetes

Table 4.5: Glucose Categorization

Range	Category
0 - 59	Bradycardia
60 - 100	Normal
> 100	Tachycardia

Table 4.6: Heart Rate Categorization

Range	Category
0	Non smoker
1 - 10	Light smoker
11 - 20	Moderate smoker
> 20	Heavy smoker

Table 4.7: Cigarettes per Day Categorization

Range	Category
0 - 12	Child
13 - 19	Teenager
20 - 35	Young Adult
36 - 54	Adult
55 - 65	Mature Adult
> 65	Senior

Table 4.8: Age Categorization

After the incorporation of domain knowledge in the dataset we used FlowOCT to extract the resulting decision tree. To use FlowOCT we first needed to binarize the data using the embedded binarize function of the odtlearn package. The resulting tree is shown below in Figure 4.2.9.

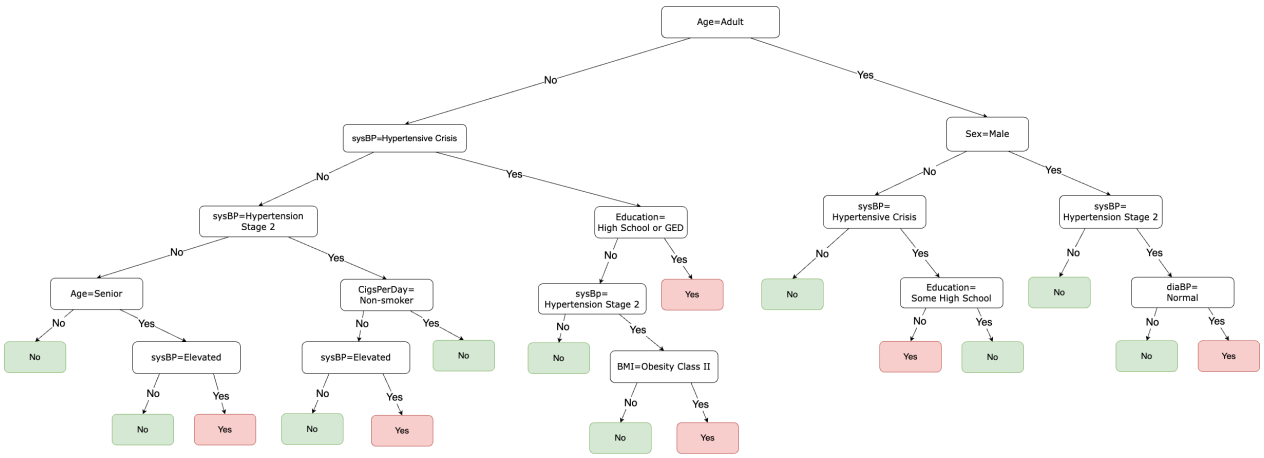

Figure 4.2.9: Framingham Categorical model

Differences Between the Trees

As it is obvious from the figures of the trees, the first model is a deep decision tree which uses continuous features for the classification. The second model is a smaller decision tree with categorical features We want to evaluate if categorical features enhance the intepretability of the tree.

4.2.5 Adult Income

The Adult income dataset [7] contains demographic (e.g., age, race, and gender), education (degree),employment (occupation, hours-per week), personal (marital status, relationship), and financial (capital gain/loss) features for 45,222 individuals. The task is to predict whether an individual's income exceeds $50K per year vs. not. It contains 14 attributes.

Objective

The primary objective of analyzing the Adult Income dataset is to classify individuals into two categories based on their annual income:

- <=**50K**: Individuals whose annual income is $50,000 or less.
- >**50K**: Individuals whose annual income exceeds $50,000.

This prediction helps in understanding the factors that contribute to higher income and addressing socio-economic disparities.

Pre-processing

The original dataset contained 47621 records of individuals with 14 features describing each individual.

- **Age**: The age of the individual (numeric).
- **Workclass**: The type of employment (e.g., Private, Self-emp-not-inc, etc.).
- **Education**: The highest level of education attained (e.g., Bachelors, Some-college, etc.).
- **Education-num**: The number of years of education (numeric).
- **Marital-status**: The marital status of the individual (e.g., Married-civ-spouse, Divorced, etc.).
- **Occupation**: The type of job (e.g., Tech-support, Craft-repair, etc.).
- **Relationship**: Relationship to the household (e.g., Own-child, Married, etc.).
- **Race**: The race of the individual (e.g., White, Asian-Pac-Islander, etc.).
- **Sex**: The sex of the individual (e.g., Male, Female).
- **Capital-gain**: Capital gains (numeric).
- **Capital-loss**: Capital losses (numeric).
- **Hours-per-week**: Hours worked per week (numeric).
- **Native-country**: Country of origin (e.g., United-States, Cambodia, etc.).

Pre-processing steps for this dataset involved:

- **Handling Missing Values**: We dropped variables with null values to ensure data integrity.
- **Balance Dataset**: The dataset often exhibits an imbalance, with significantly more instances of individuals earning less than $50,000 compared to those earning more. This imbalance can lead to models that are biased towards the majority class, resulting in poor performance on the minority class. Balancing the dataset, by undersampling the majority class, improves the model's accuracy, sensitivity, and specificity. This approach ensures that the model learns to recognize patterns in both classes, leading to fairer predictions and better generalization to new data. Moreover, balanced datasets enhance the interpretability of the model, making it easier to derive meaningful insights from the results.
- **Feature Importance Measurement**: To assess the significance of different features within our dataset, we utilized the Random Forest algorithm, a robust ensemble learning method. Specifically, we employed a Random Forest classifier with 100 estimators (trees). The choice of Random Forest for this task was motivated by its ability to handle high-dimensional data and provide insights into feature importance through its construction of multiple decision trees. The Random Forest classifier was implemented using the RandomForestClassifier from the scikit-learn library [58]. Feature importances

were derived from the trained model, which offers an aggregated view of feature significance based on the reduction of impurity (Gini impurity or entropy) across all trees in the ensemble. The resulting importance scores were visualized in Figure 4.2.10.

- **Encoding Categorical Features**: We used one-hot encoding for categorical features such as workclass, education, marital-status, etc to convert them into a format suitable for machine learning algorithms.

