UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

# Fondamenti di Intelligenza Artificiale

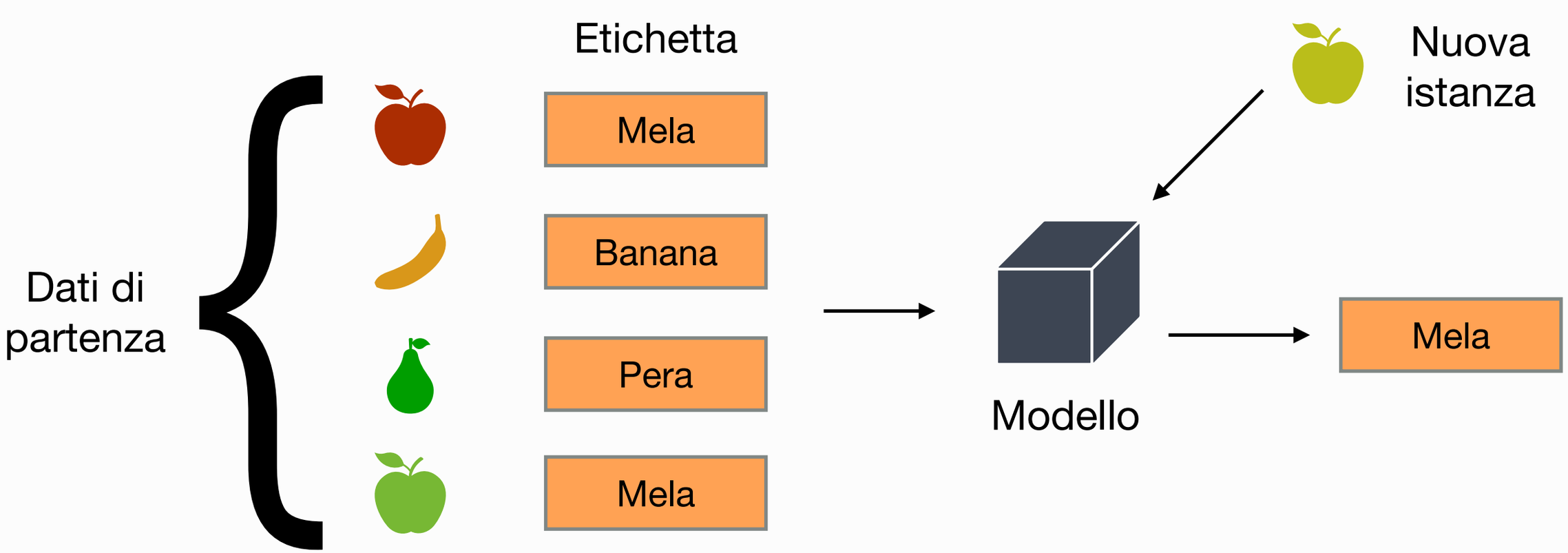
## Lezione 16 - Classificazione e classificatori

# Classificazione e Classificatori

## Problemi di classificazione

**Classificazione:** Task in cui l'obiettivo è predire il valore di una variabile categorica, chiamata variabile dipendente, target, o classe, tramite l'utilizzo di un training set, ovvero un insieme di osservazioni per cui la variabile target è nota.

I problemi di classificazione sono istanze di problemi di apprendimento supervisionato.



# Classificazione e Classificatori

## Problemi di classificazione

**Classificazione:** Task in cui l'obiettivo è predire il valore di una variabile categorica, chiamata variabile dipendente, target, o classe, tramite l'utilizzo di un training set, ovvero un insieme di osservazioni per cui la variabile target è nota.

I problemi di classificazione sono istanze di problemi di apprendimento supervisionato.

Un problema di classificazione porta alla costruzione di un *modello*, ovvero di uno strumento che fa uso di un algoritmo di apprendimento, anche detto *classificatore*, per classificare i nuovi elementi sulla base del training set.

Esistono diverse decine di classificatori, ognuno dei quali si distingue dall'altro per via delle assunzioni fatte sui dati così come delle specifiche proprietà che portano alla classificazione. Ad esempio, esistono classificatori probabilistici e classificatori basati sul concetto di entropia.

Sebbene lo scopo del corso non sia quello di presentare *tutti* i classificatori, ne approfondiremo due in particolare: *Naive Bayes* e *Decision Tree*. Più importante, però discuteremo come è possibile selezionare il *giusto* classificatore in base al problema.

Vale la pena anche sottolineare che, oltre i classificatori di base, esistono i cosiddetti *ensemble*: questi sono dei metodi che consentono di combinare insieme più classificatori. Un semplice esempio è il *Majority Voting*, un algoritmo che classifica una nuova istanza sulla base delle predizioni fatte dalla maggioranza dei classificatori di base.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

**Naive Bayes:** L'algoritmo considera le caratteristiche della nuova istanza da classificare e calcola la probabilità che queste facciano parte di una classe tramite l'applicazione del teorema di Bayes.

L'algoritmo è chiamato *naive* (ingenuo) poiché assume che le caratteristiche non siano correlate l'una all'altra. Di conseguenza, l'algoritmo non andrà a valutare in fase di classificazione la potenziale utilità data dalla combinazione di più caratteristiche.

Facciamo un esempio: una mela è un frutto di colore rosso, rotondo e avente un diametro di circa 8 centimetri. Sebbene queste caratteristiche siano dipendenti l'una dall'altra (ad esempio, la forma rotonda è collegata al fatto che il diametro sia di 8 centimetri), Naive Bayes considererà tutte queste proprietà come indipendenti per la probabilità che questo frutto sia una mela.

Tutto questo perché la classificazione viene eseguita sulla base del teorema di Bayes.

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

P(A) e P(B) rappresentano le probabilità di osservare A e B indipendentemente dall'altro.

P(B|A) rappresenta la probabilità di osservare B dato che A si è già verificato.

P(A|B) rappresenta la probabilità di osservare A dato che B si è già verificato.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Un esempio

Lanciamo due dadi e definiamo la prima ipotesi:
Evento B: *“Ottenere un solo 2 dal lancio dei dati”*;

Nello spazio degli eventi esistono 36 possibili esiti, tutti equiprobabili. Solo 11 di questi sono favorevoli al realizzarsi dell’evento B:

(2,1; 2,2; 2,3; 2,4; 2,5; 2,6; 1,2; 3,2; 4,2; 5,2; 6,2)

Quindi, P(B) = 11/36 = 30,56%;

Definiamo ora la seconda ipotesi:
Evento A: *“Ottenere 7 come somma dal lancio dei due dadi”*;

Tra i 36 possibili esiti, solo 6 sono favorevoli al realizzarsi dell’evento A:

(1,6; 2,5; 3,4; 4,3; 5,2; 6,1)

Quindi, P(A) = 6/36 = 16,67%;

Per calcolare P(B|A), occorre trovare quali delle 6 possibilità dell’evento A siano incluse nell’evento B. Nel nostro caso, (2,5; 5,2). Quindi, P(B|A) = 2/6 = 33,33%.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

### Un esempio

Applicando il teorema di Bayes, possiamo calcolare P(A|B).

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} = \frac{2/6 * 6/36}{11/36} = \frac{2}{11} = 18{,}18\,\%$$

Quindi, quasi 2 volte su 10 otteniamo come somma dal lancio dei due dadi 7, con valore 2 dal primo lancio e 5 dal secondo lancio o viceversa.

### Classificazione con Naive Bayes

In maniera molto simile all'esempio, la classificazione avviene secondo tre step:

(1) **Calcolo della probabilità della classe**: Le probabilità di classe sono semplicemente *le frequenze* delle istanze che appartengono a ciascuna classe divisa per il numero totale di istanze.

(2) **Calcolo della probabilità condizionata**: si applica il teorema di Bayes, per determinare le probabilità condizionate delle caratteristiche del problema.

(3) **Decisione**: identificata nella classe che ottiene il valore di probabilità più elevato.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

| Giocare | Meteo | Temperatura | Umidità |
|---|---|---|---|
| NO | Soleggiato | Caldo | Elevata |
| NO | Soleggiato | Caldo | Elevata |
| SI | Nuvoloso | Caldo | Elevata |
| SI | Piovoso | Mite | Elevata |
| SI | Piovoso | Freddo | Normale |
| NO | Piovoso | Freddo | Normale |
| SI | Nuvoloso | Freddo | Normale |
| NO | Soleggiato | Mite | Elevata |
| SI | Soleggiato | Freddo | Normale |
| SI | Piovoso | Mite | Normale |
| SI | Soleggiato | Mite | Normale |
| SI | Nuvoloso | Mite | Elevata |
| SI | Nuvoloso | Caldo | Normale |
| NO | Piovoso | Mite | Elevata |

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

**Step 1.** Calcoliamo la probabilità della classe *'Giocare'* e della classe *'Non Giocare'*:

Dividendo le frequenze dei 'SI' e dei 'NO' per il numero di istanze, abbiamo:

P('Giocare') = 9/14 = 64,29%;
P('Non giocare') = 5/14 = 35,71%;

**Step 2.** Per ogni caratteristica, creiamo una tabella di frequenza tramite la quale poter più facilmente applicare il teorema di Bayes. Ad esempio, nel caso di 'Meteo':

| **Tabella di frequenza** | | **Giocare** | |
|---|---|---|---|
| | | **Si** | **No** |
| Meteo | Soleggiato | 2 | 3 |
| | Nuvoloso | 4 | 0 |
| | Piovoso | 3 | 2 |

Possiamo da qui creare una tabella di probabilità.

