# Machine Learning & Knowledge Extraction

DR KONSTANTINOS KARAMPIDIS

# Πληροφορίες Μαθήματος

- Ωράριο:
  - Θεωρία: **Κάθε Τρίτη 09:00-13:00 – Αίθουσα 207**
  - Εργαστήριο: **13:00-14:00 ΕΡΓ6 (σύμφωνα με το πρόγραμμα που είναι αναρτημένο στο eclass)**
- Εργασίες
  - **1 project – Ομάδες έως 2 ατόμων – 80%**
  - **Εργαστηριακές ασκήσεις – 20%**
- Προαπαιτούμενα: Κανένα

# Περιεχόμενο Μαθήματος

- Εισαγωγή στη Μηχανική Μάθηση - τι είναι, γιατί μας ενδιαφέρει, παραδείγματα προβλημάτων, η μηχανική μάθηση ως αναζήτηση, υπόθεση επαγωγικής μάθησης
- Επεξεργασία εισόδου – Μείωση διαστατικότητας- Αξιόλογηση
- **Μέθοδοι επιβλεπόμενης μάθησης**
- Νευρωνικά Δίκτυα
- Εξελικτική Μάθηση – Γενετικοί Αλγόριθμοι
- Μηχανική Μάθηση Βασιζόμενη σε Κανόνες
- Ενισχυτική Μάθηση
- Μάθηση Αναπαράστασης
- Εξόρυξη Δεδομένων

# Τύποι ταξινόμησης

**Δυαδική ταξινόμηση – Binary Classification**

Σε μια εργασία δυαδικής ταξινόμησης, ο στόχος είναι να ταξινομηθούν τα δεδομένα εισόδου σε δύο κατηγορίες.

Τα δεδομένα εκπαίδευσης σε μια τέτοια περίπτωση επισημαίνονται σε δυαδική μορφή: αληθές και ψευδές, θετικό και αρνητικό, 0 και 1, spam και μη spam, κ.λπ. ανάλογα με το πρόβλημα που αντιμετωπίζεται.

Για παράδειγμα, μπορεί να θέλουμε να ανιχνεύσουμε αν μια δεδομένη εικόνα είναι φορτηγό ή σκάφος.

**Ταξινόμηση πολλαπλών κατηγοριών – Multiclass Classification**

Σε μια ταξινόμηση πολλαπλών κατηγοριών , ο στόχος είναι να ταξινομηθούν τα δεδομένα εισόδου σε μία από τις διαθέσιμες κατηγορίες (περισσότερες από δύο)

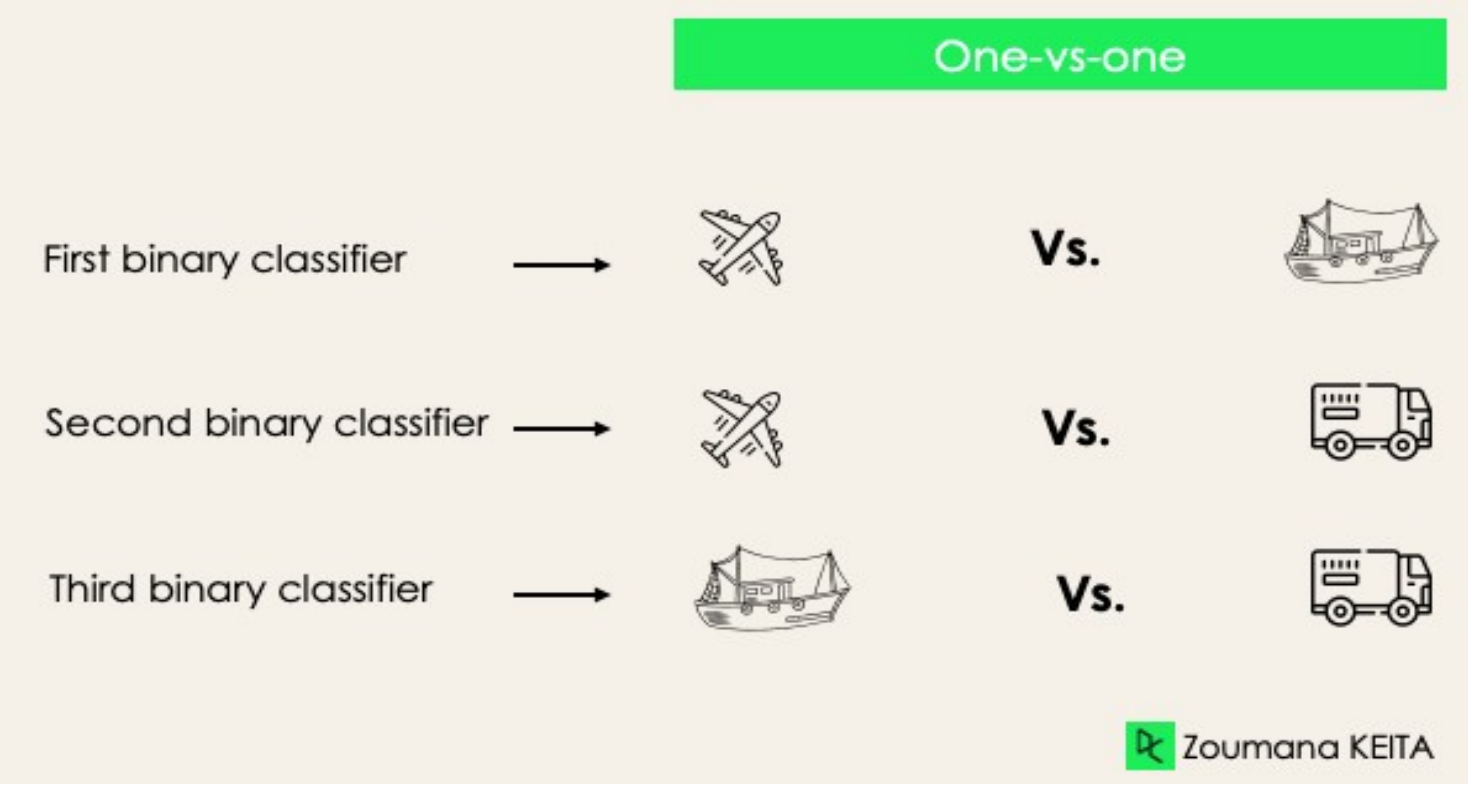
# Τύποι ταξινόμησης

Οι περισσότεροι αλγόριθμοι δυαδικής ταξινόμησης μπορούν να χρησιμοποιηθούν και σε προβλήματα ταξινόμησης πολλών κατηγοριών (multi-class).

Στην περίπτωση που κάποιος αλγόριθμος δεν υποστηρίζει multiclass classification, εφαρμόζονται προσεγγίσεις δυαδικού μετασχηματισμού, όπως οι προσεγγίσεις one-versus-one και one-versus-all.

# Τύποι ταξινόμησης

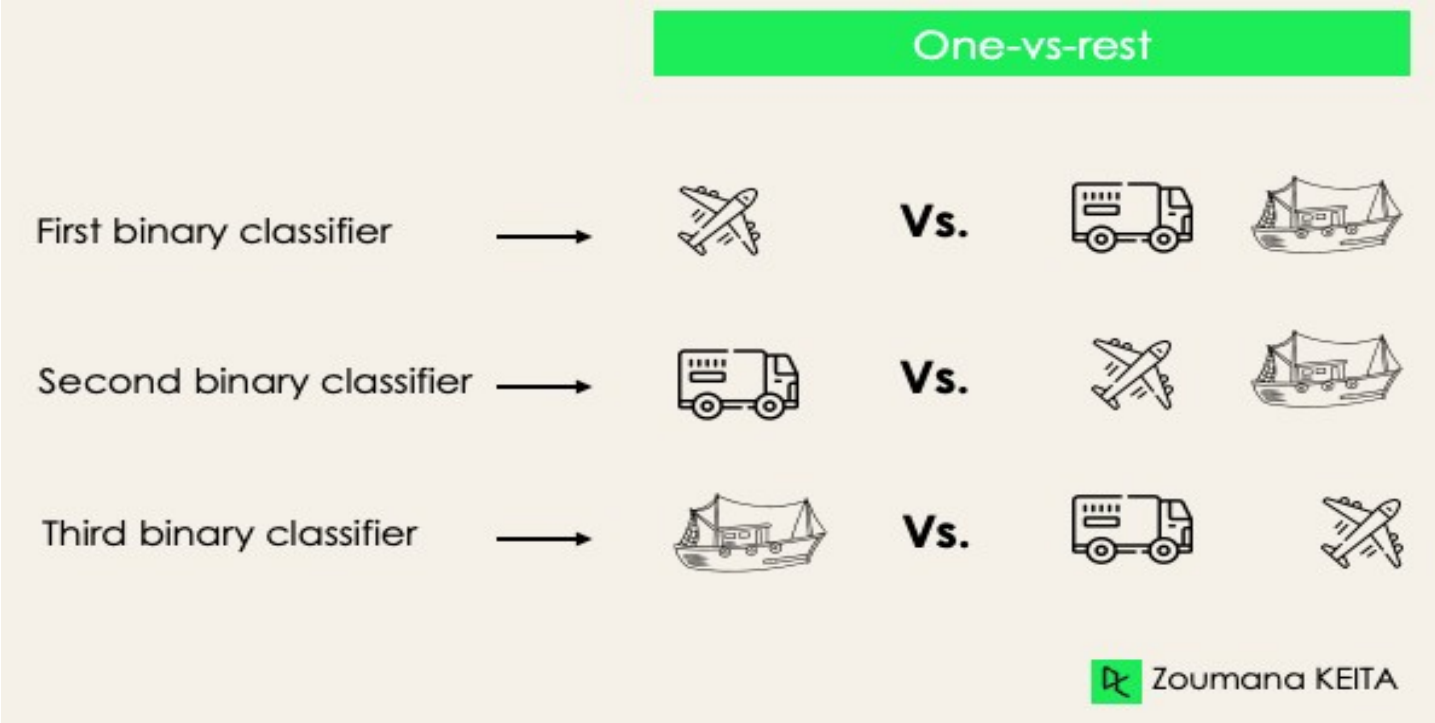
one-versus-one : αυτή η στρατηγική εκπαιδεύει τόσους ταξινομητές όσα είναι τα ζεύγη ετικετών. Εάν έχουμε μια ταξινόμηση 3 κλάσεων, θα έχουμε τρία ζεύγη ετικετών, άρα τρεις ταξινομητές.




Source: https://www.datacamp.com/blog/classification-machine-learning

# Τύποι ταξινόμησης

one-versus-all : σε αυτό το στάδιο, ξεκινάμε θεωρώντας κάθε κλάση ως ανεξάρτητη και θεωρούμε τις υπόλοιπες συνδυασμένες ως μία μόνο ετικέτα. Με 3 κλάσεις, θα έχουμε τρεις ταξινομητές.




Source: https://www.datacamp.com/blog/classification-machine-learning

# Logistic Regression

Παρά το όνομά της, η λογιστική παλινδρόμηση (logistic regression) προβλέπει την πιθανότητα ένταξης σε κλάση.

Επειδή χρησιμοποιεί μια λογιστική συνάρτηση (logit function) για να χαρακτηρίσει την πιθανότητα, μπορεί να χρησιμοποιηθεί για δυαδική κατηγοριοποίηση με έξοδο 0 ή 1.

Διακρίνονται τρείς τύποι:

Binary logistic regression

Multinomial logistic regression

Ordinal logistic regression

# Logistic Regression

**Binary logistic regression**

Στη δυαδική λογιστική παλινδρόμηση, η μεταβλητή απόκρισης μπορεί να ανήκει μόνο σε δύο κατηγορίες, όπως ναι ή όχι, 0 ή 1, ή αληθές ή ψευδές.

Για παράδειγμα, η πρόβλεψη του αν ένας πελάτης θα αγοράσει ένα προϊόν έχει μόνο δύο αποτελέσματα: ναι ή όχι.

Η δυαδική λογιστική παλινδρόμηση είναι ένας από τους πιο συχνά χρησιμοποιούμενους ταξινομητές για δυαδική ταξινόμηση και η πιο συχνά χρησιμοποιούμενη μέθοδος στη λογιστική παλινδρόμηση.

# Logistic Regression

**Multinomial logistic regression**

Αυτός ο τύπος λογιστικής παλινδρόμησης χρησιμοποιείται όταν η μεταβλητή απόκρισης μπορεί να ανήκει σε μία από τρεις ή περισσότερες κατηγορίες και δεν υπάρχει φυσική διάταξη μεταξύ των κατηγοριών.

Ένα παράδειγμα πρόβλεψης του είδους μιας ταινίας που ένας θεατής είναι πιθανό να παρακολουθήσει από ένα σύνολο επιλογών.

