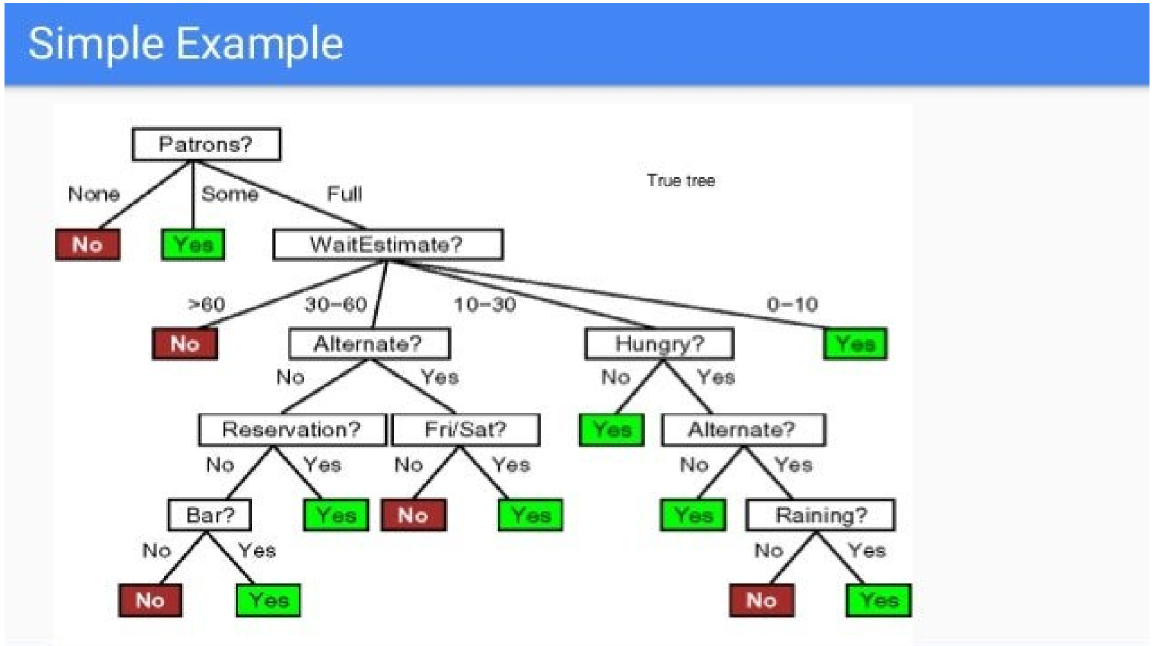
Κεφάλαιο 3



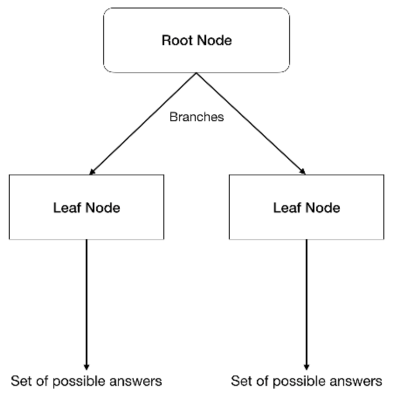
Δένδρα Απόφασης/Ταξινόμησης
Decision/Classification Trees

Εισαγωγή

- Οι αλγόριθμοι μάθησης δένδρων απόφασης (classification /decision trees) είναι από τους πιο δημοφιλείς αλγόριθμους μάθησης
  - Έχουν εφαρμοστεί αποτελεσματικά σε διάφορους τομείς, όπως διάγνωση ιατρικών περιστατικών και αξιολόγηση ρίσκου αποδοχής αίτησης για πιστωτική κάρτα.
- Είναι μια μέθοδος (ο βασικός αλγόριθμος) για την προσέγγιση συναρτήσεων στόχων, που έχουν ως έξοδο διακριτές τιμές (*ταξινόμηση*) αλλά υπάρχουν επεκτάσεις του για συνεχείς τιμές (Decision Tree – Regression)
- Η προσέγγιση (μοντέλο) που μαθαίνει το σύστημα αναπαρίσταται ως ένα δένδρο που ονομάζεται *Δένδρο Απόφασης ή Ταξινόμησης* (*Decision/Classification tree*).
  - Εναλλακτικά μπορεί να αναπαρασταθεί και ως σύνολο κανόνων if-then, για τη βελτίωση της αναγνωσιμότητας.
- Κατάλληλοι για προβλήματα όπου
  - Τα δεδομένα εκπαίδευσης περιέχουν σφάλματα
  - Κάποιες τιμές χαρακτηριστικών στα δεδομένα εκπαίδευσης λείπουν

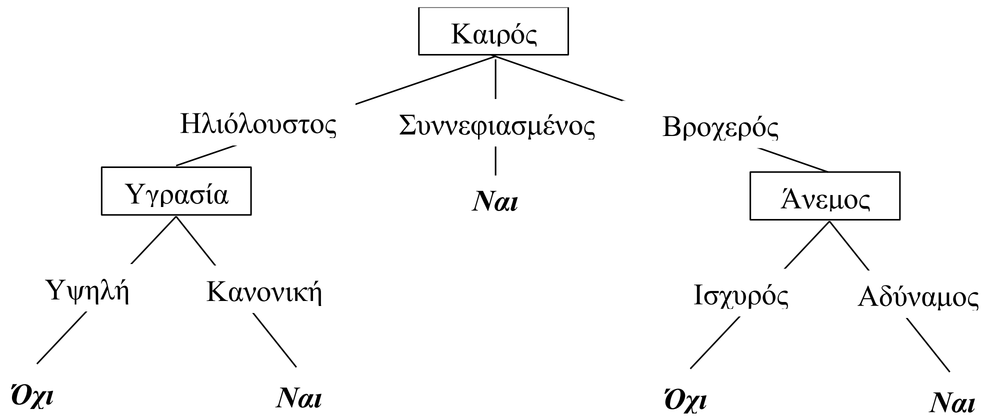
Αναπαράσταση Δένδρων Απόφασης

- Τα δένδρα ταξινόμησης χρησιμοποιούνται για να προβλέψουν, με κάποιο βαθμό ακρίβειας, την τιμή της μεταβλητής που μοντελοποιούν με βάση τις τιμές των ανεξάρτητων μεταβλητών (χαρακτηριστικών).
- Κάθε κόμβος στο δένδρο ορίζει μια συνθήκη ελέγχου της τιμής κάποιου *χαρακτηριστικού* (*attribute ή feature*) των *περιπτώσεων* (*instances*), και κάθε κλαδί που φεύγει από τον κόμβο αυτό αντιστοιχεί σε μια διαφορετική διακριτή τιμή του χαρακτηριστικού αυτού
- Ταξινόμηση άγνωστης περίπτωσης:
    - Μια (άγνωστη) *περίπτωση* ταξινομείται αρχίζοντας από τη ρίζα και ακολουθώντας τα κλαδιά του δένδρου προς κάποιο φύλλο, το οποίο περιέχει και μια διακριτή τιμή της κατηγορίας (κλάσης)
    - Σε κάθε κόμβο ελέγχεται η τιμή της περίπτωσης για το *χαρακτηριστικό* του κόμβου και ακολουθείται το αντίστοιχο κλαδί
- Ένα σημαντικό πλεονέκτημα τους είναι η ευκολία με την οποία ερμηνεύονται.



Παράδειγμα Δένδρου Απόφασης

- Δένδρο για την έννοια στόχο "καλή μέρα για τένις":
- Αντίστοιχη αναπαράσταση ως διάζευξη συζεύξεων:
  - Καλή μέρα για τένις, αν:
    (Καιρός = Ηλιόλουστος /\ Υγρασία = Κανονική) \/
    (Καιρός = Συννεφιασμένος) \/
    (Καιρός = Βροχερός /\ Άνεμος = Αδύναμος)



- Κάθε μονοπάτι από τη ρίζα προς ένα φύλλο αντιστοιχεί σε συζεύξεις (λογικό ΚΑΙ) περιορισμών στις τιμές των χαρακτηριστικών
- Τα διάφορα τέτοια μονοπάτια (οι παραπάνω κανόνες δηλαδή) συνδέονται μεταξύ τους με διάζευξη (λογικό 'Η)
- Το δένδρο συνολικά εκφράζει τη διάζευξη αυτών των συζεύξεων, αφού αποτελείται από όλα τα εναλλακτικά μονοπάτια

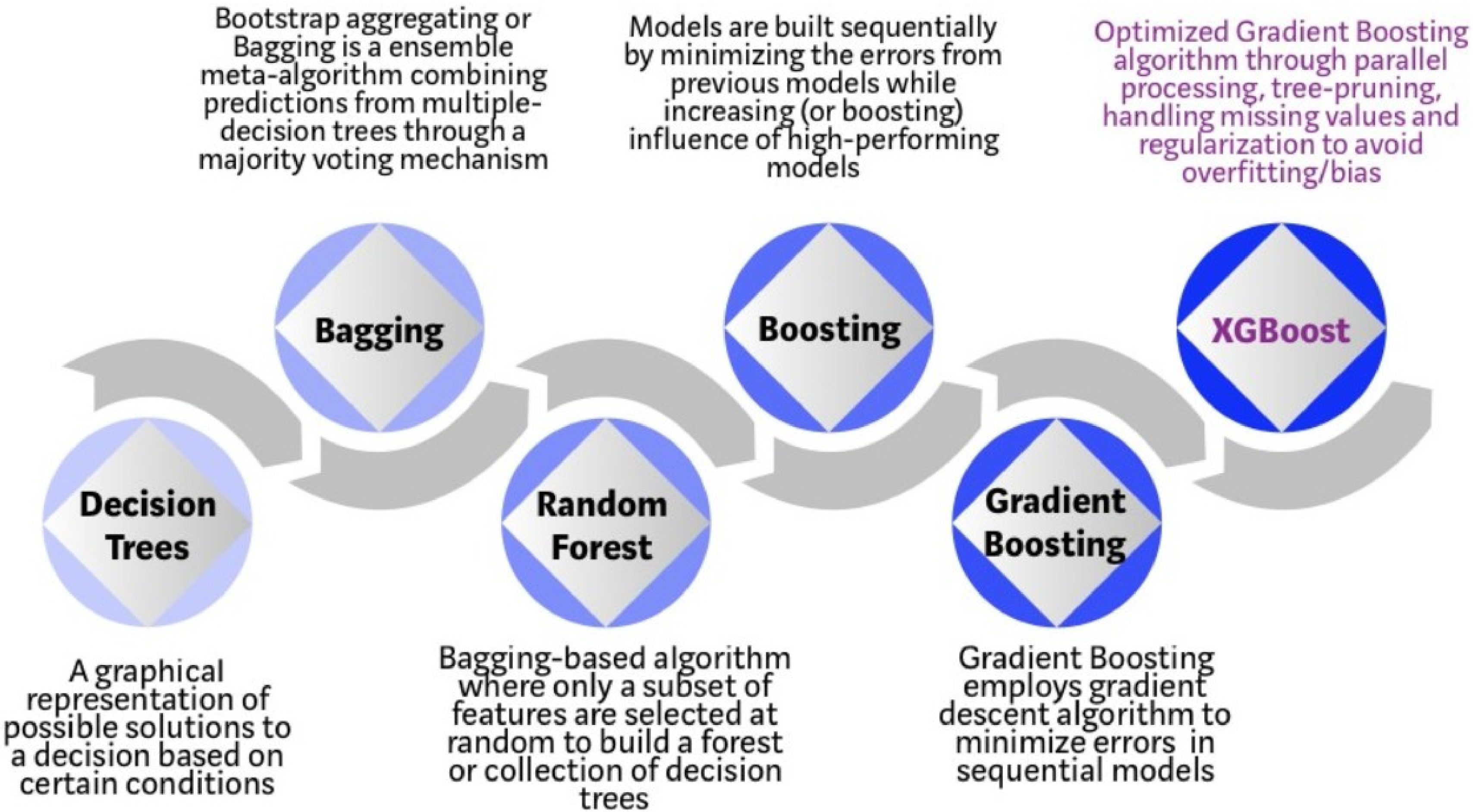
Αλγόριθμοι μάθησης Δένδρων Απόφασης/Ταξινόμησης

- Οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί για μάθηση δένδρων απόφασης (Decision Tree training algorithm) είναι παραλλαγές ενός βασικού αλγορίθμου, ο οποίος κάνει μη-πλήρη αναζήτηση στο χώρο των πιθανών δένδρων απόφασης χτίζοντας το υποψήφιο δένδρο:
  - Από πάνω (ρίζα) προς τα κάτω (φύλλα), και
  - *Άπληστα* (*greedy*), επιλέγοντας κάθε φορά ως παράμετρο διακλάδωσης το καλύτερο τοπικά χαρακτηριστικό
- Παραδείγματα του βασικού αυτού αλγορίθμου αποτελούν οι ακόλουθοι αλγόριθμοι:

- ID3 (Iterative Dichotomiser 3)[1]. Φτιάχνει το δένδρο χρησιμοποιώντας ως κριτήριο διαχωρισμού (splitting criterion) το μέγεθος *κέρδος πληροφορίας* (information gain) που βασίζεται στην *εντροπία πληροφορίας* (information entropy) κάθε κόμβου. Υποστηρίζει μόνο κατηγορικά χαρακτηριστικά.
- C4.5 (εξέλιξη του ID3) Υποστηρίζει επιπλέον, χαρακτηριστικά με συνεχείς αλλά και ελλιπείς τιμές, ενώ ενσωματώνει και μηχανισμό κλαδέματος για αποφυγή υπερπροσαρμογής
  - ✓ καθώς και ο διάδοχος του See5 (η έκδοση για τα windows 7/8/10) και C5.0 (η έκδοση για το Linux) που είναι ταχύτεροι με δυνατότητα πολυνηματικής (multithreaded) εκτέλεσης (σύγκριση με τον C4.5)
- CART (Classification And Regression Tree)[2] Χρησιμοποιείται για κατασκευή δυαδικών (binary) δένδρων, για ταξινόμηση και παρεμβολή. Κριτήριο διαχωρισμού ο δείκτης Gini ή entropy (για ταξινόμηση) ή το άθροισμα τετραγωνικού σφάλματος ή την απόλυτη τιμή σφάλματος (για παρεμβολή). Υποστηρίζει μόνο αριθμητικά χαρακτηριστικά.
- CHAID (CHi-squared Automatic Interaction Detector). Κριτήριο διαχωρισμού το στατιστικό μέτρο χ-τετράγωνο (Chi-Square) για τον υπολογισμό της στατιστικής διαφοράς μεταξύ κόμβου-πατέρα και κόμβων-παιδιών
- Conditional Inference Trees. Χρησιμοποιεί στατιστικές μεθόδους για να αποφασίσει τις διχοτομήσεις και όχι εντροπία πληροφορίας. Λύνει κάποια προβλήματα των CART trees. Δύσκολη επεξήγηση του δένδρου.
- Random Forest Μέθοδος συλλογικής μάθησης (ensemble learning) που κατασκευάζει πολλά δένδρα και λαμβάνει απόφαση βάσει πλειοψηφίας (ταξινόμηση) ή βάσει μέσης τιμής (σε προβλήματα παρεμβολής).
  - ✓ Όλες οι έννοιες που αναφέρθηκαν πριν, αναλύονται και εξηγούνται στη συνέχεια ενώ από τους προαναφερθέντες αλγορίθμους περιγράφονται ο ID3, οι βελτιώσεις που εισάγει ο C4.5 και ο Random Forest.
  - ✓ Σε επόμενο κεφάλαιο θα παρουσιαστούν τεχνικές ενίσχυσης (*Boosting*) του αρχικού μοντέλου (δένδρου).
  - ✓ (Δες επόμενο slide). Στο τέλος του κεφαλαίου δίνεται μια σύντομη περιγραφή.