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

**Step 1.** Calcoliamo la probabilità della classe *'Giocare'* e della classe *'Non Giocare'*:

Dividendo le frequenze dei 'SI' e dei 'NO' per il numero di istanze, abbiamo:

P('Giocare') = 9/14 = 64,29%;
P('Non giocare') = 5/14 = 35,71%;

**Step 2.** Per ogni caratteristica, creiamo una tabella di frequenza tramite la quale poter più facilmente applicare il teorema di Bayes. Ad esempio, nel caso di 'Meteo':

| **Tabella di frequenza** | | **Giocare** | |
|---|---|---|---|
| | | **Si** | **No** |
| Meteo | Soleggiato | 2/9 = 22,22% | 3 |
| | Nuvoloso | 4 | 0 |
| | Piovoso | 3 | 2 |

Possiamo da qui creare una tabella di probabilità.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

**Step 1.** Calcoliamo la probabilità della classe *'Giocare'* e della classe *'Non Giocare'*:

Dividendo le frequenze dei 'SI' e dei 'NO' per il numero di istanze, abbiamo:

P('Giocare') = 9/14 = 64,29%;
P('Non giocare') = 5/14 = 35,71%;

**Step 2.** Per ogni caratteristica, creiamo una tabella di frequenza tramite la quale poter più facilmente applicare il teorema di Bayes. Ad esempio, nel caso di 'Meteo':

| **Tabella di frequenza** | | **Giocare** | |
|---|---|---|---|
| | | **Si** | **No** |
| Meteo | Soleggiato | 2/9 | 3/5 |
| | Nuvoloso | 4/9 | 0/5 |
| | Piovoso | 3/9 | 2/5 |

Possiamo da qui creare una tabella di probabilità.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

**Step 1.** Calcoliamo la probabilità della classe *'Giocare'* e della classe *'Non Giocare'*:

Dividendo le frequenze dei 'SI' e dei 'NO' per il numero di istanze, abbiamo:

P('Giocare') = 9/14 = 64,29%;
P('Non giocare') = 5/14 = 35,71%;

**Step 2.** Per ogni caratteristica, creiamo una tabella di frequenza tramite la quale poter più facilmente applicare il teorema di Baye

Sommando i valori delle righe, possiamo identificare la probabilità del meteo.

| Tabella di frequenza | | | | |
|---|---|---|---|---|
| | | Si | No | |
| Me | Soleggiato | 2/9 | 3/5 | P('Soleggiato') = 5/14 |
| | | | 0/5 | P('Nuvoloso') = 4/14 |
| | Piovoso | 3 | 2/5 | P('Piovoso') = 5/14 |
| | | P('Giocare') = 9/14 | P('Non Giocare') = 5/14 | |

Banalmente, sommando i casi positivi/ negativi rispetto alla somma dei casi, avremo P('Giocare') e P('Non Giocare').

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

| Tabella di frequenza | | Giocare | | |
|---|---|---|---|---|
| | | Si | No | |
| Meteo | Soleggiato | 2/9 | 3/5 | P('Soleggiato') = 5/14 |
| | Nuvoloso | 4/9 | 0/5 | P('Nuvoloso') = 4/14 |
| | Piovoso | 3/9 | 2/5 | P('Piovoso') = 5/14 |
| | | P('Giocare') = 9/14 | P('Non Giocare') = 5/14 | |

Applicando il teorema di Bayes, avremo allora:

P (Sì | Soleggiato) = P (Soleggiato | Si) * P (Si) / P (Soleggiato) = (0,2222 x 0,6429) /0,3571 = 0,4;
P (No | Soleggiato) = P (Soleggiato | No) * P (No) / P (Soleggiato) = (0,6 x 0,3571) /0,3571 = 0,6.

Lo stesso meccanismo avverrà per tutte le altre variabili (nuvoloso, piovoso) e caratteristiche del problema (temperatura, umidità).

Avendo calcolato tutte le probabilità condizionate, possiamo fare predizioni.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

**Step 1.** Calcoliamo la probabilità della classe *'Giocare'* e della classe *'Non Giocare'*:

Dividendo le frequenze dei 'SI' e dei 'NO' per il numero di istanze, abbiamo:

P('Giocare') = 9/14 = 64,29%;
P('Non giocare') = 5/14 = 35,71%;

**Step 2.** Per ogni caratteristica, creiamo una tabella di frequenza tramite la quale poter più facilmente applicare il teorema di Bayes. Ad esempio, nel caso di 'Meteo':

P (Sì | Soleggiato) = P (Soleggiato | Si) * P (Si) / P (Soleggiato) = (0,2222 x 0,6429) /0,3571 = 0,4;
P (No | Soleggiato) = P (Soleggiato | No) * P (No) / P (Soleggiato) = (0,6 x 0,3571) /0,3571 = 0,6.

**Step 3.** Supponiamo che domani il meteo preveda i seguenti valori:

Meteo = Pioggia
Temperatura = Caldo
Umidità = Elevata
Giocare = ?

Probabilità di "Sì" = P (Meteo = Pioggia | Sì) * P (Temperatura = Caldo | Si) * P (Umidità = Elevata | Sì) = 0,36

Probabilità di "No" = P (Meteo = Pioggia | No) * P (Temperatura = Caldo | No) * P (Umidità = Elevata | No) = 0,64

In questo caso, Naive Bayes suggerirà di non andare a giocare.

# Classificazione e Classificatori

## Classificazione probabilistica: Il classificatore Naive Bayes

Consideriamo un problema di classificazione che permette di identificare se possiamo giocare a tennis un giorno oppure no a seconda di diverse caratteristiche climatiche.

**Step 1.** Calcoliamo la probabilità della classe *'Giocare'* e della classe *'Non Giocare'*:

Dividendo le frequenze dei 'SI' e dei 'NO' per il numero di istanze, abbiamo:

P('Giocare') = 9/14 = 64,29%;
P('Non giocare') = 5/14 = 35,71%;

**Step 2.** Per ogni caratteristica, creiamo una tabella di frequenza tramite la quale poter più facilmente applicare il teorema di Bayes. Ad esempio, nel caso di 'Meteo':

P (Sì | Soleggiato) = P (Soleggiato | Sì) * P (Sì) / P (Soleggiato) = (0,2222 x 0,6429) /0,3571 = 0,4;
P (No | Soleggiato) = P (Soleggiat 571 = 0,6.

**Step 3.** Supponiamo che doma

La predizione è fatta moltiplicando le varia probabilità condizionate calcolate nello step 2

Meteo = Pioggia
Temperatura = Caldo
Umidità = Elevata
Giocare = ?