# Logistic Regression

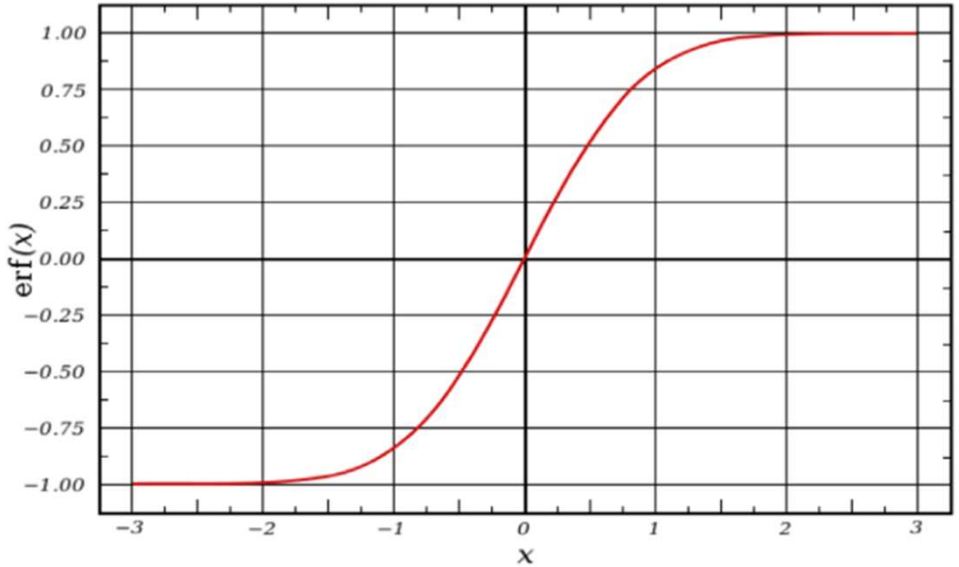
**Ordinal logistic regression**

Αυτός ο τύπος παλινδρόμησης είναι κατάλληλος όταν η μεταβλητή απόκρισης ανήκει σε μία από τρεις ή περισσότερες κατηγορίες και υπάρχει φυσική διάταξη μεταξύ τους. Για παράδειγμα, μια εταιρεία θα μπορούσε να χρησιμοποιήσει τη λογιστική παλινδρόμηση κατά σειρά για να προβλέψει αν τα επίπεδα ικανοποίησης των πελατών θα είναι χαμηλά, μεσαία ή υψηλά.

# Logistic Regression

Η λογιστική παλινδρόμηση χρησιμοποιείται για την πρόβλεψη της πιθανότητας εμφάνισης ενός γεγονότος προσαρμόζοντας τα δεδομένα της μελέτης στην εξίσωση της λογιστικής καμπύλης.

Η καμπύλη αυτή έχει σιγμοειδή μορφή και χαρακτηρίζεται από ένα στάδιο εκθετικής ανάπτυξης στο οποίο ο ρυθμός αύξησης επιβραδύνεται βαθμιαία και περατώνεται στο ασυμπτωτικό στάδιο κορεσμού της ανάπτυξης (η ευθεία βαίνει τελικά παράλληλα στον άξονα Χ).



# Logistic Regression

Η πιο διαδεδομένη βιβλιογραφικά έκφραση της λογιστικής παλινδρόμησης είναι

$\ln(\text{ods})=\beta_0+\beta_1 x_1+ \beta_2 x_2 +\ldots\ldots\ldots+ \beta_k x_k$

Όπου odds= p/(1-p), $\beta_0$ είναι το ύψος της κλίσης της γραμμής παλινδρόμησης, ενώ $\beta_i$ είναι οι συντελεστές παλινδρόμησης καθένας των οποίων εκφράζει το μέγεθος συνεισφοράς της αντίστοιχης μεταβλητής.

Ο όρος odds εναλλακτικά ονομάζεται logit και ο όρος Prob εκφράζει την πιθανότητα του συμβάντος του γεγονότος. Οι συντελεστές των ανεξάρτητων μεταβλητών στην εξίσωση παλινδρόμησης εκτιμώνται με βάση τη μεθόδου Μεγίστης Πιθανοφάνειας (Maximum Likehood Estimate). Σύμφωνα με τη μέθοδο αυτή η τιμή των συντελεστών των ανεξάρτητων μεταβλητών είναι αυτή που κάνει τις παρατηρηθείσες τιμές της εξαρτημένης μεταβλητής πιο πιθανές, βάσει του συνόλου (set) των ανεξαρτήτων μεταβλητών.

# Naive Bayes algorithm

Οι μέθοδοι Naive Bayes είναι ένα σύνολο αλγορίθμων επιβλεπόμενης μάθησης που βασίζονται στην εφαρμογή του θεωρήματος Bayes με την «αφελή» (naïve) υπόθεση της υπό συνθήκη ανεξαρτησίας μεταξύ κάθε ζεύγους χαρακτηριστικών δεδομένης της τιμής της μεταβλητής κλάσης.

# Naive Bayes algorithm

Το θεώρημα Bayes διατυπώνεται μαθηματικά από την ακόλουθη εξίσωση:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

όπου :

P(A|B) είναι η πιθανότητα να συμβεί το γεγονός A όταν το B είναι αληθές,

P(B|A) είναι η πιθανότητα να συμβεί το γεγονός B όταν το A είναι αληθές,

P(A) είναι η πιθανότητα να συμβεί το γεγονός A,

P(B) είναι η πιθανότητα να συμβεί το γεγονός B.

Στον ταξινομητή Naive Bayes, θέλουμε να βρούμε την κλάση που μεγιστοποιεί την υπό όρους πιθανότητα δεδομένου του διανύσματος εισόδου X

# Naive Bayes algorithm

Το θεώρημα Bayes διατυπώνεται μαθηματικά από την ακόλουθη εξίσωση:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

όπου :

P(A|B) είναι η πιθανότητα να συμβεί το γεγονός A όταν το B είναι αληθές,

P(B|A) είναι η πιθανότητα να συμβεί το γεγονός B όταν το A είναι αληθές,

P(A) είναι η πιθανότητα να συμβεί το γεγονός A,

P(B) είναι η πιθανότητα να συμβεί το γεγονός B.

Στον ταξινομητή Naive Bayes, θέλουμε να βρούμε την κλάση που μεγιστοποιεί την υπό όρους πιθανότητα δεδομένου του διανύσματος εισόδου X

# Naive Bayes algorithm

**Παράδειγμα**

Δεδομένου του ακόλουθου συνόλου δεδομένων, ψάχνουμε να βρούμε ποιες είναι οι πιθανότητες να είναι άνδρας ή γυναίκα κάποιος που ονομάζεται Drew.

| Name | Sex |
|---|---|
| Drew | Male |
| Claudia | Female |
| Drew | Female |
| Drew | Female |
| Alberto | Male |
| Karin | Female |
| Nina | Female |
| Sergio | Male |

$$p(\text{male} \mid drew) = \frac{1/3 * 3/8}{3/8} = \frac{0.125}{3/8}$$

$$p(\text{female} \mid drew) = \frac{2/5 * 5/8}{3/8} = \frac{0.250}{3/8}$$

$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \ldots * p(d_n|c_j)$$

Όταν υπάρχουν πολλά χαρακτηριστικά σε κάθε περίπτωση:
d1,d2,...,dn είναι τα χαρακτηριστικά για κάθε περίπτωση d και cj η κλάση.

# Naive Bayes algorithm

## Παράδειγμα

Ας υποθέσουμε ότι στο προηγούμενο παράδειγμα εκτός από το όνομα, λαμβάνουμε υπόψη και άλλα χαρακτηριστικά για κάθε περίπτωση, όπως το ύψος (αν είναι πάνω ή κάτω από 1,70μ.), το χρώμα των ματιών και το μήκος των μαλλιών.
Μας λένε επίσης ότι ο Drew έχει μπλε μάτια, ύψος πάνω από 1,70 m και μακριά μαλλιά και θέλουμε να βρούμε τι πιθανότητες υπάρχουν να είναι άνδρας ή γυναίκα.

| Name | Over 170cm | Eye | Hair length | Sex |
|---|---|---|---|---|
| Drew | No | Blue | Short | Male |
| Claudia | Yes | Brown | Long | Female |
| Drew | No | Blue | Long | Female |
| Drew | No | Blue | Long | Female |
| Alberto | Yes | Brown | Short | Male |
| Karin | No | Blue | Long | Female |
| Nina | Yes | Brown | Short | Female |
| Sergio | Yes | Blue | Long | Male |

p(Drew | cj) = p(over_170cm = yes|cj) * p(eye =blue | cj) * p(Hair_length=Long | cj)
p(Drew | Female) = 2/5 * 3/5 * 4/5
p(Drew | Male) = 2/3 * 2/3 * 1/3

# Support Vector Machines

Το Support Vector Machine (SVM) είναι ένα εργαλείο πρόβλεψης ταξινόμησης και παλινδρόμησης που χρησιμοποιεί τη θεωρία μάθησης μηχανών για να μεγιστοποιήσει την προβλεπτική ακρίβεια ενώ αποφεύγει αυτόματα την υπερβολική προσαρμογή (overfitting) στα δεδομένα.

Το SVM έχει μια τεχνική που ονομάζεται κόλπο πυρήνα (kernel trick). Πρόκειται για functions που λαμβάνουν ως είσοδο έναν χώρο χαμηλών διαστάσεων και το μετατρέπουν σε ένα υψηλότερο χώρο διαστάσεων. Μετατρέπουν ουσιαστικά το μη διαχωρίσιμο πρόβλημα σε διαχωρίσιμο και ονομάζονται kernels

# Support Vector Machines

**Πλεονεκτήματα**

Αποτελεσματικά σε χώρους μεγάλης διάστασης

Αποτελεσματικά στις περιπτώσεις όπου ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων

Ευέλικτο: μπορούν να καθοριστούν διαφορετικές συναρτήσεις πυρήνα για τη συνάρτηση απόφασης

**Μειονέκτημα**

Εάν ο αριθμός των χαρακτηριστικών είναι πολύ μεγαλύτερος από τον αριθμό των δειγμάτων, υπάρχει κίνδυνος για υπερβολικό overfitting και αυτό πρέπει να αποφευχθεί μέσω της επιλογής σωστής συνάρτησης πυρήνα

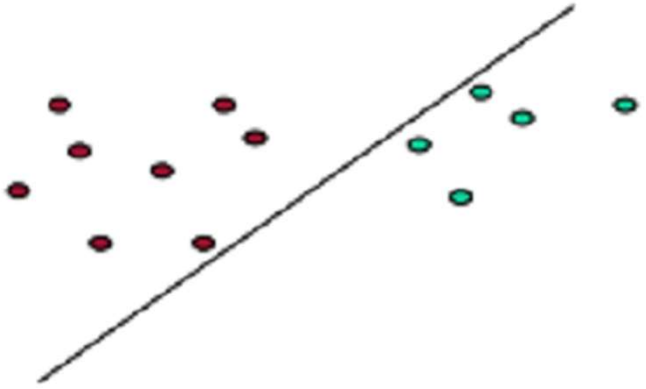
# Support Vector Machines

- Προβάλλουν τα σημεία του συνόλου εκπαίδευσης σε έναν χώρο περισσοτέρων διαστάσεων και βρίσκουν το υπερεπίπεδο το οποίο διαχωρίζει βέλτιστα τα σημεία των δύο τάξεων
- Τα άγνωστα σημεία ταξινομούνται σύμφωνα με την πλευρά του υπερεπίπεδου στην οποία βρίσκονται
- Τα διανύσματα τα οποία ορίζουν το υπερεπίπεδο που χωρίζει τις δύο τάξεις ονομάζονται διανύσματα υποστήριξης (support vectors)

# Support Vector Machines

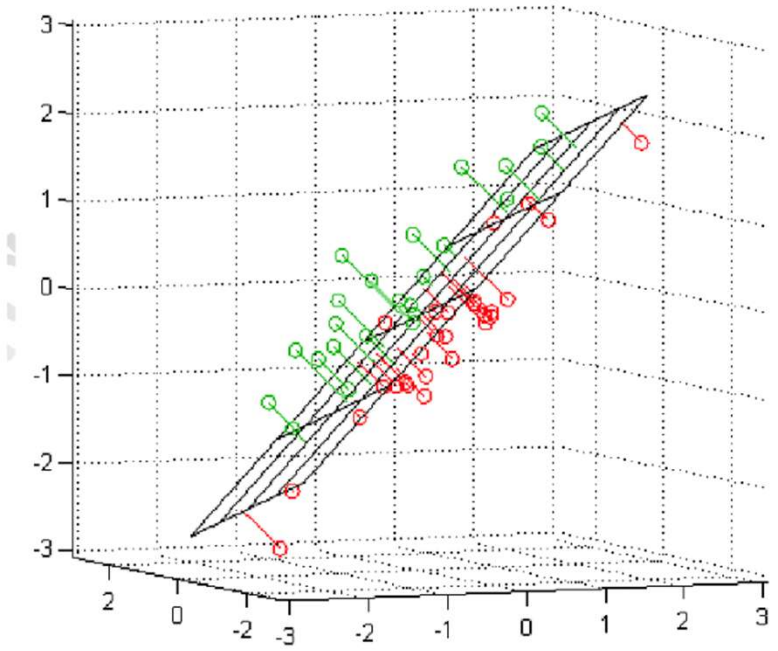
Στα SVM χρησιμοποιείται συχνά η έννοια του υπερεπιπέδου.