[1] Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106

[2] *Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Softw. ISBN 978-0-412-04841-8.*


Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

Ο αλγόριθμος ID3 - Γενική Περιγραφή

- Ο πιο γνωστός αλγόριθμος μάθησης δένδρων ταξινόμησης
  - Κατασκευάζει το δένδρο άπληστα από πάνω προς τα κάτω
  - Αρχικά επιλέγει το πιο κατάλληλο χαρακτηριστικό των δεδομένων για έλεγχο στη ρίζα.
    - ✓ Η επιλογή βασίζεται σε κάποιο στατιστικό μέτρο που υπολογίζεται από τα παραδείγματα εκπαίδευσης
  - Στη συνέχεια, για κάθε δυνατή τιμή του χαρακτηριστικού δημιουργούνται οι απόγονοι της ρίζας
    - ✓ Τα δεδομένα μοιράζονται στους νέους κόμβους ανάλογα με την τιμή που έχουν για το χαρακτηριστικό που ελέγχεται στη ρίζα
  - Η όλη διαδικασία επαναλαμβάνεται για κάθε νέο κόμβο
    - ✓ Η επιλογή του κατάλληλου χαρακτηριστικού για έλεγχο σε ένα νέο κόμβο υπολογίζεται και πάλι στατιστικά, αυτή τη φορά όμως χρησιμοποιώντας μόνο τα δεδομένα που ανήκουν σε αυτόν τον κόμβο
    - ✓ Η διαδικασία τερματίζει όταν οι κόμβοι γίνουν *τερματικοί* (*φύλλα του δένδρου*)
- Ένας κόμβος γίνεται τερματικός (φύλλο) όταν:
  - Όλα τα δεδομένα που ανήκουν σε αυτόν ανήκουν στην ίδια κατηγορία/κλάση (*αμιγής κόμβος - pure node*).
    - ✓ Αυτή η κατηγορία (*κλάση*) γίνεται και η τιμή του κόμβου.
  - Όταν τελειώσουν τα χαρακτηριστικά προς έλεγχο
    - ✓ Η πρόβλεψη του κόμβου τότε προκύπτει πλειοψηφικά με βάση τα δεδομένα του κόμβου αυτού

Αμιγή δένδρα

- Αμιγές ή καθαρό δένδρο (*pure tree*) είναι αυτό που έχει όλους τους τερματικούς του κόμβους αμιγείς.
  - Περιγράφει απόλυτα τα δεδομένα εκπαίδευσης. Αυτό όμως δε σημαίνει ότι θα προβλέπει σωστά και οποιαδήποτε άλλα (μελλοντικά) δεδομένα.
  - Γενικά τα αμιγή δένδρα δεν είναι ούτε συνηθισμένα αλλά ούτε και επιθυμητά γιατί εμφανίζουν περιορισμένη ικανότητα γενίκευσης, άρα ελλοχεύει σε αυτά ο κίνδυνος *υπερπροσαρμογής*.
  - Για αποφυγή κάτι τέτοιου, συνήθως εφαρμόζονται τεχνικές *κλαδέματος* (*pruning*).
- Στην πραγματικότητα τα περισσότερα δένδρα δεν προκύπτουν αμιγή ακόμα και μετά από εξέταση όλων των ανεξάρτητων μεταβλητών.
  - Αυτό κατ' αρχήν φαίνεται περίεργο αλλά μπορεί να συμβεί όταν για παράδειγμα υπάρχουν αλληλοσυγκρουόμενα δεδομένα εκπαίδευσης (*conflicting data*) στα οποία αν και οι τιμές στις ανεξάρτητες μεταβλητές είναι ίδιες εν τούτοις η τιμή της εξαρτημένης μεταβλητής είναι διαφορετική.
  - Σε τέτοιες περιπτώσεις η πρόβλεψη του κόμβου προκύπτει πλειοψηφικά.
  - Ταυτόχρονα όμως υπάρχει και μια ένδειξη ότι τα δεδομένα εκπαίδευσης πιθανώς δεν είναι καλά (π.χ. περιέχουν θόρυβο) ή ότι ίσως υπάρχουν και άλλα χαρακτηριστικά (ανεξάρτητες μεταβλητές) που πρέπει να ληφθούν υπ' όψη.

Ο αλγόριθμος ID3

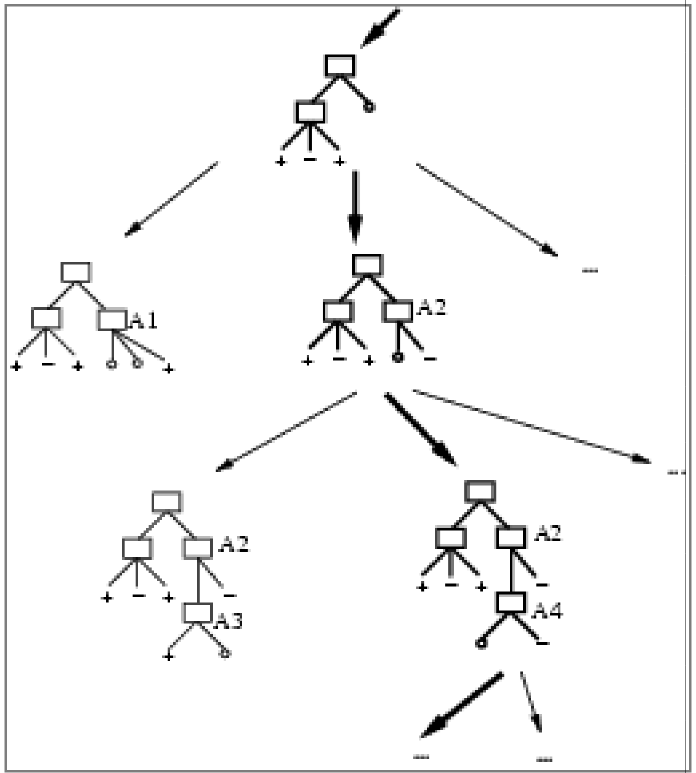
- Αλγοριθμική Περιγραφή:

```
ΑΛΓΟΡΙΘΜΟΣ: ID3(S, y, Attributes)
Είσοδος:  Σύνολο παραδειγμάτων εκπαίδευσης S
                    Σύνολο χαρακτηριστικών Attributes
                    Μεταβλητή στόχος y
Έξοδος:   Δένδρο root

Αρχή
   Φτιάξε τη ρίζα του δένδρου root
   Αν όλα τα s στο S είναι θετικά,
      επέστρεψε το δένδρο-ρίζα root με ετικέτα +
   Αν όλα τα s στο S είναι αρνητικά,
      επέστρεψε το δένδρο-ρίζα root με ετικέτα -
   Αν το X είναι άδειο,
      επέστρεψε το δένδρο-ρίζα root με ετικέτα την πιο κοινή τιμή της y στο S
   Αλλιώς
      Θέσε A το "καλύτερο" χαρακτηριστικό του Attributes
      Το χαρακτηριστικό απόφασης για τη ρίζα γίνεται το A
      Για κάθε διαφορετική τιμή, v_i του A:
         Πρόσθεσε ένα καινούριο κλαδί κάτω από τη ρίζα, για τον έλεγχο A = v_i
         Θέσε το S_vi να περιέχει τα παραδείγματα του S για τα οποία A = v_i
         Αν το S_vi είναι άδειο,
            πρόσθεσε ένα φύλλο με ετικέτα την πιο κοινή τιμή της y για όλα τα s στο S.
         Αλλιώς
            πρόσθεσε το δένδρο ID3(S_vi, y, Attributes – {A})
   Επέστρεψε το δένδρο root
Τελος
```

Η αναζήτηση του ID3 στο χώρο των υποθέσεων

- Το βασικότερο στάδιο του αλγορίθμου είναι η επιλογή της ανεξάρτητης μεταβλητής πάνω στην οποία θα συνεχιστεί η ανάπτυξη του δένδρου.
  - Απαιτείται ο ορισμός κάποιου μηχανισμού ο οποίος θα καθοδηγήσει την αναζήτηση προς το κατ' εκτίμηση καλύτερο δένδρο (περιγραφή) μέσα στο σύνολο των δυνατών δένδρων.
  - Ο χώρος υποθέσεων στον οποίο κάνει αναζήτηση ο αλγόριθμος ID3 απαρτίζεται από όλα τα πιθανά δένδρα αποφάσεων.
  - Η αναζήτηση ξεκινά με ένα άδειο δένδρο, ενώ στη συνέχεια ο αλγόριθμος το επεκτείνει προοδευτικά με στόχο να βρει ένα δένδρο που ταξινομεί σωστά τα δεδομένα εκπαίδευσης.



- Η στρατηγική αναζήτησης είναι *[αναρρίχηση λόφων]* ([hill climbing]) γιατί σε κάθε κύκλο λειτουργίας επεκτείνει το τρέχον δένδρο με τον τοπικά καλύτερο τρόπο και συνεχίζει χωρίς δυνατότητα οπισθοδρόμησης.
  - Αυτό τον κάνει αφενός εξαιρετικά αποδοτικό, αφετέρου ισχυρά εξαρτώμενο από το μηχανισμό διαχωρισμού που θα επιλεγεί.
- Ταυτόχρονα, αυτή η μη-πλήρης αναζήτηση στο χώρο των δένδρων, τον κάνει να έχει μεροληψία προτίμησης, με την έννοια ότι προτιμά δένδρα που ευνοούνται από το κριτήριο που χρησιμοποιεί για να τα κατασκευάσει.

Κριτήριο διαχωρισμού με βάση την εντροπία

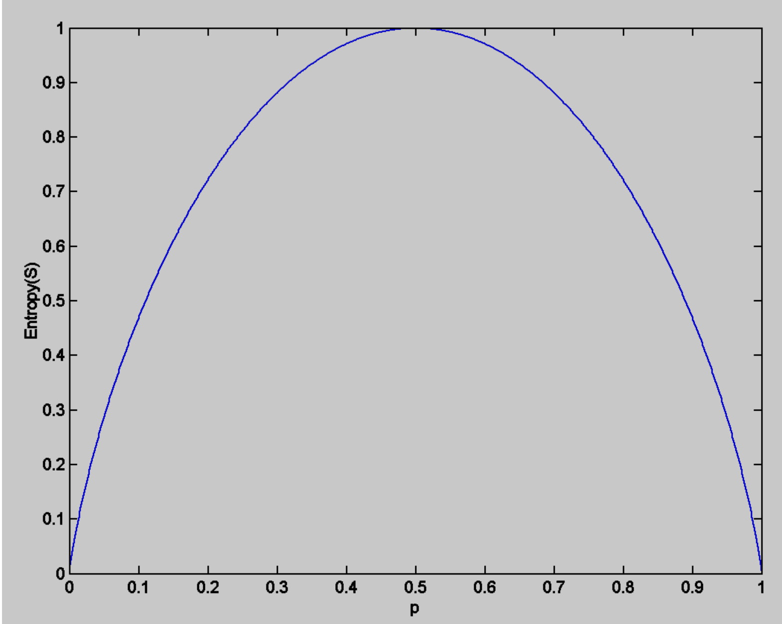
- Ένα από τα πιο διαδεδομένα κριτήρια διαχωρισμού βασίζεται στην εντροπία της πληροφορίας (information entropy) και επιλέγει εκείνη την ανεξάρτητη μεταβλητή που οδηγεί σε περισσότερο συμπαγές δένδρο.
  - Η εντροπία χαρακτηρίζει την *ανομοιογένεια* (*impurity*) μιας συλλογής παραδειγμάτων.
- Η τιμή της εντροπίας της πληροφορίας για δύο κλάσεις (κατηγορίες), θετική και αρνητική, δίνεται από τη σχέση:

$$E(S) = -p_+ \, log_2 \, p_+ - p_- \, log_2 \, p_-$$

  - όπου S είναι το σύνολο των δεδομένων εκπαίδευσης στο στάδιο (κόμβο) του διαχωρισμού, p+ είναι το κλάσμα των θετικών παραδειγμάτων του S και p- είναι το κλάσμα των αρνητικών παραδειγμάτων του S.
  - Σε όλους τους υπολογισμούς της εντροπίας, θα θεωρούμε ότι "0 $\log_2$ 0" είναι ίσο με "0".
  - Έχει τις ρίζες της στη θεωρία πληροφοριών ("A Mathematical Theory of Communication", Shannon, 1984)

Εντροπία

- Παράδειγμα υπολογισμού εντροπίας
  - Έστω S ένα σύνολο με 9 θετικά και 5 αρνητικά παραδείγματα [9+,5-]
  - $E(S) = -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) = 0.940$



- Ενδιαφέρουσες ιδιότητες
  - Η εντροπία είναι 0 αν όλα τα μέλη του S ανήκουν στην ίδια κατηγορία.
  - Η εντροπία είναι 1 αν τα μισά μέλη ανήκουν στη μια και τα άλλα μισά στην άλλη κατηγορία.
- Γενικότερος ορισμός για *c* διαφορετικές κατηγορίες
  - Έστω p_i το ποσοστό των παραδειγμάτων του S που ανήκουν στην κατηγορία i.

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

- Το κριτήριο της εντροπίας ευνοεί διαχωρισμούς που δημιουργούν πολλά παρακλάδια με μικρούς ομοιογενείς πληθυσμούς (Δηλαδή χαρακτηριστικά με πολλές τιμές).

Ο δείκτης Gini (Gini Index)

- Ένα άλλο, απλούστερο στην υλοποίηση κριτήριο διαχωρισμού, είναι ο δείκτης Gini (Gini Index).
- Για c διαφορετικές κατηγορίες, ο δείκτης Gini ορίζεται από τη σχέση:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

  - όπου το pi ορίζεται όπως στη σχέση της εντροπίας.
- Ο δείκτης Gini υπολογίζεται ταχύτερα, ευνοεί μεγαλύτερους διαχωρισμούς
  - Χρησιμοποιείται στον αλγόριθμο CART (Gini Split)

.

Splitting Criteria (Σύνοψη)

- Criteria Based on Impurity
  - Entropy, Gini index, and RSS criteria decrease impurity.
  - The impurity of a parent node τ is defined as $i(\tau)$, a non negative number that is equal to zero for a pure node.
  - *Gini* impurity and Information *Gain Entropy* are pretty much the same, and people do use the values interchangeably.
    - ✓ Given a choice, Gini impurity is preferable, as it doesn't require to compute logarithmic functions, which are computationally intensive.
  - Below are the formulae of both:

1. $Gini : Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$
2. $Entropy : H(E) = -\sum_{j=1}^{c} p_j \log p_j$

  - In statistics, the *Residual Sum of Squares* (RSS), also known as the *Sum of Squared Residuals* (SSR) or the *Sum of Squared Errors of prediction* (SSE), is the sum of the squares of residuals (deviations predicted from actual empirical values of data).
    - ✓ It is a measure of the discrepancy between the data and an estimation model.

Criteria Based on Statistical Test

- The chi-square, F test, CHAID, and FastCHAID criteria are defined by statistical tests.
- These criteria calculate the worth of a split by testing for a significant difference in the response variable across the branches defined by a split.
- The worth is defined as *-log(p)*, where p is the p-value of the test.
- You can adjust the p-values for these criteria by specifying the BONFERRONI option in the GROW statement.
- The criteria based on statistical tests compute the worth of a split as follows:
  - ✓ Chi-square criterion: For categorical response variables, the worth is based on the p-value for the Pearson chi-square test that compares the frequencies of the levels of the response across the child nodes.

Conclusion

- For categorical responses, the available criteria are CHAID, CHISQUARE, ENTROPY, FASTCHAID, and GINI, and the default criterion is ENTROPY.
- For continuous responses, the available criteria are CHAID, FTEST, and RSS, and the default criterion is RSS.

