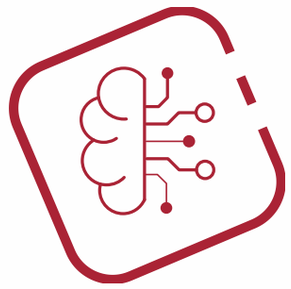
ITSAR ISTITUTO TECNICO SUPERIORE
ANGELO RIZZOLI

# Algoritmi per il Machine Learning

Ing Andrea Colleoni



# Algoritmo

Dette:

- x = {x1, …, xj}: il vettore di attributi
- Θ : valore soglia di un attributo xj in cui partizionare il set di dati
- y il vettore obiettivo

1. Se il dataset D soddisfa un criterio di terminazione, non fare nulla. Altrimenti:
   a) per ogni combinazione j; Θ :
      I. Calcola i due sotto-dataset Dxj< Θ e Dxj >= Θ;
      II. Calcola l'impurità attesa;
      III. Se l'impurità è la minore trovata finora (IG più elevato), ricorda i valori ottimali j* e Θ*;
   b) Associa i parametri migliori (j* e Θ*) alla radice e genera due figli, sinistro e destro, associati rispettivamente a Dxj< Θ e Dxj>= Θ.
   c) Applica ricorsivamente la procedura ai figli sinistro e destro.

# Algoritmo ID3

Se la variabile xj in ingresso è categorica a due valori {v1,v2}, l'algoritmo già descritto richiede la sola sostituzione della domanda "xj< θ" con la domanda "xj=v1".

Se la variabile ha più di due valori, sono possibili due varianti:

1. la domanda assume la forma "xj=vk", quindi pone un valore specifico della classe contro tutti gli altri, oppure
2. l'albero non è più binario, ma il nodo ha tanti sottoalberi quanti sono i valori che xj può assumere.

Quest'ultima variante, quand'è applicata a variabili di ingresso puramente categoriche, è solitamente nota come "algoritmo ID3".

Si osservi come, discendendo l'albero risultante, ogni colonna di input xj venga usata una sola volta, perché a monte di una domanda il valore di xj è determinato, quindi l'albero non può avere profondità maggiore del numero di colonne n del dataset.