Ας υποθέσουμε ότι έχουμε δύο ομάδες σημείων, τα πράσινα και τα κόκκινα. Αυτό που προσπαθούμε να επιτύχουμε είναι να διαχωρίσουμε τα κόκκινα από τα πράσινα σημεία. Όπως είναι εμφανές και από το παρακάτω σχήμα υπάρχει ευθεία, η οποία διαχωρίζει τις δύο ομάδες.



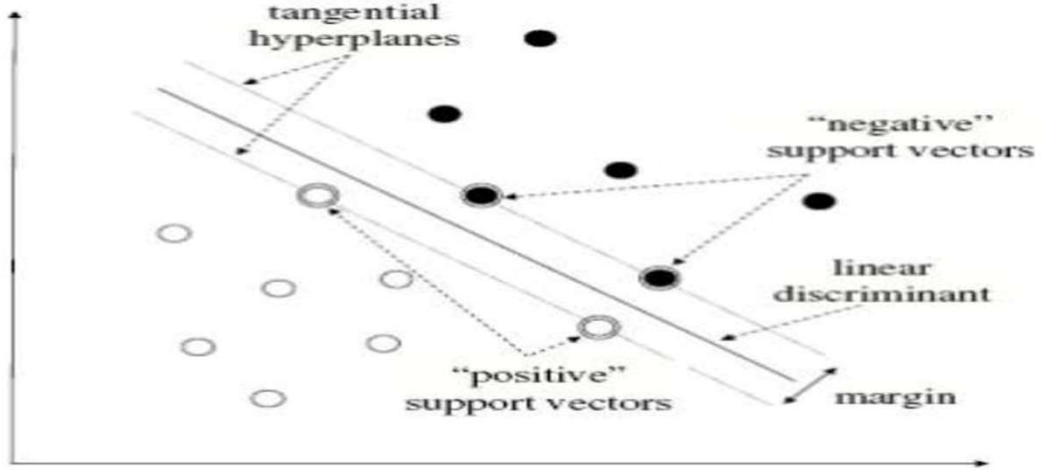
# Support Vector Machines

Έτσι λοιπόν σε δύο διαστάσεις τα σημεία μπορούν να χωρισθούν από μία ευθεία, δηλαδή ένα υπερεπίπεδο μίας διάστασης. Σε τρεις διαστάσεις όπως φαίνεται και στο παρακάτω σχήμα μπορούν να χωρισθούν από ένα επίπεδο, δηλαδή ένα υπερεπίπεδο δύο διαστάσεων.



# Support Vector Machines

- Ο κύριος σκοπός αυτής της μεθόδου είναι να βρούμε το βέλτιστο υπερεπίπεδο το οποίο διαχωρίζει καλύτερα τα σημεία μας
- Το βέλτιστο υπερεπίπεδο ονομάζεται υπερεπίπεδο μέγιστου εύρους (maximum margin hyperplane)
- Δημιουργούμε δύο παράλληλα υπερεπίπεδα τέτοια ώστε να μην υπάρχουν ανάμεσά τους δεδομένα του συνόλου εκπαίδευσης



# Support Vector Machines

Η εξίσωση η οποία συμβολίζει το υπερεπίπεδο έχει την μορφή:

$w \bullet d+b=0$

όπου w και b είναι οι παράμετροι του μοντέλου

D={d1 , d2 , …, dn } το σύνολο των δεδομένων εκπαίδευσης

C={c1 , c2 } σύνολο των κατηγοριών

ci $\in$ {-1,+1} με 1 γνήσια δηλωση και -1 εσφαλμένη δήλωση.

Για όσων δεδομένων τα διανύσματα τους βρίσκονται πάνω στο υπερεπίπεδο θα επαληθεύουν την εξίσωση $w \bullet d+b=0$

ενώ τα διανύσματα των υπόλοιπων δεδομένων θα επαληθεύουν την εξίσωση $w \bullet d+b=m$

# Support Vector Machines

$\boldsymbol{w} \cdot \boldsymbol{d} + b > 0$ τα δεδομένα βρίσκονται πάνω από το υπερεπίπεδο-όριο

$\boldsymbol{w} \cdot \boldsymbol{d} + b < 0$ τα δεδομένα βρίσκονται κατω από το υπερεπίπεδο-όριο

τα παράλληλα υπερεπίπεδα εκφράζονται

$$\boldsymbol{w} \cdot \boldsymbol{d} + b = 1$$
$$\boldsymbol{w} \cdot \boldsymbol{d} + b = -1$$

$d_1$ βρίσκεται πάνω από το υπερεπίπεδο-όριο

$d_2$ βρίσκεται κατω από το υπερεπίπεδο-όριο

εύρος $(margin)$ $\left.\begin{matrix} \boldsymbol{w} \cdot \boldsymbol{d_1} + b = 1 \\ \boldsymbol{w} \cdot \boldsymbol{d_2} + b = -1 \end{matrix}\right\} \Leftrightarrow \boldsymbol{w}(\boldsymbol{d_1} - \boldsymbol{d_2}) = 2$

$$\text{Margin} = \frac{2}{\| \vec{w} \|^2}$$

# Support Vector Machines

Βάσει της προσεγγίσεως των Support Vector Machines, ο«βέλτιστος διαχωριστής» είναι αυτός για τον οποίο, για τα κοντινότερα αντικείμενα προς ταξινόμηση ισχύει

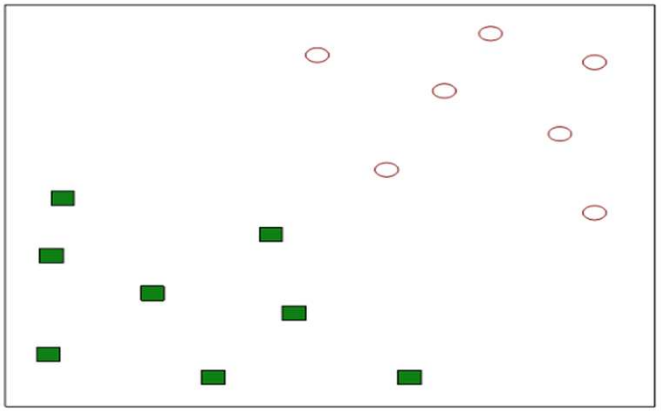
$$f(x) = w \cdot x + b \quad \Rightarrow \quad f(x) = \pm 1 .$$

Η απόσταση d ενός αντικειμένου από τη διαχωριστική υπερεπιφάνεια

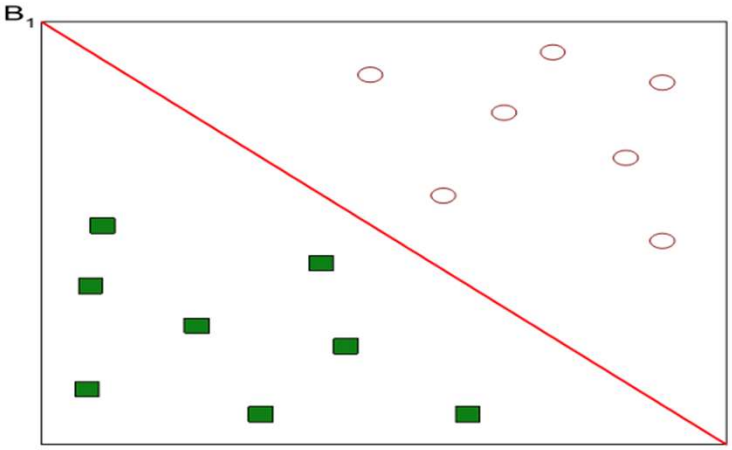
$$d = \frac{w \cdot x + b}{\|w\|}$$

Στόχος της μεθοδολογίας είναι η μεγιστοποίηση της απόστασης αυτής.