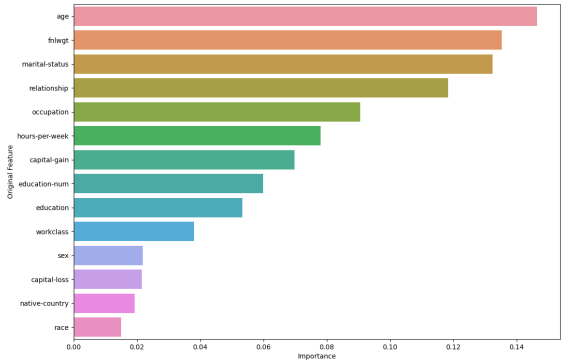

Figure 4.2.10: Adult Income Features Importance

Decision Trees Created

For the Adult Income dataset, we created two decision tree models using different algorithms to compare their interpretability:

Model 1: Created using the DecisionTreeClassifier from scikit-learn python library [58] which uses an optimized version of the CART algorithm. The model with the highest accuracy was selected. The resulting tree structure involved binary splits typical of the CART algorithm, and depth $d = 6$

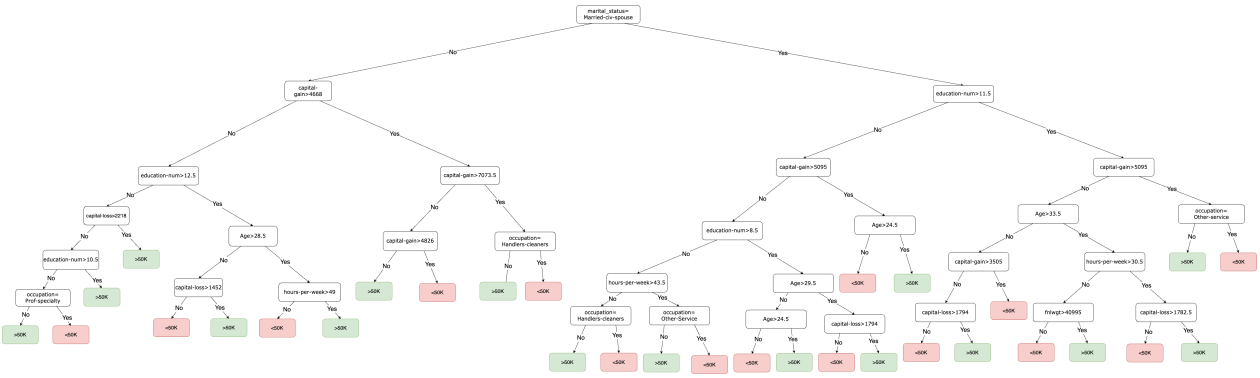

Figure 4.2.11: Adult Income CART wide model

Model 2: For this model we also used CART but we chose a more deep tree with depth $d = 8$.

Differences Between the Trees

As it is obvious from the figures of the trees, the first model is a wide decision tree and the second model is deeper. We want to evaluate if wide trees are more interpretable than deep trees.

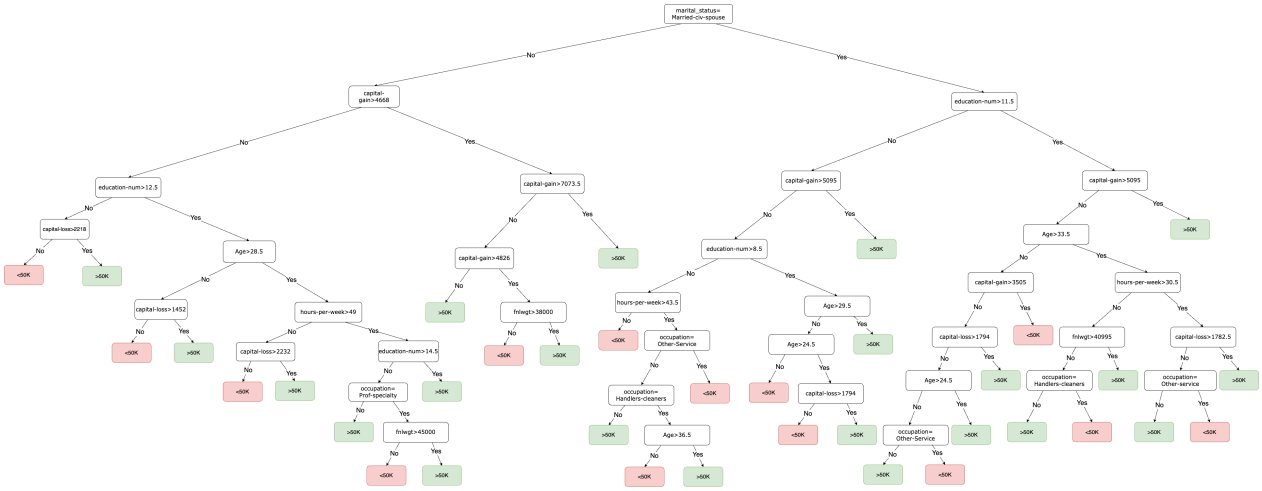

Figure 4.2.12: Adult Income CART deep model

4.3 Results

4.3.1 General Results

As we can see in figures 4.3.1, 4.3.2, 4.3.3, 4.3.4 we have high accuracy as we expected. The COMPAS dataset has the lowest accuracy as it was the first dataset that users classified.

