

Machine Learning Alberi di Decisione Random Forest

Francesco Pugliese, PhD
neural1977@gmail.com

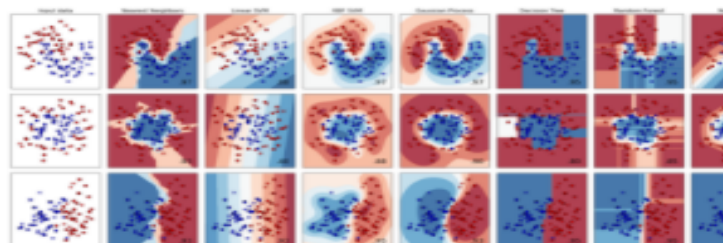
Machine Learning e Universo di Scikit-Learn

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

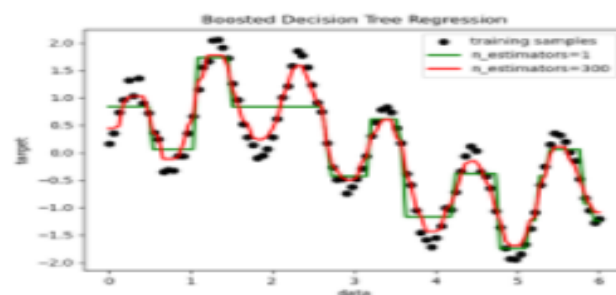


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...

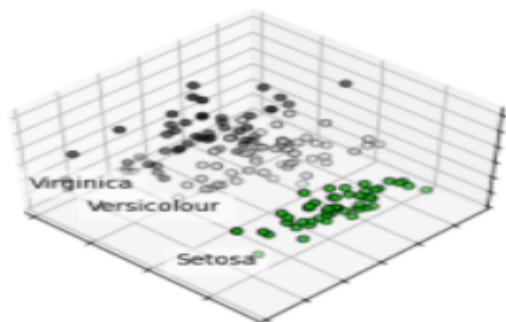


Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

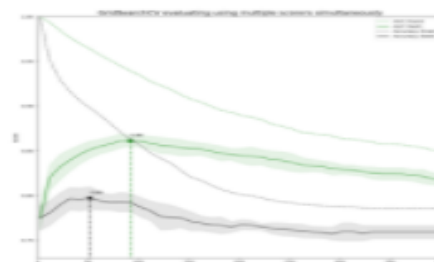


Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...

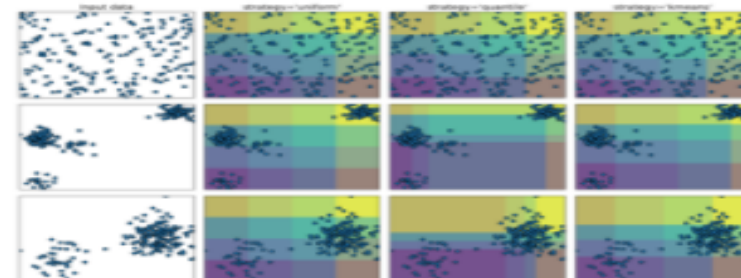


Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

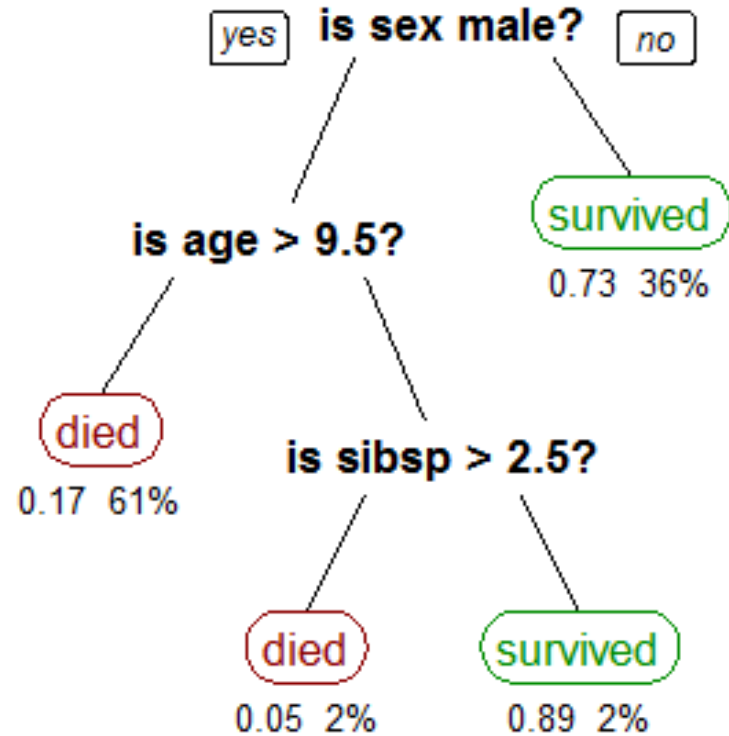
Algorithms: preprocessing, feature extraction, and more...



Alberi di Decisione

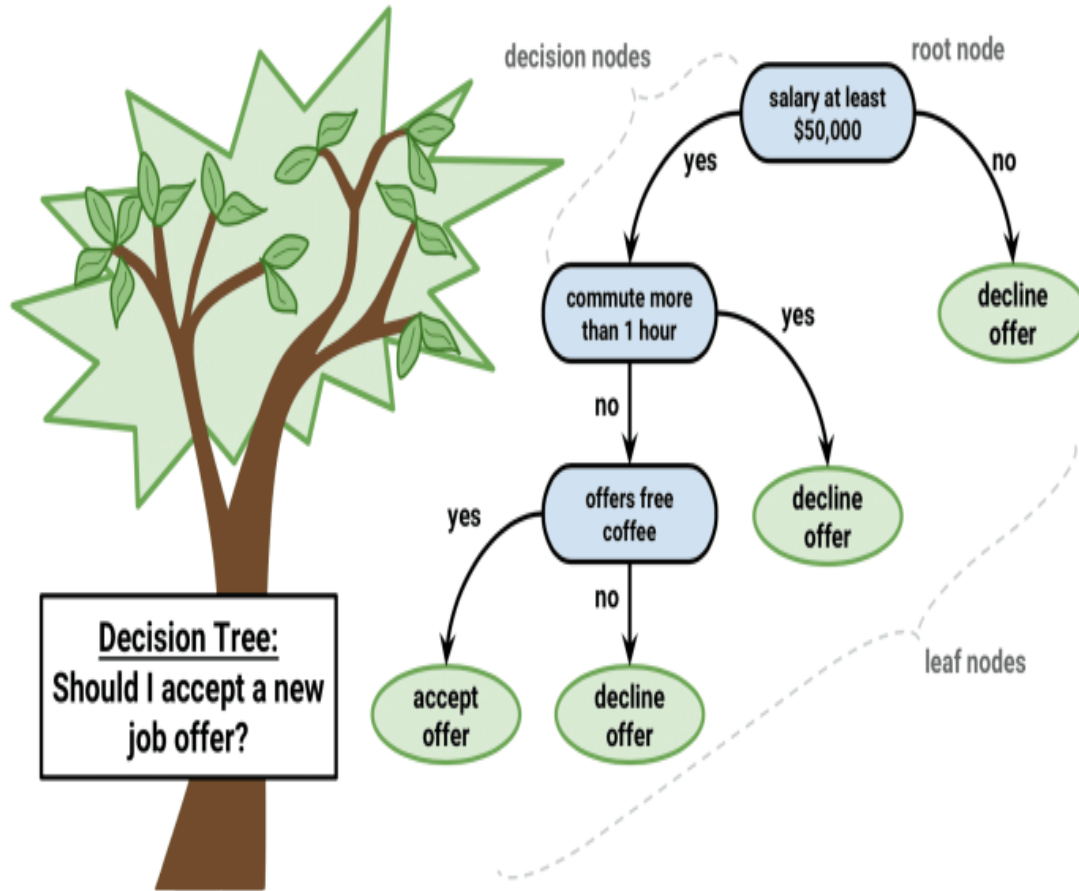
- ✓ Un albero di decisione è un tool di **supporto alle decisioni** che usa un modello ad albero per il processo di decision-making process e tutte le possibili **conseguenze** derivanti dalla decisione.
- ✓ Esso copre risultati di eventi, riduzione di costo e utilità nelle decisioni.
- ✓ Ricorda un algoritmo o un **diagramma di flusso** che contiene solo statement di controllo condizionale.
- ✓ Ciascun albero di decisione è costituito da 3 parti principali: un nodo **root**, nodi **foglia**, e **rami**.
- ✓ In un albero di decisione, ciascun nodo interno rappresenta un test o un evento. Diciamo un testa o croce nel lancio di una moneta.
- ✓ Ciascun **ramo** rappresenta **l'output** del test e ciascun **nodo** rappresenta una label di **classe – ossia una decisione presa dopo l'elaborazione di tutti gli attribute**.
- ✓ I **percorsi** dalla root ai nodi foglia rappresentano le **regole di classificazione**.