# Support Vector Machines

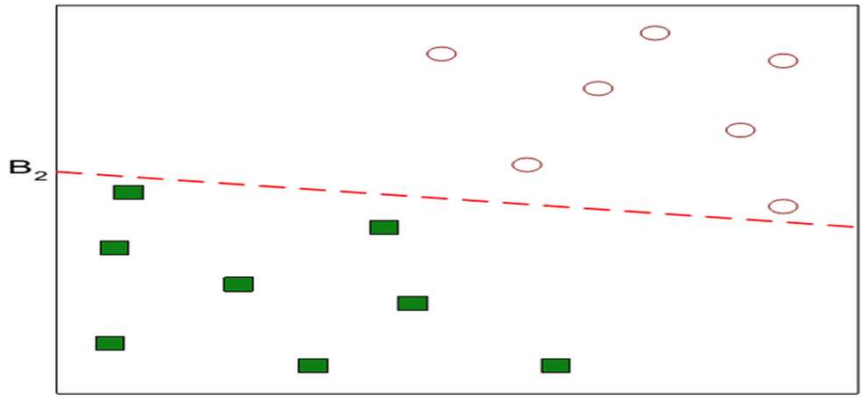


Εύρεση ενός γραμμικού ορίου απόφασης (hyperplane) που διαχωρίζει τα δεδομένα

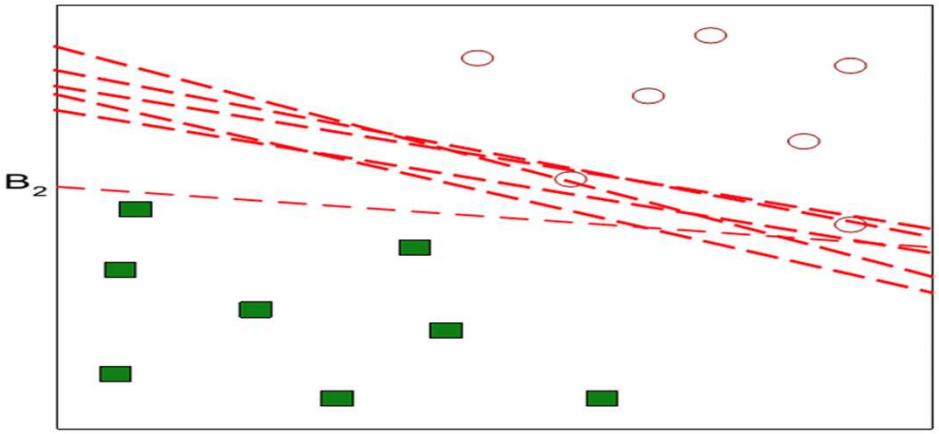



Μια πιθανή λύση

# Support Vector Machines

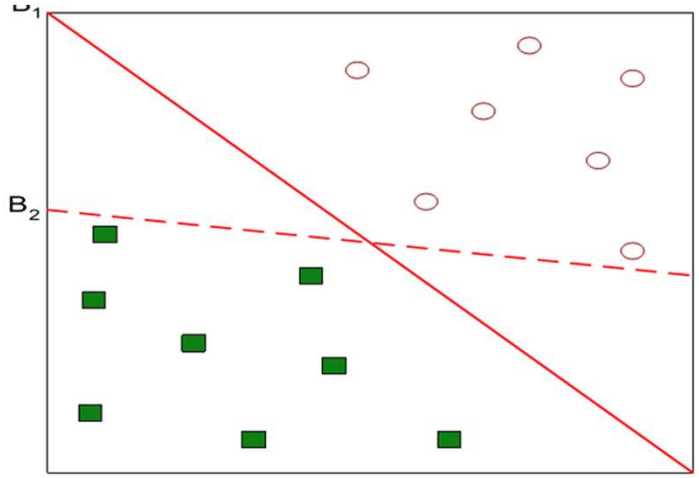



Μια εναλλακτική λύση

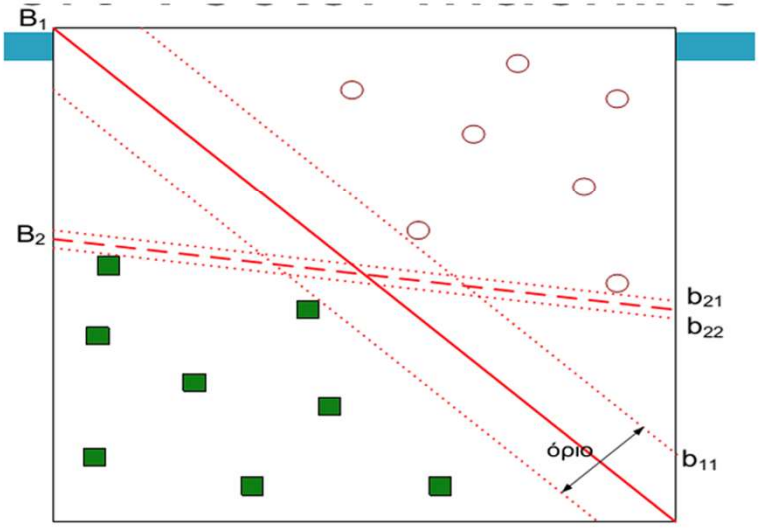



πιθανές λύσεις

# Support Vector Machines

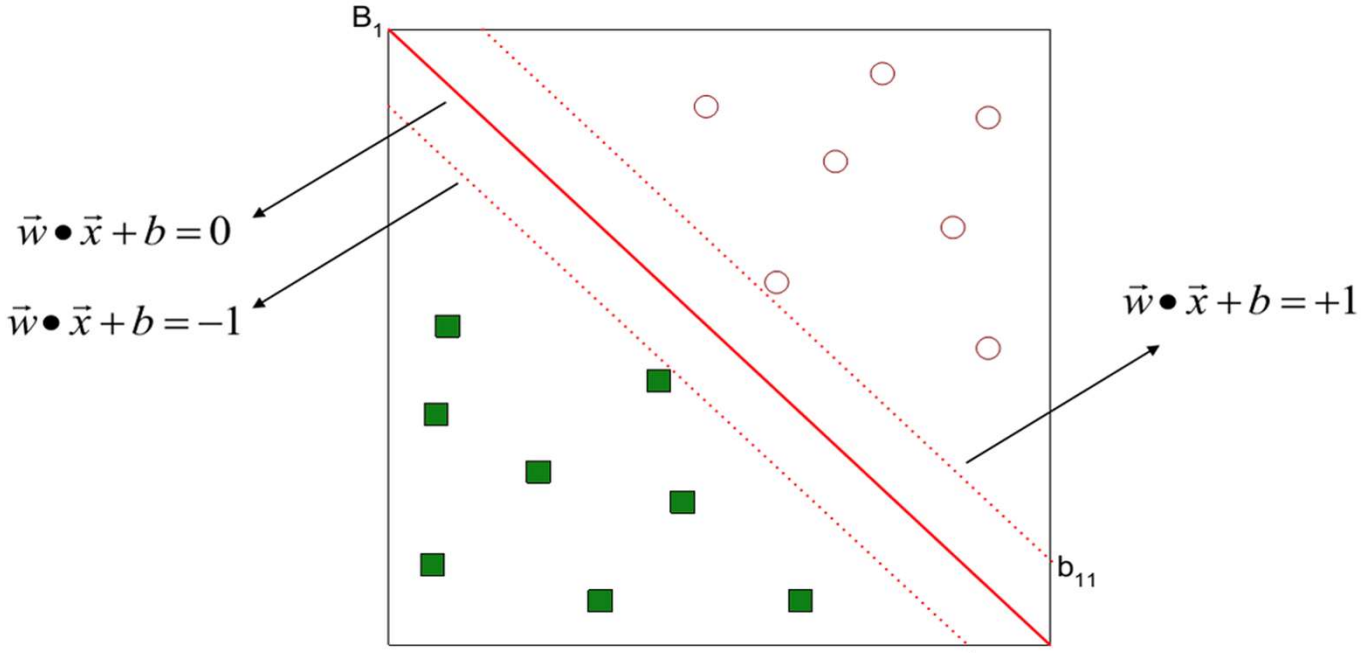



Ποια είναι καλύτερη λύση; Τη B1 ή B2?




Εύρεση του ορίου που μεγιστοποιεί το όριο (margin) που διαχωρίζει τα δεδομένα
Άρα B1 καλύτερη του B2

# Support Vector Machines




$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

1 κλάση κύκλος

-1 κλάση τετράγωνο

$$\text{Margin} = \frac{2}{\| \vec{w} \|^2}$$

# KNN algorithm

Ο αλγόριθμος k-κοντινότεροι γείτονες (KNN) είναι ένας μη παραμετρικός ταξινομητής μάθησης με επίβλεψη, ο οποίος χρησιμοποιεί την απόσταση για να κάνει ταξινομήσεις ή προβλέψεις σχετικά με την ομαδοποίηση ενός μεμονωμένου σημείου δεδομένων. Είναι ένας από τους δημοφιλέστερους και απλούστερους ταξινομητές ταξινόμησης και παλινδρόμησης που χρησιμοποιούνται σήμερα στη μηχανική μάθηση.

Αν και ο αλγόριθμος KNN μπορεί να χρησιμοποιηθεί είτε για προβλήματα παλινδρόμησης είτε για προβλήματα ταξινόμησης, συνήθως χρησιμοποιείται ως αλγόριθμος ταξινόμησης, βασιζόμενος στην υπόθεση ότι παρόμοια σημεία μπορούν να βρεθούν κοντά το ένα στο άλλο.

# KNN algorithm

Για προβλήματα ταξινόμησης, μια ετικέτα κλάσης αποδίδεται με βάση την ψηφοφορία πλειοψηφίας - δηλαδή χρησιμοποιείται η ετικέτα που αντιπροσωπεύεται συχνότερα γύρω από ένα δεδομένο σημείο δεδομένων.

Αν και τεχνικά αυτό θεωρείται «ψηφοφορία πληθικότητας», ο όρος «ψηφοφορία πλειοψηφίας» χρησιμοποιείται συχνότερα στη βιβλιογραφία.

Η διάκριση μεταξύ αυτών των ορολογιών είναι ότι η «ψηφοφορία πλειοψηφίας» τεχνικά απαιτεί πλειοψηφία μεγαλύτερη του 50%, η οποία λειτουργεί κυρίως όταν υπάρχουν μόνο δύο κατηγορίες.

Όταν όμως έχετε πολλές κατηγορίες - π.χ. τέσσερις κατηγορίες, δεν χρειάζεστε απαραίτητα το 50% των ψήφων για να βγάλετε ένα συμπέρασμα σχετικά με μια κατηγορία- θα μπορούσατε να αποδώσετε μια ετικέτα κατηγορίας με ψήφο μεγαλύτερη από 25%.

# KNN algorithm

Για προβλήματα ταξινόμησης, μια ετικέτα κλάσης αποδίδεται με βάση την ψηφοφορία πλειοψηφίας - δηλαδή χρησιμοποιείται η ετικέτα που αντιπροσωπεύεται συχνότερα γύρω από ένα δεδομένο σημείο δεδομένων.

Αν και τεχνικά αυτό θεωρείται «ψηφοφορία πληθικότητας», ο όρος «ψηφοφορία πλειοψηφίας» χρησιμοποιείται συχνότερα στη βιβλιογραφία.

Η διάκριση μεταξύ αυτών των ορολογιών είναι ότι η «ψηφοφορία πλειοψηφίας» τεχνικά απαιτεί πλειοψηφία μεγαλύτερη του 50%, η οποία λειτουργεί κυρίως όταν υπάρχουν μόνο δύο κατηγορίες.

Όταν όμως έχετε πολλές κατηγορίες - π.χ. τέσσερις κατηγορίες, δεν χρειάζεστε απαραίτητα το 50% των ψήφων για να βγάλετε ένα συμπέρασμα σχετικά με μια κατηγορία- θα μπορούσατε να αποδώσετε μια ετικέτα κατηγορίας με ψήφο μεγαλύτερη από 25%.

# KNN algorithm

Ο στόχος του αλγορίθμου k-κοντινότερου γείτονα είναι να εντοπίσει τους πλησιέστερους γείτονες ενός δεδομένου σημείου ερώτησης, έτσι ώστε να μπορέσουμε να αποδώσουμε μια ετικέτα κλάσης σε αυτό το σημείο. Για να το επιτύχει αυτό, ο KNN έχει ορισμένες απαιτήσεις:

**Καθορισμός μετρικής απόστασης**

Προκειμένου να προσδιοριστεί ποια σημεία δεδομένων είναι πλησιέστερα σε ένα δεδομένο σημείο ερώτησης, θα πρέπει να υπολογιστεί η απόσταση μεταξύ του σημείου ερώτησης και των άλλων σημείων δεδομένων. Αυτές οι μετρικές απόστασης βοηθούν στη διαμόρφωση ορίων απόφασης, τα οποία χωρίζουν τα σημεία ερωτήματος σε διαφορετικές περιοχές.

# KNN algorithm

**Καθορισμός μετρικής απόστασης**

Euclidean distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Manhattan distance: είναι επίσης μια άλλη δημοφιλής μετρική απόστασης, η οποία μετρά την απόλυτη τιμή μεταξύ δύο σημείων. Αναφέρεται επίσης ως cityblock.

$$\text{Manhattan Distance} = d(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|\right)$$

# KNN algorithm

**Καθορισμός μετρικής απόστασης**

Minkowski distance : Αυτό το μέτρο απόστασης είναι η γενικευμένη μορφή των μετρικών της Ευκλείδειας απόστασης και της απόστασης Μανχάταν. Η παράμετρος, p, στον παρακάτω τύπο, επιτρέπει τη δημιουργία άλλων μετρικών απόστασης. Η ευκλείδεια απόσταση αναπαρίσταται από αυτόν τον τύπο όταν το p είναι ίσο με δύο, ενώ η απόσταση Μανχάταν συμβολίζεται με p ίσο με ένα.

$$\text{Minkowski Distance} = \left(\sum_{i=1}^{n} \left|x_i - y_i\right|\right)^{1/p}$$

# KNN algorithm

**Καθορισμός μετρικής απόστασης**

Hamming distance : Αυτή η τεχνική χρησιμοποιείται συνήθως με διανύσματα Boolean ή συμβολοσειρών, εντοπίζοντας τα σημεία όπου τα διανύσματα δεν ταιριάζουν. Ως αποτέλεσμα, αναφέρεται επίσης ως μετρική επικάλυψης (overlap).

$$\text{Hamming Distance} = D_H = \left( \sum_{i=1}^{k} \left| x_i - y_i \right| \right)$$

$x=y \quad D=0$

$x \neq y \quad D \neq 1$

# KNN algorithm

**Καθορισμός k**

Η τιμή k στον αλγόριθμο k-NN καθορίζει πόσοι γείτονες θα ελεγχθούν για να καθοριστεί η ταξινόμηση ενός συγκεκριμένου σημείου ερώτησης.

Για παράδειγμα, εάν k=1, το παράδειγμα θα καταταχθεί στην ίδια κλάση με τον μοναδικό πλησιέστερο γείτονά του.

Διαφορετικές τιμές μπορεί να οδηγήσουν σε overfitting ή underfitting.

# KNN algorithm

**Καθορισμός k**

Εάν το K έχει ρυθμιστεί σε μια πολύ χαμηλή τιμή, όπως 1, ο αλγόριθμος γίνεται πολύ ευαίσθητος στο θόρυβο και τις ακραίες τιμές στα δεδομένα.

Σε τέτοιες περιπτώσεις, ο αλγόριθμος μπορεί να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης και να αποτυγχάνει να γενικεύσει καλά σε μη εμφανείς περιπτώσεις.

Από την άλλη πλευρά, εάν το K έχει ρυθμιστεί σε μια πολύ υψηλή τιμή, ο αλγόριθμος μπορεί να χάσει την ικανότητα να καταγράφει τοπικά μοτίβα και μπορεί αντ' αυτού να βασίζεται στην καθολική δομή των δεδομένων, οδηγώντας σε υποκατάσταση.

# KNN algorithm

**Καθορισμός k**

Συνιστάται να έχετε έναν περιττό αριθμό για το k για να αποφύγετε τις ισοπαλίες στην ταξινόμηση (voting procedure).

Μια κοινή μέθοδος εύρεση της βέλτιστης τιμής του k, είναι η χρήση τεχνικών διασταυρούμενης επικύρωσης (k-fold cross validation. Η διασταυρούμενη επικύρωση περιλαμβάνει τη διαίρεση των δεδομένων εκπαίδευσης σε k υποσύνολα ή πτυχές. Στη συνέχεια, ο αλγόριθμος εκπαιδεύεται στις πτυχές k-1 και αξιολογείται στην υπόλοιπη πτυχή, επαναλαμβάνοντας αυτή τη διαδικασία k φορές. Η απόδοση του αλγορίθμου υπολογίζεται κατά μέσο όρο σε όλες τις επαναλήψεις και επιλέγεται η τιμή του K που αποδίδει την καλύτερη απόδοση.