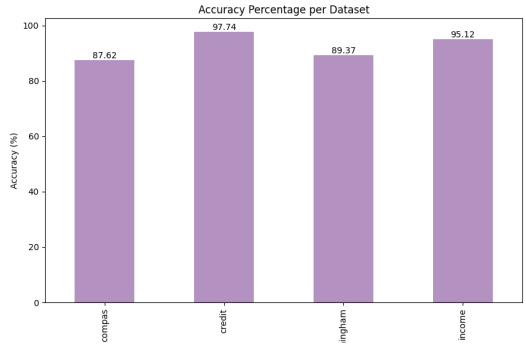

Figure 4.3.1: Accuracy per Dataset

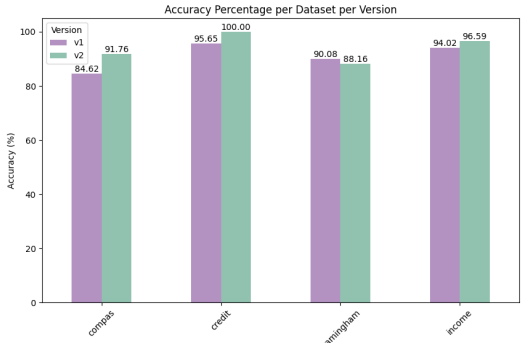

Figure 4.3.2: Accuracy per Dataset per Version

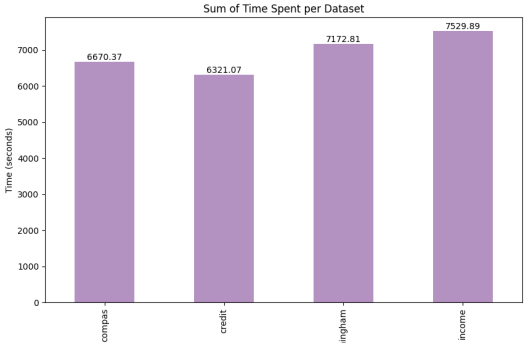

Figure 4.3.3: Time needed per Dataset

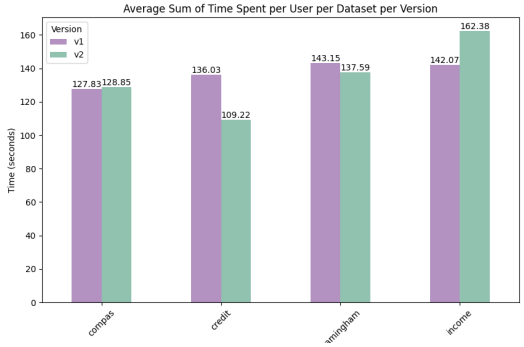

Figure 4.3.4: Time needed per Dataset per Version

In figures 4.3.5, 4.3.6, 4.3.7 we display the answer confidence, path clarity and the simplicity of tree structure.

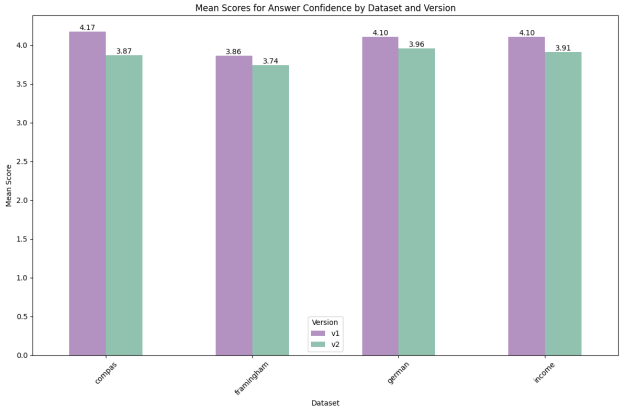

Figure 4.3.5: Answer confidence per Dataset per Version

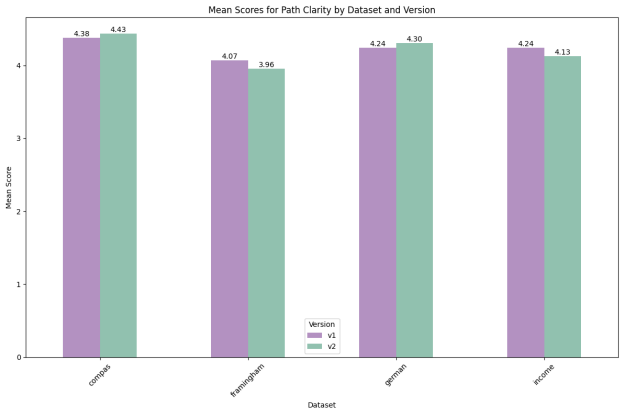

Figure 4.3.6: Path Clarity per Dataset per Version

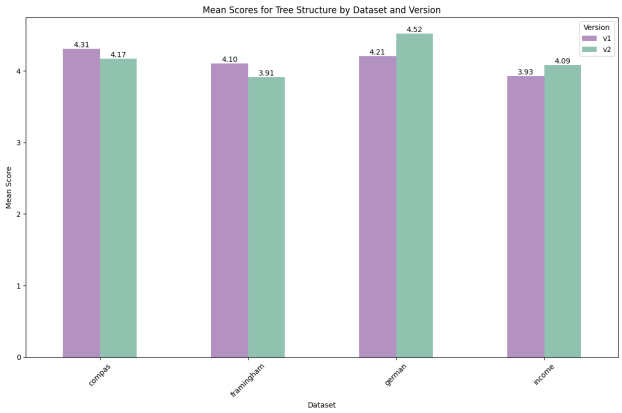

Figure 4.3.7: Simplicity of Tree Structure per Dataset per Version

We will analyse the results per dataset and extract conclusions by comparing the decision trees and the results below.

4.3.2 COMPAS

In this section, we compare the decision trees created based on the COMPAS dataset. The first decision tree employs a single numeric threshold for splits at decision nodes, while the second utilizes binned features for splitting nodes. Binned features offer a reduction in tree size as they require fewer decision nodes to describe a condition. Based on the results shown in Figures 4.3.8, 4.3.9 and 4.3.10, we observe that the FlowOCT decision tree results in an increase in accuracy, albeit with a slight increase in the time needed for classification.

Furthermore, the data reveals that users exhibit lower answer confidence in the second model, despite their answers being correct. The path clarity is better in the second model due to shorter paths; however, the tree structure is more complex because the binned features are not intuitive for users.

From the user descriptions of the trees, it is evident that users focused on the apparent bias against African-Americans in the dataset. The binned model is perceived as more complicated by users, whereas the numeric model suffers from the drawback of feature repetition.