Probabilità di "Sì" = P (Meteo = Pioggia | Sì) * P (Temperatura = Caldo | Sì) * P (Umidità = Elevata | Sì) = 0,36

Probabilità di "No" = P (Meteo = Pioggia | No) * P (Temperatura = Caldo | No) * P (Umidità = Elevata | No) = 0,64

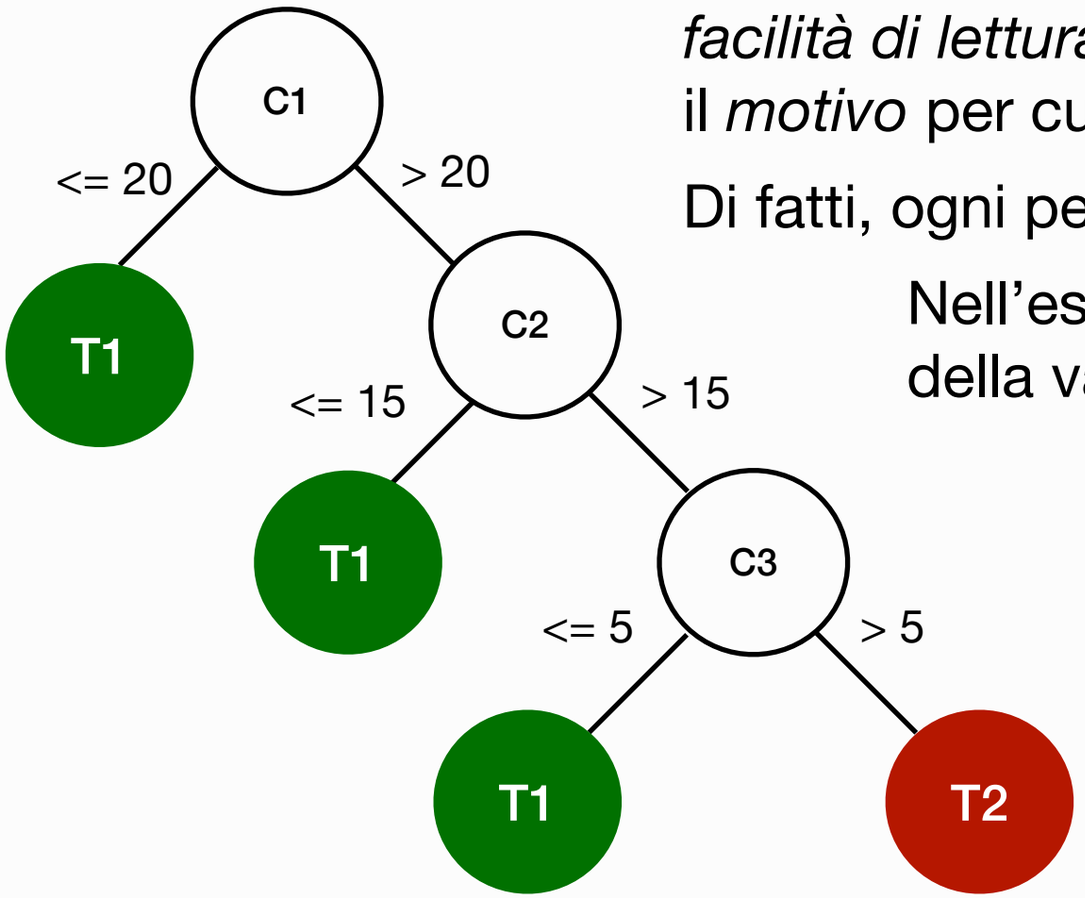
In questo caso, Naive Bayes suggerirà di non andare a giocare.

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

L'obiettivo di un albero decisionale è predire il valore di una variabile target apprendendo semplici *regole di decisione* inferite dai dati di training.




Gli alberi decisionali sono particolarmente utili per la loro *facilità di lettura*: navigando l'albero è possibile comprendere il *motivo* per cui è stata fatta una determinata predizione.

Di fatti, ogni percorso dell'albero corrisponde ad una regola.

Nell'esempio, potremmo motivare la predizione della variabile target T2 nel seguente modo:

$$C1 > 20 \wedge C2 > 15 \wedge C3 > 5 \Rightarrow T2$$

Inoltre, gli alberi decisionali possono essere utilizzati sia per problemi di classificazione che per problemi di regressione.

Il loro funzionamento è abbastanza semplice e si compone di tre passi.

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Mela | Verde | 57 | 62 |
| Banana | Giallo | 40 | 180 |
| Mela | Verde | 69 | 72 |
| Arancia | Arancione | 35 | 30 |
| Arancia | Arancione | 45 | 35 |
| Arancia | Arancione | 50 | 45 |
| Banana | Giallo | 40 | 170 |
| Banana | Giallo | 30 | 140 |
| Mela | Giallo | 60 | 62 |
| Mela | Giallo | 52 | 58 |

Vogliamo predire il tipo di frutto con le seguenti caratteristiche:

| | | | |
|---|---|---|---|
| ? | Giallo | 58 | 63 |

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

In questo caso, il colore sembra essere la caratteristica più discriminante.

| Frutta | Colore | | |
|---|---|---|---|
| Mela | Verde | | |
| Banana | Giallo | 40 | 180 |
| Mela | Verde | 69 | 72 |
| Arancia | Arancione | 35 | 30 |
| Arancia | Arancione | 45 | 35 |
| Arancia | Arancione | 50 | 45 |
| Banana | Giallo | 40 | 170 |
| Banana | Giallo | 30 | 140 |
| Mela | Giallo | 60 | 62 |
| Mela | Giallo | 52 | 58 |

Vogliamo predire il tipo di frutto con le seguenti caratteristiche:

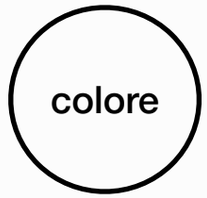
| | | | |
|---|---|---|---|
| ? | Giallo | 58 | 63 |

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.



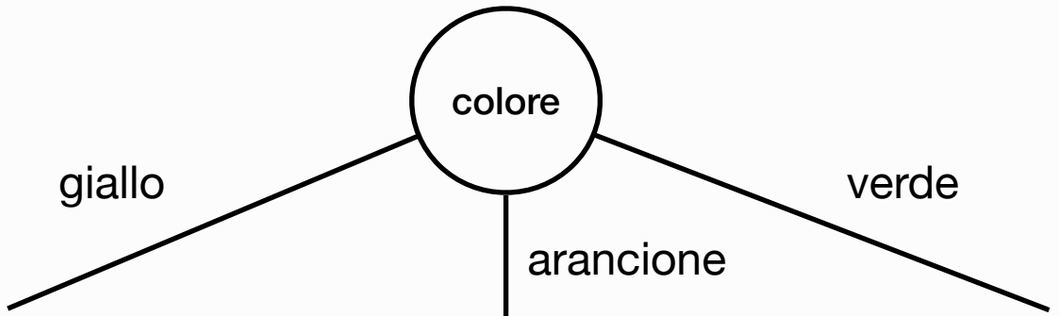
# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.




| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Banana | Giallo | 40 | 180 |
| Banana | Giallo | 40 | 170 |
| Banana | Giallo | 30 | 140 |
| Mela | Giallo | 60 | 62 |
| Mela | Giallo | 52 | 58 |

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Arancia | Arancione | 35 | 30 |
| Arancia | Arancione | 45 | 35 |
| Arancia | Arancione | 50 | 45 |

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Mela | Verde | 57 | 62 |
| Mela | Verde | 69 | 72 |

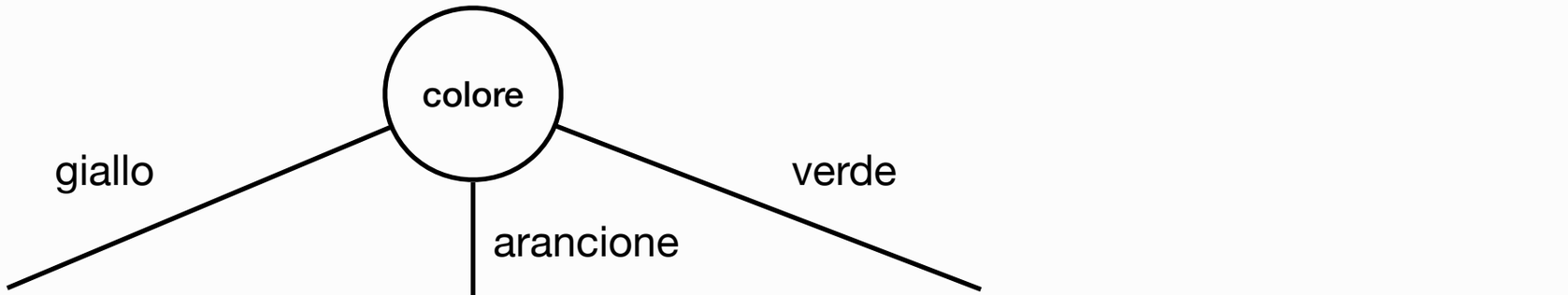
# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.




| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Banana | Giallo | 40 | 180 |
| Banana | Giallo | 40 | 170 |
| Banana | Giallo | 30 | 140 |
| Mela | Giallo | 60 | 62 |
| Mela | Giallo | 52 | 58 |

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Mela | Verde | 57 | 62 |
| Mela | Verde | 69 | 72 |

Un **sottoinsieme puro** contiene tutti e soli gli elementi di una classe (ad esempio, tutte le banane, tutte le mele, tutte le arance). Questo sottoinsieme **non è puro**: quindi, abbiamo ancora incertezza nella classificazione e dobbiamo necessariamente procedere a dividere ulteriormente il dataset.