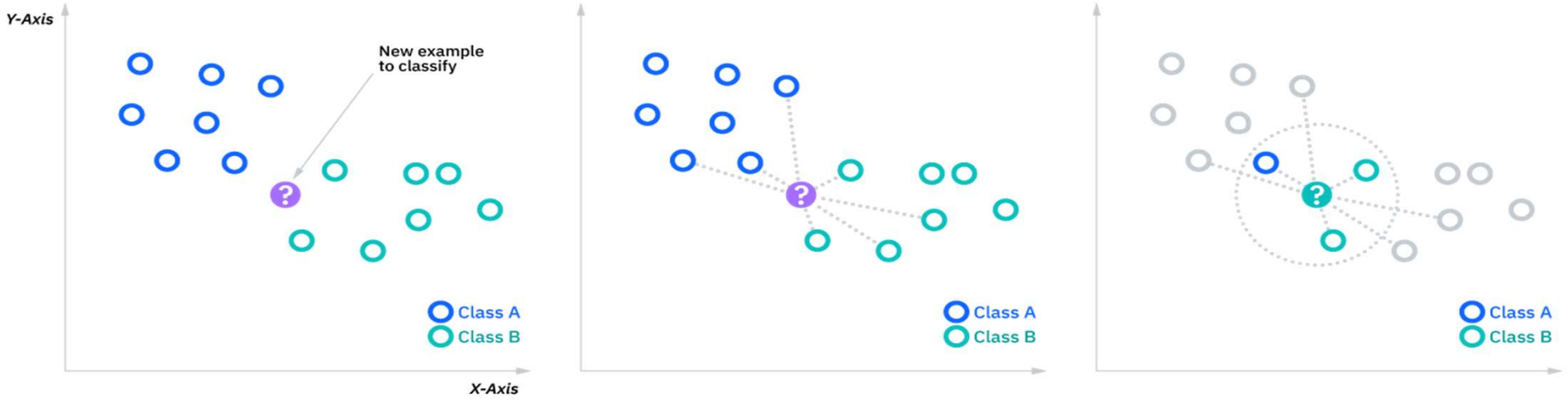
# KNN algorithm

**Καθορισμός k**

Μια άλλη προσέγγιση είναι η χρήση της αναζήτησης πλέγματος (grid search), η οποία περιλαμβάνει την αξιολόγηση της απόδοσης του αλγορίθμου για διαφορετικές τιμές του K σε ένα προκαθορισμένο εύρος. Η αναζήτηση πλέγματος αναζητά εξαντλητικά όλους τους πιθανούς συνδυασμούς τιμών παραμέτρων και επιλέγει την καλύτερη απόδοση. Αυτή η μέθοδος μπορεί να είναι υπολογιστικά ακριβή, ειδικά για μεγάλα σύνολα δεδομένων ή χώρους χαρακτηριστικών υψηλών διαστάσεων, αλλά παρέχει έναν συστηματικό και αξιόπιστο τρόπο προσδιορισμού της βέλτιστης τιμής του k.

Τα δεδομένα που χρησιμοποιούν κανονικοποίηση και κλιμάκωση χαρακτηριστικών μπορούν να βελτιώσουν την απόδοση του αλγορίθμου.

# KNN algorithm




Source: https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN,used%20in%20machine%20learning%20today.

# KNN algorithm

**Πλεονεκτήματα**

- Εύκολη εφαρμογή: Ο αλγόριθμος είναι απλός και ακριβής.

- Προσαρμόζεται εύκολα: Καθώς προστίθενται νέα δείγματα εκπαίδευσης, ο αλγόριθμος προσαρμόζεται ώστε να λαμβάνει υπόψη του κάθε νέο δεδομένο, δεδομένου ότι όλα τα δεδομένα εκπαίδευσης αποθηκεύονται στη μνήμη.

- Λίγες υπερπαραμέτρους: Ο KNN απαιτεί μόνο μια τιμή k και μια μετρική απόστασης.

# KNN algorithm

**Μειονεκτήματα**

- Δεν κλιμακώνεται καλά: Δεδομένου ότι ο KNN είναι ένας τεμπέλης αλγόριθμος, καταλαμβάνει περισσότερη μνήμη και αποθήκευση δεδομένων σε σύγκριση με άλλους ταξινομητές.

- Κατάρα της διαστατικότητας (Curse of dimensionality): Ο αλγόριθμος KNN δεν αποδίδει καλά με δεδομένα εισόδου υψηλής διάστασης.

- Είναι επιρρεπής στην υπερπροσαρμογή (overfitting): Λόγω της «κατάρας της διαστατικότητας», ο KNN είναι επίσης πιο επιρρεπής στην υπερπροσαρμογή.

# Δέντρα απόφασης - Decision trees

Τα Δέντρα Αποφάσεων είναι μια μη παραμετρική μέθοδος μάθησης με επίβλεψη που χρησιμοποιείται για ταξινόμηση και παλινδρόμηση. Ο στόχος είναι η δημιουργία ενός μοντέλου που προβλέπει την τιμή μιας μεταβλητής-στόχου με την εκμάθηση απλών κανόνων απόφασης που προκύπτουν από τα χαρακτηριστικά των δεδομένων.

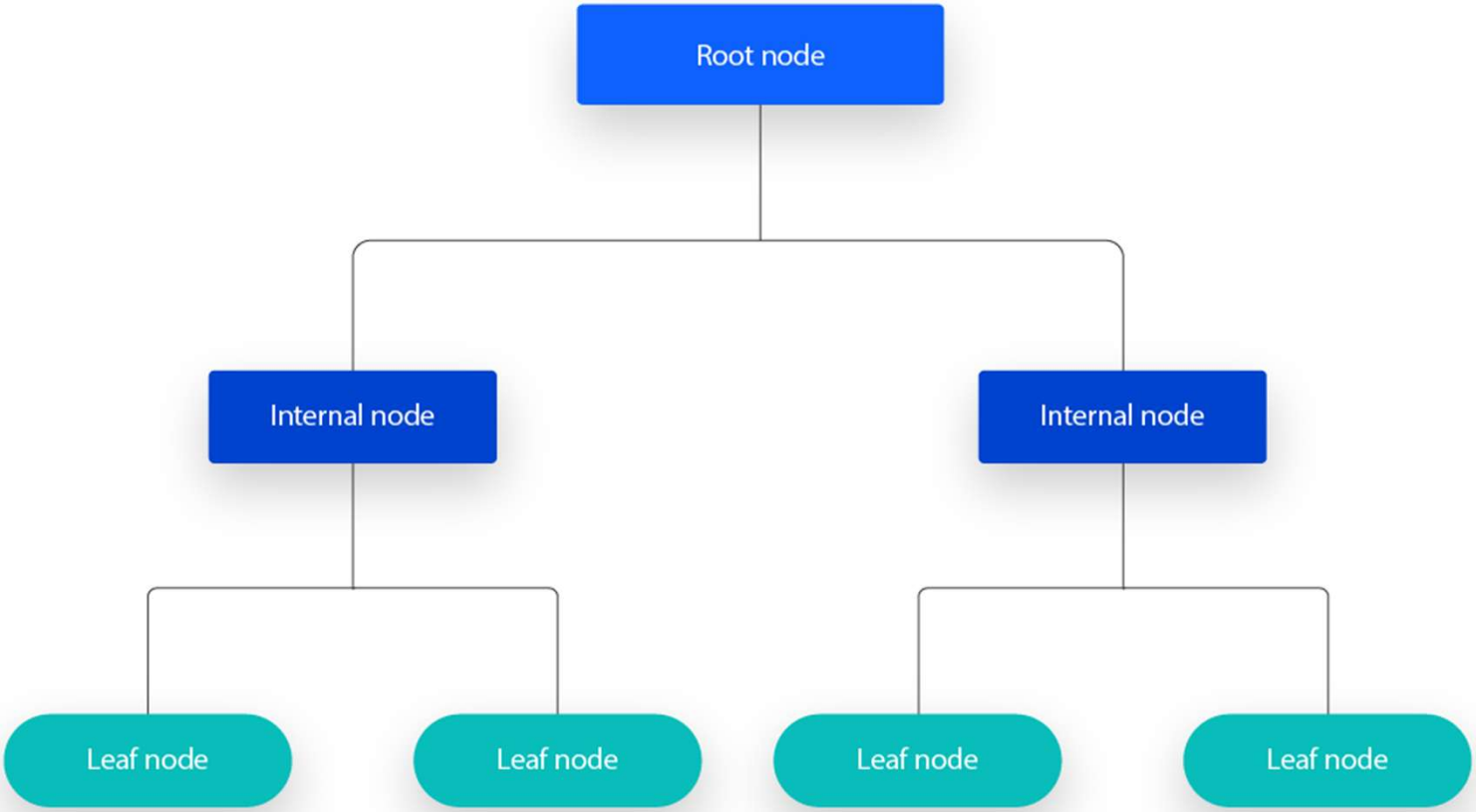
# Δέντρα απόφασης - Decision trees

Ένα δέντρο αποφάσεων ξεκινά με έναν κόμβο-ρίζα (root node), ο οποίος δεν έχει εισερχόμενους κλάδους.

Οι εξερχόμενοι κλάδοι από τον κόμβο ρίζας τροφοδοτούν στη συνέχεια τους εσωτερικούς κόμβους, γνωστούς και ως κόμβους απόφασης (decision nodes). Με βάση τα διαθέσιμα χαρακτηριστικά, και οι δύο τύποι κόμβων διεξάγουν αξιολογήσεις για να σχηματίσουν ομοιογενή υποσύνολα, τα οποία συμβολίζονται με κόμβους φύλλων (leaf nodes) ή τερματικούς κόμβους.

Οι κόμβοι φύλλων αντιπροσωπεύουν όλα τα πιθανά αποτελέσματα εντός του συνόλου δεδομένων.

# Δέντρα απόφασης - Decision trees




Source: https://www.ibm.com/topics/decision-trees

# Δέντρα απόφασης - Decision trees

Η εκμάθηση δέντρων αποφάσεων χρησιμοποιεί μια στρατηγική «διαίρει και βασίλευε» με τη διεξαγωγή μιας άπληστης αναζήτησης για τον εντοπισμό των βέλτιστων σημείων διαχωρισμού μέσα σε ένα δέντρο. Αυτή η διαδικασία διάσπασης επαναλαμβάνεται στη συνέχεια με αναδρομικό τρόπο από πάνω προς τα κάτω, έως ότου όλες ή η πλειοψηφία των εγγραφών ταξινομηθούν σε συγκεκριμένες ετικέτες κλάσης.

Το κατά πόσον όλα τα σημεία δεδομένων ταξινομούνται ως ομοιογενή σύνολα εξαρτάται σε μεγάλο βαθμό από την πολυπλοκότητα του δέντρου απόφασης. Τα μικρότερα δέντρα είναι πιο εύκολα σε θέση να επιτύχουν αμιγείς κόμβους φύλλων - δηλαδή σημεία δεδομένων σε μία μόνο κλάση.

# Δέντρα απόφασης - Decision trees

Ωστόσο, καθώς ένα δέντρο μεγαλώνει σε μέγεθος, γίνεται όλο και πιο δύσκολο να διατηρηθεί αυτή η καθαρότητα και αυτό συνήθως έχει ως αποτέλεσμα να εμπίπτουν πολύ λίγα δεδομένα σε ένα δεδομένο υποδέντρο. Όταν συμβαίνει αυτό, είναι γνωστό ως κατακερματισμός δεδομένων και μπορεί συχνά να οδηγήσει σε υπερπροσαρμογή.

Τα δέντρα αποφάσεων θα πρέπει να προσθέτουν πολυπλοκότητα μόνο αν είναι απαραίτητο, καθώς η απλούστερη εξήγηση είναι συχνά η καλύτερη. Για να μειωθεί η πολυπλοκότητα και να αποφευχθεί η υπερπροσαρμογή, χρησιμοποιείται συνήθως το κλάδεμα (pruning), που πρόκειται για μια διαδικασία η οποία αφαιρεί κλάδους που χωρίζονται σε χαρακτηριστικά με χαμηλή σημασία. Η προσαρμογή του μοντέλου μπορεί στη συνέχεια να αξιολογηθεί μέσω της διαδικασίας της διασταυρούμενης επικύρωσης (cross validation).

# Δέντρα απόφασης - Decision trees

Οι πιο διαδεδομένοι αλγόριθμοι είναι :

- ID3: που είναι συντομογραφία για το «Iterative Dichotomiser 3» (Επαναληπτικός Διχοτομητής 3). Αυτός ο αλγόριθμος αξιοποιεί την εντροπία και το κέρδος πληροφορίας ως μετρικές για την αξιολόγηση των υποψήφιων διαχωρισμών.