Alberi di Decisione



- In un **Albero di Decisione**, per predire (assegnare) una etichetta di **classe** per un dato record (riga o data point) possiamo partire dalla **radice** dell'albero. Confrontiamo i valori dell'attributo radice con gli attributi del record. Sulla base del confronto, seguiamo il ramo corrispondente a quel valore e saltiamo al nodo successivo.
- Gli **Alberi di Decisione** come suggerisce il nome lavorano su un insieme di decisioni derivate dai dati e dal suo comportamento.
- Essi non usano un classificatore lineare o un regressore, pertanto le sue prestazioni sono indipendenti dalla natura lineare dei dati o meno.

Alcuni Concetti relative agli Alberi di Decisione



1.Nodo Radice: Esso rappresenta l'intera popolazione o un campo e questo ulteriormente si divide in due o più insiemi omogenei.

2.Splitting: Rappresenta il processo di divisione di un nodo in due o più sotto-nodi.

3.Nodo Decisionale: Quando un sotto-nodo si suddivide in ulteriori sotto-nodi, questo prende il nome di nodo decisionale.

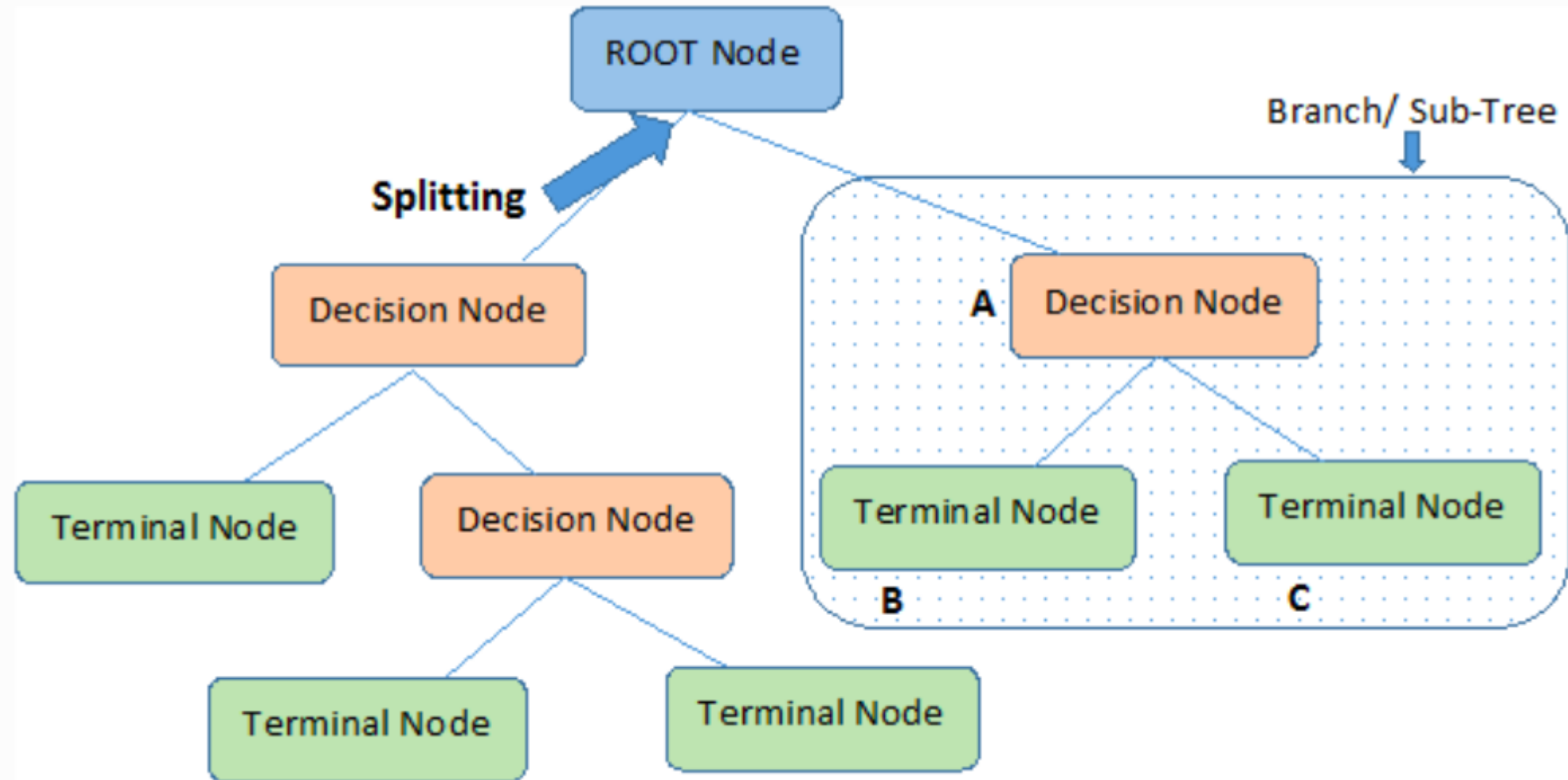
4.Nodi Foglia o Terminali: I nodi che non sono splittati in ulteriori nodi prendono il nome di Nodi Foglia e Nodi Terminali.

5.Potatura dell'Albero (Pruning): Quando rimuoviamo sotto-nodi da un nodo decisionale, questo processo prende il nome di pruning (potatura). In altre parole è l'opposto del processo di splitting.

6.Branch / Sotto-Albero: Una sottosezione di un intero albero viene chiamata branch o sotto albero.

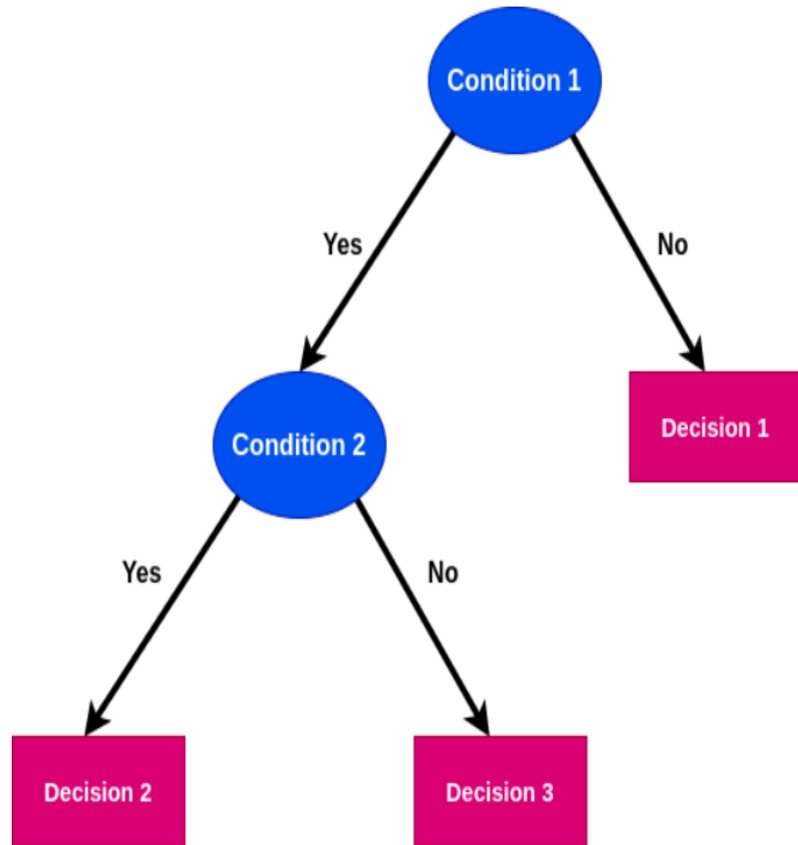
7.Nodo Padre e Nodo Figlio: Un nodo, che viene diviso in sotto-nodi viene chiamato nodo padre mentre I suoi sotto-nodi prendono il nome di nodi figli.

