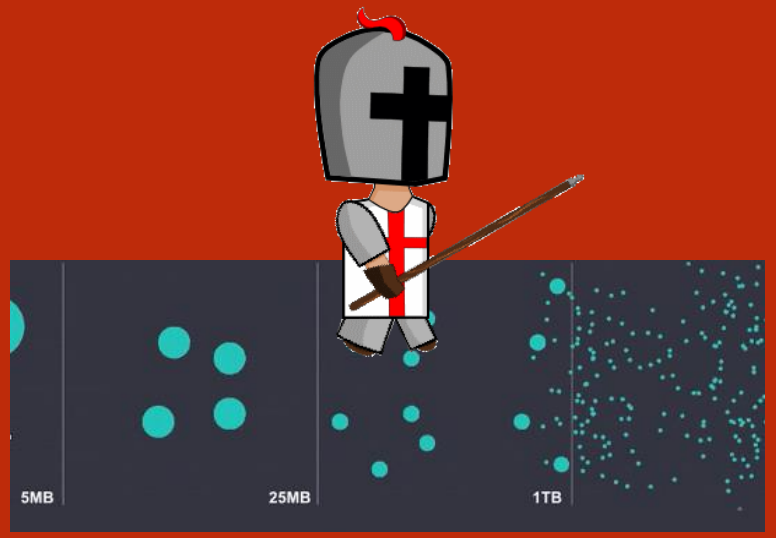


The (big) data scientist path

Lorenzo Stacchio

Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Il percorso di un (big) data scientist