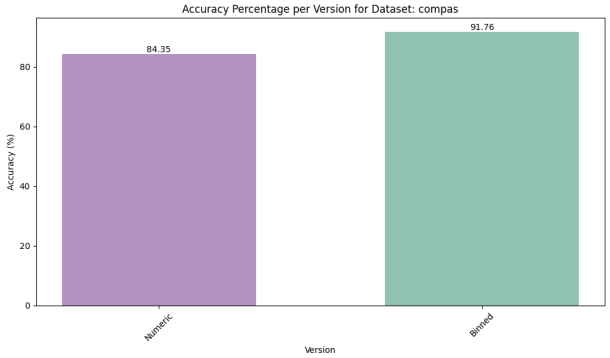

Figure 4.3.8: COMPAS Accuracy

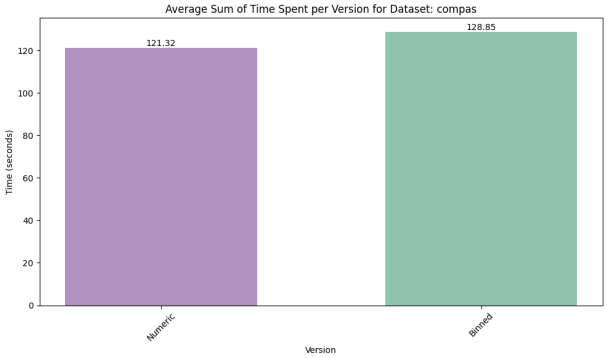

Figure 4.3.9: COMPAS Time needed

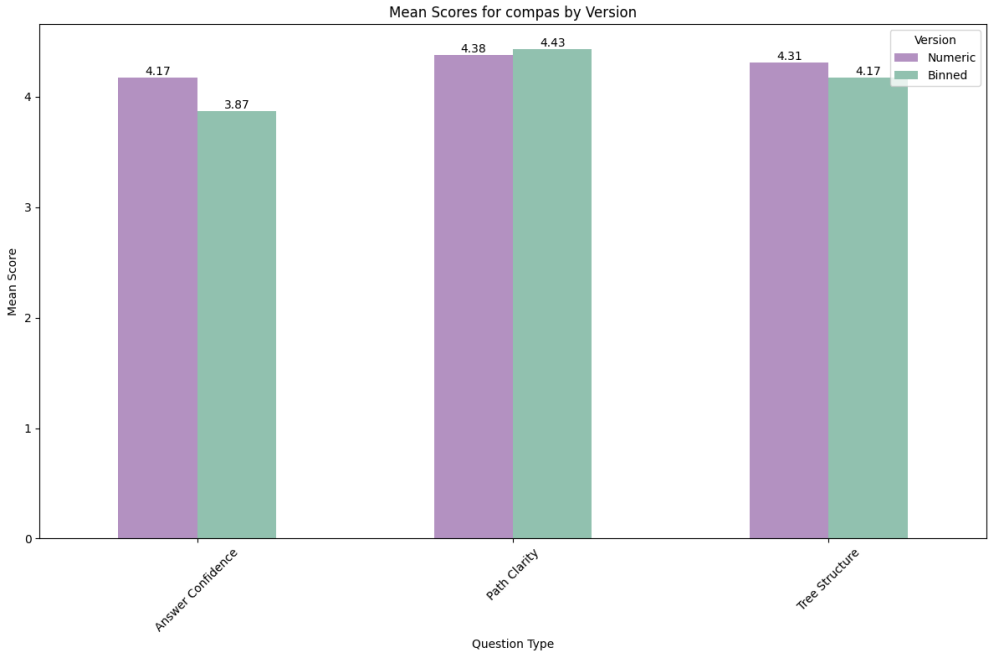

Figure 4.3.10: COMPAS Evaluation Metrics: Answer Confidence, Decision Path Clarity, and Tree Structure

4.3.3 German Credit

In this section, we compare the decision trees created based on the German Credit dataset. The first decision tree uses binary splits at decision nodes, while the second utilizes multiway splits for decision nodes. Multiway splits offer a reduction in tree size as they require fewer decision nodes to describe a condition. Based on the results shown in Figures 4.3.11, 4.3.12 and 4.3.13, we observe that the multiway decision tree results in a perfect accuracy score with a significant time reduction for classification.

Furthermore, the data reveals that users exhibit lower answer confidence in the second model, despite the high accuracy of their answers. This happens due to the large branching factor (3.8 instead of 2 for binary model) of each node. The path clarity and the simplicity is better in the second model due to shorter paths and wider tree structure.

From the user descriptions of the trees, it is evident that users focused on the apparent bias against females in the dataset. The binary tree, utilizing the term "Other," enabled users to draw some conclusions about the decision-making process. While the multiway split is considered more intuitive compared to the binned feature in the COMPAS dataset, it is also perceived as more complex due to the large branching factor.

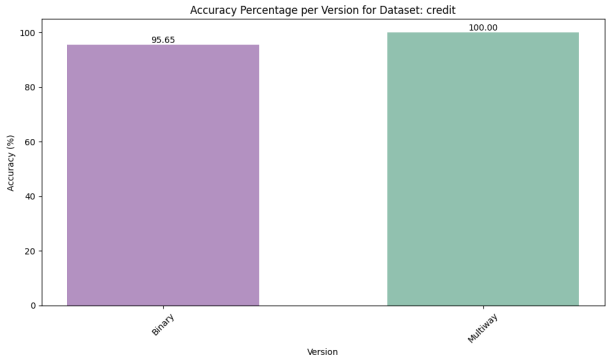

Figure 4.3.11: German Credit Accuracy

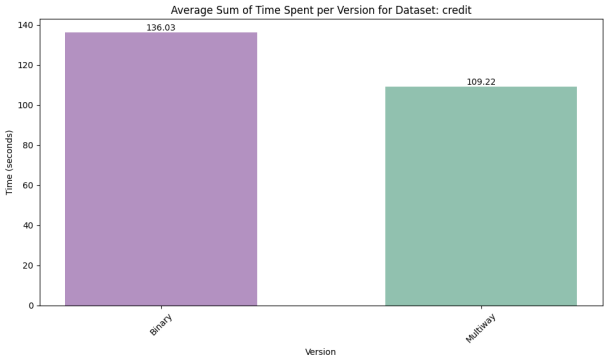

Figure 4.3.12: German Credit Time needed

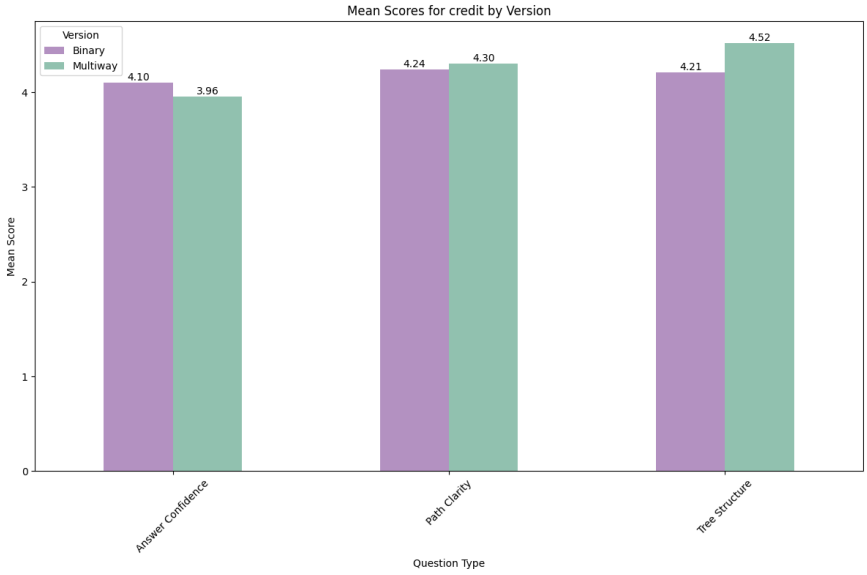

Figure 4.3.13: German Credit Evaluation Metrics: Answer Confidence, Decision Path Clarity, and Tree Structure

4.3.4 Framingham Heart Study

In this section, we compare the decision trees created based on the Framingham Heart Study dataset. The first decision tree uses categorical features with binary splits at decision nodes, while the second employs numeric features with a single numeric threshold for binary splits at decision nodes. Categorical splits offer a reduction in tree size, requiring fewer decision nodes to describe a condition, and significantly enhance model comprehension. Based on the results shown in Figures 4.3.14, 4.3.15 and 4.3.16, we observe that the categorical decision tree achieves a high accuracy score with a slight increase in classification time.