Alcuni Concetti relative agli Alberi di Decisione



Note:- A is parent node of B and C.

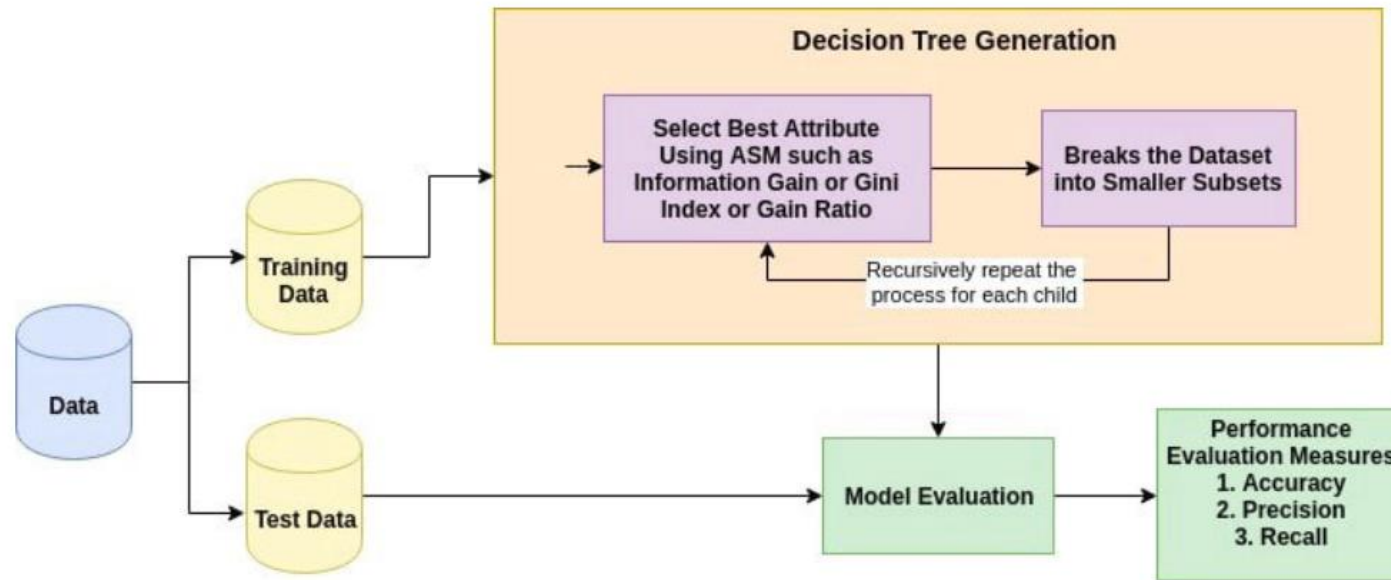
Addestramento di un Albero di Decisione



- Gli **alberi di decisione** usano un algoritmo **CART** (Classification and Regression Trees). In entrambi i casi, le decisioni sono basate su condizioni su ognuna della feature.
- I nodi interni rappresentano le condizioni e i nodi foglia rappresentano la decisione basata sulle condizioni.
- Un **albero di decisione** è una rappresentazione grafica di tutte le possibili soluzioni ad una decisione basata su certe condizioni.
- Su ciascuno step o nodo di un **albero di decisione**, usato per la classificazione, cerchiamo di creare una condizione sulle feature per separare tutte le label o classi contenute nel dataset dalla **purezza (purity)** più completa.

Addestramento di un Albero di Decisione

- La creazione dei sotto-nodi incrementa **l'omogeneità** dei sotto-nodi risultanti. In altre parole, possiamo dire che la purezza del nodo incrementa con la variabile di target presa in esame.
- Un albero di decisione divide i nodi su tutte le variabili disponibili e seleziona la ripartizione che sfocia nella maggior parte dei sotto-nodi **omogenei**.



Addestramento di un Albero di Decisione



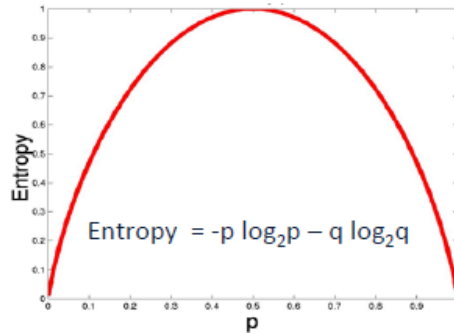
La selezione di un algoritmo è basata sul tipo di variabili di target.

Alcuni algoritmi usati negli Alberi di Decisione sono:

- **ID3** → (estensione di D3)
- **C4.5** → (successore di ID3)
- **CART** → (Classification And Regression Tree)
- **CHAID** → (Detect dell'interazione automatica Chi-quadriforms multi-level trees)
- **MARS** → (Curve di regressione adattiva multivariate)

Addestramento degli Alberi di Decisione

Entropia



a) Entropia del Target (Entropia che usa la tabella delle frequenze su un solo attributo)

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

b) Entropia su ciascun attributo (Entropia che usa la tabella delle frequenze di due attributi insieme)

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

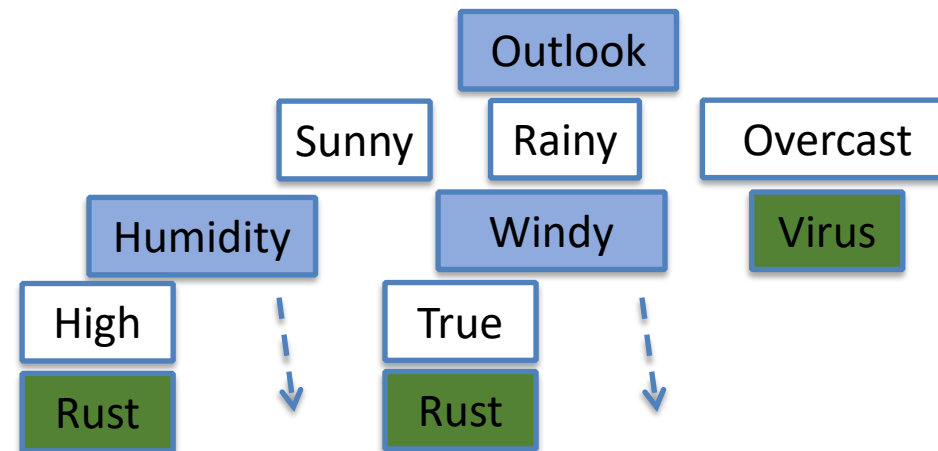
Guadagno di Informazione

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

Algoritmo di Addestramento ID3 (Quinlan, 1986)

Passi:

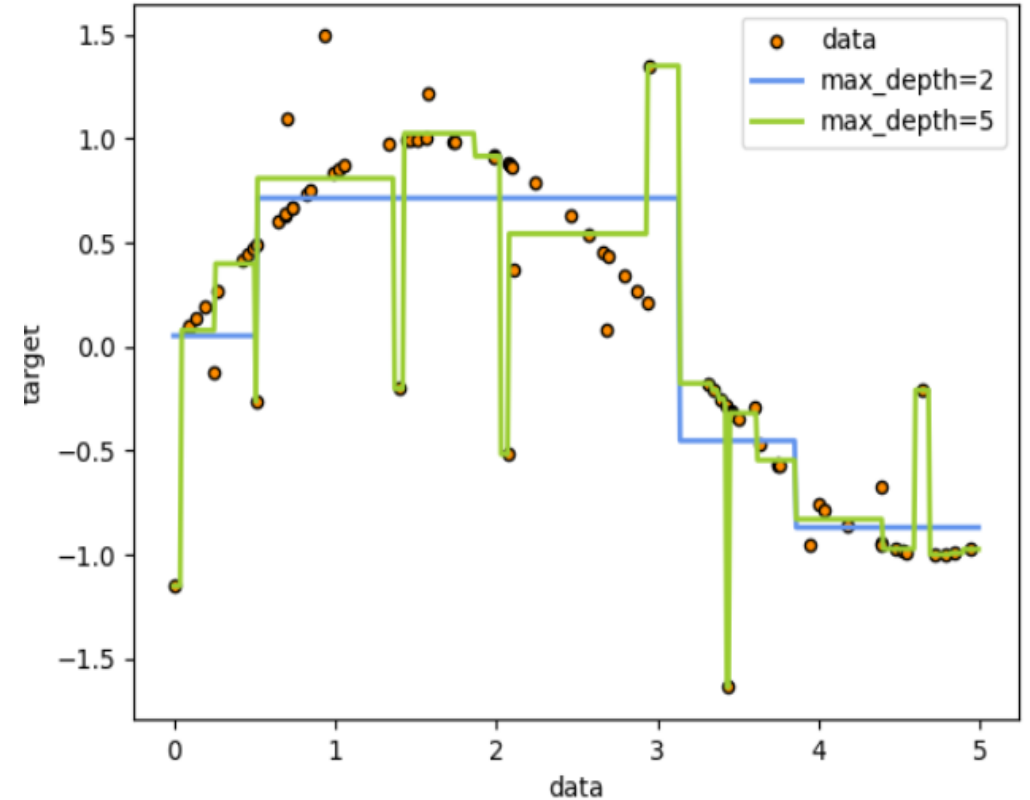
1. Calcolare l'entropia del target.
2. Calcolare il **Guadagno di Informazione** per ciascun attributo. .
3. Scegliere l'attributo con il più grande Guadagno di Informazione possibile, come nodo radice dell'Albero.
4. Ricorsivamente avviare l'algoritmo sui sottoalberi non foglia.



Algoritmo di addestramento degli Alberi di Decisione ID3

Passi dell'algoritmo ID3 (Iterative Dichotomiser 3) usato per generare un albero di decisione da un dataset. ID3 è il precursore dell'algoritmo C4.5:

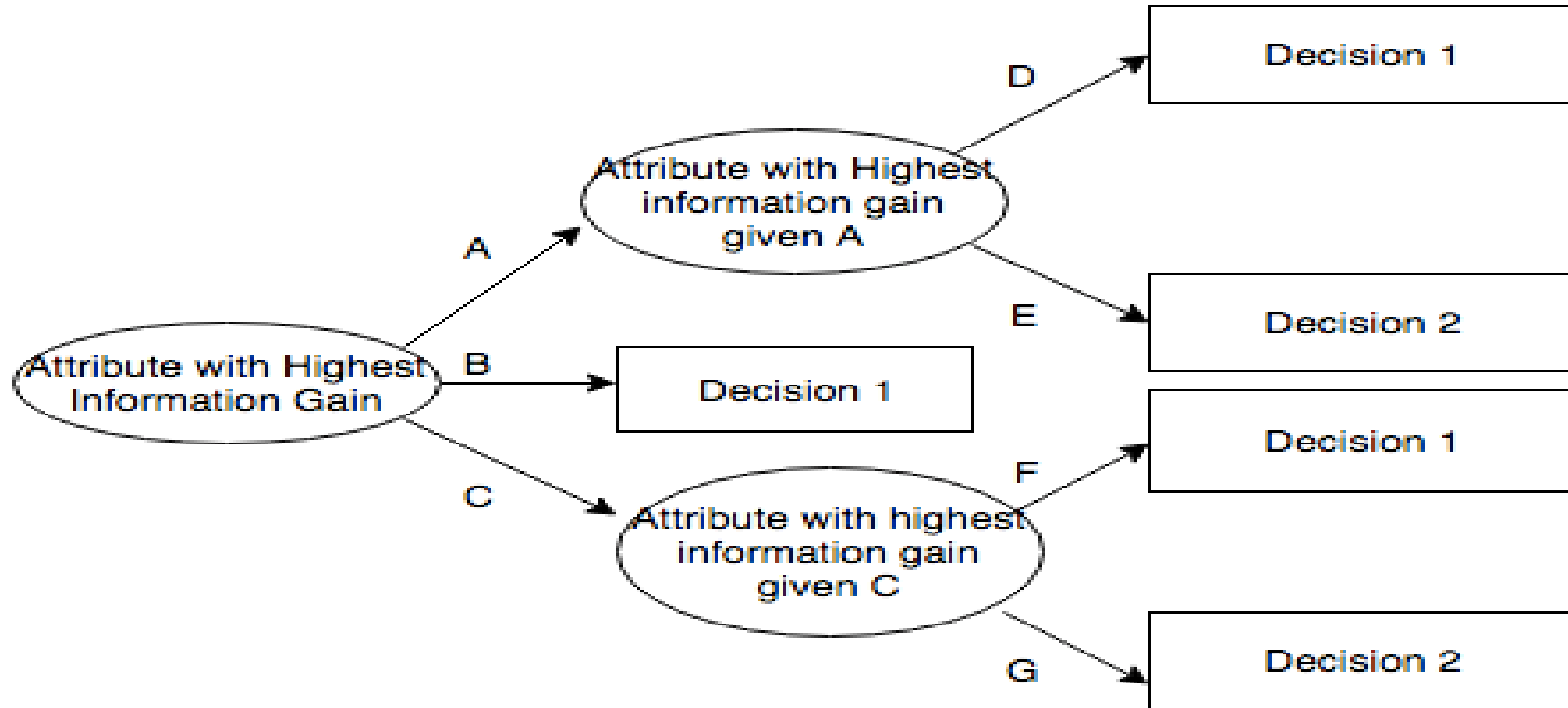
1. Inizia con un insieme d'origine **S** come nodo radice.
2. Per ciascuna iterazione dell'algoritmo, esso itera attraverso gli attribute mai usati di **S** e calcola **Entropy(H)** e **Information Gain(IG)** di questo attribute.
3. Esso poi selezione l'attribute che ha la minore **entropia** o il più grande **Guadagno d'Informazione**.
4. L'insieme **S** è poi suddiviso (splitting) dall'attribute selezionato che funge da spartiacque al fine di produrre un più sottoinsiemi di dati.
5. L'algoritmo continua ricorsivamente su ciascun sottoinsieme, considerando solo attribute mai selezionati prima.



Algoritmo di Addestramento di Alberi di Decisione ID3

- Per esempio, un nodo può essere splittato in nodi figli a seconda dei sottoinsiemi della popolazione le cui età sono meno di 50, tra 50 e 100 e più grandi di 100. L'algoritmo continua la ricorsione su ciascun sottoinsieme, considerando solo attributi mai selezionati prima.
- ✓ La Ricorsione su un insieme si ferma quando:
- 1) Ogni elemento nel sottoinsieme appartiene alla medesima classe: in questo caso il nodo viene convertito in nodo foglia ed etichettato con la classe degli esempi.
 - 2) Non ci sono più attributi da selezionare, ma gli esempi ancora non appartengono tutti alla stessa classe. In questo caso, il nodo è reso un nodo foglia ed etichettato con la classe maggioritaria degli esempi nel sotto insieme.
 - 3) Non ci sono più esempi nel sottoinsieme, il che accade quando nessun esempio nell'insieme padre è stato trovato per matchare uno specific valore dell'attributo selezionato.