Κέρδος Πληροφορίας (Information Gain)

- Στην πράξη, ο ID3 χρησιμοποιεί το κέρδος πληροφορίας, G(S, A) που αναπαριστά τη μείωση της εντροπίας του συνόλου εκπαίδευσης S αν επιλεγεί ως παράμετρος διαχωρισμού η μεταβλητή A
- Όταν μειώνεται η πληροφοριακή εντροπία, αυξάνεται η πυκνότητα πληροφορίας και άρα η περιγραφή γίνεται περισσότερο συμπαγής.
  - Έστω ένα σύνολο *S* και ένα χαρακτηριστικό *A* με σύνολο τιμών *V*(*A*). Το κέρδος πληροφορίας σε σχέση με αυτό το χαρακτηριστικό είναι:

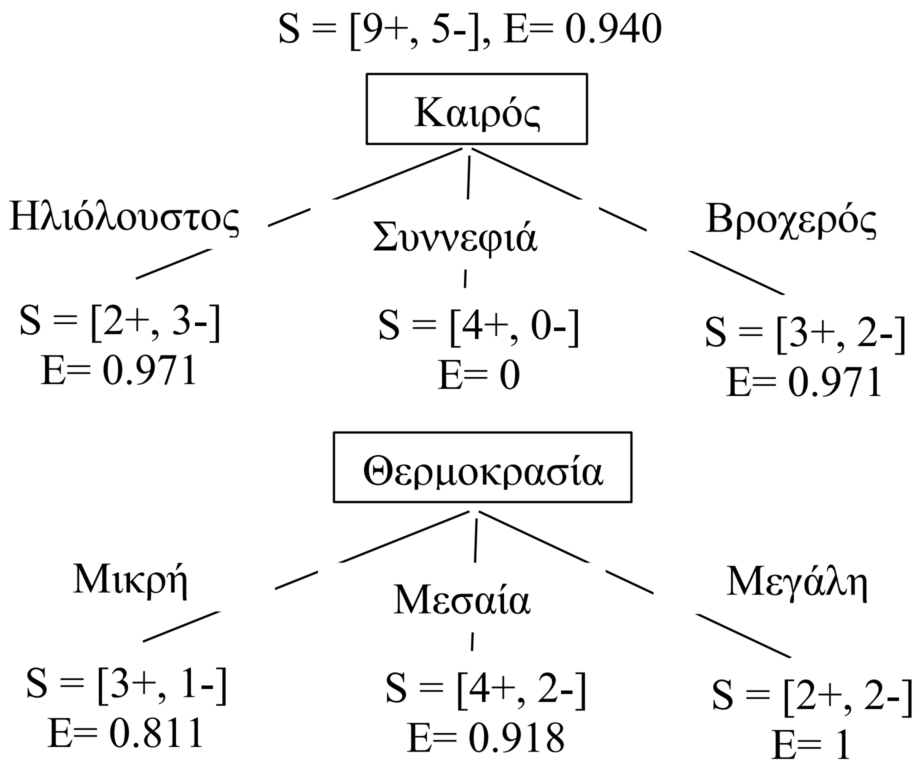
$$G(S,A) = E(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} E(S_v)$$

    - ✓ |S| το πλήθος των παραδειγμάτων του υπό εξέταση κόμβου,
    - ✓ E(S) η εντροπία πληροφορίας του υπό εξέταση κόμβου,
    - ✓ A η ανεξάρτητη μεταβλητή την οποία και αξιολογούμε ως πιθανή επιλογή για την επόμενη διακλάδωση,
    - ✓ V(A) το σύνολο των τιμών που μπορεί να πάρει η A, και v μία από αυτές,
    - ✓ $|S_v|$ το πλήθος των παραδειγμάτων με A=v,
    - ✓ $E(S_v)$ η εντροπία πληροφορίας του υπό εξέταση κόμβου ως προς A= v.
  - Ουσιαστικά, ο δεύτερος όρος (το άθροισμα) είναι η εντροπία των παραδειγμάτων μετά το διαχωρισμό τους σε τόσες υπο-ομάδες όσες και οι διαφορετικές τιμές του A (δηλ. το άθροισμα της εντροπίας των υποομάδων).

Παράδειγμα εκτέλεσης ID3 (1/6)

- Πρόβλεψη καλού καιρού για τένις

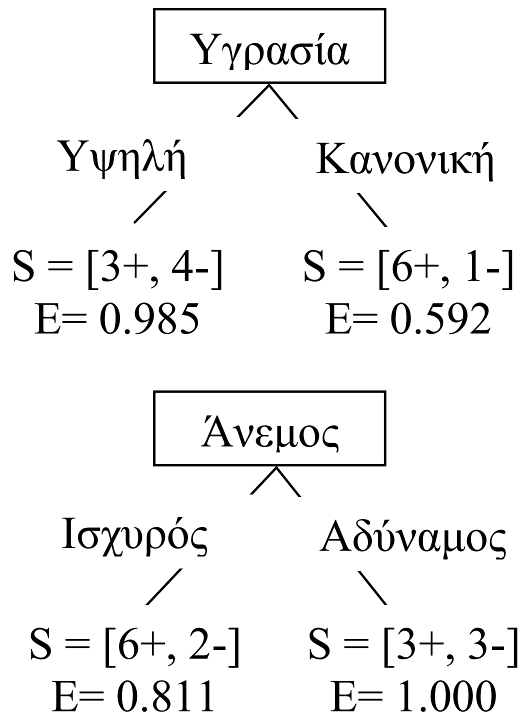
Ημέρα	Καιρός	Θερμοκρασία	Υγρασία	Άνεμος	Τένις
1	Ηλιόλουστος	Μεγάλη	Υψηλή	Αδύναμος	Όχι
2	Ηλιόλουστος	Μεγάλη	Υψηλή	Ισχυρός	Όχι
3	Συννεφιασμένος	Μεγάλη	Υψηλή	Αδύναμος	Ναι
4	Βροχερός	Μεσαία	Υψηλή	Αδύναμος	Ναι
5	Βροχερός	Μικρή	Κανονική	Αδύναμος	Ναι
6	Βροχερός	Μικρή	Κανονική	Ισχυρός	Όχι
7	Συννεφιασμένος	Μικρή	Κανονική	Ισχυρός	Ναι
8	Ηλιόλουστος	Μεσαία	Υψηλή	Αδύναμος	Όχι
9	Ηλιόλουστος	Μικρή	Κανονική	Αδύναμος	Ναι
10	Βροχερός	Μεσαία	Κανονική	Αδύναμος	Ναι
11	Ηλιόλουστος	Μεσαία	Κανονική	Ισχυρός	Ναι
12	Συννεφιασμένος	Μεσαία	Υψηλή	Ισχυρός	Ναι
13	Συννεφιασμένος	Μεγάλη	Κανονική	Αδύναμος	Ναι
14	Βροχερός	Μεσαία	Υψηλή	Ισχυρός	Όχι



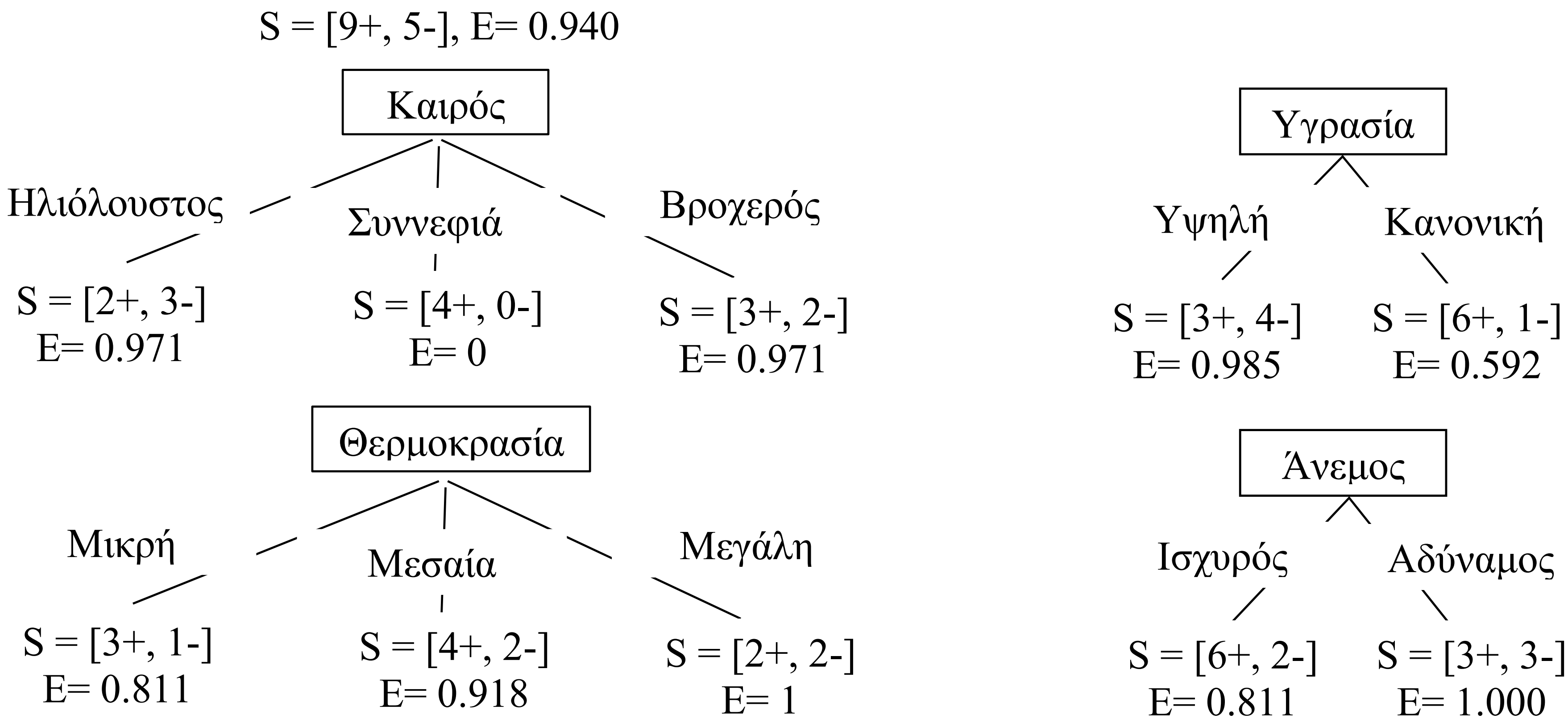
Παράδειγμα εκτέλεσης ID3 (2/6)

- Πρόβλεψη καλού καιρού για τένις

Ημέρα	Καιρός	Θερμοκρασία	Υγρασία	Άνεμος	Τένις
1	Ηλιόλουστος	Μεγάλη	Υψηλή	Αδύναμος	Όχι
2	Ηλιόλουστος	Μεγάλη	Υψηλή	Ισχυρός	Όχι
3	Συννεφιασμένος	Μεγάλη	Υψηλή	Αδύναμος	Ναι
4	Βροχερός	Μεσαία	Υψηλή	Αδύναμος	Ναι
5	Βροχερός	Μικρή	Κανονική	Αδύναμος	Ναι
6	Βροχερός	Μικρή	Κανονική	Ισχυρός	Όχι
7	Συννεφιασμένος	Μικρή	Κανονική	Ισχυρός	Ναι
8	Ηλιόλουστος	Μεσαία	Υψηλή	Αδύναμος	Όχι
9	Ηλιόλουστος	Μικρή	Κανονική	Αδύναμος	Ναι
10	Βροχερός	Μεσαία	Κανονική	Αδύναμος	Ναι
11	Ηλιόλουστος	Μεσαία	Κανονική	Ισχυρός	Ναι
12	Συννεφιασμένος	Μεσαία	Υψηλή	Ισχυρός	Ναι
13	Συννεφιασμένος	Μεγάλη	Κανονική	Αδύναμος	Ναι
14	Βροχερός	Μεσαία	Υψηλή	Ισχυρός	Όχι



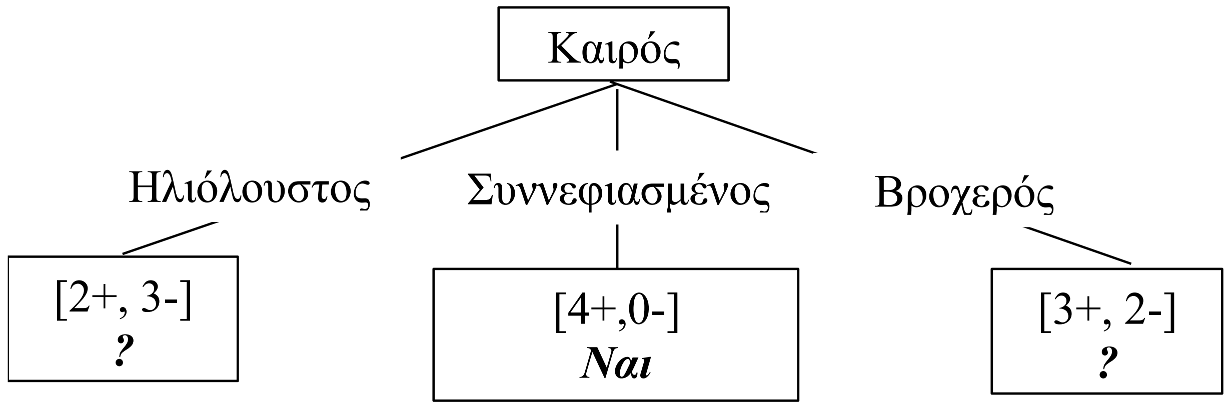
Παράδειγμα εκτέλεσης ID3 (3/6)



- Κέρδος πληροφορίας
  - G(S, Καιρός) = 0.940-(5/14)*0.971-(4/14)*0-(5/14)*0.971 = **0.246**
  - G(S, Θερμοκρασία) = 0.940-(4/14)*0.811-(6/14)*0.918-(4/14)*1 = 0.029
  - G(S, Υγρασία) = 0.940-(7/14)*0.985-(7/14)* 0.592 = 0.151
  - G(S, Άνεμος) = 0.940-(8/14)*0.811-(6/14)*1.0 = 0.048

Παράδειγμα εκτέλεσης ID3 (4/6)

- Άρα επιλέγουμε για διαχωρισμό το χαρακτηριστικό "Καιρός", επειδή έχει το μέγιστο κέρδος πληροφορίας.

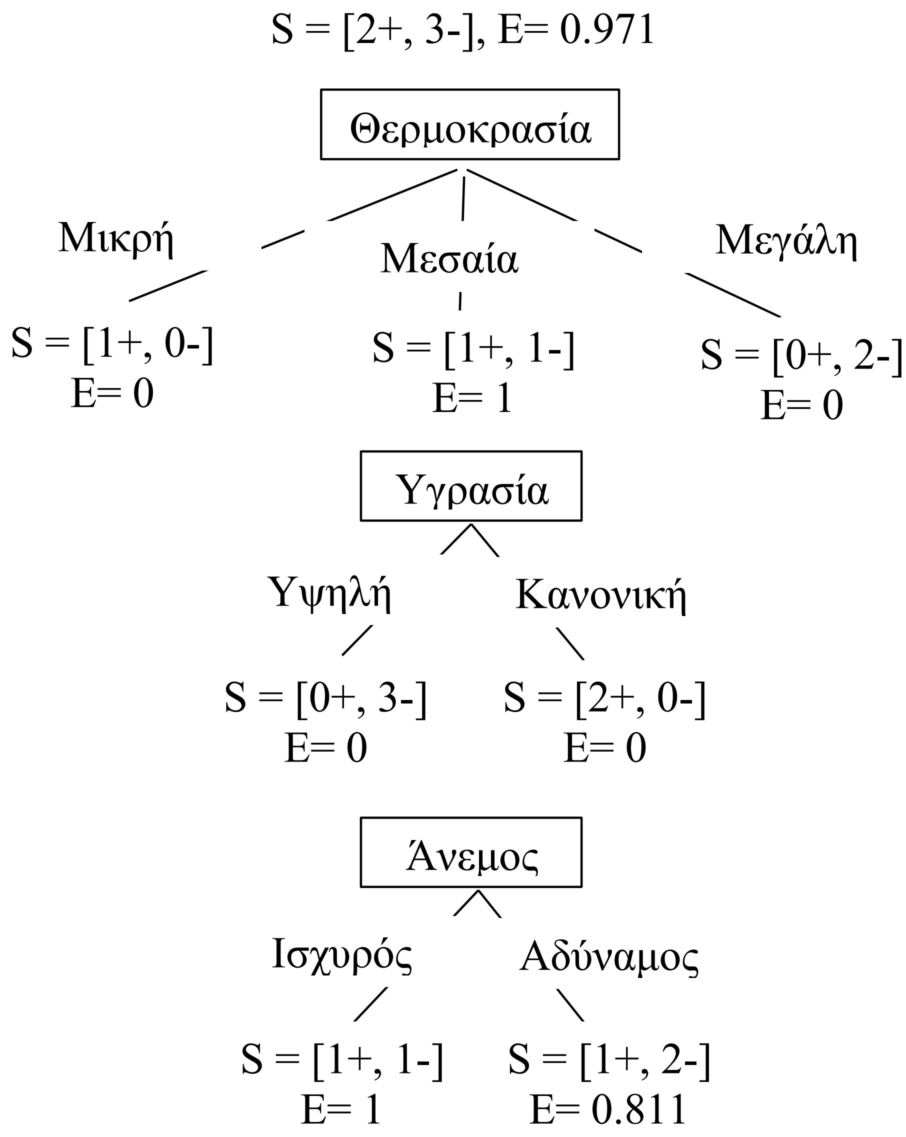


- Συνεχίζουμε τον αλγόριθμο επαναλαμβάνοντας τη διαδικασία μόνο για τους κόμβους στους οποίους οδηγούν τα κλαδιά "Καιρός=Ηλιόλουστος" και "Καιρός=Βροχερός", αφού ο κόμβος "Καιρός=Συννεφιασμένος" είναι αμιγής.
- Για την εύρεση του χαρακτηριστικού με το οποίο θα γίνει ο διαχωρισμός στον κόμβο που οδηγεί το κλαδί "Καιρός=Ηλιόλουστος" θα χρησιμοποιήσουμε <u>μόνο τις 5 περιπτώσεις</u> που ανήκουν στον κόμβο αυτό.