- C4.5: Αυτός ο αλγόριθμος θεωρείται μεταγενέστερη επανάληψη του ID3, ο οποίος αναπτύχθηκε επίσης από τον Quinlan. Μπορεί να χρησιμοποιήσει κέρδος πληροφορίας ή αναλογίες κέρδους για την αξιολόγηση των σημείων διαχωρισμού εντός των δέντρων απόφασης.

- CART: Ο όρος, CART, είναι συντομογραφία για τα «δέντρα ταξινόμησης και παλινδρόμησης» και εισήχθη από τον Leo Breiman.

# Δέντρα απόφασης - Decision trees

Πώς να επιλέξετε το καλύτερο χαρακτηριστικό σε κάθε κόμβο

Δύο μέθοδοι, το κέρδος πληροφορίας (Information gain) και η Gini impurity, λειτουργούν ως δημοφιλές κριτήριο διαχωρισμού για τα μοντέλα δέντρων αποφάσεων. Βοηθούν στην αξιολόγηση της ποιότητας κάθε δοκιμαστικής συνθήκης και του πόσο καλά θα μπορέσει να ταξινομήσει τα δείγματα σε μια κλάση.

# Δέντρα απόφασης - Decision trees

$$\text{Entropy}(S) = -\sum_{c \in C} p(c)\log_2 p(c)$$

Το S αντιπροσωπεύει το σύνολο δεδομένων που υπολογίζεται η εντροπία
c αντιπροσωπεύει τις κλάσεις στο σύνολο S
p(c) αντιπροσωπεύει την αναλογία των σημείων δεδομένων που ανήκουν στην κλάση c προς τον αριθμό των συνολικών σημείων δεδομένων στο σύνολο S

# Δέντρα απόφασης - Decision trees

Οι τιμές της εντροπίας μπορούν να κυμαίνονται μεταξύ 0 και 1. Εάν όλα τα δείγματα στο σύνολο δεδομένων, S, ανήκουν σε μία κλάση, τότε η εντροπία θα είναι ίση με μηδέν.

Εάν τα μισά δείγματα ταξινομούνται σε μία κλάση και τα άλλα μισά σε άλλη κλάση, η εντροπία θα είναι στο υψηλότερο σημείο της, στο 1.

Προκειμένου να επιλεγεί το καλύτερο χαρακτηριστικό για διαχωρισμό και να βρεθεί το βέλτιστο δέντρο απόφασης, θα πρέπει να χρησιμοποιηθεί το χαρακτηριστικό με το μικρότερο ποσό εντροπίας.

# Δέντρα απόφασης - Decision trees

Το κέρδος πληροφορίας αντιπροσωπεύει τη διαφορά στην εντροπία πριν και μετά τη διάσπαση σε ένα δεδομένο χαρακτηριστικό.

Το χαρακτηριστικό με το μεγαλύτερο κέρδος πληροφορίας θα παράγει την καλύτερη διάσπαση, καθώς κάνει την καλύτερη δουλειά στην ταξινόμηση των δεδομένων εκπαίδευσης σύμφωνα με την ταξινόμηση-στόχο.

Το κέρδος πληροφορίας αναπαρίσταται συνήθως με τον ακόλουθο τύπο, όπου:

Information Gain= entropy(parent) –[average entropy(children)]

# Δέντρα απόφασης - Decision trees

Παράδειγμα

| Day | Outlook | Temp | Humidity | Wind | Tennis |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Weak | Yes |

# Δέντρα απόφασης - Decision trees

Παράδειγμα

Για αυτό το σύνολο δεδομένων, η εντροπία είναι 0,94. Αυτό μπορεί να υπολογιστεί βρίσκοντας το ποσοστό των ημερών όπου το «Play Tennis» είναι «Yes», το οποίο είναι 9/14, και το ποσοστό των ημερών όπου το «Play Tennis» είναι «No», το οποίο είναι 5/14. Στη συνέχεια, αυτές οι τιμές μπορούν να εισαχθούν στον τύπο εντροπίας.

Entropy (Tennis) = -(9/14) log2(9/14) – (5/14) log2 (5/14) = 0.94

| Day | Outlook | Temp | Humidity | Wind | Tennis |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Weak | Yes |

# Δέντρα απόφασης - Decision trees

Παράδειγμα

Στη συνέχεια, μπορούμε να υπολογίσουμε το κέρδος πληροφορίας για κάθε ένα από τα χαρακτηριστικά ξεχωριστά. Για παράδειγμα, το κέρδος πληροφορίας για το χαρακτηριστικό «Humidity» θα είναι το ακόλουθο:
Gain (Tennis, Humidity) = (0.94)-(7/14)*(0.985) – (7/14)*(0.592) = 0.151

7/14 αντιπροσωπεύει το ποσοστό των τιμών όπου η υγρασία είναι ίση με «υψηλή» προς το συνολικό αριθμό τιμών υγρασίας. Στην περίπτωση αυτή, ο αριθμός των τιμών όπου η υγρασία είναι ίση με «υψηλή» είναι ο ίδιος με τον αριθμό των τιμών όπου η υγρασία είναι ίση με «κανονική».

- 0,985 είναι η εντροπία όταν η υγρασία = «υψηλή».

- 0,59 είναι η εντροπία όταν η υγρασία = «κανονική»

| Day | Outlook | Temp | Humidity | Wind | Tennis |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Weak | Yes |

# Δέντρα απόφασης - Decision trees

Παράδειγμα

Στη συνέχεια, επαναλάβετε τον υπολογισμό του κέρδους πληροφορίας για κάθε χαρακτηριστικό στον παραπάνω πίνακα και επιλέξτε το χαρακτηριστικό με το μεγαλύτερο κέρδος πληροφορίας ως το πρώτο σημείο διαχωρισμού στο δέντρο αποφάσεων. Σε αυτή την περίπτωση, το outlook παράγει το υψηλότερο κέρδος πληροφορίας. Από εκεί και πέρα, η διαδικασία επαναλαμβάνεται για κάθε υποδέντρο.

| Day | Outlook | Temp | Humidity | Wind | Tennis |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Weak | Yes |

# Δέντρα απόφασης - Decision trees

Gini Impurity

Είναι ένα μέτρο που χρησιμοποιείται στους αλγορίθμους δέντρων απόφασης (όπως ο CART — Classification and Regression Trees) για να εκτιμήσει πόσο «καθαρός» ή «μικτός» είναι ένας κόμβος, δηλαδή πόσο καλά χωρίζει τα δεδομένα σε ξεχωριστές κατηγορίες.
Παρόμοια με την εντροπία, εάν το σύνολο, S, είναι καθαρό - δηλαδή ανήκει σε μία κλάση - τότε, το Gini Impurity είναι μηδέν. Αυτό συμβολίζεται με τον ακόλουθο τύπο:

$$\text{Gini Impurity} = 1 - \sum_i (p_i)^2$$

όπου $p_i$ :το ποσοστό των δειγμάτων στον κόμβο $t$που ανήκουν στην κατηγορία $i$.

# Δέντρα απόφασης - Decision trees

Gini Impurity

Αν $G(t)$=0: ο κόμβος είναι καθαρός (όλα τα δείγματα ανήκουν στην ίδια κατηγορία).

Αν $G(t)$ είναι μεγάλος, σημαίνει ότι οι κατηγορίες είναι ανακατεμένες μέσα στον κόμβο.

Η μέγιστη τιμή για δυαδική ταξινόμηση (2 κατηγορίες) είναι 0.5 — όταν οι κατηγορίες είναι 50%-50%.

# Δέντρα απόφασης - Decision trees

Πώς χρησιμοποιείται στα δέντρα απόφασης

Κατά την κατασκευή του δέντρου:

- Ο αλγόριθμος δοκιμάζει διάφορα πιθανά διαχωριστικά (splits).
- Υπολογίζει το μέτρο για τους νέους κόμβους.
- Υπολογίζει το σταθμισμένο μετρο μετά τον διαχωρισμό.
- Επιλέγει το split που μειώνει περισσότερο την ακαθαρσία.

# Δέντρα απόφασης - Decision trees

**Παράδειγμα**
Ας υποθέσουμε ότι σε έναν κόμβο έχουμε:
4 δείγματα της **Κατηγορίας Α**
6 δείγματα της **Κατηγορίας Β**
Τότε:

$$p_A = \frac{4}{10} = 0.4, p_B = \frac{6}{10} = 0.6$$

Άρα:

$$G = 1 - \left(0.4^2 + 0.6^2\right) = 1 - (0.16 + 0.36) = 0.48$$

Ο κόμβος αυτός έχει σχετικά υψηλό Gini impurity (είναι «μικτός»).

# Δέντρα απόφασης - Decision trees

**Πλεονεκτήματα**

- Απλό στην κατανόηση και την ερμηνεία. Τα δέντρα μπορούν να απεικονιστούν.

- Απαιτεί μικρή προετοιμασία δεδομένων. Άλλες τεχνικές απαιτούν συχνά κανονικοποίηση των δεδομένων, πρέπει να δημιουργηθούν εικονικές μεταβλητές και να αφαιρεθούν οι κενές τιμές. Ορισμένοι συνδυασμοί δέντρων και αλγορίθμων υποστηρίζουν τις ελλείπουσες τιμές.

- Ευέλικτο: Το δένδρο αποφάσεων μπορεί να χρησιμοποιηθεί τόσο για εργασίες ταξινόμησης όσο και για εργασίες παλινδρόμησης, καθιστώντας το πιο ευέλικτο από ορισμένους άλλους αλγορίθμους. Αυτό σημαίνει ότι εάν δύο μεταβλητές συσχετίζονται σε μεγάλο βαθμό, ο αλγόριθμος θα επιλέξει μόνο ένα από τα χαρακτηριστικά για διαχωρισμό.

# Δέντρα απόφασης - Decision trees

Πλεονεκτήματα

- Δυνατότητα χειρισμού προβλημάτων πολλαπλών εξόδων.
- Αποτελεί ένα white box model . Εάν μια δεδομένη κατάσταση είναι παρατηρήσιμη σε ένα μοντέλο, η εξήγηση της κατάστασης εξηγείται εύκολα με τη λογική boolean. Αντίθετα, σε ένα black box model (π.χ. σε ένα τεχνητό νευρωνικό δίκτυο), τα αποτελέσματα μπορεί να είναι πιο δύσκολο να ερμηνευθούν.

# Δέντρα απόφασης - Decision trees

Πλεονεκτήματα

- Είναι δυνατή η επικύρωση ενός μοντέλου με τη χρήση στατιστικών δοκιμών. Αυτό καθιστά δυνατή τη συνεκτίμηση της αξιοπιστίας του μοντέλου.

- Αποδίδει γενικά καλά.

# Δέντρα απόφασης - Decision trees

Μειονεκτήματα

- Μπορεί να δημιουργηθούν υπερβολικά πολύπλοκα δέντρα που δεν γενικεύουν καλά τα δεδομένα (overfitting).

- Τα δέντρα απόφασης μπορεί να είναι ασταθή, επειδή μικρές μεταβολές στα δεδομένα μπορεί να οδηγήσουν στη δημιουργία ενός εντελώς διαφορετικού δέντρου

- Υπάρχουν έννοιες που μαθαίνονται δύσκολα επειδή τα δέντρα αποφάσεων δεν τις εκφράζουν εύκολα, όπως τα προβλήματα XOR, ισοτιμίας (parity) ή πολυπλέκτη (multiplexer).
- Υπάρχει πρόβλημα αν κάποιες κλάσεις κυριαρχούν. Συνεπώς, συνιστάται η εξισορρόπηση του συνόλου δεδομένων πριν από την προσαρμογή με το δέντρο απόφασης.

# Ensemble methods

Η μάθηση συνόλου (ensemble learning) είναι μια τεχνική μηχανικής μάθησης που συγκεντρώνει δύο ή περισσότερα μοντέλα (π.χ. μοντέλα παλινδρόμησης, νευρωνικά δίκτυα) προκειμένου να παράγει καλύτερες προβλέψεις.

Με άλλα λόγια, ένα ensemble model συνδυάζει πολλά μεμονωμένα μοντέλα για να παράγει ακριβέστερες προβλέψεις από ό,τι ένα μεμονωμένο μοντέλο από μόνο του.