Furthermore, the data reveals that categorical features result in higher scores across all metrics: answer confidence, path clarity, and simplicity of tree structure.

From the user descriptions of the trees, it is evident that users had difficulty understanding medical terms, as they are not field experts. The numeric tree is perceived as more complex due to its depth and repetitiveness. While both trees are considered challenging to comprehend, the categorical tree appears to be clearer.

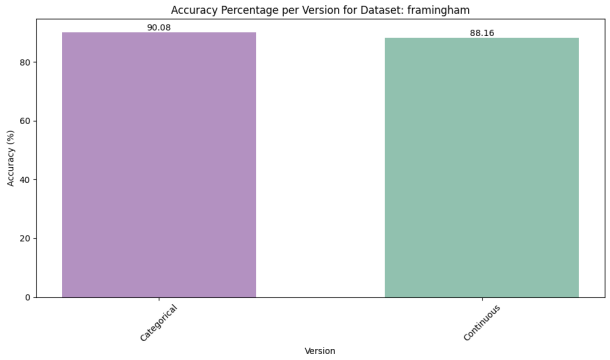

Figure 4.3.14: Framingham Heart Study Accuracy

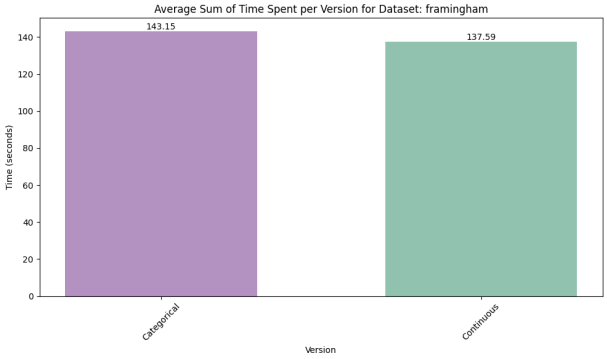

Figure 4.3.15: Framingham Heart Study Time needed

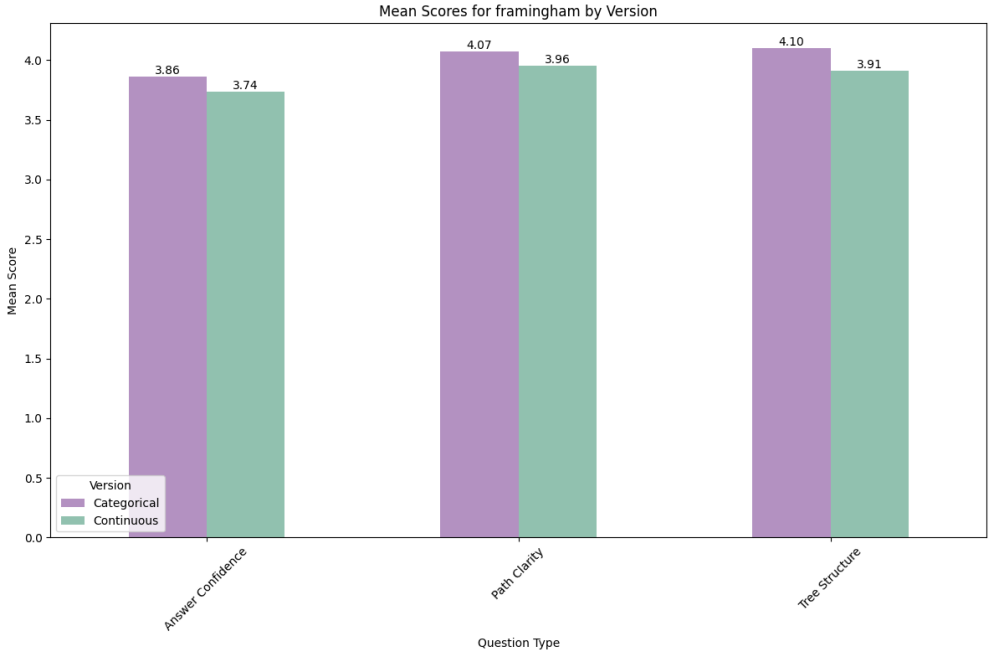

Figure 4.3.16: Framingham Evaluation Metrics: Answer Confidence, Decision Path Clarity, and Tree Structure

4.3.5 Adult Income

In this section, we compare the decision trees created based on the Adult Income dataset. The first decision tree is deeper versus the second one which is swallower and wider. Deep trees often result to feature repetition that are a bit tiring and difficult to comprehend by humans. Based on the results shown in Figures 4.3.17, 4.3.18 and 4.3.19, we observe that the wider decision tree results in an increase in accuracy, albeit with a slight increase in the time needed for classification.

Furthermore, the data reveals that users exhibit lower answer confidence in the wide model, despite the high accuracy of their answers. The differences in path clarity and simplicity of structure are minimal and could be attributed to statistical error.

From the user descriptions of the trees, it is evident that the size of the tree was noted by the users. Many users who used the deep tree commented on its size and how it made the tree difficult to comprehend. They compared the size with the previous models and we can see that the limit of depth 7 is a true limit for human cognition.