Παράδειγμα εκτέλεσης ID3 (5/6)

- Πρόβλεψη καλού καιρού για τένις

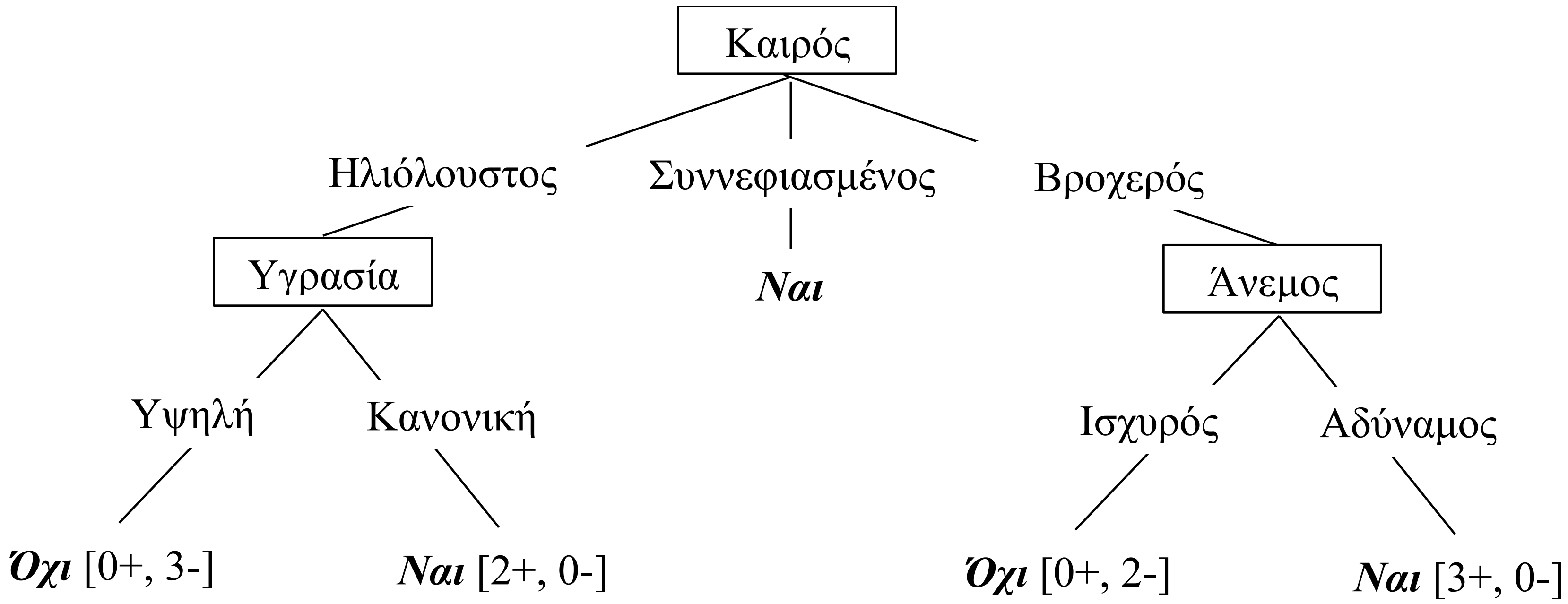
Ημέρα	Καιρός	Θερμοκρασία	Υγρασία	Άνεμος	Τένις
1	Ηλιόλουστος	Μεγάλη	Υψηλή	Αδύναμος	Όχι
2	Ηλιόλουστος	Μεγάλη	Υψηλή	Ισχυρός	Όχι
3	Συννεφιασμένος	Μεγάλη	Υψηλή	Αδύναμος	Ναι
4	Βροχερός	Μεσαία	Υψηλή	Αδύναμος	Ναι
5	Βροχερός	Μικρή	Κανονική	Αδύναμος	Ναι
6	Βροχερός	Μικρή	Κανονική	Ισχυρός	Όχι
7	Συννεφιασμένος	Μικρή	Κανονική	Ισχυρός	Ναι
8	Ηλιόλουστος	Μεσαία	Υψηλή	Αδύναμος	Όχι
9	Ηλιόλουστος	Μικρή	Κανονική	Αδύναμος	Ναι
10	Βροχερός	Μεσαία	Κανονική	Αδύναμος	Ναι
11	Ηλιόλουστος	Μεσαία	Κανονική	Ισχυρός	Ναι
12	Συννεφιασμένος	Μεσαία	Υψηλή	Ισχυρός	Ναι
13	Συννεφιασμένος	Μεγάλη	Κανονική	Αδύναμος	Ναι
14	Βροχερός	Μεσαία	Υψηλή	Ισχυρός	Όχι



- Κέρδος Πληροφορίας
  - G(S, Θερμοκρασία) = 0.971-(1/5)*0-(2/5)*1-(2/5)*0 = 0.571
  - G(S, Υγρασία) = 0.971-(3/5)*0-(2/5)* 0 = **0.971**
  - G(S, Άνεμος) = 0.971-(2/5)*1-(3/5)*0.918 = 0.020

Παράδειγμα εκτέλεσης ID3 (6/6)

- Άρα επιλέγουμε για διαχωρισμό το χαρακτηριστικό "Υγρασία", επειδή έχει το μέγιστο κέρδος πληροφορίας.
- Ομοίως συνεχίζουμε και για το κόμβο στον οποίο οδηγεί το κλαδί "Καιρός=Βροχερός", για τον οποίο, αν κάνουμε τους υπολογισμούς, προκύπτει ότι το μέγιστο κέρδος πληροφορίας προσφέρει το χαρακτηριστικό "Άνεμος".
- Το τελικό δένδρο είναι:



Μειονεκτήματα της αναζήτησης του ID3

- Διατηρεί κατά την αναζήτηση μόνο μια υπόθεση (δένδρο) συμβατή με τα δεδομένα. Επομένως δεν μπορεί να βρει όλα τα δένδρα που είναι συμβατά με τα δεδομένα.
- Δεν κάνει οπισθοδρόμηση (backtracking) κατά την αναζήτηση του.
  - Άπαξ και επιλέξει ένα χαρακτηριστικό για έλεγχο σε κάποιο κόμβο, δεν επιστρέφει ποτέ πίσω για να αλλάξει την επιλογή αυτή
  - Αυτό σημαίνει ότι διατρέχει τον κίνδυνο εύρεσης τοπικά μόνο βέλτιστων δένδρων
- Παρακάτω θα μελετηθεί μια επέκταση του ID3 που κάνει ένα είδος οπισθοδρόμησης (κλάδεμα του δένδρου μετά την πλήρη ανάπτυξη του).

Επαγωγική Μεροληψία (inductive bias) του ID3

- Πως μπορεί ο ID3 και γενικεύει ώστε να ταξινομεί νέα δεδομένα;
  - Μια προσέγγιση: "Τα μικρότερα δένδρα είναι προτιμότερα από τα μεγαλύτερα".
  - Μια καλύτερη προσέγγιση: "Τα μικρότερα δένδρα είναι προτιμότερα από τα μεγαλύτερα και προτιμώνται περισσότερο τα δένδρα που εξετάζουν πιο κοντά στη ρίζα τα χαρακτηριστικά με μεγαλύτερο κέρδος πληροφορίας"
- Μεροληψία περιορισμού και προτίμησης
  - Ο αλγόριθμος ID3 έχει ***μεροληψία προτίμησης*** επειδή ψάχνει έναν πλήρη χώρο υποθέσεων, αλλά εκτελεί μη πλήρη αναζήτηση σε αυτόν (εκτελεί *ευρετική* αναζήτηση).
    - ✓ Η μεροληψία προκύπτει από την στρατηγική αναζήτησης και όχι από τον χώρο υποθέσεων
- Είναι καλό να προτιμάται η απλούστερη υπόθεση; (Occam's razor)
  - Μοιάζει ενδιαφέρουσα άποψη, αλλά ίσως μην είναι το καλύτερο που μπορεί να γίνει για όλες τις περιπτώσεις.

Ο Αλγόριθμος C4.5

- Αποτελεί βελτίωση του ID3 αντιμετωπίζοντας:
  - την προτίμηση του ID3 σε χαρακτηριστικά με πολλές διακριτές τιμές
  - υποστηρίζοντας και χαρακτηριστικά με *ελλιπείς* (*missing*) τιμές,
  - χαρακτηριστικά με μεγάλη διαφορά κόστους,
  - χαρακτηριστικά με συνεχείς τιμές, καθώς και
  - κλάδεμα του παραγόμενου δένδρου για αποφυγή *υπερπροσαρμογής*.

- Υλοποίηση του C4.5 σε Python:
  - ✓ https://pypi.org/project/c45-decision-tree/ and https://github.com/geerk/C45algorithm
- To scikit-learn υποστηρίζει τον CART which is very similar to C4.5
  - ✓ https://stackoverflow.com/questions/66627436/decision-tree-in-python-with-sklearn-change-sklearn-to-use-c4-5

⇒ **J48** is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool.
⇒ Binary tree, multiclass. Example IRIS dataset.

1. Άλλα στατιστικά μεγέθη αξιολόγησης χαρακτηριστικών

- Το κέρδος πληροφορίας έχει την μεροληψία ότι προτιμά χαρακτηριστικά με πολλές διακριτές τιμές σε σχέση με αυτά που έχουν λίγες γιατί παράγουν το μέγιστο κέρδος πληροφορίας.
  - Δημιουργούνται περισσότερες ομάδες που προφανώς έχουν μικρότερη εντροπία από τις μεγαλύτερες
    - ✓ Ακραίο παράδειγμα είναι η ύπαρξη χαρακτηριστικού ημερομηνίας στα δεδομένα του παραδείγματος με το Τένις.
    - ✓ Αυτό θα χώριζε τέλεια τα δεδομένα σε φύλλα που έχουν μόνο ένα δεδομένο και επομένως μηδέν εντροπία
    - ✓ Ποια η χρησιμότητα όμως της ημερομηνίας για πρόβλεψη;
- Ο λόγος κέρδους (Gain Ratio) είναι ένα εναλλακτικό στατιστικό που αντιμετωπίζει αυτό το πρόβλημα. Ορίζεται ως εξής:

$$GR(S,A) = \frac{G(S,A)}{SI(S,A)}$$

  - Βασίζεται στο στατιστικό μέγεθος ***Πληροφορία Διαχωρισμού*** (***Split Information, SI***), το οποίο είναι ευαίσθητο στο εύρος και την ομοιομορφία διαχωρισμού των δεδομένων από ένα χαρακτηριστικό:

$$SI(S,A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}$$

- Έχουν προταθεί επίσης διάφορα άλλα στατιστικά

2. Χειρισμός δεδομένων με ελλιπείς τιμές

- Σε πολλές περιπτώσεις (π.χ. ιατρικά δεδομένα) είναι πιθανό να λείπουν οι τιμές για κάποια χαρακτηριστικά.
- Πως υπολογίζεται το κέρδος πληροφορίας για ένα τέτοιο χαρακτηριστικό;
  - Του αναθέτουμε την πιο κοινή τιμή που έχουν τα δεδομένα του κόμβου.
  - Του αναθέτουμε την πιο κοινή τιμή που έχουν τα δεδομένα του κόμβου με την ίδια τιμή για το χαρακτηριστικό πρόβλεψης.
- Μια πιο σύνθετη στρατηγική που ακολουθεί ο C4.5:
  - Μπορούμε να αναθέσουμε πιθανότητες για την κάθε ξεχωριστή τιμή του χαρακτηριστικού με βάση το ποσοστό εμφάνισης των τιμών αυτών στα δεδομένα του κόμβου.
  - Οι πιθανότητες συνεισφέρουν στον υπολογισμό του κέρδους πληροφορίας, ενώ το δεδομένο μοιράζεται σε όλους τους κόμβους που ακολουθούν μαζί με την αντίστοιχη πιθανότητα.
    - ✓ Έστω ότι 35 περιπτώσεις πρέπει να διαχωριστούν στο χαρακτηριστικό "Φύλο" με τιμές "άνδρας" (10 περιπτώσεις) και "γυναίκα" (20 περιπτώσεις), ενώ υπάρχουν και 5 περιπτώσεις που δεν έχουν τιμή στο "Φύλο". Οι 5 αυτές περιπτώσεις θα διοχετευτούν και στους δύο υπο-κόμβους που θα δημιουργηθούν, με βαρύτητα 10/30 για την τιμή "άνδρας" και 20/30 για την τιμή "γυναίκα".
- Πως αποφασίζουμε για μια νέα περίπτωση που της λείπουν τιμές;
  - Η νέα περίπτωση μπορεί τμηματικά να κατέβει από διάφορα κλαδιά του δένδρου και αντί να καταλήξουμε σε έναν κόμβο-φύλλο καταλήγουμε σε περισσότερους, αλλά με κάποια πιθανότητα. Στο τέλος αθροίζουμε τις πιθανότητες για την κάθε απόφαση.