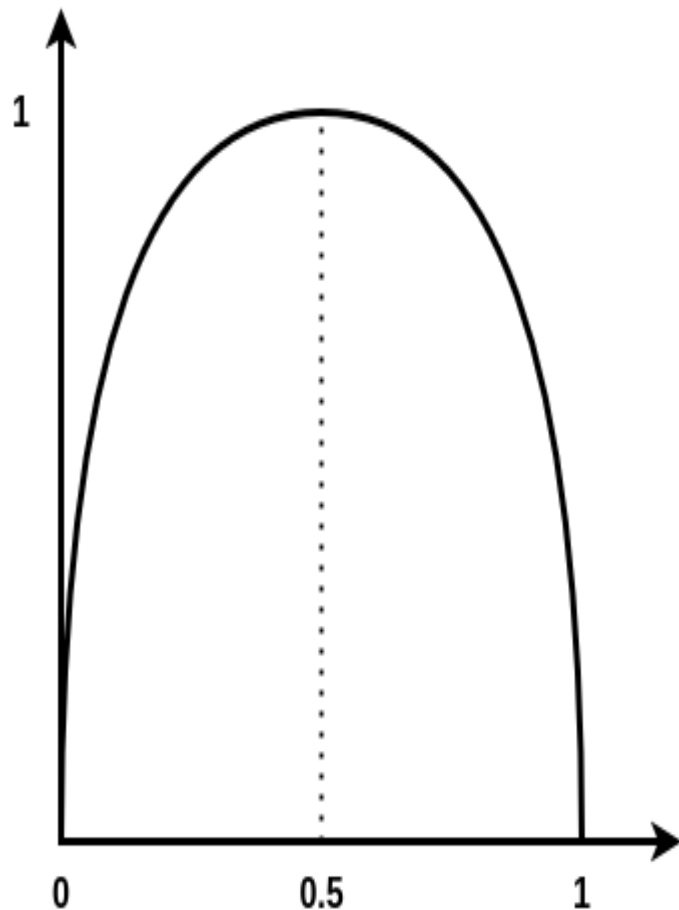
Addestramento degli Alberi di Decisione



Addestramento degli Alberi di Decisione: Entropia

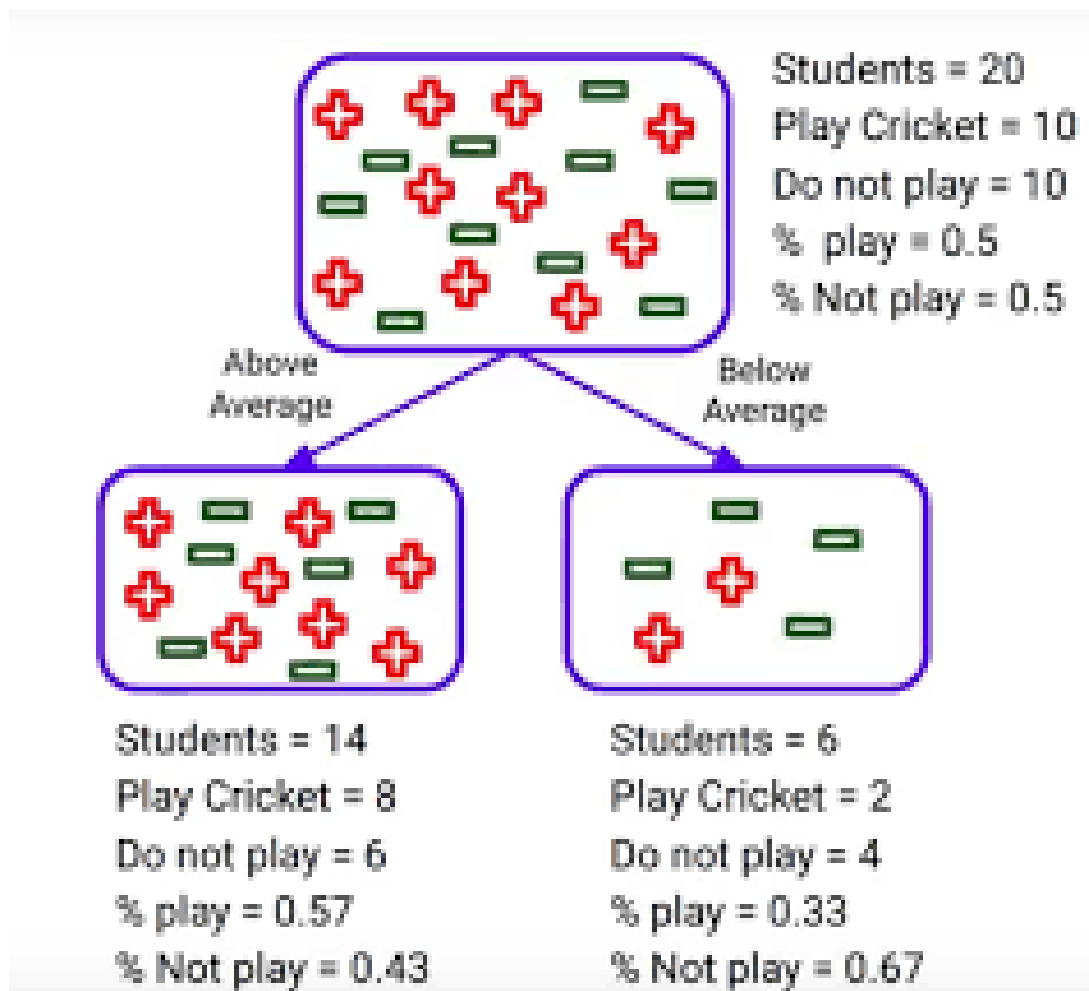
- **Entropia:** essa fornisce la misura dell'impurità o casualità nei dati. Essa è data dalla seguente formula:

$$\text{Entropia} = - P(\text{class 1}) \times \text{Log}(P(\text{class 1})) - P(\text{class 2}) \times \text{Log}(P(\text{class 2})) \quad \text{dove } P \text{ denota la probabilità.}$$



- Se ci sono solo due classi, classe 1 e classe 2, bilanciate, ovvero il numero di dati nella classe 1 è uguale al numero di entry nella classe 2, e selezioniamo casualmente una entry (dato), essa apparterrà a qualsiasi delle classi 1 o 2 con una probabilità del 50% ciascuna. In tali casi, l'entropia sarà alta, praticamente massima, non è possibile decidere.
- Se un certo datase ha tutti i dati appartenenti alla classe 1 o alla classe 2, l'entropia ottenuta è 0, dal momento che in quel caso $P(\text{classe 1})$ or $P(\text{classe 2})$ saranno uguali a 0. Se $P(\text{classe 1}) = 0$ allora $P(\text{class2})$ dovrebbe essere uguale a 1.
- In questo modo è evidente che l'entropia sarà alta solo se abbiamo label di classi impure o mixate in un dataset.
- Il diagramma di lato mostra la variazione della probabilità delle etichette al variare dell'entropia. Possiamo vedere che se la probabilità di un'etichetta (label) è 0.5, allora l'entropia sarà massima e secondo ID3 quell'attributo sarà scartato, perchè impuro, quindi poco significativo ai fini della decisione.

Addestramento di Alberi di Decisione: Guadagno di Informazione




Il **Guadagno di Informazione (Information gain)** è usato per decidere quale feature suddividere (splitting) a ciascuno step della costruzione dell'albero.

Semplicità vuol dire meglio, pertanto vogliamo mantenere il nostro albero più piccolo possibile. Per fare questo, a ciascuno step dovremmo scegliere lo split che risulta il più puro possibile. Per ciascun nodo dell'albero il Guadagno di Informazione, misura quanta informazione una feature ci fornisce nei riguardi di una classe. Uno split con il più grande guadagno di informazione sarà considerato prioritariamente come primo split e il processo continuerà finchè tutti i nodi figli saranno puri, o finchè il guadagno di informazione è 0.


Addestramento di un Albero di Decisione: Guadagno di Informazione

Feature 1	Feature 2	Label
1	1	lab_1
1	1	lab_1
2	1	lab_2
3	2	lab_2
3	3	lab_2
3	3	lab_1
2	3	lab_2
1	2	lab_1
1	3	lab_2
2	2	lab_2

- **Guadagno di Informazione:** Il guadagno di informazione è pari al decremento di entropia dopo che il dataset viene splittato sulla base di un attributo a guadagno massimo. La costruzione di un Albero di Decisione consiste nel trovare l'attributo che restituisce il più alto guadagno d'Informazione. Ciò aiuta a scegliere quale feature o attributo sarà usato per creare il nodo interno di decisione ad un certo punto.