# Random Forest

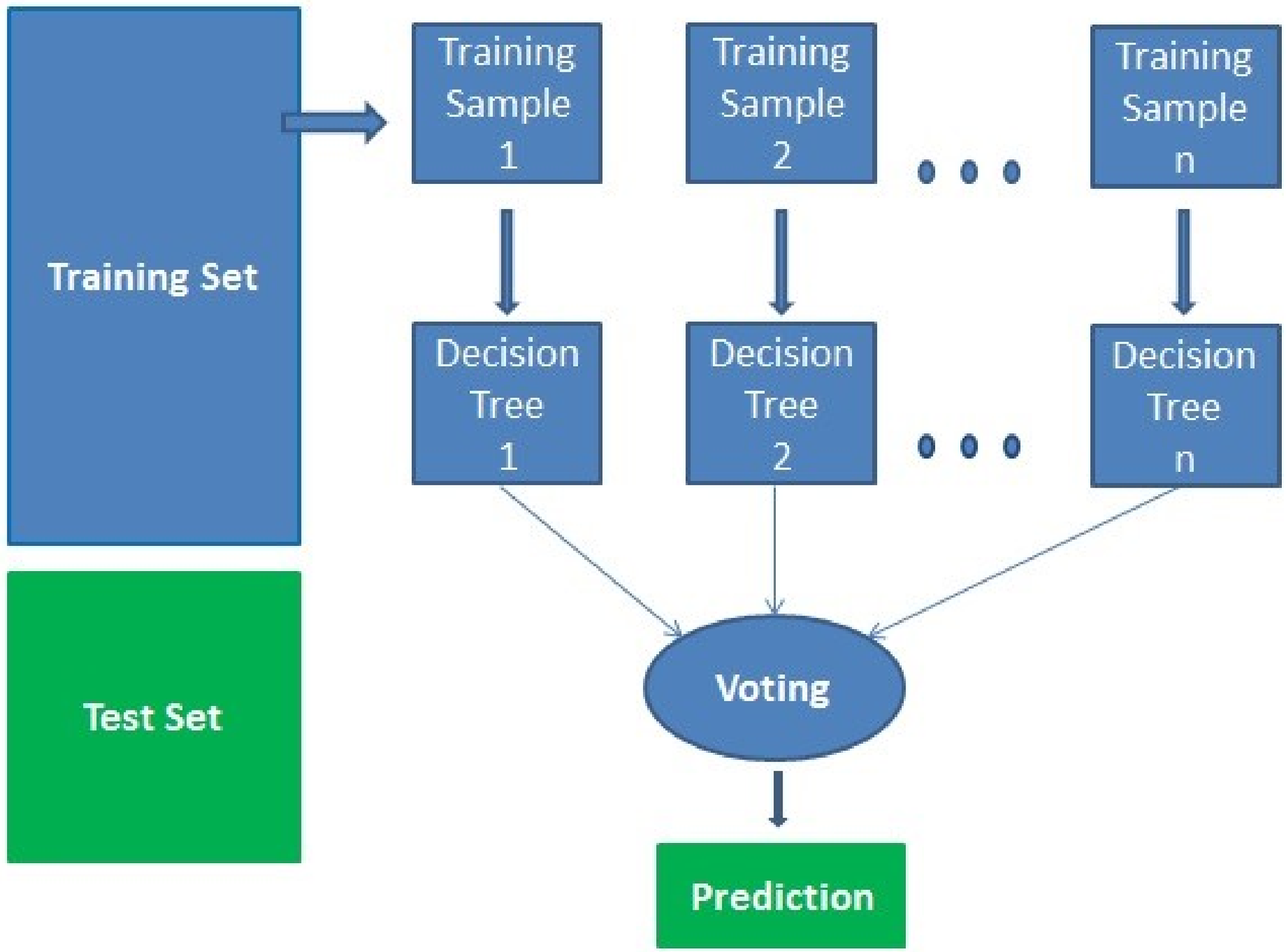
Random Forest is a powerful and versatile **supervised machine learning algorithm** that grows and combines multiple decision trees to create a “forest.” It can be used for both classification and regression problems in R and Python.

Random Forest grows multiple decision trees which are merged together for a more accurate prediction.

The logic behind the Random Forest model is that multiple uncorrelated models (the individual decision trees) perform much better as a group than they do alone.

When using Random Forest for **classification**, each tree gives a classification or a “vote.” The forest chooses the classification with the majority of the “votes.” When using Random Forest for **regression**, the forest picks the average of the outputs of all trees.

# Random Forest



# Training a Random Forest: bagging

**Bagging = Bootstrap + Aggregation**

- **Bootstrap** randomly performs row sampling and feature sampling from the dataset to form sample datasets for every model.
- **Aggregation** reduces these sample datasets into summary statistics based on the observation and combines them. Bootstrap Aggregation can be used to reduce the variance of high variance algorithms such as decision trees.

# Random Forest from scratch

**https://github.com/SebastianMantey/Random-Forest-from-Scratch**

# Selezione degli attributi

Spesso, l'insieme degli attributi presentati da un dataset è sovrabbondante: molti attributi possono essere irrilevanti per la predizione dell'output, altri possono essere inutilmente ripetitivi.

La necessità di trattare un numero eccessivo di attributi comporta un certo numero di problemi:

- i tempi di apprendimento (ovvero di costruzione del modello) si allungano;
- il rischio di overfit è maggiore, in quanto il modello ha più informazioni sulle quali "specializzarsi";
- più dati possono comportare più rumore, quindi predizioni meno precise.