| | | | |
|---|---|---|---|
| Arancia | Arancione | 50 | 45 |

# Classificazione e Classificatori

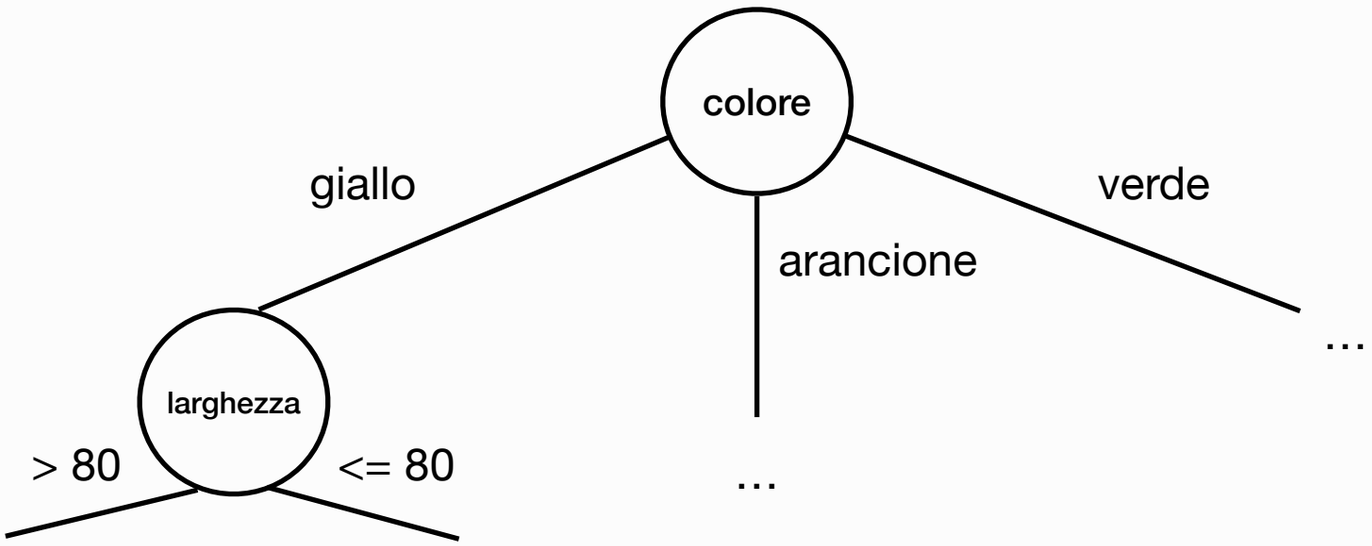
## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.

**Step 3.** Ripeti.




| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Banana | Giallo | 40 | 180 |
| Banana | Giallo | 40 | 170 |
| Banana | Giallo | 30 | 140 |

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Mela | Giallo | 60 | 62 |
| Mela | Giallo | 52 | 58 |

# Classificazione e Classificatori

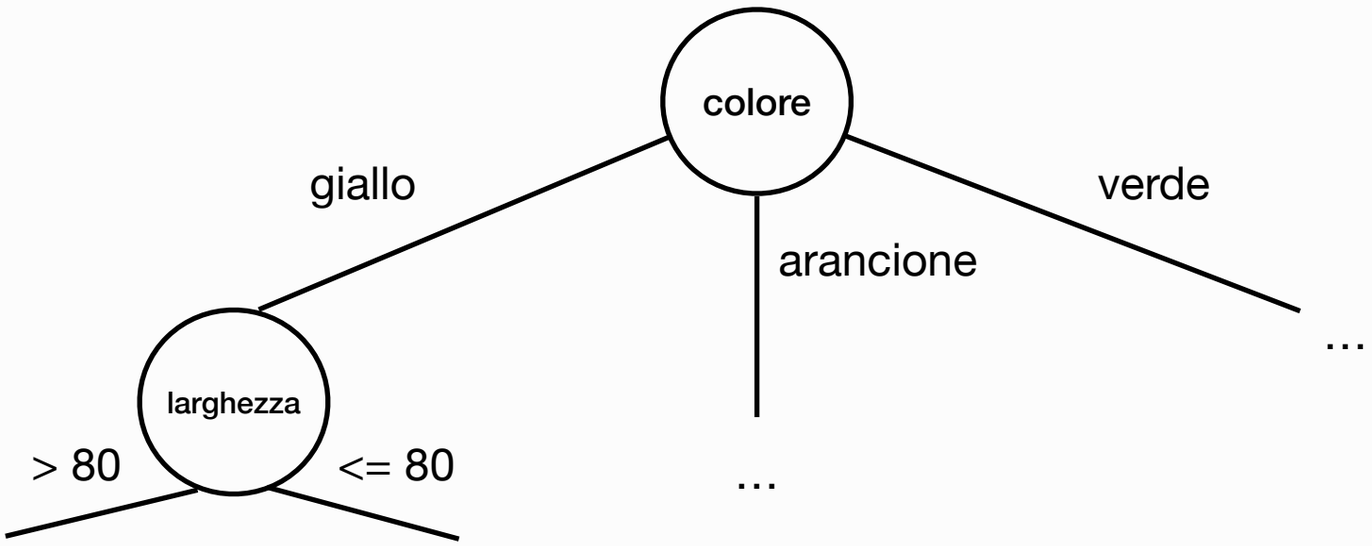
## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.

**Step 3.** Ripeti.




| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Banana | Giallo | 40 | 180 |
| Banana | Giallo | 40 | 170 |
| Banana | Giallo | 30 | 140 |

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| Mela | Giallo | 60 | 62 |
| Mela | Giallo | 52 | 58 |

Entrambi i sottoinsiemi sono adesso **puri**, per cui possiamo terminare ed assegnare le foglie.

# Classificazione e Classificatori

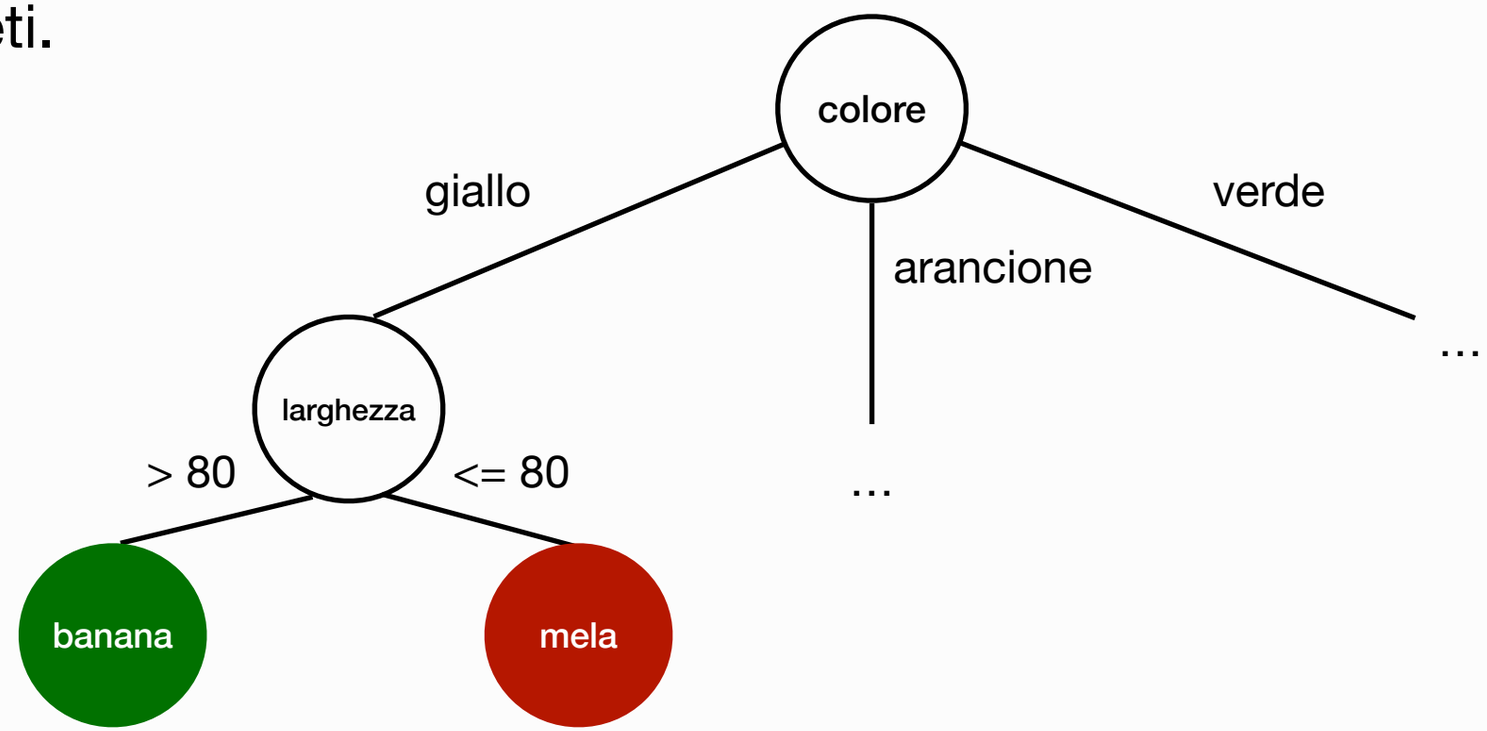
## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.