Feature 1==1		Feature 1==2		Feature 1==3
labels		labels		labels
lab_1		lab_2		lab_2
lab_1		lab_2		lab_2
lab_1		lab_2		lab_1
lab_2				



Feature 2==1		Feature 2==2		Feature 2==3
labels		labels		labels
lab_1		lab_2		lab_2
lab_1		lab_1		lab_1
lab_2		lab_2		lab_2
				lab_2

- $\text{Information gain} = \text{Entropy}(s) - [(\text{Media Pesata}) \times (\text{Entropia di ciascuna feature})]$

Information Gain Ft1= 0.313

Information Gain Ft2 = 0.1

Addestramento di un Albero di Decisione: Impurità di GINI

- **Puro**

Puro significa, in un campione selezionato del dataset tutti i dati appartengono alla stessa classe (Puro)

- **Impuro**

Impuro significa, che i dati sono una mistura di classi differenti.

- Definizione di Indice di Impurità di Gini

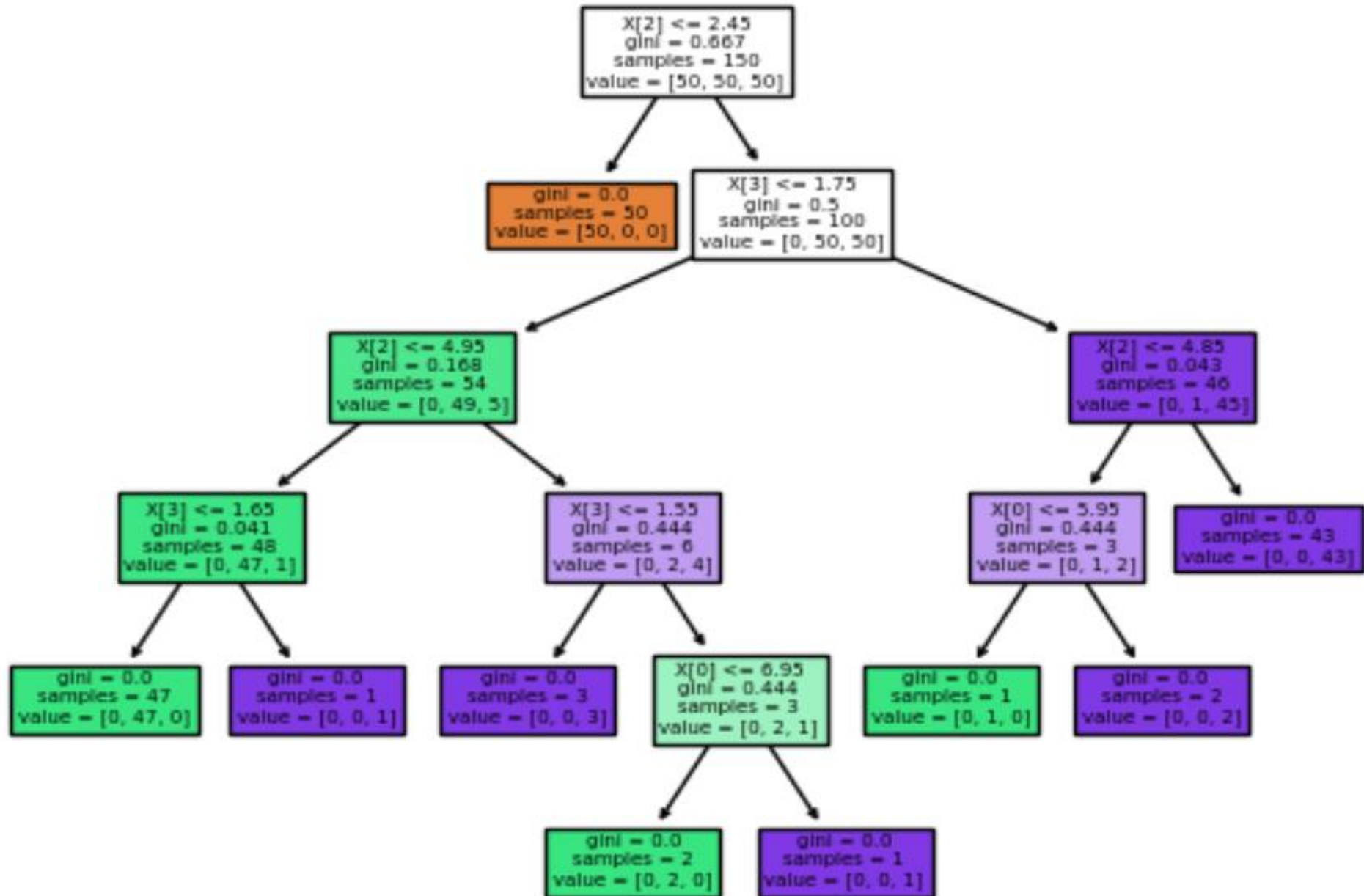
L'Impurità di Gini è una misura della probabilità di una classificazione incorretta di una nuova istanza di una variabile casuale, se quella nuova istanza è stata casualmente classificata a seconda di una distribuzione di etichette di classe da un dataset.

Se il nostro dataset è Puro allora la probabilità di classificazione errata è 0. Se il nostro campione è una mistura di classi differenti allora la probabilità di classificazione errata sarà alta.

$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2$$

Quando usiamo l'indice di Gini come criterio per l'algoritmo di addestramento per la selezione del nodo radice, si prende in considerazione il Gini Index più basso possibile in pratica, in quanto coinciderà a dataset puro.

Addestramento di un Albero di Decisione



Alberi di Decisione

- ✓ Possono rappresentare un potente algoritmo di Machine Learning per la **classificazione** e la **regressione**.
- ✓ L'albero di **Classificazione** lavora sul target al fine di classificare i dati in etichette di una variabile categorical discreta.
- ✓ Invece gli Alberi di **Regressione** sono rappresentati in modo simile, ma predicono valori **continui** come i **prezzi delle case** nel vicinato.
- ✓ Il miglior aspetto degli alberi di decisione:
 - ✓ Gestiscono sia dati **numerici** che **categorici**
 - ✓ Gestiscono problem **multi-output**
 - ✓ Gli alberi di Decisione richiedono uno sforzo relativamente inferior nella preparazione dei dati in quanto gestiscono direttamente dati simbolici senza necessità di **normalizzazione** o **standardizzazione** per esempio.
 - ✓ Le relazioni **non lineari** tra i parametri non influenzano le prestazioni dell'albero.