# Ensemble methods

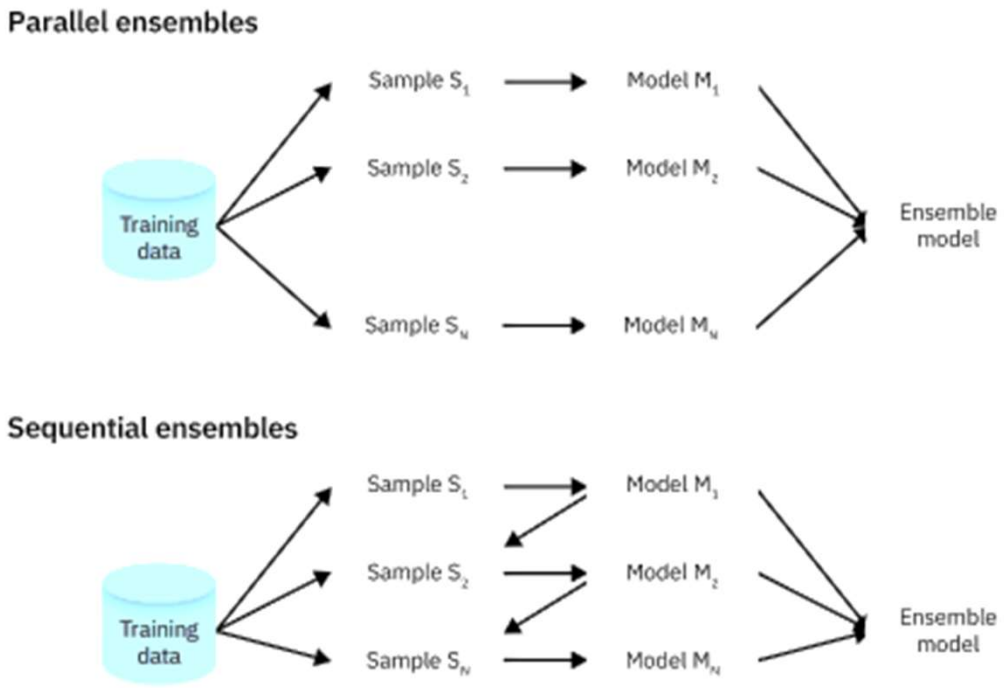
**Τύποι**

- Parallel methods : εκπαιδεύουν κάθε μοντέλο ξεχωριστά από τα υπόλοιπα.

- Sequantial methods: εκπαιδεύουν έναν νέο μοντέλο έτσι ώστε να ελαχιστοποιεί τα σφάλματα που έγιναν από το προηγούμενο μοντέλο που εκπαιδεύτηκε στο προηγούμενο βήμα. Με άλλα λόγια, οι διαδοχικές μέθοδοι κατασκευάζουν μοντέλα βάσης διαδοχικά σε στάδια

# Ensemble methods

**Τύποι**




Source: https://www.ibm.com/topics/ensemble-learning

# Ensemble methods

**Voting**

Η ψηφοφορία πλειοψηφίας λαμβάνει υπόψη την πρόβλεψη κάθε μοντέλου για μια δεδομένη περίπτωση δεδομένων και εξάγει μια τελική πρόβλεψη που καθορίζεται από ό,τι προβλέπει η πλειοψηφία των μοντέλων. Για παράδειγμα, σε ένα πρόβλημα δυαδικής ταξινόμησης, η ψηφοφορία πλειοψηφίας λαμβάνει τις προβλέψεις από κάθε ταξινομητή βάσης για μια δεδομένη περίπτωση δεδομένων και χρησιμοποιεί την πρόβλεψη της πλειοψηφίας ως τελική πρόβλεψη.

Η σταθμισμένη ψηφοφορία πλειοψηφίας είναι μια επέκταση αυτής της τεχνικής που δίνει μεγαλύτερη βαρύτητα στις προβλέψεις ορισμένων μοντέλων έναντι άλλων.

# Ensemble methods

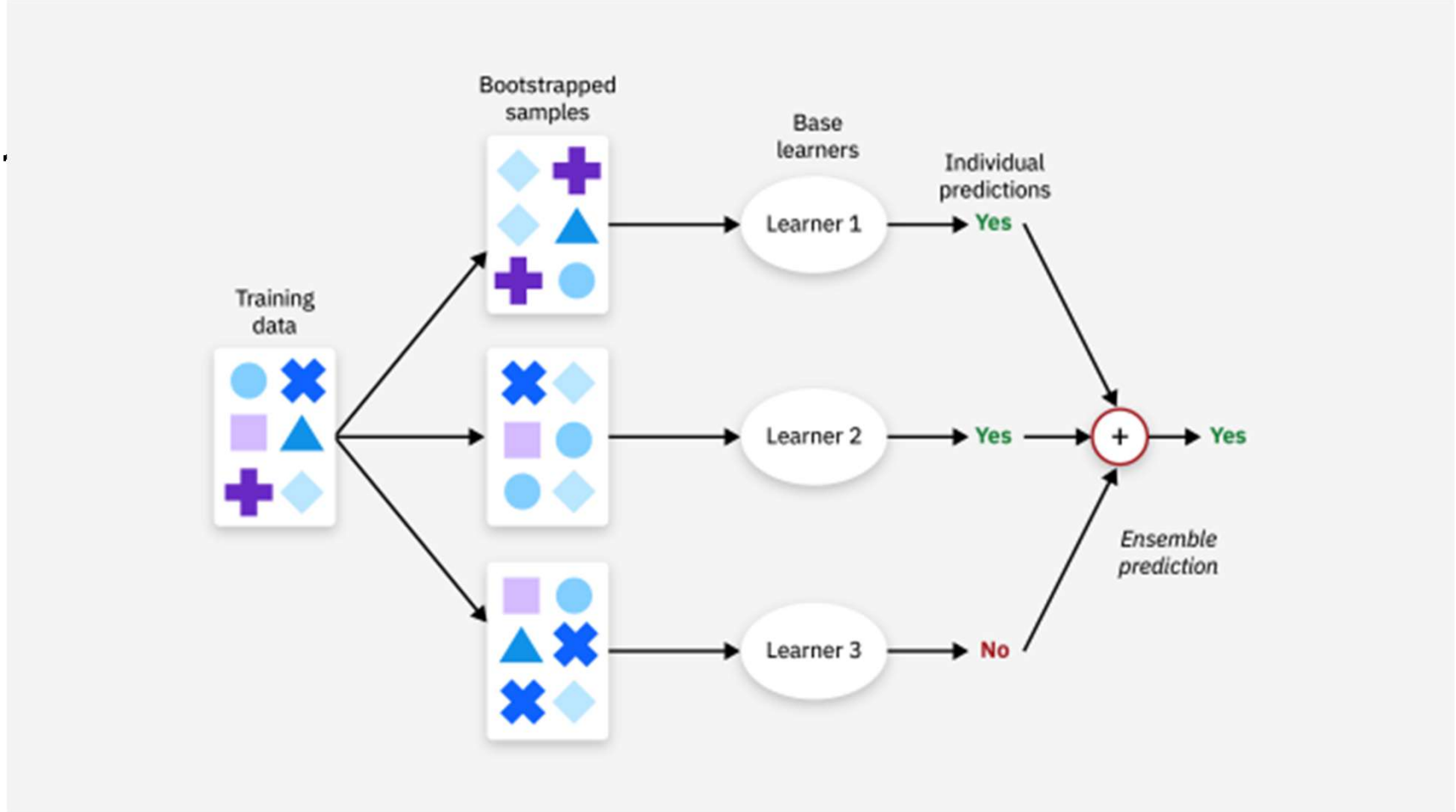
**Τεχνικές**

Bagging

Η μέθοδος bagging χρησιμοποιεί μια τεχνική που ονομάζεται bootstrap resampling για την εξαγωγή πολλαπλών νέων συνόλων δεδομένων από ένα αρχικό σύνολο δεδομένων εκπαίδευσης, προκειμένου να εκπαιδευτούν πολλαπλοί learners.

Ας υποθέσουμε ότι ένα σύνολο δεδομένων εκπαίδευσης περιέχει n παραδείγματα εκπαίδευσης. Η επαναδειγματοληψία bootstrap αντιγράφει n παραδείγματα δεδομένων από αυτό το σύνολο σε ένα νέο υποδειγματικό σύνολο δεδομένων, με ορισμένα αρχικά παραδείγματα να εμφανίζονται περισσότερες από μία φορές και άλλα να αποκλείονται εντελώς. Αυτά είναι τα δείγματα bootstrap. Η επανάληψη αυτής της διαδικασίας x φορές παράγει x επαναλήψεις του αρχικού συνόλου δεδομένων, καθεμία από τις οποίες περιέχει n δείγματα από το αρχικό σύνολο.

# Ensemble methods




Source: https://www.ibm.com/topics/ensemble-learning

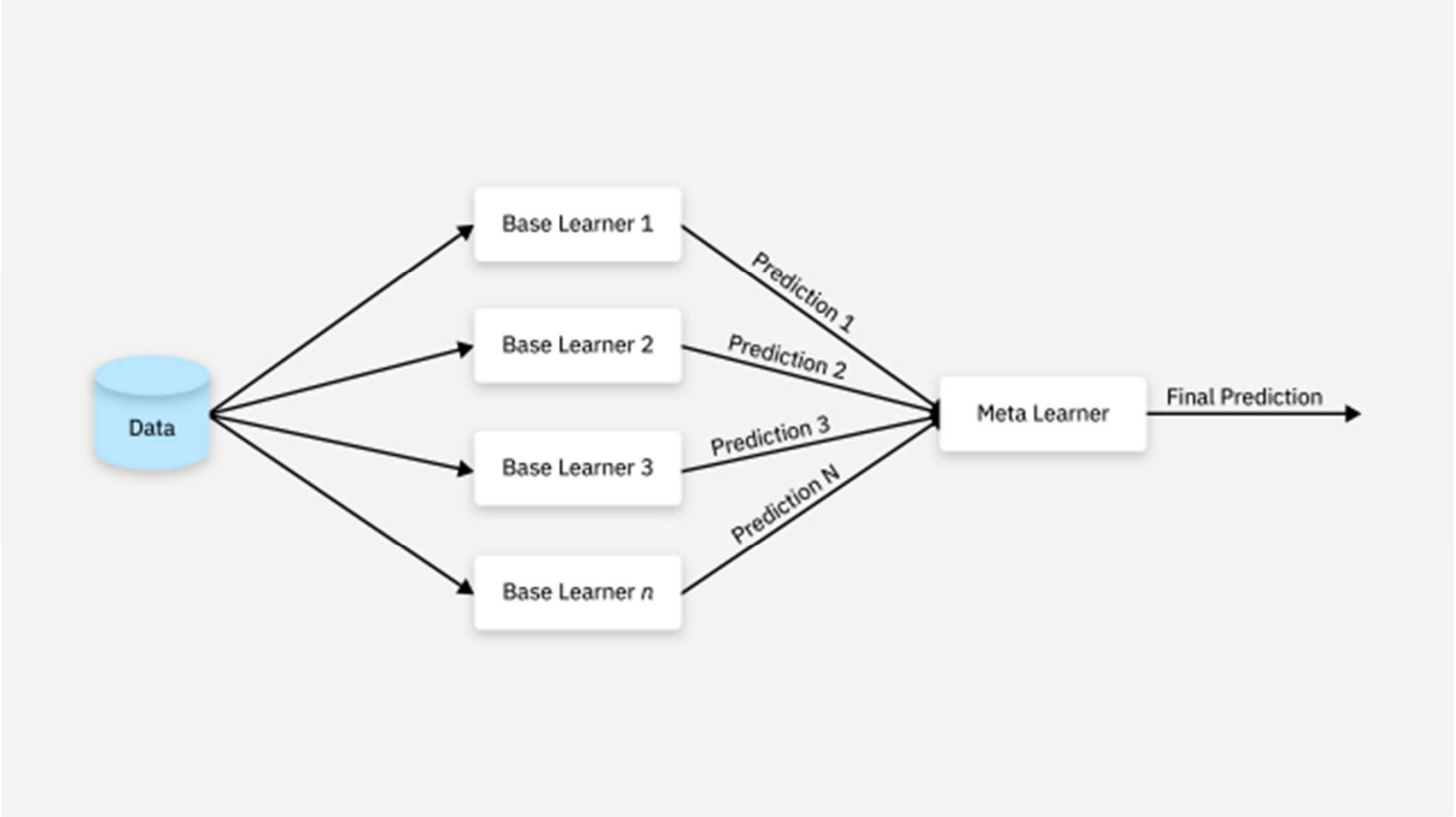
# Ensemble methods

**Τεχνικές**

Stacking

Εκπαιδεύει συγκεκριμένα πολλούς base learners από το ίδιο σύνολο δεδομένων χρησιμοποιώντας διαφορετικό αλγόριθμο εκπαίδευσης για κάθε εκπαιδευόμενο. Κάθε base learner κάνει προβλέψεις σε ένα test set. Αυτές οι προβλέψεις των συγκεντρώνονται και χρησιμοποιούνται για την εκπαίδευση ενός τελικού μοντέλου.