**Step 3.** Ripeti.



# Classificazione e Classificatori

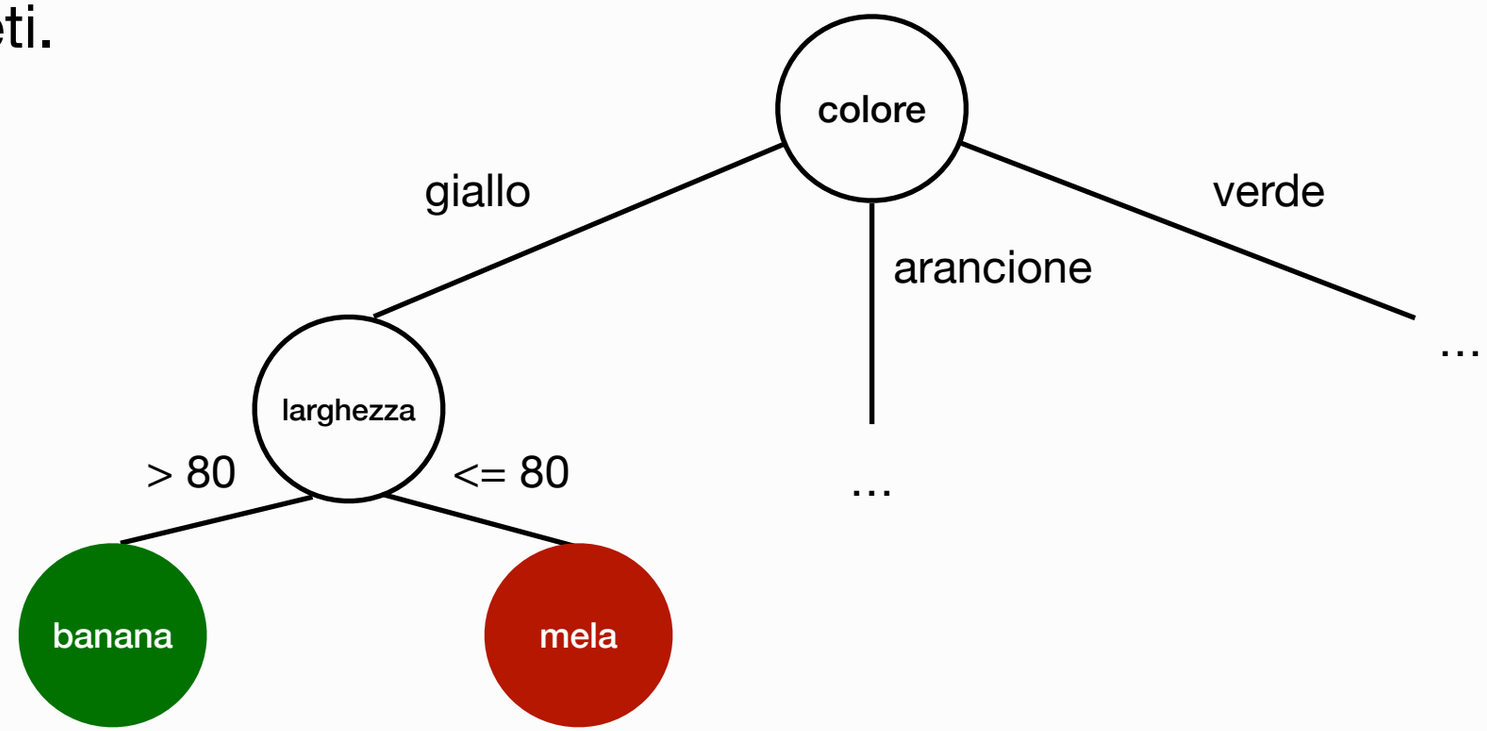
## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.

**Step 3.** Ripeti.




Una nuova istanza non nota verrà classificata come una mela se:

$$colore =' giallo' \wedge larghezza <= 80 \Rightarrow' mela'$$

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| ? | Giallo | 58 | 63 |

# Classificazione e Classificatori

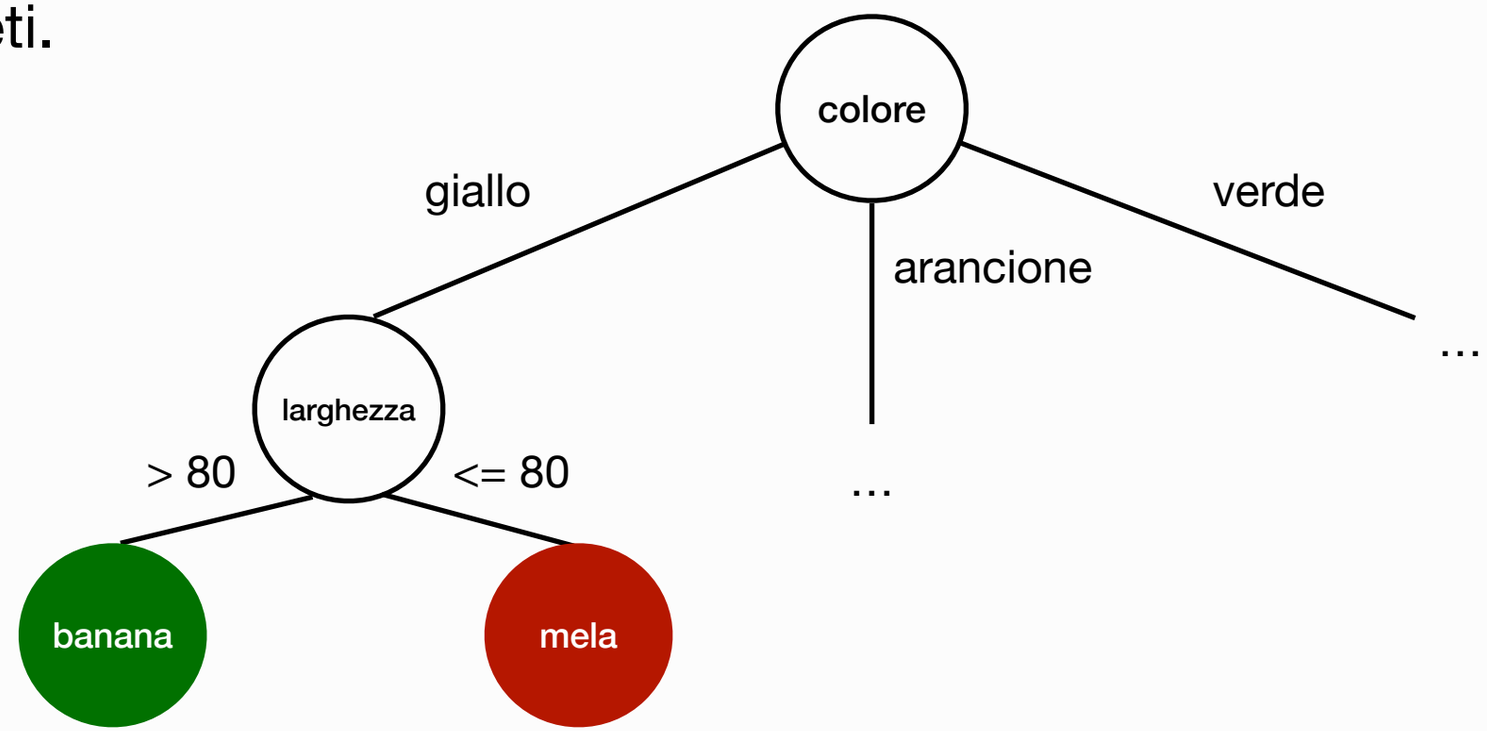
## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Step 1.** Posiziona la miglior caratteristica del training set come radice dell'albero.

**Step 2.** Dividi il training set in sotto-insiemi.

**Step 3.** Ripeti.




Una nuova istanza non nota verrà classificata come una mela se:

$$colore =' giallo' \wedge larghezza <= 80 \Rightarrow' mela'$$

| Frutta | Colore | Altezza (mm) | Larghezza (mm) |
|---|---|---|---|
| **Mela** | **Giallo** | **58** | **63** |

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

L'algoritmo di un albero decisionale:

(1) Posiziona la miglior caratteristica del training set come radice dell'albero;

(2) Dividi il training set in sotto-insiemi. Ogni sotto-insieme dovrebbe essere composto di valori simili per una certa caratteristica;

(3) Ripeti gli step (1) e (2) su ogni sotto-insieme fin quando non viene raggiunto un nodo foglia in ogni sotto-albero.