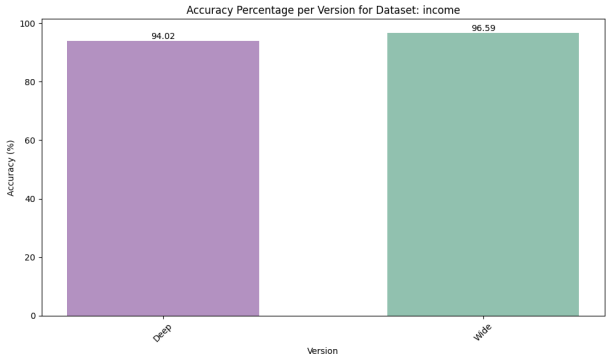

Figure 4.3.17: Adult Income Accuracy

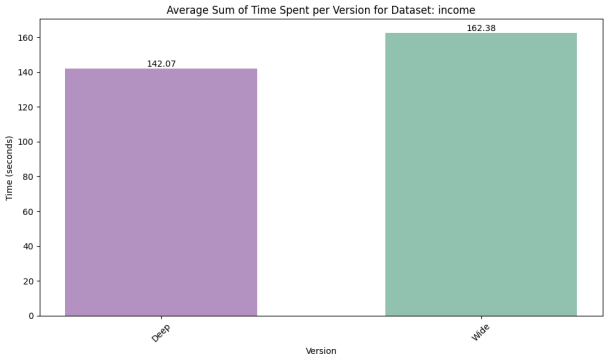

Figure 4.3.18: Adult Income Time needed

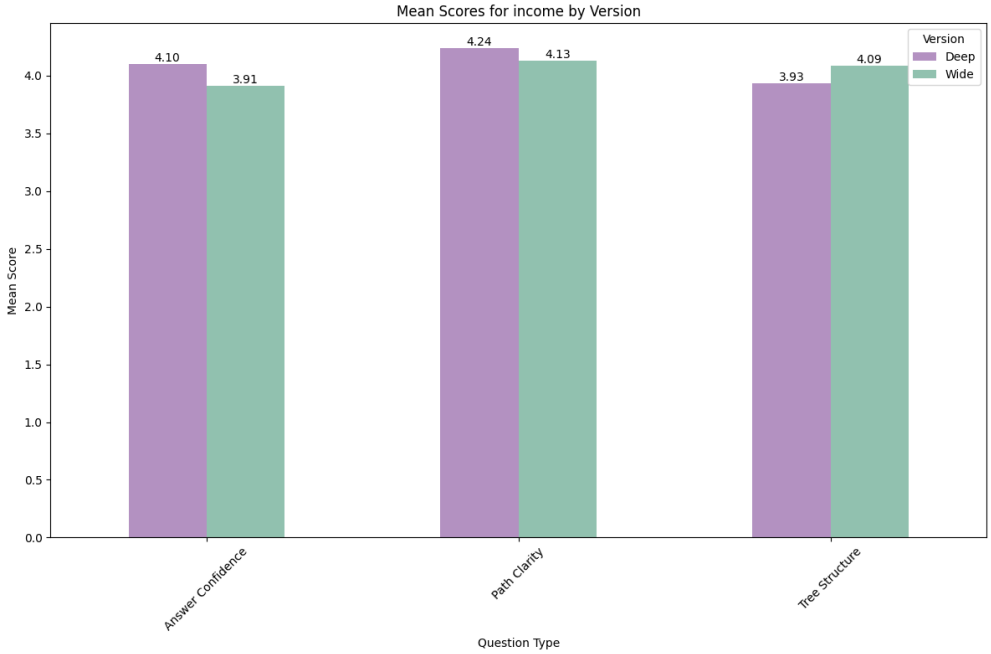

Figure 4.3.19: Adult Income Evaluation Metrics: Answer Confidence, Decision Path Clarity, and Tree Structure

4.4 Conclusions

4.4.1 Evaluation of Tree structure

Based on the results of our user study, we observed notable differences in the performance and interpretability of decision trees across various datasets and tree structures. The study revealed that decision trees utilizing categorical splits generally achieved higher accuracy and better interpretability metrics, such as answer confidence, path clarity, and tree structure simplicity. This was evident in the Framingham Heart Study dataset, where the categorical decision tree outperformed the numeric one in user ratings despite the slight increase in classification time. Also the incorporation of domain knowledge seemed to increase the comprehensibility of the model. We also saw that the grouping of the least important feature values helped users extract general conclusions for the model.

Conversely, decision trees with numeric splits, particularly those that are deeper, often resulted in feature repetition and increased complexity, making them more difficult for users to comprehend and lowering the user satisfaction. This was highlighted in the Adult Income dataset, where the deeper tree was noted for its size and complexity, impacting user confidence despite its accuracy. A tree with depth close to 7 seemed extremely complicated to the users. Additionally, the COMPAS dataset illustrated the challenges users face with binned features.

Overall, the study underscores the importance of considering both accuracy and interpretability when designing decision trees. While categorical splits can enhance user understanding and confidence, careful attention must be paid to the trade-offs between tree complexity and comprehensibility to ensure effective and user-friendly decision-making tools.

4.4.2 Evaluation of Interpretability Metrics

Based on the results of our study, we evaluated the interpretability of the decision trees using several key metrics: model size, number of features used, sparsity of nodes, leaf purity, tree balance, and rule length. Our findings indicate that while smaller model size and shallower tree depth generally contributed to better user comprehension, this was not always the case. In terms of the number of features used, models that incorporated fewer, more relevant features were easier for users to understand, as seen in the Framingham Heart Study dataset. Tree balance played a significant role in interpretability, with well-balanced trees from the COMPAS dataset being easier for users to navigate and understand compared to more imbalanced counterparts. Finally, shorter rule length, as observed in the categorical splits of the Framingham Heart Study dataset, facilitated easier comprehension, reinforcing the importance of concise and straightforward decision rules. Overall, our study underscores that a combination of these metrics, rather than any single measure, is essential for enhancing the interpretability of decision tree models.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we investigated the interpretability of decision trees, highlighting the importance of interpretability in modern AI systems and the challenges associated with achieving it. We discussed well-known interpretable models and demonstrated why decision trees are particularly valuable for interpretable decision-making. Additionally, we explored human decision-making processes and illustrated how decision trees align with these processes but also researched ways to increase their interpretability based on human cognition.

We discussed interpretability measures from both a human perspective, including transparency and user satisfaction, and a technical perspective, such as tree balance and rule length. To enhance interpretability, we proposed several preprocessing techniques, including feature selection, the incorporation of domain knowledge, and the supervised discretization of numeric data.

Furthermore, we advocated for the use of the modern FlowOCT model for classification, as it generates balanced trees with high accuracy. We also suggested a modification of the C4.5 algorithm to group feature values into categories, thereby minimizing the branching factor of the tree and implement multiway splits for numeric features to reduce the depth of the trees.