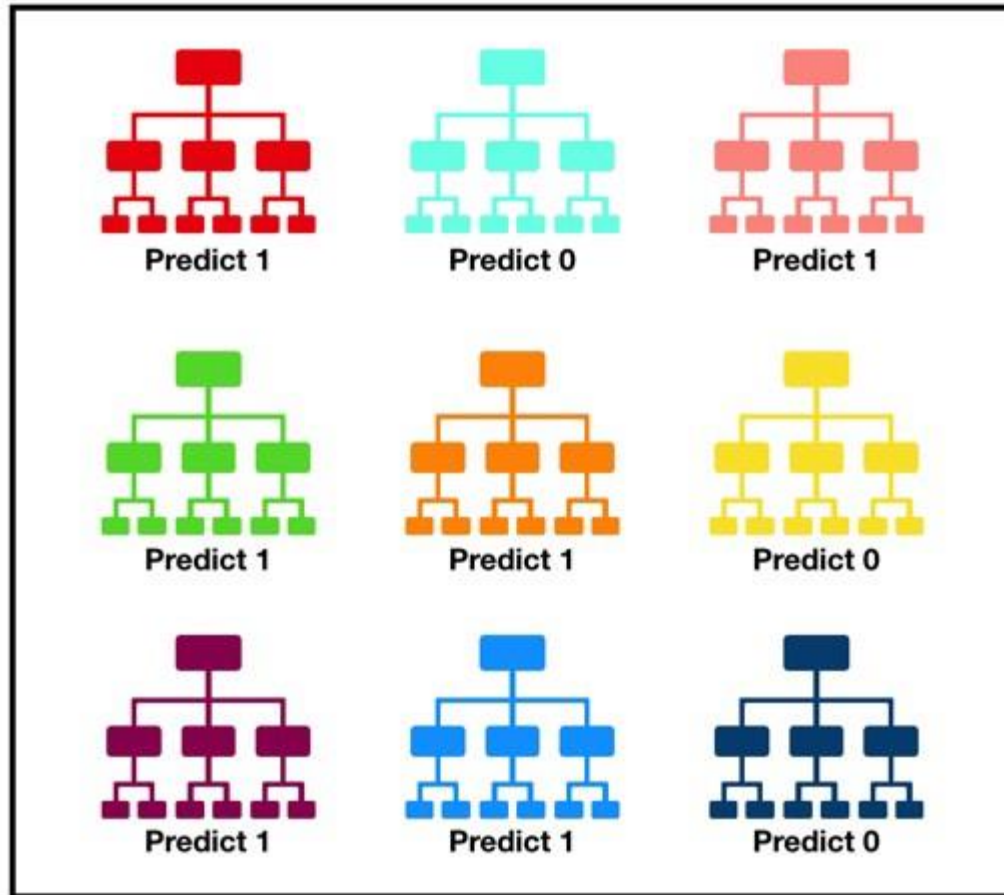
Applicazioni degli Alberi Decisionali

- ✓ Decidere un volo per un viaggio
- ✓ Predire l'occupazione delle date per hotel
- ✓ Decidere il numero di drogherie nelle vicinanze di interesse particolare per il cliente X
- ✓ Classificazione di cellule cancerogene vs cellule non-cancerogene
- ✓ Suggestire ad un cliente quale automobile acquistare

Random Forest

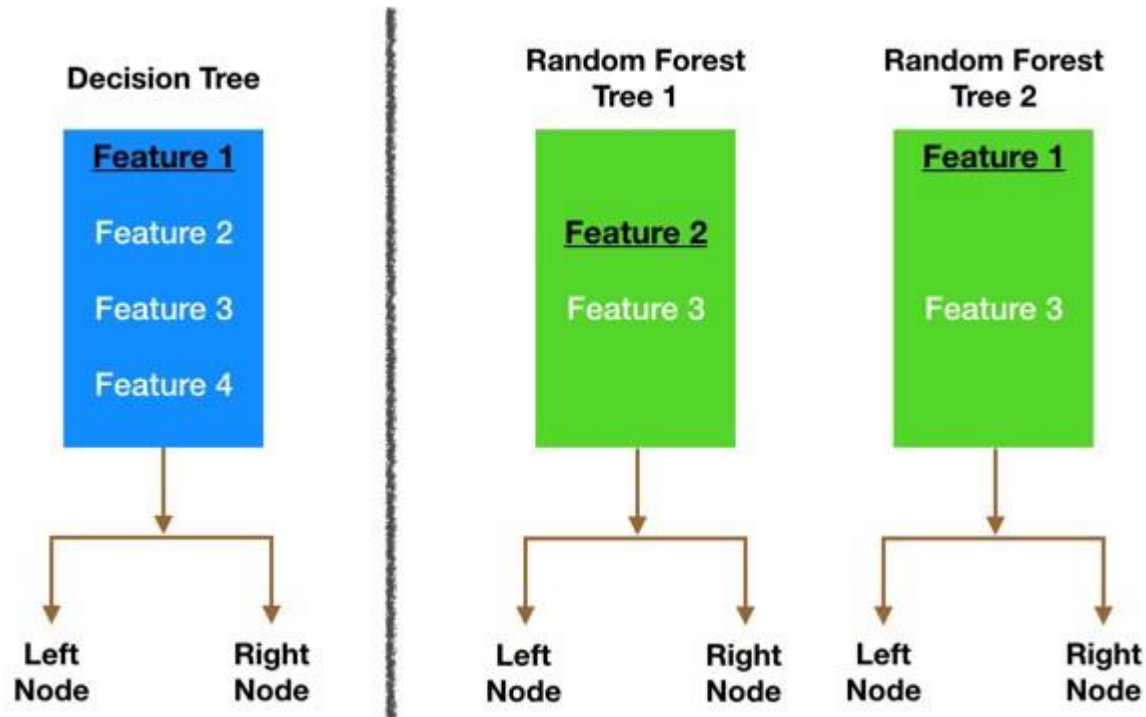
- ✓ E' una tecnica di "**ensemble learning**" comprendente classificazione, regressione e altre operazioni che dipendono da una moltitudine di alberi decisionali a tempo di addestramento.
- ✓ E' veloce, flessibile, e rappresenta un robusto approccio al mining di dati ad alta dimensionalità ed è un'estensione degli alberi di decisione per classificazione e regressione di cui abbiamo parlato poco fa.
- ✓ L' "Ensemble learning", in generale, può essere definito come un modello che effettua delle predizioni combinando i risultati di modelli individuali.
- ✓ Il modello ensemble tende ad essere molto più flessibile con meno bias e meno varianza.
- ✓ L'Ensemble Learning ha due metodi popolari come:
 - ✓ **Bagging**: Ciascun albero individuale viene addestrato su un campione casuale del dataset risultante in diversi alberi.
 - ✓ **Boosting**: Ciascun albero/modello individuale apprende da errori fatti dal modello precedente e migliora di volta in volta.

Random Forest



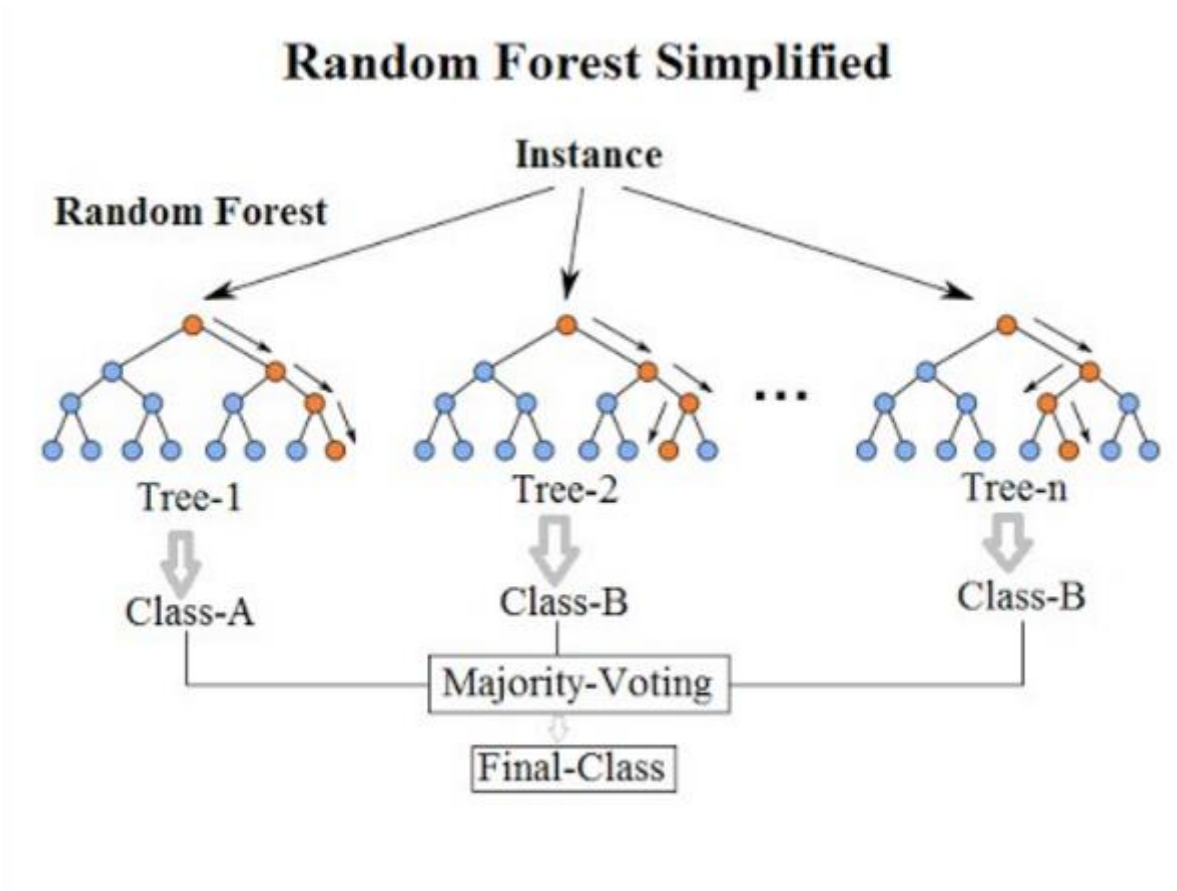
- **Random forest**, come implica il nome stesso, è costituito da un gran numero di alberi di decisione individuali (foresta) che operano come un ensemble di modelli. Ciascun albero individuale all'interno della foresta casuale produce una predizione della classe e la classe con un'elezione a maggioranza (classe più frequente) diventa la predizione del modello ensemble.
- Un gran numero di modelli relativamente non correlate tra loro (alberi) operano come un comitato che supera in prestazioni qualsiasi altro modello individuale costituente il comitato (ensemble).
- **Bagging (Bootstrap Aggregation)** — Gli Alberi di Decisione sono molto sensibili ai dati su cui vengono addestrati, dunque piccoli cambiamenti sul training set possono generare strutture di alberi differenti. Random forest trae vantaggio da questo permettendo a ciascun albero individuale di campionare casualmente il dataset con rimpiazzo, generando differenti alberi. Questo processo è noto come **bagging**.