Capiamo il meccanismo di funzionamento con un esempio.

**Domanda.** Come decidiamo con quale attributo dividere il dataset?

**Information Gain:** Misura che indica il *grado di purezza* di un attributo, ovvero quanto un certo attributo sarà in grado di dividere adeguatamente il dataset.

Nella teoria dell'informazione, l'entropia indica in che misura un messaggio è ambiguo e difficile da capire:

$$H(D) = -\sum_{c} p(c) \bullet log_2 p(c)$$ dove p(c) è la proporzione della classe c nel dataset D.

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

L'algoritmo di un albero decisionale:

(1) Posiziona la miglior caratteristica del training set come radice dell'albero;

(2) Dividi il training set in sotto-insiemi. Ogni sotto-insieme dovrebbe essere composto di valori simili per una certa caratteristica;

(3) Ripeti gli step (1) e (2) su ogni sotto-insieme fin quando non viene raggiunto un nodo foglia in ogni sotto-albero.

Capiamo il meccanismo di funzionamento con un esempio.

**Domanda.** Come decidiamo con quale attributo dividere il dataset?

**Information Gain:** Misura che indica il *grado di purezza* di un attributo, ovvero quanto un certo attributo sarà in grado di dividere adeguatamente il dataset.

Maggiore è l'entropia, minore sarà l'ammontare di informazione del messaggio.

Nella teoria dell'infor messaggio è ambiguo e difficile da capire:

$$H(D) = -\sum_{c} p(c) \bullet log_2 p(c)$$ dove p(c) è la proporzione della classe c nel dataset D.

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Information Gain:** Misura che indica il *grado di purezza* di un attributo, ovvero quanto un certo attributo sarà in grado di dividere adeguatamente il dataset.

Nei decision tree, l'entropia è utilizzata come base per l'analisi dell'information gain.

Partendo dalla radice dell'albero decisionale, usiamo l'entropia per dividere i dati in sottoinsiemi che contengono istanze simili (o omogenee).

In particolare, per ogni attributo del dataset calcoleremo il suo gain:

$$Gain(D, A) = H(D) - \sum_{v \in values(A)} \frac{|D_v|}{|D|} \bullet H(D_v)$$

dove
- D è l'entropia del dataset;
- $D_v$ è il sottoinsieme di D per cui l'attributo A ha valore v;
- $|D_v|$ è il numero di elementi di $D_v$;
- $|D|$ è il numero di elementi del dataset.

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Information Gain:** Misura che indica il *grado di purezza* di un attributo, ovvero quanto un certo attributo sarà in grado di dividere adeguatamente il dataset.

| Frutta | Colore |
|---|---|
| Mela | Verde |
| Banana | Giallo |
| Mela | Verde |
| Arancia | Arancione |
| Arancia | Arancione |
| Arancia | Arancione |
| Banana | Giallo |
| Banana | Giallo |
| Mela | Giallo |
| Mela | Giallo |

|D| = 10;  D = {Mela (x4), Banana (x3), Arancia (x3)};

$$H(D) = -\frac{4}{10} \bullet log_2\frac{4}{10} - \frac{3}{10} \bullet log_2\frac{3}{10} - \frac{3}{10} \bullet log_2\frac{3}{10}$$

$$H(D) = 0{,}53 + 0{,}52 + 0{,}52 = 1{,}57$$

A = "Colore";  values(A) = {Verde (x2), Giallo (x5), Arancione (x3)};

$$H(D_{giallo}) = -\frac{2}{5} \bullet log_2\frac{2}{5} - \frac{3}{5} \bullet log_2\frac{3}{5} = 0{,}97$$

Ricordiamo che $D_v$ è il sottoinsieme di D per cui l'attributo A ha valore v —> Nel caso del giallo, abbiamo due sottoinsiemi: il primo composto dai due elementi {giallo;mela}, il secondo composto dai tre elementi {giallo;banana}.

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

**Information Gain:** Misura che indica il *grado di purezza* di un attributo, ovvero quanto un certo attributo sarà in grado di dividere adeguatamente il dataset.

| Frutta | Colore |
|---|---|
| Mela | Verde |
| Banana | Giallo |
| Mela | Verde |
| Arancia | Arancione |
| Arancia | Arancione |
| Arancia | Arancione |
| Banana | Giallo |
| Banana | Giallo |
| Mela | Giallo |
| Mela | Giallo |

|D| = 10;  D = {Mela (x4), Banana (x3), Arancia (x3)};

$$H(D) = -\frac{4}{10} \bullet log_2\frac{4}{10} - \frac{3}{10} \bullet log_2\frac{3}{10} - \frac{3}{10} \bullet log_2\frac{3}{10}$$

$$H(D) = 0{,}53 + 0{,}52 + 0{,}52 = 1{,}57$$

A = "Colore";  values(A) = {Verde (x2), Giallo (x5), Arancione (x3)};

$$H(D_{giallo}) = -\frac{2}{5} \bullet log_2\frac{2}{5} - \frac{3}{5} \bullet log_2\frac{3}{5} = 0{,}97$$

$$H(D_{verde}) = -\frac{2}{2} \bullet log_2\frac{2}{2} = 0$$

$$H(D_{arancione}) = -\frac{3}{3} \bullet log_2\frac{3}{3} = 0$$

$$Gain(D, A) = H(D) - \sum_{v \in values(A)} \frac{|D_v|}{|D|} \bullet H(D_v) = 1{,}57 - (\frac{5}{10} \bullet 0{,}97 + \frac{2}{10} \bullet 0 + \frac{3}{10} \bullet 0) = 1{,}09$$

# Classificazione e Classificatori

## Classificazione basata su entropia: Gli alberi decisionali

**Albero decisionale:** L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni.

In base ai concetti di entropia ed information gain, possiamo raffinare la procedura di creazione di un albero decisionale:

(1) Calcola l'entropia per ogni attributo del dataset;

(2) Dividi il training set in sotto-insiemi, utilizzando l'attributo per cui l'entropia sia minimizzata o, in maniera equivalente, l'information gain è massimizzata;

(3) Crea un nodo dell'albero decisionale contenente quell'attributo;

(4) Ripeti i passi precedenti fin quando tutti i sottoinsiemi definiti non siano puri.

# Classificazione e Classificatori

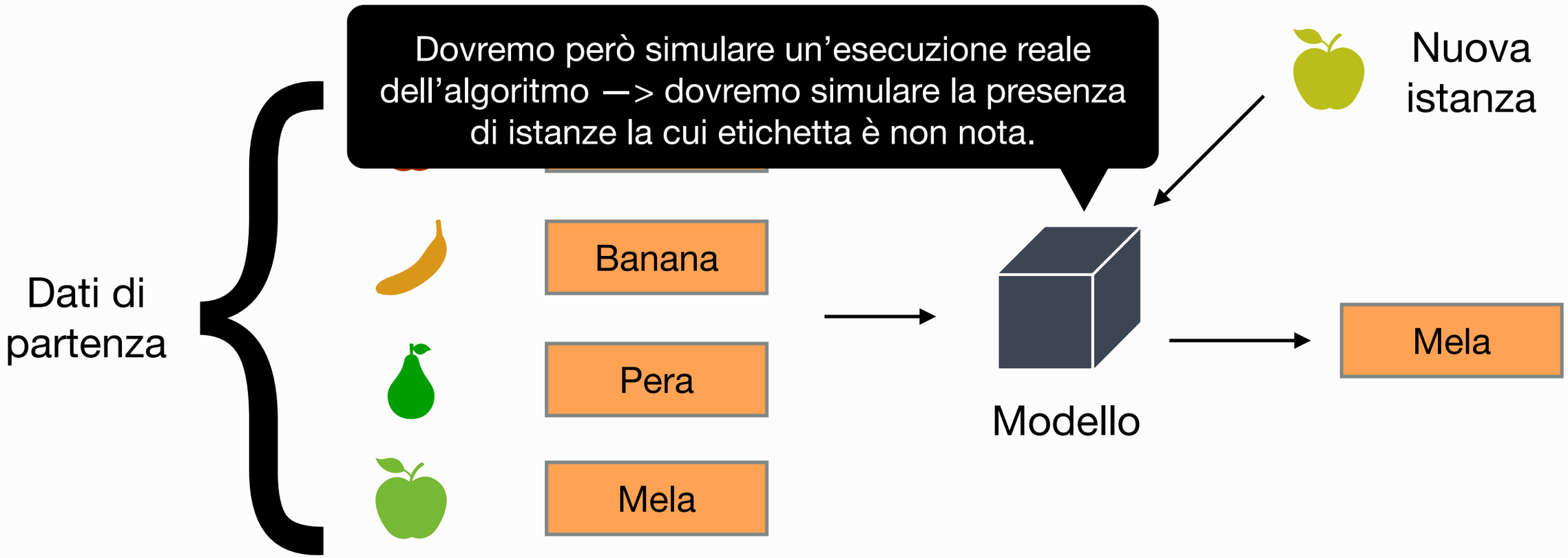
## Ok, ma come decidere qual è il classificatore da usare per un determinato problema?