3. Χαρακτηριστικά με διαφορετικό κόστος

- Σε κάποιες περιπτώσεις μπορεί οι τιμές των χαρακτηριστικών να έχουν κάποιο διαφορετικό κόστος καταγραφής από χαρακτηριστικό σε χαρακτηριστικό.
  - Για παράδειγμα σε ιατρικά δεδομένα το χρηματικό κόστος κάποιων εξετάσεων ή η επίπτωση τους στην υγεία του ασθενή μπορεί να διαφέρουν.
- Σε τέτοιες περιπτώσεις μπορεί να προτιμώνται τα χαρακτηριστικά χαμηλού κόστους έναντι αυτών με το μεγαλύτερο κόστος, εκτός και αν η μεγάλη ακρίβεια πρόβλεψης του δένδρου είναι πολύ σημαντική.
  - Ένα στατιστικό αξιολόγησης χαρακτηριστικών που λαμβάνει υπόψη το κόστος:
  - Διαίρεση του κέρδους πληροφορίας με το κόστος του χαρακτηριστικού ώστε να προτιμώνται περισσότερο αυτά με το μικρότερο κόστος. $$\frac{G(S,A)^2}{Cost(A)}$$
- Πιο εξεζητημένα στατιστικά αξιολόγησης χαρακτηριστικών:
  - Αναγνώριση αντικειμένων από διάφορους αισθητήρες αφής ρομπότ. Το κόστος είναι τα δευτερόλεπτα που χρειάζεται για να ληφθεί μια μέτρηση λόγω της κίνησης του βραχίονα.
  - Ιατρική διάγνωση με βάση εργαστηριακά τεστ με διαφορετικό κόστος. Το w είναι μια σταθερά [0..1] που ορίζει τη σχέση κέρδους και κόστους. $$\frac{2^{G(S,A)}-1}{\left(Cost(A)+1\right)^w}$$

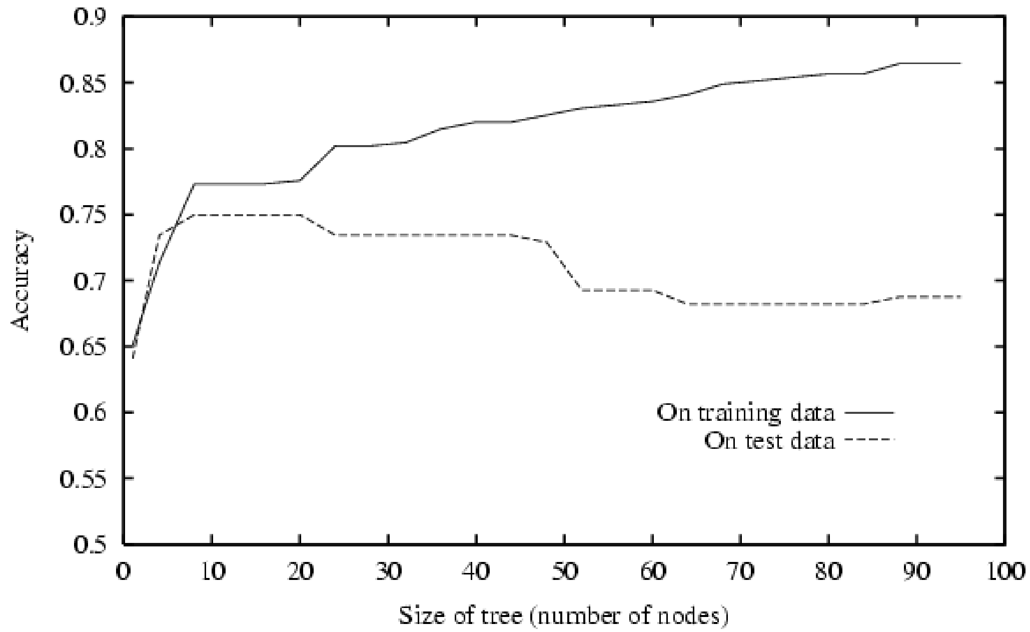
4. Υπερπροσαρμογή ή Υπερμοντελοποίηση των δεδομένων (1/2)

- Σε ένα χώρο υποθέσεων *H*, μια υπόθεση $h \in H$ **υπερπροσαρμόζεται** στα ή ***υπερμοντελοποιεί*** τα (***overfits***) δεδομένα εκπαίδευσης, αν υπάρχει μια άλλη υπόθεση $h' \in H$ με μεγαλύτερο σφάλμα από την *h* στα δεδομένα εκπαίδευσης, αλλά μικρότερο σε όλο το σύνολο των περιπτώσεων.
- Η υπερμοντελοποίηση ή υπερπροσαρμογή είναι ένα πολύ συχνό και σημαντικό πρόβλημα στη μηχανική μάθηση και πάντα κάποιος που ασχολείται με το σχεδιασμό συστημάτων μάθησης θα πρέπει να το διερευνά, ασχέτως αν χρησιμοποιεί δένδρα απόφασης ή κάποια άλλη μέθοδο μάθησης.[3]
- Στον αλγόριθμο ID3, η επέκταση του δένδρου στηρίζεται αποκλειστικά και μόνο στα δεδομένα εκπαίδευσης που υπάρχουν στον εκάστοτε κόμβο. Έτσι όμως ελλοχεύει ο κίνδυνος της υπερπροσαρμογής όταν:
  - Υπάρχει θόρυβος στα δεδομένα εκπαίδευσης.
    - ✓ Θόρυβος είναι τα λάθη είτε στις τιμές των χαρακτηριστικών είτε ακόμα χειρότερα στην τιμή του χαρακτηριστικού πρόβλεψης
  - Τα δεδομένα αυτά δεν αποτελούν ένα αντιπροσωπευτικό δείγμα της έννοιας στόχου.
    - ✓ Αν πολύ λίγα δεδομένα σχετίζονται με έναν κόμβο του δένδρου τότε μπορεί κατά τύχη ένα χαρακτηριστικό που είναι άσχετο με την έννοια στόχο να διαχωρίζει πολύ καλά τα δεδομένα αυτά

[3] Στα Νευρωνικά Δίκτυα: Dropout in Neural Networks

Υπερπροσαρμογή των δεδομένων (2/2)

- Παράδειγμα υπερπροσαρμογής
  - Αξονας x: Αριθμός κόμβων του δένδρου
  - Άξονας y: Ακρίβεια πρόβλεψης
  - Η συνεχόμενη γραμμή εκφράζει την ακρίβεια στα δεδομένα εκπαίδευσης.
    - ✓ Καθώς επεκτείνεται το δένδρο, η ακρίβεια αυξάνεται
  - Η διακεκομμένη γραμμή εκφράζει την ακρίβεια σε ένα ανεξάρτητο σύνολο δεδομένων ελέγχου.
- Μια πειραματική μελέτη του ID3 με 5 θορυβώδη σύνολα δεδομένων έδειξε ότι η υπερπροσαρμογή μείωσε την ακρίβεια των δένδρων απόφασης περίπου 10-25%.



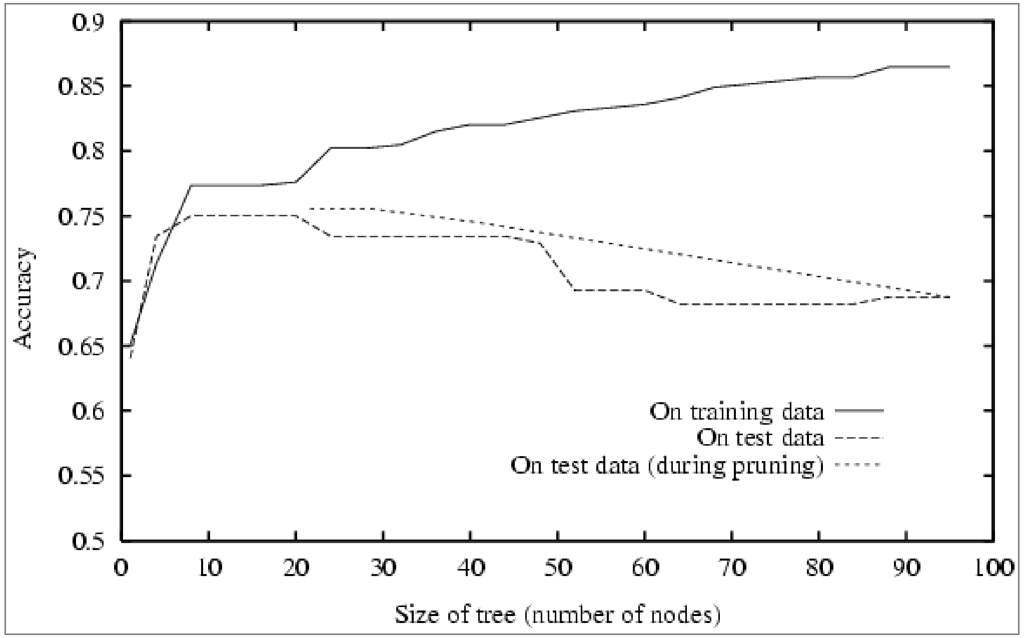
Αποφυγή της υπερπροσαρμογής

- 2 κατηγορίες μεθόδων αποφυγής
  - Αυτές που σταματούν την ανάπτυξη του δένδρου, πριν τις φυσικές συνθήκες τερματισμού, στο σημείο που σταματά να βελτιώνεται η απόδοσή του.
  - Αυτές που το αφήνουν να μεγαλώσει πλήρως και μετά το κλαδεύουν.
    - ✓ Στην πράξη αυτή έχει αποδειχθεί αποτελεσματικότερη
- 2 προσεγγίσεις στην επιλογή του κατάλληλου μεγέθους δένδρου.
  - Χρήση ενός υποσυνόλου των δεδομένων για εκπαίδευση (συνήθως τα 2/3) και των υπόλοιπων δεδομένων (συνήθως το 1/3) για την αξιολόγηση της χρησιμότητας προσθήκης ή κλαδέματος κάποιου κόμβου (*σύνολο επικύρωσης* – *validation set*).
  - Χρήση όλων των δεδομένων για εκπαίδευση, αλλά χρήση ενός στατιστικού τεστ για την αξιολόγηση της χρησιμότητας προσθήκης ή κλαδέματος κάποιου κόμβου.
    - ✓ Ένα γνωστό στατιστικό τεστ που έχει χρησιμοποιηθεί είναι το chi-square τεστ (Quinlan, 1986).
  - Στην πράξη χρησιμοποιείται πιο συχνά η πρώτη προσέγγιση, που ταιριάζει περισσότερο με την δεύτερη μέθοδο αποφυγής υπερπροσαρμογής (αναλύεται στη συνέχεια).
    - ✓ Στον CART ορίζεις το βάθος του δένδρου.

Κλάδεμα μείωσης σφάλματος (1/2)

- Ο αλγόριθμος κλαδέματος μείωσης σφάλματος (*reduced error pruning*) εφαρμόζεται μετά το τέλος της ανάπτυξης του δένδρου και στηρίζεται στη χρήση ενός υποσυνόλου των δεδομένων για *εκπαίδευση* και ενός για *επικύρωση* (*validation*).
- Το κλάδεμα ενός κόμβου συνίσταται στην αφαίρεση όλου του υποδένδρου που κρέμεται από τον κόμβο αυτό.
    - Ο κόμβος μετατρέπεται σε φύλλο και παίρνει ως τιμή (απόφαση) την τιμή του χαρακτηριστικού πρόβλεψης που υπάρχει στην πλειοψηφία των παραδειγμάτων που ανήκουν στο φύλλο αυτό.
- Ο αλγόριθμος κλαδέματος
    - Υπολογίζει το κέρδος στην ακρίβεια πρόβλεψης του δένδρου που προκύπτει από το κλάδεμα του κάθε κόμβου στα δεδομένα επικύρωσης
    - Στη συνέχεια κλαδεύει αυτόν που θα φέρει το μεγαλύτερο κέρδος
    - Η διαδικασία επαναλαμβάνεται μέχρις ότου το περαιτέρω κλάδεμα χειροτερέψει την ακρίβεια του δένδρου στα δεδομένα επικύρωσης
- Έχει το μειονέκτημα ότι λόγω της απαίτησης ξεχωριστού συνόλου δεδομένων επικύρωσης, υπάρχει πρόβλημα σε περιπτώσεις όπου τα δεδομένα είναι λίγα.
- Η χρήση συνόλου επικύρωσης για εντοπισμό φαινομένων υπερπροσαρμογής είναι κάτι συνηθισμένο στη μηχανική μάθηση, όταν τα μοντέλα και οι αλγόριθμοι που δοκιμάζονται έχουν παραμέτρους που απαιτούν ρύθμιση

Κλάδεμα μείωσης σφάλματος (2/2) – Παράδειγμα



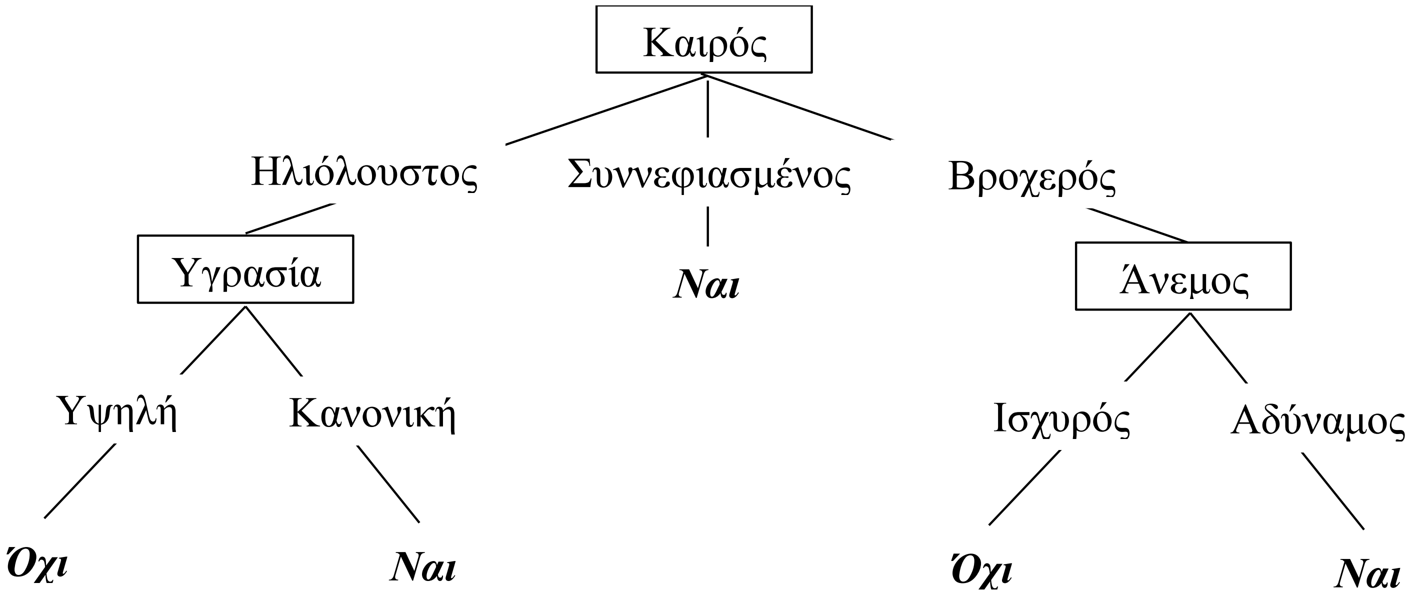
- Άξονας x: Αριθμός κόμβων του δένδρου
- Άξονας y: Ακρίβεια πρόβλεψης
- Η συνεχόμενη καμπύλη εκφράζει την ακρίβεια στα δεδομένα εκπαίδευσης.
  - ✓ Αναμενόμενα, καθώς επεκτείνεται το δένδρο αυξάνεται η ακρίβεια
- Η διακεκομμένη καμπύλη εκφράζει την ακρίβεια σε ένα ανεξάρτητο σύνολο δεδομένων ελέγχου.
- Η νέα γραμμή (διακεκομμένη με τελείες) εκφράζει την ακρίβεια στο σύνολο δεδομένων ελέγχου μετά το κλάδεμα.
  - ✓ Η ακρίβεια ανεβαίνει κατά τη διάρκεια του κλαδέματος (αν δούμε στο σχήμα στον άξονα των x από το 90 προς το 30).
- Η ακρίβεια δεν ανεβαίνει όμοια με το αρχικό δένδρο, καθώς κάθε φορά κλαδεύεται ο πιο προσοδοφόρος κόμβος και όχι αυτός που αναπτύχθηκε τελευταία.
  - ✓ Σημείωση: τα δεδομένα χωρίστηκαν σε 3 σύνολα: *εκπαίδευσης, επικύρωσης, ελέγχου*
  - ✓ Τα δεδομένα *επικύρωσης* (*validation set*) χρησιμοποιούνται για την επιλογή των κόμβων για κλάδεμα ενώ τα δεδομένα *ελέγχου* (*test set*) για την αξιολόγηση του τελικού δένδρου που παράγεται.

Κλάδεμα κανόνων (1/3)

- Ο αλγόριθμος κλαδέματος κανόνων (*rule post-pruning*) είναι ένας πολύ πετυχημένος αλγόριθμος (χρησιμοποιείται από τον C4.5) που εφαρμόζεται μετά το τέλος της ανάπτυξης του δένδρου αλλά δεν απαιτεί τη χρήση υποσυνόλων εκπαίδευσης και επικύρωσης.
- Το κλάδεμα κανόνων αποτελείται από τα ακόλουθα 4 στάδια:
  - Εκπαίδευση του δένδρου μέχρι τα δεδομένα εκπαίδευσης να μοντελοποιηθούν όσο καλύτερα γίνεται, επιτρέποντας την υπερμοντελοποίηση.
  - Μετατροπή του δένδρου σε ένα ισοδύναμο σύνολο κανόνων, μέσω της δημιουργίας ενός κανόνα για κάθε μονοπάτι από τη ρίζα σε φύλλο.
  - Κλάδεμα (γενίκευση) κάθε κανόνα μέσω της αφαίρεσης κάθε συνθήκης του κανόνα που οδηγεί στη βελτίωση της ακρίβειας του.
  - Ταξινόμηση των κανόνων που προκύπτουν κατά φθίνουσα σειρά ακρίβειας και χρήση τους με αυτή τη σειρά για πρόβλεψη νέων δεδομένων.