# Algoritmi per la selezione

- Metodi “wrapper”: dato un modello, lo utilizzano per capire in che modo ogni attributo o combinazione di attributi influiscono sul risultato; sono computazionalmente costosi, ma visto che possono prendere decisioni basate sulle prestazioni del modello sono molto efficaci. Ad esempio, un sistema che riaddestra una rete neurale con sottoinsiemi di attributi diversi cercando di minimizzare l’errore.
- Metodi “filter”: applicati prima di addestrare un modello, utilizzano misure statistiche (correlazione, informazione mutua) per cercare di prevedere l’influenza dei vari attributi sulla variabile di uscita.
- Metodi “embedded”: sono parte integrante dell’algoritmo di addestramento del modello; ad esempio, la selezione dell’attributo da utilizzare a un certo nodo di un albero di decisione può rientrare in questa categoria, e una volta che l’albero è stato addestrato gli attributi “selezionati” sono quelli che vengono effettivamente utilizzati in qualche nodo.

# Dati continui: il coefficiente di correlazione

Se X e Y sono continue, la loro covarianza può essere stimata come:

$$\sigma_{X,Y} \sim \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y}),$$

dove $\bar{x}$ e $\bar{y}$ sono i valori medi delle osservazioni di X e Y rispettivamente. La covarianza tende ad essere positiva quando gli scarti delle $x_i$ e delle $y_i$ rispetto alla media tendono ad avere segno concorde, negativa se tendono ad essere discordi: cattura quindi possibili dipendenze lineari (affini) fra X e Y.

La covarianza però dipende molto dalla magnitudine di X e Y, e quindi le covarianze fra variabili diverse non sono sempre confrontabili.

Per migliorare la confrontabilità, possiamo normalizzare la covarianza rispetto alla variabilità delle due variabili casuali, misurata in base allo scarto quadratico medio, ottenendo il coefficiente di correlazione di Pearson.

$$\rho_{X,Y} = \frac{\sigma_{X,Y}}{\sigma_X \sigma_Y}.$$

Il coefficiente varia tra −1 e 1. I valori estremi significano che i punti (xi, yi) sono perfettamente allineati. Un valore pari a 0 non significa che non vi siano dipendenze, ma solo che queste non sono lineari.

# Variabili categoriche: l'informazione mutua

Abbiamo già visto il concetto di entropia H(Y) di una variabile casuale Y. Data una seconda variabile casuale X, l'entropia *condizionata* H(Y|X) risponde alla domanda "Quanti bit servono mediamente per comunicare l'esito di Y se si è già a conoscenza dell'esito di X?" Ovviamente, se X e Y non sono del tutto indipendenti, allora l'osservazione di X conterrà qualche informazione sull'esito di Y, quindi in generale:

$$H(Y|X) \leq H(Y),$$

con l'uguaglianza verificata quando X e Y sono indipendenti.
L'entropia condizionata `e il valore atteso dell'entropia di Y al variare di X fra i suoi possibili valori:

$$H(Y|X) = \sum_{x \in X} \Pr(X = x) H(Y|X = x).$$

Infine, l'informazione mutua I(X;Y) di X e Y è la diminuzione dell'entropia di Y data dalla conoscenza di X:

$$I(X;Y) = H(Y) - H(Y|X).$$

# Calcolo dell'informazione mutua

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)},$$

Dove x e y variano rispettivamente nel dominio delle variabili casuali X e Y, p(x,y) = Pr(X=x ∧ Y=y) è la probabilità congiunta, mentre p(x) e p(y) sono le probabilità marginali.

Dalla definizione stessa, è chiaro che l'informazione mutua varia fra 0 e l'entropia diY2.

Alcuni casi particolari:

- I(X;Y) = 0 significa che H(Y) =H(Y|X), ossia che la conoscenza della sola X è ininfluente sulla determinazione di Y;
- I(X;Y) =H(Y) significa che H(Y|X) = 0, quindi l'osservazione di X determina esattamente il valore di Y, e non c'è bisogno di informazioni aggiuntive per comunicarne il valore.

In generale, più alto è il valore di I(X;Y) più possiamo concludere che la conoscenza di X è importante per determinare il valore di Y.