Sicuramente, potremmo ragionare sulle proprietà del problema e valutare quale classificatore si adatti meglio… ma questo richiede una profonda conoscenza degli algoritmi di machine learning, il ché non è banale!

Un'alternativa più pragmatica è quella di sperimentare più classificatori e valutare le prestazioni di classificazione ottenute.

Chiaramente, non possiamo valutare un classificatore su dati che non conosciamo, altrimenti non potremmo misurare le sue prestazioni.

Ma, avendo un dataset di partenza etichettato, possiamo sfruttare questa conoscenza per capire come un classificatore classifica questi dati.



# Classificazione e Classificatori

## Validare un modello di machine learning: Training e Test set

Addestrare e validare un modello di machine learning sullo stesso dataset è un (grosso) errore metodologico che porta ad avere risultati totalmente inaffidabili.

| Giocare | Meteo | Temperatura | Umidità |
|---|---|---|---|
| NO | Soleggiato | Caldo | Elevata |
| NO | Soleggiato | Caldo | Elevata |
| SI | Nuvoloso | Caldo | Elevata |
| SI | Piovoso | Mite | Elevata |
| SI | Piovoso | Freddo | Normale |
| NO | Piovoso | Freddo | Normale |
| SI | Nuvoloso | Freddo | Normale |
| NO | Soleggiato | Mite | Elevata |
| SI | Soleggiato | Freddo | Normale |
| SI | Piovoso | Mite | Normale |
| SI | Soleggiato | Mite | Normale |
| SI | Nuvoloso | Mite | Elevata |
| SI | Nuvoloso | Caldo | Normale |
| NO | Piovoso | Mite | Elevata |

Possiamo però dividere il dataset di partenza in maniera tale da considerare alcune delle istanze come non note.

Creeremo quindi due insiemi:

- Il *training set*, che sarà composto delle istanze che l'algoritmo utilizzerà per l'addestramento;

- il *test set*, che sarà composto delle istanze per cui l'algoritmo addestrato dovrà predire la classe di appartenenza.

Esistono diversi modi di dividere training e test set.

Ad esempio, addestrare un modello considerando il 67% del dataset e validare le sue prestazioni sul restante 33%.

Un altro metodo è chiamato *convalida incrociata*.

# Classificazione e Classificatori

## Validare un modello di machine learning: La convalida incrociata (k-fold cross validation)

**Convalida incrociata:** Metodo statistico che consiste nella ripetuta partizione e valutazione dell'insieme dei dati di partenza.

La convalida incrociata prevede i seguenti passi:

(1) Mischiare in maniera casuale i dati di partenza;

(2) Dividere i dati in *k* gruppi;

(3) Per ogni gruppo:

(3.1) Considerare il gruppo come test set;

(3.2) Considerare i rimanenti *k-1* gruppi come training set;

(3.3) Addestrare il modello con i dati del training set;

(3.4) Valutare le prestazioni del modello ed eliminarlo.

# Classificazione e Classificatori

## Validare un modello di machine learning: La convalida incrociata (k-fold cross validation)

**Convalida incrociata:** Metodo statistico che consiste nella ripetuta partizione e valutazione dell'insieme dei dati di partenza

La convalida incrociata prevede i s

(1) Mischiare in maniera casuale

(2) Dividere i dati in *k* gruppi;

(3) Per ogni gruppo:

(3.1) Considerare il gruppo come test set;

(3.2) Considerare i rimanenti *k-1* gruppi come training set;

(3.3) Addestrare il modello con i dati del training set;

(3.4) Valutare le prestazioni del modello ed eliminarlo.

E' abbastanza comune utilizzare un *k*=10. Questo caso si definisce come 10-fold cross validation.

# Classificazione e Classificatori

## Validare un modello di machine learning: La convalida incrociata (k-fold cross validation)

**Convalida incrociata:** Metodo statistico che consiste nella ripetuta partizione e valutazione dell'insieme dei dati di partenza

La convalida incrociata prevede i s

(1) Mischiare in maniera casuale

(2) Dividere i dati in *k* gruppi;

(3) Per ogni gruppo:

(3.1) Considerare il gruppo co

(3.2) Considerare i rimanenti gruppi come training set;

(3.3) Addestrare il modello con i dati del training set;

(3.4) Valutare le prestazioni del modello ed eliminarlo.

E' abbastanza comune utilizzare un *k*=10. Questo caso si definisce come 10-fold cross validation.

**NB: Tutti i processi di normalizzazione (es., feature selection, data balancing) vanno effettuati ad ogni addestramento!!!**

# Classificazione e Classificatori

## Validare un modello di machine learning: La convalida incrociata (k-fold cross validation)

**Convalida incrociata:** Metodo statistico che consiste nella ripetuta partizione e valutazione dell'insieme dei dati di partenza

La convalida incrociata prevede i s

(1) Mischiare in maniera casuale

(2) Dividere i dati in *k* gruppi;

(3) Per ogni gruppo:

(3.1) Considerare il gruppo co

(3.2) Considerare i rimanenti gruppi come training set;

(3.3) Addestrare il modello con i dati del training set;

(3.4) Valutare le prestazioni del modello ed eliminarlo.

E' abbastanza comune utilizzare un *k*=10. Questo caso si definisce come 10-fold cross validation.

**NB: Tutti i processi di normalizzazione (es., feature selection, data balancing) vanno effettuati ad ogni addestramento!!!**

E' importante notare che ogni istanza sarà assegnata ad un **unico** gruppo durante l'intera procedura di validazione —> altrimenti, mischieremmo i dati di training con quelli di test, influenzando le capacità predittive del classificatore.

**Data leakage:** Problema che si presenta quando un modello è capace di lavorare accuratamente in fase di addestramento, ma non in fase di rilascio.

Mischiare dati di training e test porta ad un altro caso di data leakage, noto come *leaky validation strategy*!

## Validare un modello di machine learning: Altre tipologie di convalida e data leaking

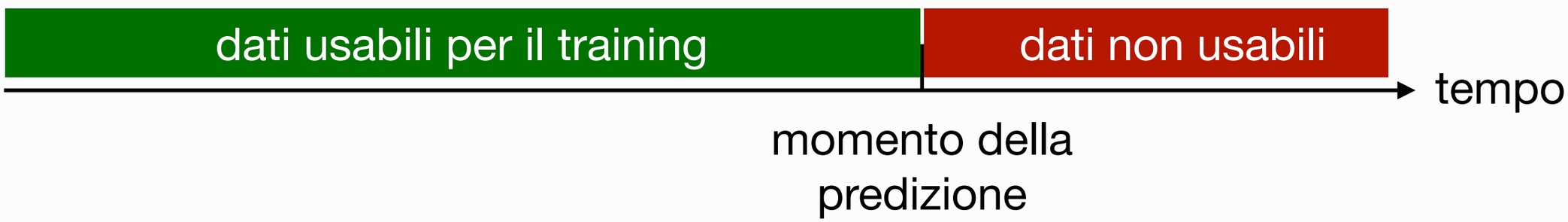
La convalida incrociata può però avere alcuni problemi:

(1) Essendo casuale, il primo passo della validazione potrebbe inavvertitamente portare un *vantaggio* al classificatore, ad esempio una divisione dei gruppi irrealistica.

—> **Soluzione 1.** Ripetere la validazione *N* volte, in modo da limitare l'influenza della casualità del primo passo. In questo caso, parliamo di *N*-times *k*-fold validation.

—> **Soluzione 2.** Modificare il primo passo della validazione in modo da avere un campionamento stratificato, che porta i sottoinsiemi creati ad avere un numero simile di istanze delle diverse classi. Qui parliamo di *stratified* k-fold validation.

(2) La convalida incrociata non può essere usata facilmente nel caso in cui i dati seguano un ordine temporale.




In questo caso, una convalida incrociata mischierebbe dati temporali, il ché porta ad un caso irrealistico. Nella realtà, non potremo usare dati futuri per predire dati passati!

L'utilizzo della convalida cross-fold classica crea un caso di leaky validation.