Κλάδεμα κανόνων (2/3)

- Δένδρο



- Κανόνες για την έννοια "καλή μέρα για τένις"
  - **ΑΝ** (Καιρός=Ηλιόλουστος) **ΚΑΙ** (Υγρασία=Υψηλή) **ΤΟΤΕ** Όχι
  - **ΑΝ** (Καιρός=Ηλιόλουστος) **ΚΑΙ** (Υγρασία=Κανονική) **ΤΟΤΕ** Ναι
  - **ΑΝ** (Καιρός=Συννεφιασμένος) **ΤΟΤΕ** Ναι
  - **ΑΝ** (Καιρός=Βροχερός) **ΚΑΙ** (Άνεμος=Ισχυρός) **ΤΟΤΕ** Όχι
  - **ΑΝ** (Καιρός= Βροχερός) **ΚΑΙ** (Άνεμος =Αδύναμος) **ΤΟΤΕ** Ναι

Κλάδεμα κανόνων (3/3)

- Γιατί να μετατρέπουμε το δένδρο σε κανόνες;
  - Υπάρχει μεγαλύτερη ευελιξία στο κλάδεμα των κανόνων γιατί ο αλγόριθμος μπορεί εναλλακτικά να κλαδέψει οποιαδήποτε συνθήκη ανεξάρτητα από το που βρίσκεται στον κανόνα.
    - ✓ Αντίθετα στο δένδρο όταν κλαδεύεται ένας κόμβος, κλαδεύονται και όλες οι συνθήκες που κρέμονται κάτω από αυτόν.
  - Κάνει πιο εύκολη τη διαδικασία κλαδέματος γιατί δεν χρειάζεται να διατηρηθεί μια δομή δένδρου μετά το κλάδεμα.
    - ✓ Στο κλάδεμα δένδρων αν για παράδειγμα είναι να κλαδευτεί η ρίζα χρειάζεται πολύ δουλειά στην αναδιοργάνωση του δένδρου.
  - Βελτιώνεται η αναγνωσιμότητα της γνώσης που παράγεται.
    - ✓ Οι άνθρωποι καταλαβαίνουν τους κανόνες πολύ πιο εύκολα από ότι ένα σύνθετο δένδρο

5. Χειρισμός δεδομένων με συνεχείς τιμές

- Χαρακτηριστικά με συνεχείς τιμές μπορούν να ληφθούν υπ' όψιν από τον αλγόριθμο με τη δυναμική δημιουργία αντίστοιχων διακριτών χαρακτηριστικών που χωρίζουν τις συνεχείς τιμές σε διακριτά διαστήματα (*διακριτοποίηση* - Discretization).
  - Για ένα συνεχές χαρακτηριστικό A ο αλγόριθμος μπορεί να δημιουργήσει δυναμικά ένα λογικό χαρακτηριστικό A_c, το οποίο θα παίρνει αληθή τιμή αν $A < c$ και ψευδή αλλιώς.
- Πως επιλέγουμε το κατώφλι c;
  - Αυτό που θα μας επιφέρει το μέγιστο κέρδος πληροφορίας αν το χρησιμοποιήσουμε.
  - Το κέρδος πληροφορίας δεν θα το υπολογίσουμε για όλες τις δυνατές τιμές του A, αλλά θα τις ταξινομήσουμε σε αύξουσα σειρά και θα δούμε για ποιες γειτονικές τιμές αλλάζει η τιμή του χαρακτηριστικού πρόβλεψης.
  - Έχει αποδειχθεί ότι το μέγιστο κέρδος πληροφορίας το δίνουν τιμές οι οποίες είναι στο ενδιάμεσο των παραπάνω γειτονικών τιμών.
- Παράδειγμα
  - Στον διπλανό πίνακα με 6 δεδομένα για την θερμοκρασία, ταξινομημένα κατά αύξουσα σειρά, υποψήφιες τιμές είναι α) 54, β) 85

Θερμοκρασία	40	48	60	72	80	90
Τένις	Όχι	Όχι	Ναι	Ναι	Ναι	Όχι

Example: Decision Tree Classification

Data set: Iris Plants Database

Number of Instances: 150 (50 in each of three classes)

Number of Attributes: 4 numeric, predictive attributes and the class

Attributes:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class: (Iris Setosa, Iris Versicolour, Iris Virginica)

Missing Attribute Values: None

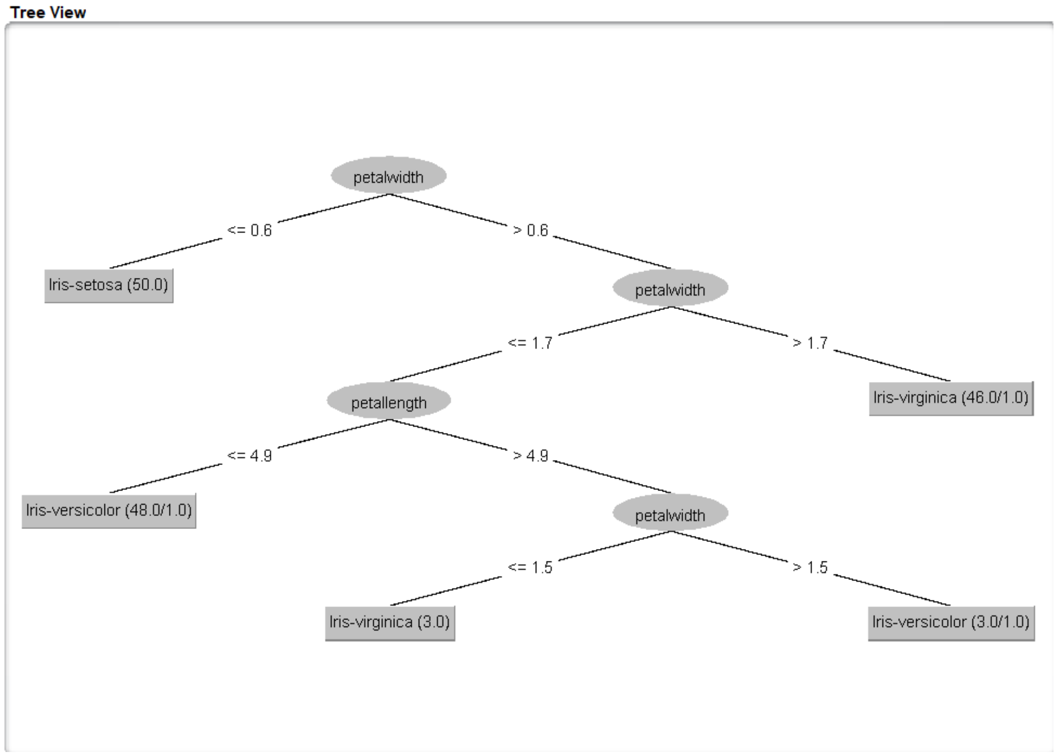
@data
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.8,2.8,5.1,2.4,Iris-virginica
6.4,3.2,5.3,2.3,Iris-virginica
6.5,3.0,5.5,1.8,Iris-virginica
……………………….
………………………….

Weka Algorithm: **J48**

Total Number of Instances 150

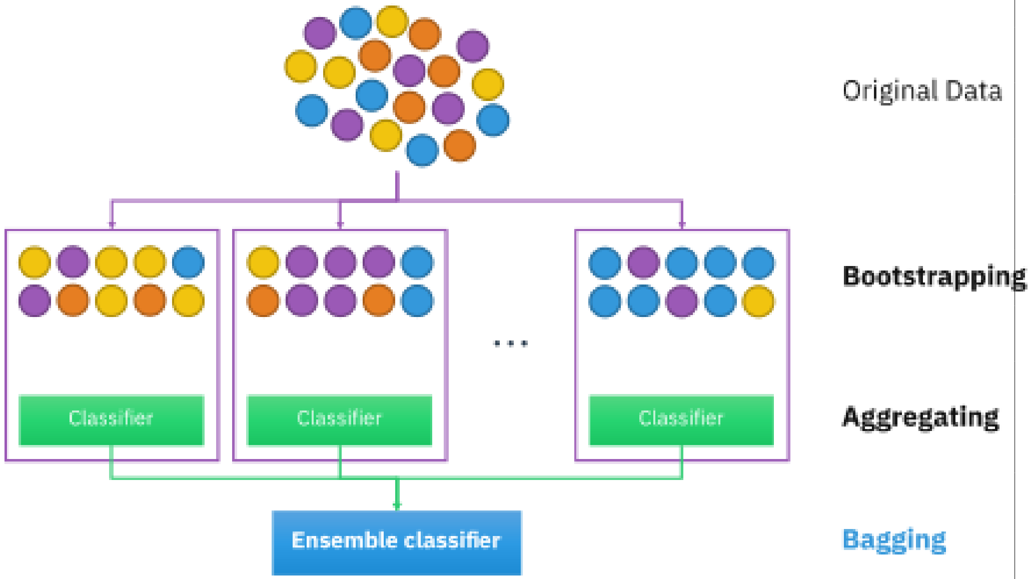
Evaluation Metrics

Correctly Classified Instances	144	96 %
Incorrectly Classified Instances	6	4 %
Precision	1,0	
Recall	0,98	
F-Measure	0,99	



Bootstraping

- Η αξιοπιστία μιας μεθόδου θα μπορούσε να βελτιωθεί, αν μπορούσαμε να εφαρμόσουμε την μέθοδο πάνω σε πολλά διαφορετικά σύνολα δεδομένων, τα οποία να είναι και ασυσχέτιστα μεταξύ τους.
- Επειδή όμως αυτό είναι δύσκολο, ενώ πολλές φορές έχουμε όχι μόνο ένα σύνολο δεδομένων, αλλά και αυτό με λίγα στοιχεία, εφαρμόζουμε την μέθοδο bootstrapping, η οποία προέρχεται από την στατιστική και η οποία επιχειρεί να εξομοιώσει το ζητούμενο αποτέλεσμα.
- Κατά την εκτέλεση της μεθόδου *bootstraping*, ξεκινώντας από ένα αρχικό σύνολο δεδομένων S το οποίο περιέχει n γραμμές (instances/examples), κατασκευάζουμε πολλά νέα σύνολα S1, S2, …, Sm, τα οποία περιέχουν επίσης n γραμμές.

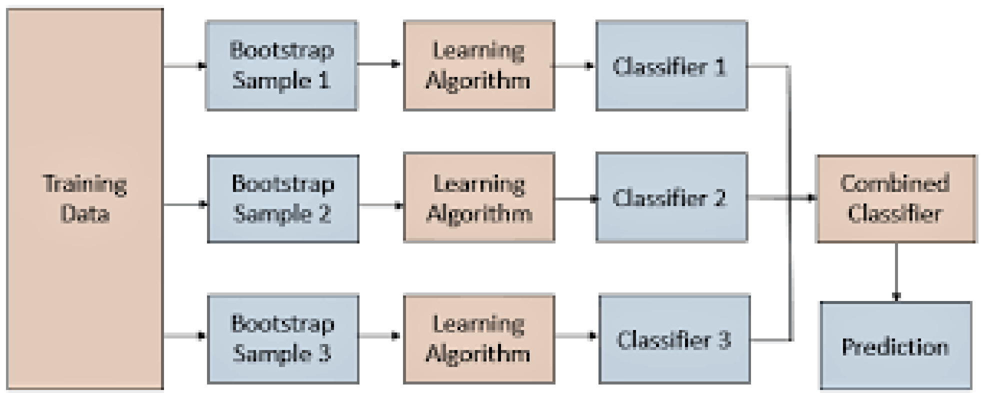


- Τα νέα σύνολα κατασκευάζονται, επιλέγοντας από το αρχικό σύνολο τυχαίες γραμμές, έως ότου το νέο σύνολο να αποκτήσει και αυτό n στοιχεία. Κάθε γραμμή μπορεί να επιλεγεί μια ή και περισσότερες φορές.
- Τα τελικά σύνολα δεδομένων που προκύπτουν δεν είναι πλήρως ασυσχέτιστα μεταξύ τους, μπορούμε όμως να θεωρήσουμε ότι είναι μια ικανοποιητική προσέγγιση.

Bagging (Bootstrap Aggregating)

Ενθυλάκωση ή συνάθροιση αυτοδυναμίας

- Χρησιμοποιώντας την μέθοδο *bootstrapping* δημιουργούνται (τυχαία) πολλά διαφορετικά σύνολα δεδομένων από το αρχικό μέσω "Δειγματοληψίας με Επανατοποθέτηση".
  - Το νέο σύνολο δεδομένων έχει ίδιο αριθμό δεδομένων με το αρχικό, αλλά κάποια δεδομένα έχουν επαναληφθεί ενώ κάποια δεν έχουν συμπεριληφθεί καθόλου.
  - Στη συνέχεια εφαρμόζεται ένας αλγόριθμο μάθησης (π.χ. δένδρα) σε όλα τα νέα σύνολα δεδομένων και παράγονται αντίστοιχα μοντέλα πρόβλεψης.
  - Η μέθοδος αυτή ονομάστηκε Bagging από τα αρχικά των λέξεων Bootstrap Aggregating και δίνει καλύτερα αποτελέσματα από το να εφαρμόσουμε μόνο τον βασικό αλγόριθμο στο αρχικό σύνολο δεδομένων μας.

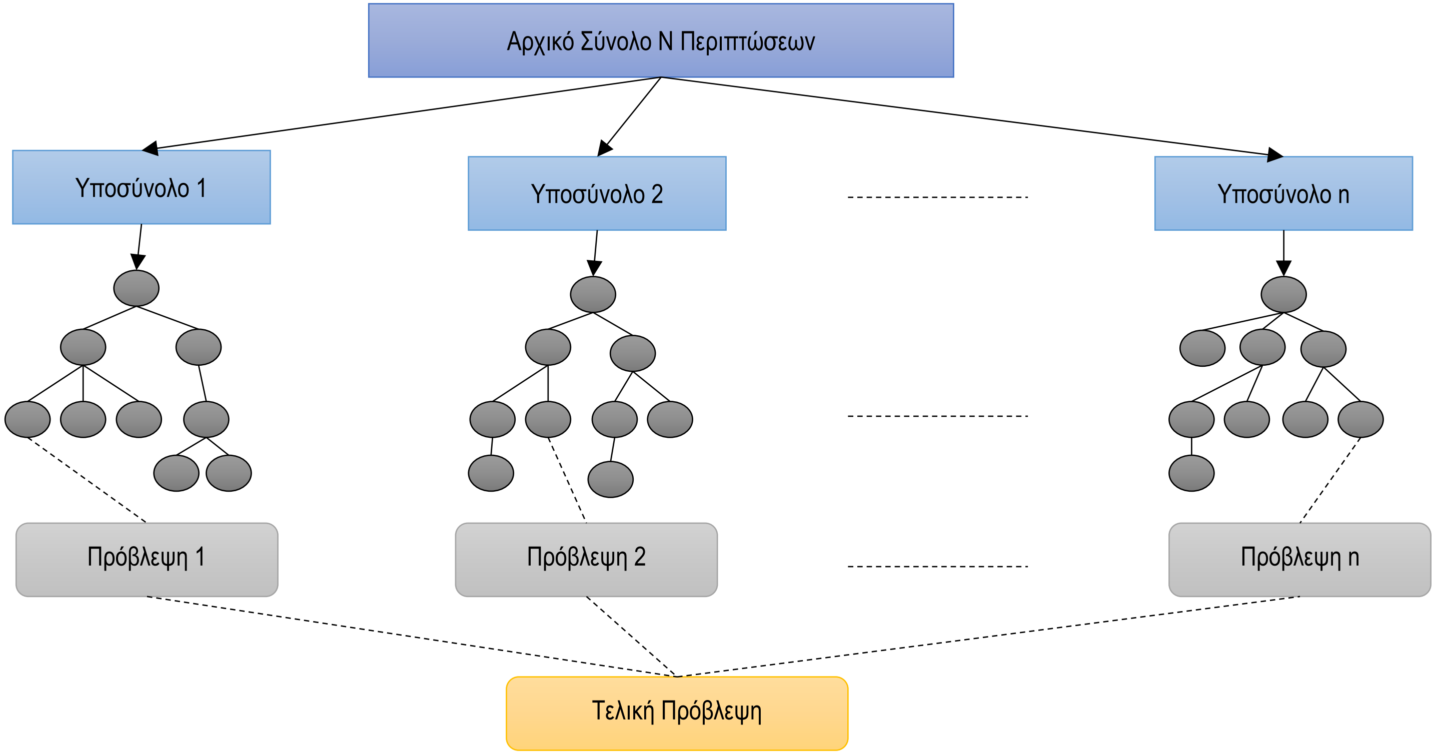


- Για τη διαδικασία πρόβλεψης λαμβάνουμε υπόψη τις αποφάσεις όλων των μοντέλων:
  - Η τελική τιμή είναι είτε η κλάση που συγκεντρώνει τις περισσότερες αποφάσεις μοντέλων (voting) είτε ο μέσος όρος των αριθμητικών προβλέψεων των διαφορετικών μοντέλων.
  - Ο αλγόριθμος που χρησιμοποιούμε ονομάζεται βασικός ταξινομητής (base classifier).
    - ✓ Αυτή η μέθοδος χρησιμοποιείται στον αλγόριθμο τυχαίου δάσους (*random forest*) ο οποίος συνδυάζει πολλά δένδρα απόφασης