To evaluate our proposals, we conducted a user study involving classifications from four different datasets, comparing two versions of decision trees to assess the impact of our suggested improvements on interpretability. The results of this study were analyzed, leading to our final conclusions.

5.2 Discussion

Our study investigated the interpretability of decision trees across various datasets, revealing significant differences in performance and user comprehension based on tree structures and splitting criteria. Decision trees utilizing categorical splits generally achieved higher accuracy and better interpretability metrics, such as answer confidence, path clarity, and simplicity, as evidenced in the Framingham Heart Study dataset. Conversely, trees with numeric splits, especially deeper ones, often resulted in feature repetition and increased complexity, making them harder for users to understand, as seen in the Adult Income dataset. The COMPAS dataset highlighted challenges with binned features, which, despite reducing tree size, complicated decision-making highlighting the importance of intuitive features.

Evaluating interpretability using metrics like model size, number of features, sparsity of nodes, leaf purity, tree balance, and rule length, we found that smaller model size and shallower depth generally improved comprehension. Models with fewer, relevant features were easier to understand, while sparsity in nodes and lower leaf purity negatively impacted clarity. Well-balanced trees, like those from the German Credit dataset, were easier to navigate, and shorter rule lengths facilitated comprehension, emphasizing the importance of concise decision rules.

5.3 Future Work

For future work, several avenues can be explored to further enhance the interpretability and usability of decision tree models. Firstly, the integration of domain-specific knowledge into the decision tree construction process could significantly improve the relevance and comprehensibility of the features used, potentially leading to more intuitive models that align closely with expert understanding. Secondly, developing more advanced visualization tools that allow users to interact with decision tree models—such as dynamic zooming and real-time feedback—could greatly enhance user engagement and understanding, making complex models more accessible. Generally a further research in post-processing techniques will be really useful. Also we would like to further evaluate the C4.5-Advanced model we suggested on more datasets and more benchmarks. Finally, a comprehensive comparison of decision tree models with other explainable AI (XAI) methods, including rule-based systems and neural network interpretability techniques, could provide deeper insights into the relative strengths and limitations of each approach, guiding the development of more effective and user-friendly AI decision support systems.

Chapter 6

Bibliography

[1] Aghaei, S., Azizi, M. J., and Vayanos, P. "Learning optimal and fair decision trees for non-discriminative decision-making". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 1418–1426.

[2] Aghaei, S., Gómez, A., and Vayanos, P. "Strong optimal classification trees". In: *arXiv preprint arXiv:2103.15965* (2021).

[3] AHMED, N. A. and Alpkoçak, A. "A quantitative evaluation of explainable AI methods using the depth of decision tree". In: *Turkish Journal of Electrical Engineering and Computer Sciences* 30.6 (2022), pp. 2054–2072.

[4] Aıvodji, U. et al. "Learning fair rule lists". In: *arXiv preprint arXiv:1909.03977* (2019).

[5] Allahyari, H. and Lavesson, N. "User-oriented assessment of classification model understandability". In: *11th scandinavian conference on Artificial intelligence*. IOS Press. 2011.

[6] Azuma, R., Daily, M., and Furmanski, C. "A review of time critical decision making models and human cognitive processes". In: (2006).

[7] Becker, B. and Kohavi, R. *Adult*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20. 1996.

[8] Bellazzi, R. and Zupan, B. "Predictive data mining in clinical medicine: current issues and guidelines". In: *International journal of medical informatics* 77.2 (2008), pp. 81–97.

[9] Bertsimas, D. and Dunn, J. "Optimal classification trees". In: *Machine Learning* 106 (2017), pp. 1039–1082.

[10] Bertsimas, D. and Dunn, J. *Machine learning under a modern optimization lens*. Dynamic Ideas LLC Charlestown, MA, 2019.

[11] Bhardwaj, A. *Framingham heart study dataset*. url: https://www.kaggle.com/dsv/3493583. 2022. DOI: 10.34740/KAGGLE/DSV/3493583.

[12] Bramer, M. *Principles of Data Mining*. Jan. 2007. ISBN: 978-1-84628-765-7. DOI: 10.1007/978-1-84628-766-4.

[13] Breiman, L. et al. *Classification and Regression Trees*. Taylor & Francis, 1984. ISBN: 9780412048418. URL:

[14] Breiman, L. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[15] Burkart, N. and Huber, M. F. "A survey on the explainability of supervised machine learning". In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317.

[16] Cendrowska, J. "PRISM: An algorithm for inducing modular rules". In: *International Journal of Man-Machine Studies* 27.4 (1987), pp. 349–370.

[17] Childs, C. M. and Washburn, N. R. "Embedding domain knowledge for machine learning of complex material systems". In: *MRS Communications* 9.3 (2019), pp. 806–820.

[18] Confalonieri, R. et al. "Using ontologies to enhance human understandability of global post-hoc explanations of black-box models". In: *Artificial Intelligence* 296 (2021), p. 103471.

[19] Cowan, N. "The magical mystery four: How is working memory capacity limited, and why?" In: *Current directions in psychological science* (2010).

[20] Cowan, N. *Working memory capacity*. Psychology press, 2012.