Random Forest: Bagging



- Non andiamo a divider il training set in piccoli gruppi e addestrare ciascun albero su un diverso chunk. Piuttosto, se abbiamo un campione di dimensione N , cerchiamo di alimentare ciascun albero con un training set di dimensione N (a meno che non sia specificato altrimenti). Tuttavia invece dei dati di training originali, prendiamo un campione casuale di dimensione N con rimpiazzo.
- Per esempio, se i nostril training data fossero $[1, 2, 3, 4, 5, 6]$ allora potremmo alimentare uno dei nostril alberi con la seguente lista di elementi $[1, 2, 2, 3, 6, 6]$. Nota che entrambe le liste sono di lunghezza 6, tuttavia sia il "2" che l'elemento "6" sono entrambi ripetuti nei dati di training estratti casualmente e poi forniti in pasto all'albero (questo vuol dire che campioniamo con rimpiazzo).

Random Forest



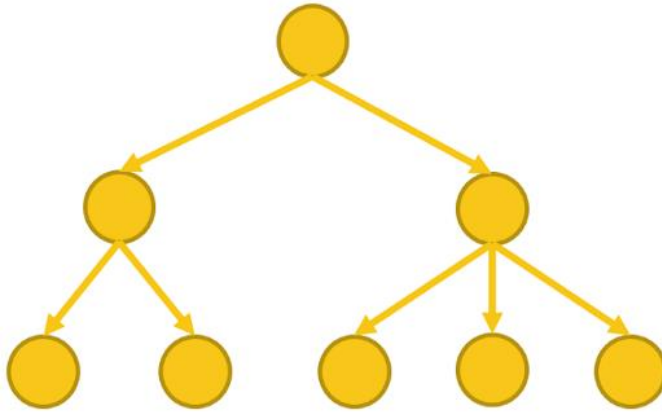
Casualità delle Feature — In un normale albero di decisione, quando è il mometo di splittare un nodo consideriamo ogni possibile feature ed estraiamo quella che produce la migliore separazione tra le osservazioni nel nodo di sinistra e quelle che sono nel nodo di destra. Al contrario, ciascun albero in un Random Forest può estrarre solo un sottoinsieme random di feature. Questo forza la presenza anche di una maggiore variazione tra gli alberi all'interno del modello e alla fine produce una minore correlazione tra gli alberi ed una maggiore diversificazione.

Random Forest

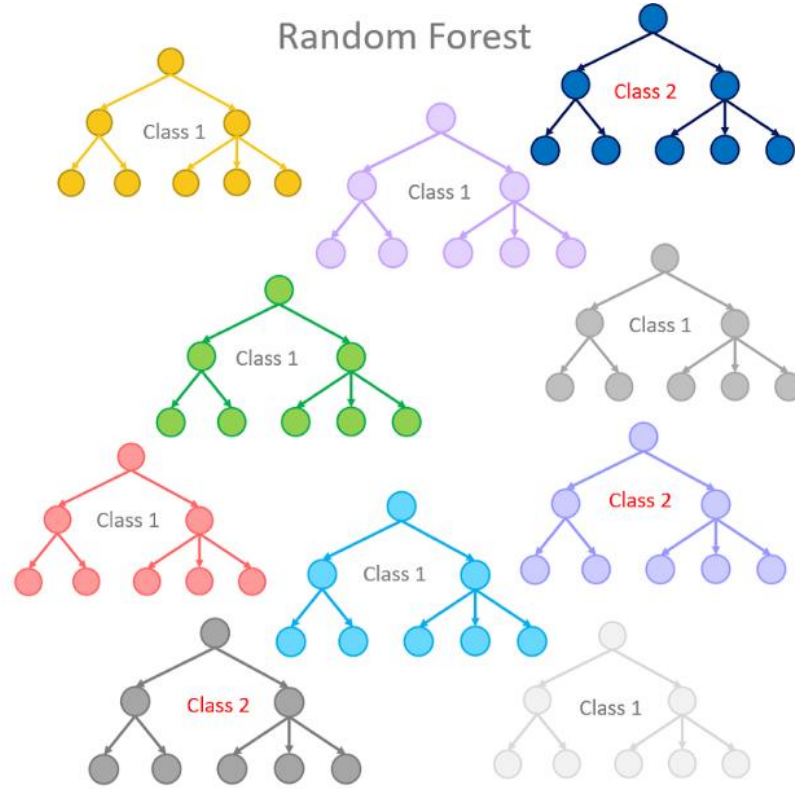
- ✓ I tempi di esecuzione di Random Forest sono piuttosto veloci.
- ✓ Questo algoritmo è piuttosto efficiente con i dati che presentano dati mancanti o dati incorretti.
- ✓ Sui dati negative, RF non può predire oltre l'intervallo definite nei dati di addestramento, e può over-fittare i dataset che sono particolarmente rumorosi.
- ✓ Un Rando Forest dovrebbe avere un numeri di alberi tra **64–128**.
- ✓ **Random Forest vs Decision Tree:**
 - ✓ Random Forest è essenzialmente una collezione di Alberi di Decisione.
 - ✓ Un albero di decisione è costruito su un intero dataset, che usa tutte le feature/variabili di interesse, mentre un un random forest **randomicamente** selezione **osservazioni/righe** e **specifiche features/variabili** per costruire alberi di decisione multipli e poi **mediarne** i risultati.

Random Forest

Single Decision Tree



Random Forest



Applicazioni di Random Forest

- ✓ Fraud detection per conti correnti bancari, carte di credito.
- ✓ Individuare e predire la sensibilità alle droghe di una medicina.
- ✓ Identificare la malattia di un paziente analizzando i loro registri medici.
- ✓ Predire la perdita stimata o il profitto dopo l'acquisto di una particolare azione finanziaria.

Bibliografia

Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. *Nature biotechnology*, 26(9), 1011-1013.

Cheng, J., Fayyad, U. M., Irani, K. B., & Qian, Z. (1988). Improved decision trees: a generalized version of id3. In *Machine Learning Proceedings 1988* (pp. 100-106). Morgan Kaufmann.

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012, July). How many trees in a random forest?. In *International workshop on machine learning and data mining in pattern recognition* (pp. 154-168). Springer, Berlin, Heidelberg.

Francesco Pugliese