Ο Αλγόριθμος Τυχαίου Δάσους - Random (Decision) Forest

- Μέθοδος *συλλογικής μάθησης* (*ensemble learning*) που κατασκευάζει και συνδυάζει πολλά δένδρα
  - Ταξινόμηση και παρεμβολή
  - Ο πρώτος τέτοιος αλγόριθμος προτάθηκε από τον Tin Kam Ho το 1995
- Για N δεδομένα εκπαίδευσης με M χαρακτηριστικά λειτουργεί ως εξής:
  - Από τα N δεδομένα φτιάχνονται *n* υποσύνολα μεγέθους N, με τυχαία επιλογή και επανατοποθέτηση.
    - ✓ ή όπως αλλιώς λέγεται, μέσω δειγματοληψίας με επανατοποθέτηση (sampling with replacement)
    - ✓ Κάθε ένα από αυτά τα υποσύνολα θα δημιουργήσει ένα από τα δένδρα του δάσους
  - Ορίζεται ένας αριθμός m<M που καθορίζει το πλήθος των τυχαίων m χαρακτηριστικών από τα M, που θα ληφθούν υπόψη για την κατασκευή κάθε δένδρου (διατηρείται σταθερός)
  - Κατασκευάζεται το δένδρο για κάθε ένα από τα n υποσύνολα στο μέγιστο βάθος και χωρίς κλάδεμα (Bagging)
  - Το χαρακτηριστικό διαχωρισμού (από τα m) σε κάθε κόμβο αποφασίζεται με κάποιο από τα συνήθη κριτήρια (π.χ. κέρδος πληροφορίας).
- Η πρόβλεψη γίνεται με βάση την επικρατέστερη (πλειοψηφούσα-voting) απόφαση των n δένδρων για την περίπτωση ταξινόμησης ή με βάση τη μέση τιμή της αριθμητικής πρόβλεψης κάθε δένδρου, για την περίπτωση της παρεμβολής

How the random forest algorithm works in machine learning



- Το πλήθος των δένδρων που χρησιμοποιείται μπορεί να είναι της τάξης του 10^2 ή 10^3.
  - Ενδεικτικές τιμές του m για προβλήματα ταξινόμησης με M χαρακτηριστικά είναι $\sqrt{M}$ (στρογγυλοποιημένα προς τα κάτω), ενώ
  - για προβλήματα παρεμβολής προτείνεται να επιλέγονται τα M/3 (στρογγυλοποιημένα προς τα κάτω), με ένα ελάχιστο μέγεθος 5
  - Συνήθως όμως οι τιμές αυτές αποτελούν παραμέτρους που πρέπει να ρυθμιστούν.

Random Forest (συνεχ.)

- Η βασική ιδέα πίσω από τον αλγόριθμο τυχαίου δάσους είναι ότι ένα μεγάλο πλήθος από μη συσχετιζόμενα δένδρα που αποφασίζουν από κοινού (σαν επιτροπή), θα πάρουν καλύτερη απόφαση από κάθε δένδρο ξεχωριστά.
- Για να γίνει αυτό απαιτείται χαμηλή συσχέτιση μεταξύ των επιμέρους μοντέλων (δένδρων) ώστε αυτά να προστατεύονται κατά κάποιο τρόπο μεταξύ τους, για τα λάθη που κάνουν.
  - Ώστε αν κάποια δένδρα υπολογίσουν λάθος πρόβλεψη, πολλά άλλα θα υπολογίσουν τη σωστή και συνολικά η πρόβλεψη θα κινηθεί προς τη σωστή κατεύθυνση.
- Για να εξασφαλιστεί η όσο το δυνατό πιο χαμηλή συσχέτιση μεταξύ των δένδρων (άρα και των μοντέλων που κωδικοποιούν), χρησιμοποιούνται δύο τεχνικές:
  - Κάθε δένδρο χτίζεται πάνω σε ένα δικό του, δειγματοληπτικά παραγόμενο σύνολο δεδομένων, προερχόμενο από το αρχικό σύνολο δεδομένων του προβλήματος.
    - ✓ Δεδομένου ότι τα δένδρα είναι αρκετά ευαίσθητα ως προς το σύνολο εκπαίδευσης, εξασφαλίζεται ποικιλομορφία στα παραγόμενα δένδρα.
    - ✓ Η διαδικασία αυτή ονομάζεται *bagging* (*Bootstrap Aggregating*) (και εδώ ονομάζεται tree bagging)
  - Κάθε δένδρο χτίζεται με βάση ένα δικό του υποσύνολο m χαρακτηριστικών από τα M του συνόλου εκπαίδευσης. Αυτό εισάγει ακόμη μεγαλύτερη ποικιλομορφία μεταξύ των δένδρων και επομένως οδηγεί σε δένδρα χαμηλής συσχέτισης. (*feature bagging*)
- Άρα καταλήγουμε σε δένδρα που όχι μόνο είναι εκπαιδευμένα σε διαφορετικά δεδομένα, αλλά επιπλέον βασίζουν την απόφασή τους σε διαφορετικά χαρακτηριστικά.

Random Forest (σύνοψη)

- Κατατάσσονται στους κορυφαίους αλγορίθμους ταξινόμησης.
  - Ακολουθούν τη φιλοσοφία της *σοφίας του πλήθους* (*wisdom of crowd*) όπου πολλά αδύναμα μοντέλα (weak learners) συνεργαζόμενα δημιουργούν ισχυρά μοντέλα (strong learners).
- Ένα σημαντικό πλεονέκτημά τους είναι ότι
  - Δεν υπερπροσαρμόζουν σε καλά δεδομένα (μείωση της διακύμανσης χωρίς αύξηση της μεροληψίας) ενώ
  - Παράλληλα μπορούν να χειριστούν μεγάλα σύνολα εκπαίδευσης με πολλά χαρακτηριστικά.
- Η κατασκευή τους δεν είναι υπολογιστικά πολύπλοκη.
- Στα μειονεκτήματα, συγκαταλέγεται
  - Η ασυνέχεια στις τιμές πρόβλεψης σε προβλήματα παρεμβολής, κάτι αναμενόμενο δεδομένου του τρόπου που δημιουργούνται τα δένδρα και
  - Ο κίνδυνος υπερπροσαρμογής σε δεδομένα με πολύ θόρυβο και μικρό αριθμό χαρακτηριστικών (στηλών).

Example: Decision Tree Classification

The Pima Indians dataset is well-known among beginners to machine learning because it is a binary classification problem and has nice, clean data. The simplicity made it an attractive option

The Pima Indian population are based near Phoenix, Arizona (USA). They have been heavily studied since 1965 on account of high rates of diabetes.

Data set: Pima Indians Diabetes

This dataset contains measurements for 768 female subjects, all aged 21 years and above

Attributes

- preg - the number of times the subject had been pregnant
- plan - the concentration of blood plasma glucose (two hours after drinking a glucose solution)
- pres - diastolic blood pressure in mmHg
- skin - triceps skin fold thickness in mm
- insu - serum insulin (two hours after drinking glucose solution)
- mass - body mass index ((weight/height)**2)
- pedi - 'diabetes pedigree function' (a measurement I didn't quite understand but it relates to the extent to which an individual has some kind of hereditary or genetic risk of diabetes higher than the norm)
- age - in years
- class (tested_negative, tested-possitive)

@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
……………………….
………………………….

Weka Algorithm: **J48**

Total Number of Instances 286

Evaluation Metrics

Correctly Classified Instances	567	73.828 %
Incorrectly Classified Instances	201	26.171%
Precision	0,79	
Recall	0,814	
F-Measure	0,802	

Weka Algorithm: **Random Forest**
Total Number of Instances 286

Evaluation Metrics

Correctly Classified Instances	582	75.781%
Incorrectly Classified Instances	186	24.218 %
Precision	0,801	
Recall	0,836	
F-Measure	0,818	

Weka Algorithm: **LMT (Logistic model tree)**
Total Number of Instances 286

Evaluation Metrics

Correctly Classified Instances	595	77.474%
Incorrectly Classified Instances	173	22.526%
Precision	0,790	
Recall	0,890	
F-Measure	0,837	

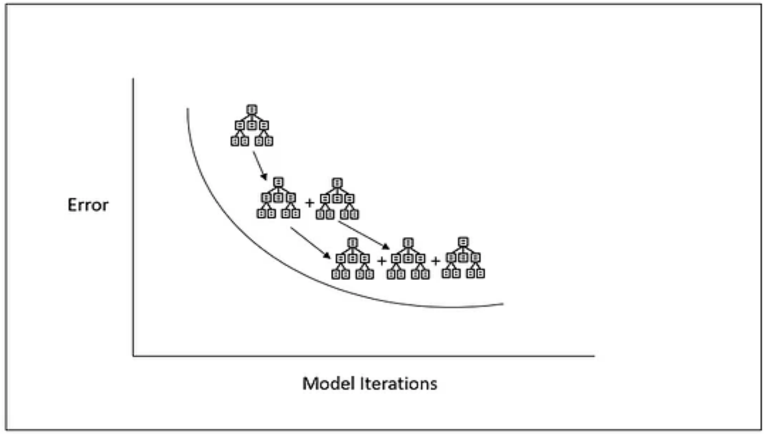
Logistic model trees (LMT), are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values.

Python (Decision Trees, Random Forest):

- sklearn uses a modified version of the CART algorithm
  - Συνεπώς και ο Random Forest αποτελείται από τέτοια δένδρα
  - Άρα υποστηρίζει μόνο αριθμητικά χαρακτηριστικά.
  - Μετατροπή των κατηγορικών σε αριθμητικά: How to fit categorical data types for random forest classification?
- Python calls (sklearn):
  - Classification:
    - ✓ sklearn.tree.**DecisionTreeClassifier**(criterion={‘gini’,‘entropy’}, max_depth, min_samples_split)
    - ✓ sklearn.ensemble.**RandomForestClassifier**(n_estimators=*n*, criterion={‘gini’, ‘entropy’}, max_depth, min_samples_split)
  - Regression:
    - ✓ sklearn.tree.**DecisionTreeRegressor**(criterion={‘squared_error’,‘absolute_error’},max_depth,min_samples_split)
    - ✓ sklearn.ensemble.**RandomForestRegressor**(n_estimators, criterion={‘squared_error’, ‘absolute_error’}, max_depth, min_samples_split)
- Visualization of a Decision Tree can be performed in two ways:
  - Using sklearn function sklearn.tree.plot_tree(dt_model)
  - Using GraphViz library with sklearn:
    1. graphviz_model = sklearn.tree.export_graphviz(dt_model, out_file=None, feature_names={feature_names}, class_names={target_names}, filled={False, True}, rounded={False, True}, special_characters={False, True})
    2. dt_graph = graphviz.Source(graphviz_model)
    3. dt_graph.render("output_file_name")
    4. Results are saved as a PDF file with the filename given above.

Gradient Boosting (Διαβαθμισμένη ενίσχυση)

- Gradient descent optimization in the machine learning world is typically used to find the parameters associated with a single model that optimizes some loss function.
- In contrast, gradient boosters are meta-models consisting of multiple weak models whose output is added together to get an overall prediction.
- The gradient descent optimization occurs on the output of the model and not the parameters of the weak models.
- In the figure we can see that gradient boosting adds sub-models incrementally to minimize a loss function.



- The boosting ensemble technique consists of three simple steps:
  - ✓ An initial model F0 is defined to predict the target variable y. This model will be associated with a residual (y – F0)
  - ✓ A new model $\mathbf{h_1}$ is fit to the residuals from the previous step
  - ✓ Now, $\mathbf{F_0}$ and $\mathbf{h_1}$ are combined to give $\mathbf{F_1}$, the boosted version of F0. The mean squared error from F_1 will be lower than that from F_0:

$$F_1(x) <- F_0(x) + h_1(x)$$

  - ✓ To improve the performance of $\mathbf{F_1}$, we could model after the residuals of $\mathbf{F_1}$ and create a new model $\mathbf{F_2}$:

$$F_2(x) <- F_1(x) + h_2(x)$$

  - ✓ This can be done for 'm' iterations, until residuals have been minimized as much as possible: $F_m(x) <- F_{m-1}(x) + h_m(x)$
- Here, the additive learners do not disturb the functions created in the previous steps. Instead, they impart information of their own to bring down the errors.

ΠΑΡΑΡΤΗΜΑ Α

ΑΣΚΗΣΕΙΣ - ΠΑΡΑΔΕΙΓΜΑΤΑ

Άσκηση 4.1

- Υπολογίστε με βάση τα δεδομένα του πίνακα
  - Την εντροπία σε σχέση με την κατηγορία (+, -)
  - Το κέρδος πληροφορίας αν χωρίσουμε τα δεδομένα με βάση το χαρακτηριστικό $α_2$

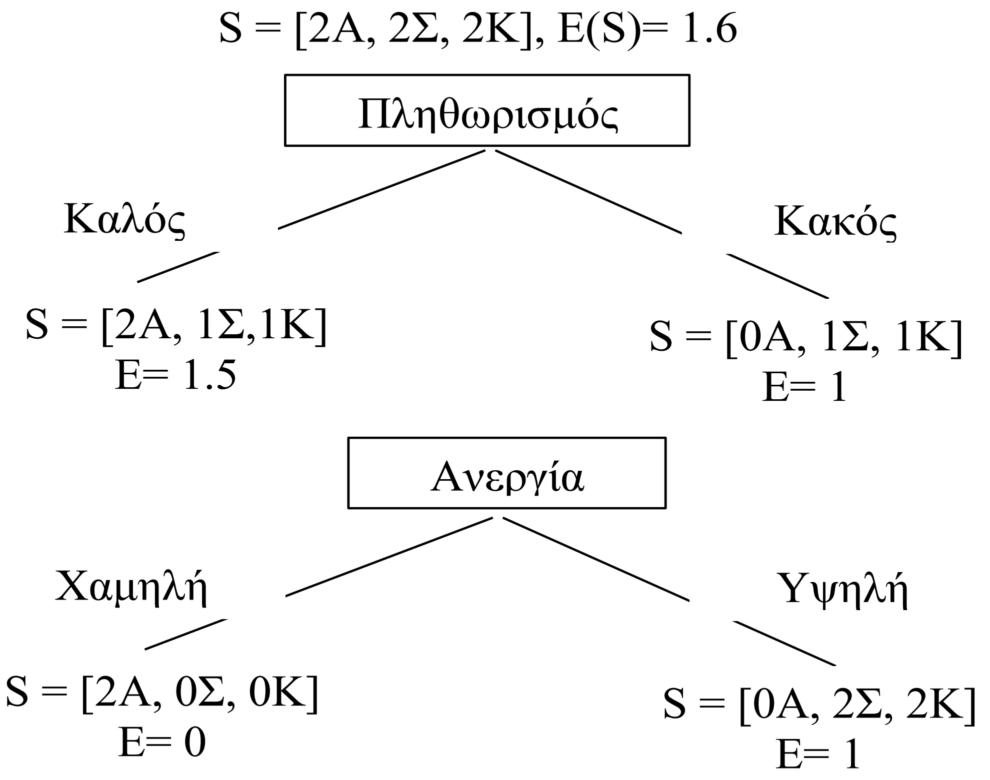
Περίπτωση	Κατηγορία	$α_1$	$α_2$
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Άσκηση 4.2