[21] Dhar, V., Chou, D., and Provost, F. "Discovering Interesting Patterns for Investment Decision Making with GLOWER—A Genetic Learner Overlaid with Entropy Reduction". In: *Data Mining and Knowledge Discovery* 4 (2000), pp. 251–280.
[22] DiSessa, A. A. "Toward an epistemology of physics". In: *Cognition and instruction* 10.2-3 (1993), pp. 105–225.
[23] Doshi-Velez, F. and Kim, B. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017).
[24] Dunn, J., Mingardi, L., and Zhuo, Y. D. "Comparing interpretability and explainability for feature selection". In: *arXiv preprint arXiv:2105.05328* (2021).
[25] Fayyad, U. M. and Irani, K. B. "Multi-interval discretization of continuous-valued attributes for classification learning". In: *Ijcai*. Vol. 93. 2. Citeseer. 1993, pp. 1022–1029.
[26] Filandrianos, G. et al. "Counterfactuals of Counterfactuals: a back-translation-inspired approach to analyse counterfactual editors". In: *arXiv preprint arXiv:2305.17055* (2023).
[27] Frank, E. and Witten, I. H. "Selecting multiway splits in decision trees". In: (1996).
[28] Freitas, A. A. *Data mining and knowledge discovery with evolutionary algorithms*. Springer Science & Business Media, 2002.
[29] Freitas, A. A. "Comprehensible classification models: a position paper". In: *ACM SIGKDD explorations newsletter* (2014).
[30] Fürnkranz, J. "Separate-and-conquer rule learning". In: *Artificial Intelligence Review* 13 (1999), pp. 3–54.
[31] Goodman, B. and Flaxman, S. "European Union Regulations on Algorithmic Decision Making and a "Right to Explanation"". In: *AI Magazine* 38.3 (2017), pp. 50–57. ISSN: 2371-9621. DOI: 10.1609/aimag.v38i3.2741.
[32] Guidotti, R. et al. "A Survey Of Methods For Explaining Black Box Models". In: *IEEE* (2018).
[33] Günlük, O. et al. "Optimal decision trees for categorical data via integer programming". In: *Journal of global optimization* 81 (2021), pp. 233–260.
[34] Guyon, I. et al. "Gene selection for cancer classification using support vector machines". In: *Machine learning* 46 (2002), pp. 389–422.
[35] Hagras, H. "Toward human-understandable, explainable AI". In: *Computer* 51.9 (2018), pp. 28–36.
[36] Halford, G. S., Cowan, N., and Andrews, G. "Separating cognitive capacity from knowledge: A new hypothesis". In: *Trends in cognitive sciences* (2007).
[37] Hoffman, R. R. et al. "Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance". In: *Frontiers in Computer Science* 5 (2023), p. 1096257.
[38] Hofmann, H. *Statlog (German Credit Data)*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77. 1994.
[39] Huysmans, J., Baesens, B., and Vanthienen, J. "Using rule extraction to improve the comprehensibility of predictive models". In: (2006).
[40] Huysmans, J. et al. "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models". In: *Decision Support Systems* 51.1 (2011), pp. 141–154.
[41] Jarvenpaa, S. L., Dickson, G. W., and DeSanctis, G. "Methodological issues in experimental IS research: experiences and recommendations". In: *MIS quarterly* (1985), pp. 141–156.
[42] Kim, H. and Loh, W.-Y. "Classification trees with unbiased multiway splits". In: *Journal of the American Statistical Association* 96.454 (2001), pp. 589–604.
[43] Kononenko, I. "On biases in estimating multi-valued attributes". In: *Ijcai*. Vol. 95. Citeseer. 1995, pp. 1034–1040.
[44] Kuhn, M., Johnson, K., et al. *Applied predictive modeling*. Vol. 26. Springer, 2013.
[45] Lage, I. et al. "Human evaluation of models built for interpretability". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 2019, pp. 59–67.
[46] Larson, J. et al. "How We Analyzed the Compas Recidivism Algorithm." In: *Edited by ProPublica.org.* (2016). url: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.
[47] Laurent, H. and Rivest, R. L. "Constructing optimal binary decision trees is NP-complete". In: *Information processing letters* 5.1 (1976), pp. 15–17.
[48] Lavrač, N. "Selected techniques for data mining in medicine". In: *Artificial intelligence in medicine* 16.1 (1999), pp. 3–23.

[49] Liartis, J. et al. “Semantic Queries Explaining Opaque Machine Learning Classifiers.” In: *DAO-XAI*. 2021.
[50] Liartis, J. et al. “Searching for explanations of black-box classifiers in the space of semantic queries”. In: *Semantic Web* Preprint (), pp. 1–42.
[51] Mastromichalakis, O. M. et al. “Rule-Based Explanations of Machine Learning Classifiers Using Knowledge Graphs”. In: *Proceedings of the AAAI Symposium Series*. Vol. 3. 1. 2024, pp. 193–202.
[52] MENIS-MASTROMICHALAKIS, O. “Explainable Artificial Intelligence: An STS perspective”. In: ().
[53] Menis-Mastromichalakis, O. et al. *Semantic Prototypes: Enhancing Transparency Without Black Boxes*. 2024. arXiv: 2407.15871 [cs.LG]. URL:
[54] Miller, G. A. “The magical number seven, plus or minus two: Some limits on our capacity for processing information.” In: *Psychological Review* (1956).
[55] Miller, T. “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial intelligence* 267 (2019), pp. 1–38.
[56] Mohseni, S., Zarei, N., and Ragan, E. D. “A multidisciplinary survey and framework for design and evaluation of explainable AI systems”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11.3-4 (2021), pp. 1–45.
[57] Pazzani, M. J. “Knowledge discovery from data?” In: *IEEE intelligent systems and their applications* 15.2 (2000), pp. 10–12.
[58] Pedregosa, F. et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.
[59] Pratama, N. K. *C45 Decision Tree*. Accessed: 2024-07-03. n.d.
[60] Quinlan, J. R. “Generating production rules from decision trees.” In: *ijcai*. Vol. 87. Citeseer. 1987, pp. 304–307.
[61] Quinlan, J. R. *C4. 5: programs for machine learning*. Elsevier, 2014.
[62] Quinlan, J. R. “Induction of decision trees”. In: *Machine learning* 1 (1986), pp. 81–106.
[63] Rudin, C. “Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead”. In: *Nature Machine Intelligence* (2019).
[64] Selvaraju, R. R. et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.
[65] Souza, V. F. et al. “Decision trees with short explainable rules”. In: *Advances in neural information processing systems* 35 (2022), pp. 12365–12379.
[66] Sternberg, R. J. “Component processes in analogical reasoning.” In: *Psychological review* 84.4 (1977).
[67] Subramanian, G. H. et al. “A comparison of the decision table and tree”. In: *Communications of the ACM* 35.1 (1992), pp. 89–94.
[68] Tegarden, D. P. “Business information visualization”. In: *Communications of the Association for Information Systems* 1.1 (1999), p. 4.
[69] Tufte, E. R. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT, 2001.
[70] Vanthienen, J. and Wets, G. “From decision tables to expert system shells”. In: *Data & Knowledge Engineering* 13.3 (1994), pp. 265–282.
[71] Verwer, S. and Zhang, Y. “Learning optimal classification trees using a binary linear program formulation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 1625–1632.
[72] Vessey, I. “Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature*”. In: *Decision Sciences* 22.2 (1991).
[73] Ware, C. *Information visualization: perception for design*. Morgan Kaufmann, 2019.
[74] Weston, J. et al. “Use of the zero norm with linear models and kernel methods”. In: *The Journal of Machine Learning Research* 3 (2003), pp. 1439–1461.
[75] Zadeh, L. A. “A new direction in AI: Toward a computational theory of perceptions”. In: *AI magazine* 22.1 (2001), pp. 73–73.