## Validare un modello di machine learning: Altre tipologie di convalida e data leaking

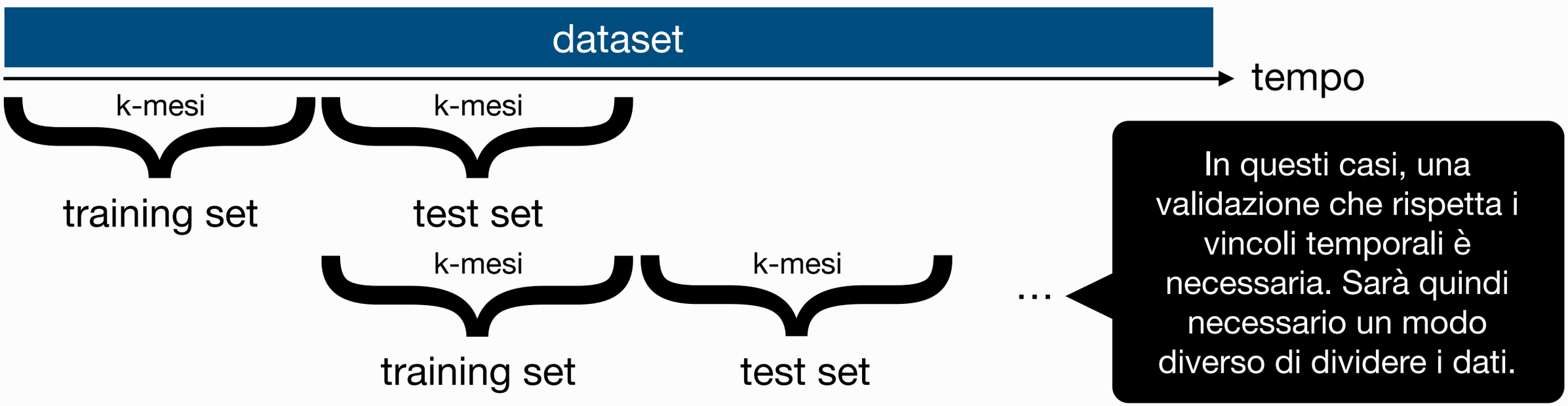
La convalida incrociata può però avere alcuni problemi:

(1) Essendo casuale, il primo passo della validazione potrebbe inavvertitamente portare un *vantaggio* al classificatore, ad esempio una divisione dei gruppi irrealistica.

—> **Soluzione 1.** Ripetere la validazione *N* volte, in modo da limitare l'influenza della casualità del primo passo. In questo caso, parliamo di *N*-times *k*-fold validation.

—> **Soluzione 2.** Modificare il primo passo della validazione in modo da avere un campionamento stratificato, che porta i sottoinsiemi creati ad avere un numero simile di istanze delle diverse classi. Qui parliamo di *stratified* k-fold validation.

(2) La convalida incrociata non può essere usata facilmente nel caso in cui i dati seguano un ordine temporale.



# Classificazione e Classificatori

## Validare un modello di machine learning: Metriche di valutazione

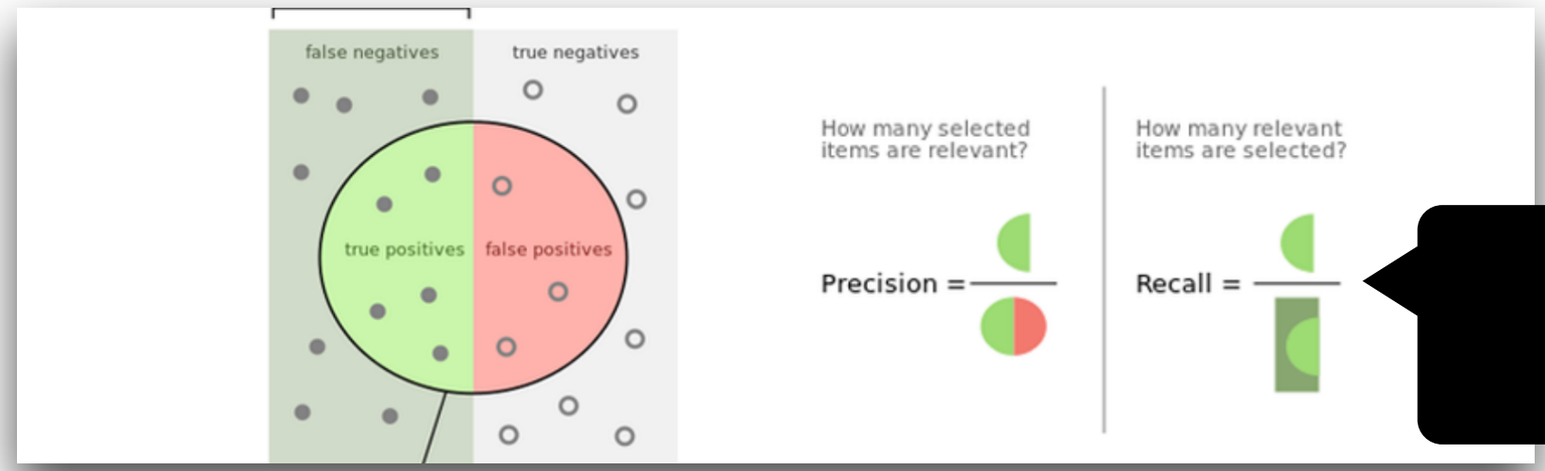
A prescindere dalla procedura di validazione che utilizzeremo, avremo bisogno di strumenti adatti per valutare la bontà delle predizioni.

Parliamo quindi della *matrice di confusione*, anche detta tabella di errata classificazione, la quale restituisce una rappresentazione dell'accuratezza di un classificatore.

| | **Istanze realmente positive** | **Istanze realmente negative** |
|---|---|---|
| **Istanze predette come positive** | **Veri positivi** | **Falsi positivi** |
| **Istanze predette come negative** | **Falsi negativi** | **Veri negativi** |

La matrice di confusione è una matrice con cui poter indicare se ed in quanti casi il classificatore ha predetto correttamente o meno il valore di un'etichetta del test set.

Sulla base dei valori della matrice di confusione, possiamo poi calcolare diverse metriche.




Precision e recall sono due delle metriche principali nei processi di classificazione di dati.

# Classificazione e Classificatori

## Validare un modello di machine learning: Metriche di valutazione

Definendo come TP il numero di veri positivi, come FP il numero di falsi positivi, come TN il numero di veri negativi, e come FN il numero di falsi negativi:

$$Precision = \frac{TP}{(TP + FP)} \qquad Recall = \frac{TP}{(TP + FN)} \qquad Specificity = \frac{TP}{(TN + FP)}$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \qquad MCC = \frac{TP \bullet TN - FP \bullet FN)}{\sqrt{(TP + FP) \bullet (TP + FN) \bullet (TN + FP) \bullet (TN + FN)}}$$

Ogni metrica fornisce un'indicazione complementare per la valutazione del modello di classificazione. Ad esempio, in un problema binario (classificazione true/false):

(1) La precision indica il numero di predizioni corrette per la classe 'true' rispetto a tutte le predizioni fatte dal classificatore. In altri termini, indica quanti errori ci saranno nella lista delle predizioni fatte dal classificatore. E' chiaramente una metrica che vogliamo massimizzare.

(2) La recall indica il numero di predizioni corrette per la classe 'true' rispetto a tutte le istanze positive di quella classe. In altri termini, indica quante istanze positive nell'intero dataset il classificatore può determinare. E' chiaramente una metrica che vogliamo massimizzare.

## Validare un modello di machine learning: Metriche di valutazione

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Un'attenzione particolare richiede però l'accuracy. Per definizione, questa indica il numero totale di predizioni corrette (sia della classe positiva che negativa).

Al numeratore abbiamo anche i veri negativi, ovvero il numero di istanze correttamente classificate come negative. Questo potrebbe creare un problema di interpretazione nel caso di dataset sbilanciati dove il numero di casi positivi è molto basso…

In questi casi, il classificatore sarà sicuramente più abile a riconoscere i veri negativi, poiché sono molti di più. E quindi, che significa avere una accuracy del 99%?

Immaginate un problema di classificazione dei melanomi, dove (per fortuna) il numero di veri positivi è molto basso: un'accuracy altissima potrebbe far pensare che il classificatore sia efficacissimo, *ma è solo un'illusione*: il classificatore riconoscerà bene chi non ha il melanoma… ma a noi interessa scoprire chi, invece, ne ha affetto!

Le metriche forniscono indicazioni, ma vanno interpretate correttamente sulla base dei problemi che si analizzano. Mai fidarsi ciecamente delle metriche e, soprattutto, attenzione all'uso che fate di queste!

UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

# Fondamenti di Intelligenza Artificiale

## Lezione 16 - Classificazione e classificatori