- Έστω το σύνολο δεδομένων που δίνεται στον πίνακα.
- Με βάση ποιο χαρακτηριστικό θα γίνει ο διαχωρισμός στη ρίζα, αν εφαρμόσουμε τον αλγόριθμο ID3;
- Εξαρτημένη μεταβλητή: Πορεία Μετοχής
  - Δίνεται: log2(1/3)=-1.6, log2(1/2)=-1, log2(1/4)= -2

Πληθωρισμός	**Ανεργία**	**Πορεία Μετοχής**
Καλός	Χαμηλή	Ανοδική
Κακός	Υψηλή	Σταθερή
Κακός	Υψηλή	Καθοδική
Καλός	Υψηλή	Σταθερή
Καλός	Υψηλή	Καθοδική
Καλός	Χαμηλή	Ανοδική

Άσκηση 2 - Απάντηση



E(S) = 3*(-2/6 log2(2/6))=1.6

E(S|Πληθ.=Καλός)=-2/4 log2(2/4) – 2*(-1/4 log2(1/4))=1.5

E(S|Πληθ.=Κακός)=0 – 2*(-1/2 log2(1/2))

E(S|Ανεργία=Χαμηλή)=… **E(S|Ανεργία=Υψηλή)=**…

- Κέρδος πληροφορίας
  - G(S, Πληθωρισμός) = 1.6-(4/6)*1.5-(2/6)*1 = 0.267
  - G(S, Ανεργία) = 1.6-(2/6)*0-(4/6)*1 = **0.933**

Άσκηση 4.3

- Έστω το παρακάτω σύνολο δεδομένων. Η εντροπία των παραδειγμάτων με τιμή *Χαμηλή* για το χαρακτηριστικό *Ανεργία* είναι 0 και η εντροπία των παραδειγμάτων με τιμή *Κακός* για το χαρακτηριστικό *Πληθωρισμός* είναι 1.
- Επίσης, είναι γνωστό ότι ο αλγόριθμος ID3, βρίσκει την *Ανεργία* να είναι καταλληλότερο χαρακτηριστικό από τον *Πληθωρισμό* για το διαχωρισμό στη ρίζα.
- Οι τιμές που μπορεί να πάρει η εξαρτημένη μεταβλητή (*Πορεία Μετοχής*) είναι *Καθοδική* και *Ανοδική*.
- Συμπληρώστε τις τιμές που λείπουν από τον πίνακα, αιτιολογώντας την απάντησή σας.

	Πληθωρισμός	**Ανεργία**	**Πορεία Μετοχής**
1	Καλός	Χαμηλή	
2	Καλός	Υψηλή	
3	Κακός	Χαμηλή	Καθοδική
4	Κακός	Υψηλή	

Άσκηση 3 - Απάντηση

- ✓ Αφού η εντροπία των παραδειγμάτων με τιμή *Χαμηλή* για το χαρακτηριστικό *Ανεργία* είναι 0 άρα η τιμής της Μετοχής και για την άλλη τιμή της Ανεργία=Χαμηλή, δηλ. η (β) θα είναι πάλι Καθοδική

	Πληθωρισμός	Ανεργία	Πορεία Μετοχής
1	Καλός	Χαμηλή	(β) Καθοδική
2	Καλός	Υψηλή	
3	Κακός	Χαμηλή	(α) Καθοδική
4	Κακός	Υψηλή	

- ✓ Αφού η εντροπία των παραδειγμάτων με τιμή Κακός για το χαρακτηριστικό Πληθωρισμός είναι 1, σημαίνει ότι θα έχουν διαφορετική τιμή στην πορεία της Μετοχής και αφού η μια τιμή, η (α) είναι Καθοδική, η άλλη (γ) θα είναι Ανοδική.

	Πληθωρισμός	Ανεργία	Πορεία Μετοχής
1	Καλός	Χαμηλή	(β) Καθοδική
2	Καλός	Υψηλή	
3	Κακός	Χαμηλή	(α) Καθοδική
4	Κακός	Υψηλή	(γ) Ανοδική

- ✓ Αφού η Ανεργία είναι καταλληλότερο χαρακτηριστικό από τον Πληθωρισμό για το διαχωρισμό στη ρίζα, άρα η τιμή της Μετοχής και για την άλλη τιμή της Ανεργία=Υψηλή, δηλ. η (δ) θα είναι πάλι Ανοδική ώστε η εντροπία της Ανεργία=Υψηλή να είναι πάλι μηδέν όπως και της Ανεργία=Χαμηλή.
- ✓ Έτσι η εντροπία του Πληθωρισμός=Καλός και =Κακός θα είναι 1 και προφανώς θα επιλεγεί το χαρακτηριστικό Ανεργία για το διαχωρισμό στη ρίζα.

	Πληθωρισμός	Ανεργία	Πορεία Μετοχής
1	Καλός	Χαμηλή	(β) Καθοδική
2	Καλός	Υψηλή	(δ) Ανοδική
3	Κακός	Χαμηλή	(α) Καθοδική
4	Κακός	Υψηλή	(γ) Ανοδική

Άσκηση 4.4[4]

Χρησιμοποιήστε τα δεδομένα του Πίνακα και υπολογίστε το Κέρδος Πληροφορίας σε περίπτωση που επιλεγεί το πεδίο «Εισόδημα» ως μεταβλητή διαχωρισμού για τη δημιουργία Δένδρου Αποφάσεων ID3.

ΕΙΣΟΔΗΜΑ	ΗΛΙΚΙΑ	ΕΓΚΡΙΣΗ
ΥΨΗΛΟ	ΜΕΓΑΛΗ	No
ΥΨΗΛΟ	ΜΕΓΑΛΗ	No
ΥΨΗΛΟ	ΜΕΣΑΙΑ	Yes
ΜΕΣΟ	ΜΕΣΑΙΑ	Yes
ΧΑΜΗΛΟ	ΜΙΚΡΗ	Yes
ΧΑΜΗΛΟ	ΜΕΓΑΛΗ	No
ΧΑΜΗΛΟ	ΜΙΚΡΗ	Yes
ΜΕΣΟ	ΜΕΓΑΛΗ	No
ΧΑΜΗΛΟ	ΜΙΚΡΗ	Yes
ΜΕΣΟ	ΜΙΚΡΗ	Yes
ΜΕΣΟ	ΜΙΚΡΗ	Yes
ΜΕΣΟ	ΜΕΣΑΙΑ	Yes
ΥΨΗΛΟ	ΜΕΣΑΙΑ	Yes
ΜΕΣΟ	ΜΕΓΑΛΗ	No

[4] Απόσπασμα από το βιβλίο: Επιχειρηματική Ευφυΐα & Εξόρυξη Δεδομένων, Ευστάθιος Γ. Κύρκος, ISBN: 978-960-603-109-0, Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράματα και Βοηθήματα, www.kallipos.gr, 2015

Άσκηση 4 - Απάντηση

Αρχικά πρέπει να υπολογιστεί η Εντροπία του συνόλου E(S). Θεωρούμε ως θετική κλάση την έγκριση του δανείου. Στο σύνολο δεδομένων υπάρχουν εννέα θετικές και πέντε αρνητικές παρατηρήσεις. Η Εντροπία υπολογίζεται ως εξής:

E(S)= - (9/14)*log2(9/14) – (5/14)*log2(5/14) = 0,94.

Εάν επιλεγεί το Εισόδημα ως μεταβλητή διαχωρισμού, τότε το σύνολο δεδομένων θα διαχωριστεί σε τρία υποσύνολα, όπου στο πρώτο υποσύνολο S1 θα περιλαμβάνονται οι υποψήφιοι με χαμηλό εισόδημα, στο δεύτερο υποσύνολο S2 οι υποψήφιοι με υψηλό εισόδημα και στο τρίτο υποσύνολο S3 οι υποψήφιοι με μεσαίο εισόδημα. Αρχικά πρέπει να υπολογιστούν οι Εντροπίες των τριών υποσυνόλων.

Το υποσύνολο S1 περιέχει τρεις θετικές και μια αρνητική παρατήρηση. Η Εντροπία του υπολογίζεται ως εξής:

E(S1) = -(3/4)*log2(3/4) – (1/4)*log2(1/4) = 0,811.

Το υποσύνολο S2 περιέχει δύο θετικές και δύο αρνητικές παρατηρήσεις. Η Εντροπία του υπολογίζεται ως εξής:

E(S2) = -(2/4)*log2(2/4) – (2/4)*log2(2/4) = 1.

Το υποσύνολο S3 περιέχει τέσσερεις θετικές και δύο αρνητικές παρατηρήσεις. Η Εντροπία του υπολογίζεται ως εξής:

E(S3) = -(4/6)*log2(4/6) – (4/6)*log2(4/6) = 0,918.

Το υποσύνολο S1 περιέχει τέσσερεις παρατηρήσεις, το υποσύνολο S2 περιέχει τέσσερεις παρατηρήσεις, το υποσύνολο S3 περιέχει έξι παρατηρήσεις, και το αρχικό σύνολο περιέχει δέκα τέσσερεις παρατηρήσεις. Η Εντροπία διαχωρισμού θα υπολογιστεί ως εξής:

E(S, Εισόδημα) = (4/14)*E(S1) + (4/14)*E(S2) + (6/14)*E(S3) = 0,911.

Το Κέρδος Πληροφορίας είναι: G(S, Εισόδημα) = E(S) - E(S, Εισόδημα) = 0,94 – 0,911 = 0,029.

Άσκηση 4.5

- Έστω ένα πρόβλημα κατασκευής ενός δένδρου απόφασης με βάση το σύνολο S των εγγραφών του παρακάτω πίνακα, στον οποίο καταγράφεται το αν έγινε ένας αθλητικός αγώνας σε σχέση με τις συνθήκες υγρασίας, ανέμου και θερμοκρασίας που επικρατούσαν.

Ημέρα	**Υγρασία**	**Άνεμος**	**Θερμοκρασία**	**Έγινε Αγώνας?**
H_1	υψηλή	ασθενής	υψηλή	όχι
H_2	υψηλή	ισχυρός	υψηλή	όχι
H_3	υψηλή	ασθενής	μέση	όχι
H_4	κανονική	ασθενής	χαμηλή	ναι
H_5	κανονική	ισχυρός	μέση	ναι

α) Χρησιμοποιώντας τα μεγέθη Εντροπία και Κέρδος και θεωρώντας ως εξαρτημένη μεταβλητή το πεδίο "Έγινε Αγώνας", να αποφασιστεί ποιο από τα πεδία υγρασία, άνεμος και θερμοκρασία είναι καταλληλότερο για τον επόμενο διαχωρισμό.

β) Να κατασκευαστεί το πλήρες δένδρο ταξινόμησης (απόφασης).

Άσκηση 4.6

- Έστω το παρακάτω σύνολο δεδομένων. Με βάση ποιο χαρακτηριστικό θα γίνει ο διαχωρισμός στη ρίζα, αν εφαρμόσουμε τον αλγόριθμο ID3;
  - Για τη διακριτοποίηση του συνεχούς χαρακτηριστικού επιλέξτε το κατάλληλο κατώφλι c.
  - Δίνονται όλοι οι λογάριθμοι που απαιτούνται.

Καιρός	**Θερμοκρασία**	**Υγρασία**	**Τένις**
Ηλιόλουστος	30	Χαμηλή	Όχι
Συννεφιασμένος	10	Υψηλή	Ναι
Ηλιόλουστος	20	Χαμηλή	Ναι
Βροχερός	5	Υψηλή	Όχι
Ηλιόλουστος	15	Χαμηλή	Ναι
Συννεφιασμένος	7	Υψηλή	Ναι
Συννεφιασμένος	2	Χαμηλή	Όχι
Βροχερός	29	Υψηλή	Όχι

Άσκηση 4.7

- Έστω το παρακάτω σύνολο δεδομένων.
  - Με βάση ποιο χαρακτηριστικό (Φύλο ή Ύψος) θα γίνει ο διαχωρισμός στη ρίζα, αν εφαρμόσουμε τον αλγόριθμο ID3;
  - Για τη διακριτοποίηση του συνεχούς χαρακτηριστικού επιλέξτε το κατάλληλο(α) κατώφλι(α) με τη μέθοδο του διαχωρισμού ίσης συχνότητας.

ID	Φύλλο	Ύψος	Χαρακτηρισμός
1	Θ	1.6	Κοντός/ή
2	Θ	1.6	Κοντός/ή
3	Θ	1.7	Κοντός/ή
4	Α	1.7	Κοντός/ή
5	Θ	1.75	Μέτριος/α
6	Θ	1.8	Μέτριος/α
7	Θ	1.8	Μέτριος/α
8	Α	1.85	Μέτριος/α
9	Θ	1.88	Μέτριος/α
10	Θ	1.9	Μέτριος/α
11	Θ	1.9	Μέτριος/α
12	Α	1.95	Μέτριος/α
13	Α	2.0	Ψηλός/ή
14	Α	2.1	Ψηλός/ή
15	Α	2.2	Ψηλός/ή

- ✓ Για τη διακριτοποίηση σε 3 διαστήματα θα μπορούσαν να οριστούν τα διαστήματα 1.6-1.75 με 5 παραδείγματα, 1.8-1.9 με 6 παραδείγματα και 1.95-2.2 με 4 παραδείγματα

Άσκηση - (Homework)

- Να χρησιμοποιήσετε το dataset Breast cancer και να εφαρμόσετε έναν classification tree αλγόριθμο δοκιμάζοντας διάφορες τιμές των παραμέτρων: split function (criterion) και maxdepth.
- Επίσης θα εφαρμόσετε τον random forest με τιμές παραμέτρων split function (criterion) και number of trees (n_estimators).
- Τα αποτελέσματα θα εμφανίζονται σε έναν πίνακα όπου να αναγράφονται οι μετρικές *Precision*, *Recall* και *F1* για κάθε περίπτωση.
- Στο τέλος να δώστε μια σύντομη παράγραφο σχολιασμού των αποτελεσμάτων.

α/α	Algorthm	Criterion	Maxdepth	Precision	Recall	F1
1						

α/α	Random Forest	Criterion	N_etsimators	Precision	Recall	F1
1						

ΠΑΡΑΡΤΗΜΑ Β

Συμπληρωματικό Υλικό

weka.classifiers.trees

DecisionStump

Class for building and using a decision stump. Usually used in conjunction with a boosting algorithm. Does regression (based on mean-squared error) or classification (based on entropy). Missing is treated as a separate value.

A decision stump is a machine learning model consisting of a one-level decision tree. That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. Sometimes they are also called 1-rules.

Hoeffding Tree

A Hoeffding tree (VFDT) is an incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams, assuming that the distribution generating examples does not change over time. Hoeffding trees exploit the fact that a small sample can often be enough to choose an optimal splitting attribute. This idea is supported mathematically by the Hoeffding bound, which quantifies the number of observations (in our case, examples) needed to estimate some statistics within a prescribed precision (in our case, the goodness of an attribute).

A theoretically appealing feature of Hoeffding Trees not shared by other incremental decision tree learners is that it has sound guarantees of performance. Using the Hoeffding bound one can show that its output is asymptotically nearly identical to that of a non-incremental learner using infinitely many examples. For more information, see:

Geoff Hulten, Laurie Spencer, Pedro Domingos: Mining time-changing data streams. In: ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 97-106, 2001.

J48

Class for generating a pruned or unpruned C4.5 decision tree.

For more information, see: Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

LMT

Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values.

For more information see:

Niels Landwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees. Machine Learning. 95(1-2):161-205.

REPTree

Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5).

RandomTree

Class for constructing a tree that considers K randomly chosen attributes at each node. Performs no pruning. Also has an option to allow estimation of class probabilities (or target mean in the regression case) based on a hold-out set (backfitting).

RandomForest

Class for constructing a forest of random trees.

For more information see: Leo Breiman (2001). Random Forests. Machine Learning. 45(1):5-32.

- In Weka 3.7.11, RandomForest is using (bagging) Weka's RandomTree.
- WEKA's RandomForest is not based on CART, but it is also not based on J48, rather a variant of REPTree modified to be include the desired randomness, and not pruned.