# Ensemble methods




Source: https://www.ibm.com/topics/ensemble-learning

# Ensemble methods

**Τεχνικές**

Boosting

Εκπαιδεύει έναν learner σε κάποιο αρχικό σύνολο δεδομένων, d. Ο learner που προκύπτει είναι συνήθως αδύναμος και ταξινομεί εσφαλμένα πολλά δείγματα στο σύνολο δεδομένων. Όπως και το bagging, έτσι και το boosting παίρνει δείγματα από το αρχικό σύνολο δεδομένων για να δημιουργήσει ένα νέο σύνολο δεδομένων (d2). Ωστόσο, σε αντίθεση με το bagging, το boosting δίνει προτεραιότητα σε εσφαλμένα ταξινομημένα παραδείγματα δεδομένων από το πρώτο μοντέλο.

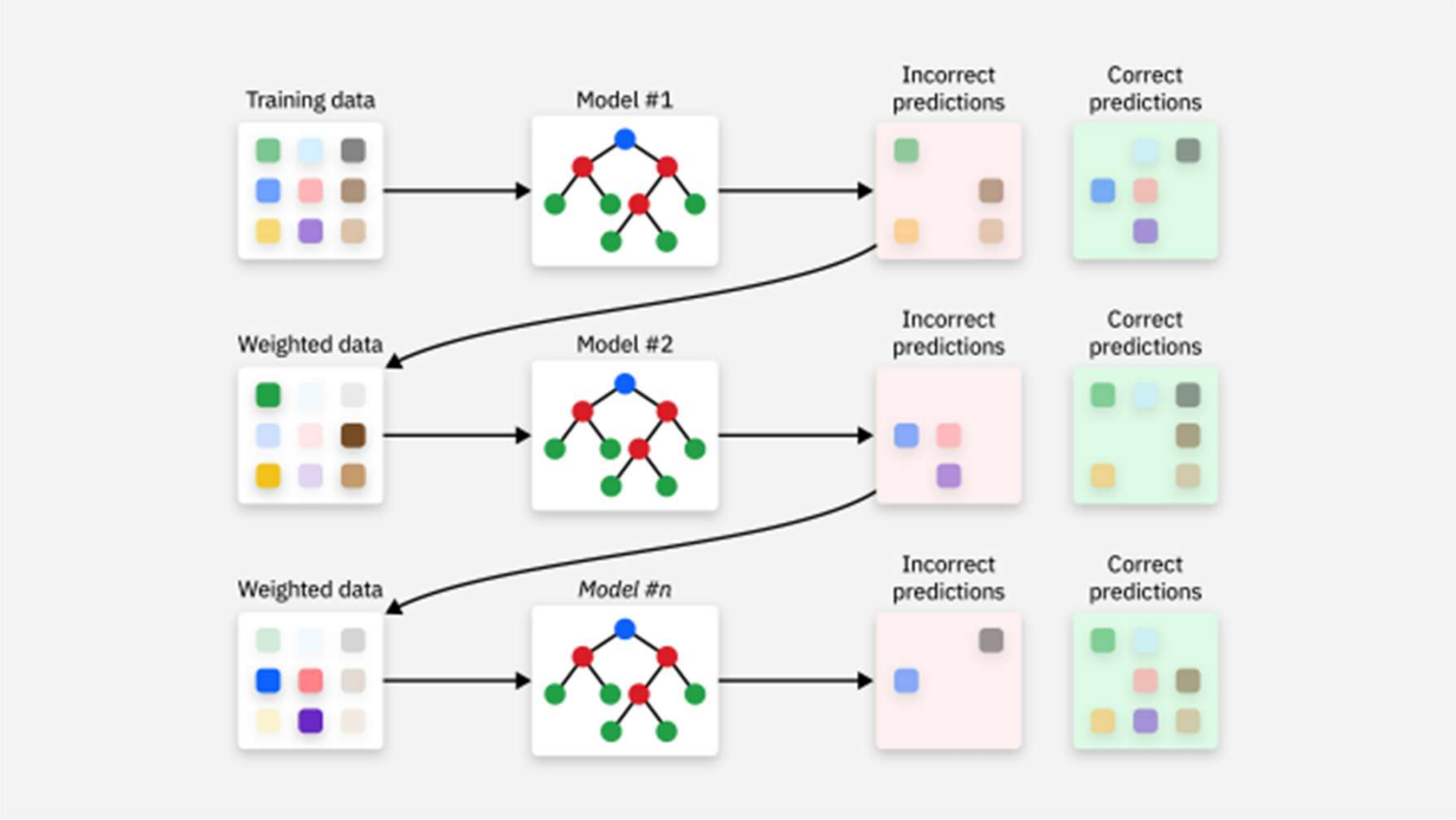
# Ensemble methods

**Τεχνικές**

Boosting

Ένας νέος learner εκπαιδεύεται σε αυτό το νέο σύνολο δεδομένων d2. Στη συνέχεια, ένα τρίτο σύνολο δεδομένων (d3) καταρτίζεται από τα d1 και d2, δίνει προτεραιότητα στα εσφαλμένα ταξινομημένα δείγματα του δεύτερου learner και στις περιπτώσεις στις οποίες τα d1 και d2 διαφωνούν. Η διαδικασία επαναλαμβάνεται n φορές για την παραγωγή n μαθητών. Στη συνέχεια, το boosting συνδυάζει και σταθμίζει όλους τους learners μαζί για να παράγει τις τελικές προβλέψεις

# Ensemble methods




Source: https://www.ibm.com/topics/ensemble-learning

# Random Forests

Ο Random Forest είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος μηχανικής μάθησης, ο οποίος συνδυάζει την έξοδο πολλαπλών δέντρων απόφασης για να καταλήξει σε ένα ενιαίο αποτέλεσμα.

Η ευκολία χρήσης και η ευελιξία του έχουν καταστήσει δημοφιλή, καθώς χειρίζεται τόσο προβλήματα ταξινόμησης όσο και παλινδρόμησης

# Random Forests

Ο αλγόριθμος random forest αποτελεί επέκταση της μεθόδου bagging, καθώς χρησιμοποιεί τόσο το bagging όσο και την τυχαιότητα των χαρακτηριστικών για τη δημιουργία ενός ασυσχέτιστου δάσους δέντρων απόφασης.

Η τυχαιότητα των χαρακτηριστικών, επίσης γνωστή ως feature bagging ή «the random subspace method», δημιουργεί ένα τυχαίο υποσύνολο χαρακτηριστικών, το οποίο εξασφαλίζει χαμηλή συσχέτιση μεταξύ των δέντρων απόφασης.

Αυτή είναι μια βασική διαφορά μεταξύ των δέντρων απόφασης και των τυχαίων δασών. Ενώ τα δέντρα αποφάσεων εξετάζουν όλες τις πιθανές διαχωρίσεις χαρακτηριστικών, τα τυχαία δάση επιλέγουν μόνο ένα υποσύνολο αυτών των χαρακτηριστικών.

# Random Forests

Οι αλγόριθμοι Random Forest έχουν τρεις κύριες υπερπαραμέτρους, οι οποίες πρέπει να οριστούν πριν από την εκπαίδευση.

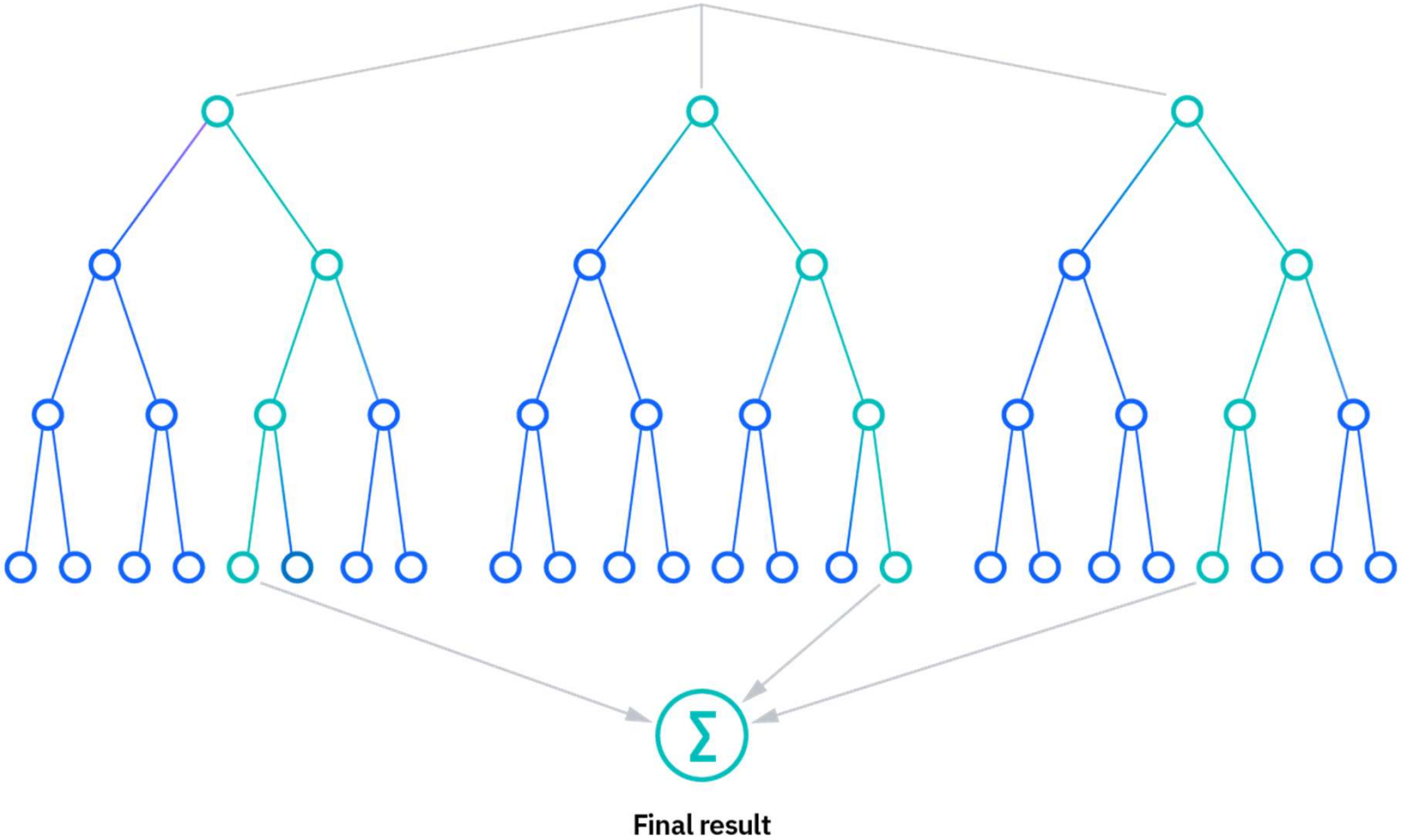
- το μέγεθος των κόμβων,
- τον αριθμό των δέντρων,
- τον αριθμό των δειγματοληπτικών χαρακτηριστικών.

Από εκεί και πέρα, ο ταξινομητής τυχαίου δάσους μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων παλινδρόμησης ή ταξινόμησης.

# Random Forests

Ο αλγόριθμος τυχαίου δάσους αποτελείται από μια συλλογή δέντρων απόφασης και κάθε δέντρο στο σύνολο αποτελείται από ένα δείγμα δεδομένων που αντλείται από ένα σύνολο εκπαίδευσης με αντικατάσταση, το οποίο ονομάζεται δείγμα bootstrap. Από αυτό το δείγμα εκπαίδευσης, το ένα τρίτο αυτού του δείγματος τίθεται στην άκρη ως δεδομένα δοκιμής, γνωστό ως δείγμα εκτός σάκου (**o**ut **o**f **b**ag -oob).

# Random Forests




Source: https://www.ibm.com/topics/random-forest

# Μηχανική Μάθηση & Εξόρυξη Γνώσης

Ερωτήσεις

?

# Βιβλιογραφία


- Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου, Τεχνητή Νοημοσύνη - Γ' Έκδοση, ISBN: 978-960-8396-64-7, Έκδοση/Διάθεση: Εκδόσεις Πανεπιστημίου Μακεδονίας, 2011
- Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Κ. Διαμαντάρας, Ι. Μπότσης, Μηχανική Μάθηση – Α' Έκδοση, ISBN: 978-960-461-955-5, Εκδόσεις Κλειδάριθμος, 2019
- P.-N. Tan, M.Steinbach,  V. Kumar, «Introduction to Data Mining», Addison Wesley, 2006