Studia il problema

Programmazione

Database

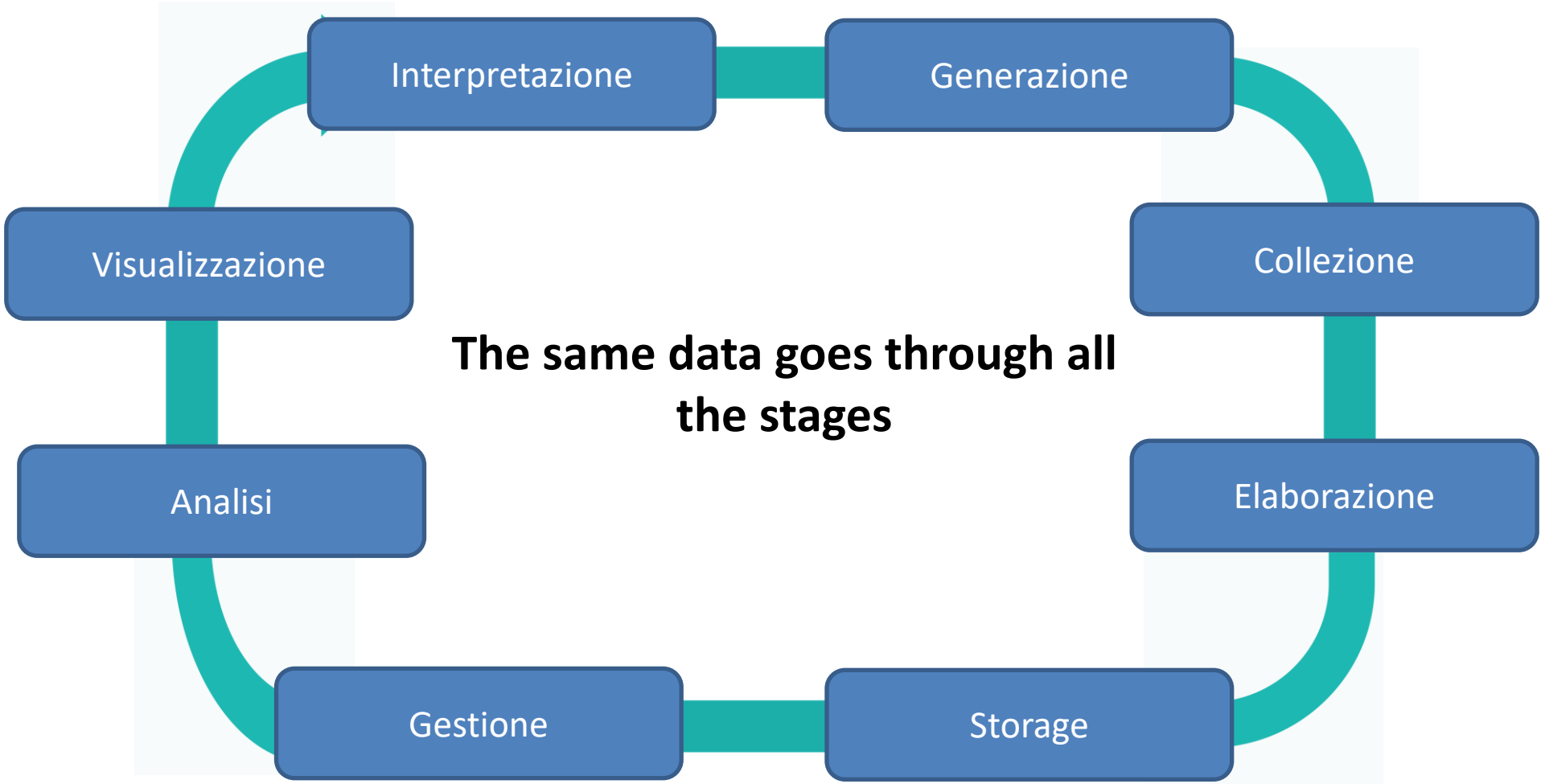
Big data

Machine & Deep Learning

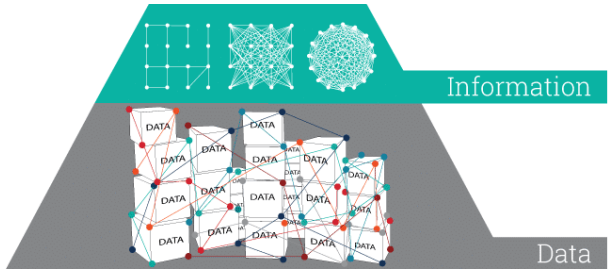
Math

Analisi, Algebra lineare, probabilità e statistica

Il ciclo dei (big) data



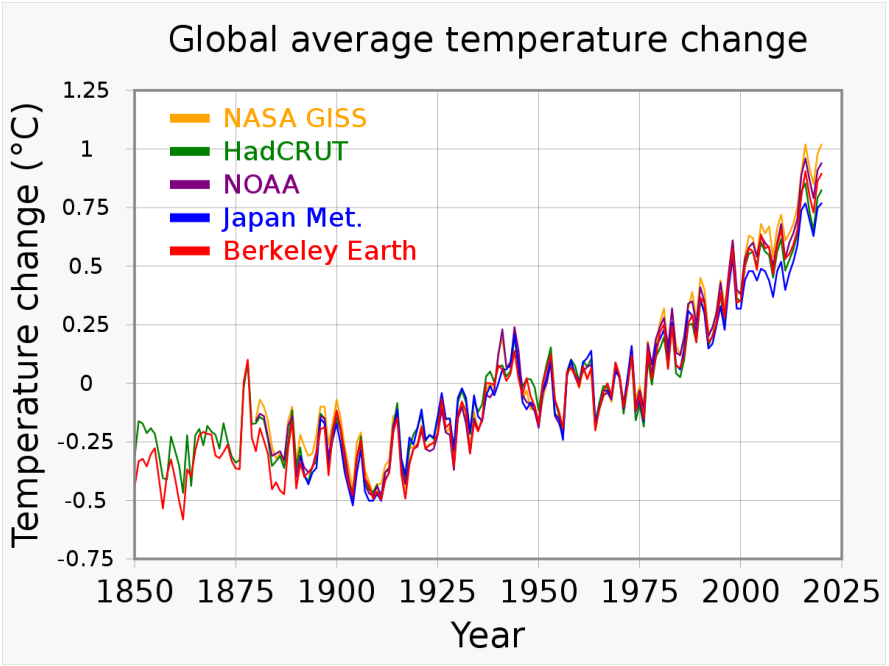
Dati, informazioni, conoscenza e saggezza



- I **dati sono una collezione di fatti grezzi e non organizzati**, come numeri e caratteri. Senza contesto, I dati hanno poco senso. **Ad esempio**, 1 è un numero, ma possiamo vederlo come **un grado di temperatura** (**1**°). Abbiamo quindi trasformato il dato in **informazione**!

- **L'informazione** è un insieme di dati «**pulito ed elaborato**» in un modo che renda più facile la manipolazione, analisi e visualizzazione; **Ad esempio**, potremmo fare un grafico per analizzare i valori della temperatura media globale negli ultimi 170 anni!

- Ponendo domande pertinenti su "chi", "cosa", "quando", "dove", ecc., possiamo ricavare informazioni preziose dalle informazioni rendendole più utili. Ma quando arriviamo alla domanda sul "**come**", siamo costretti a fare il salto dall'informazione alla **conoscenza**!

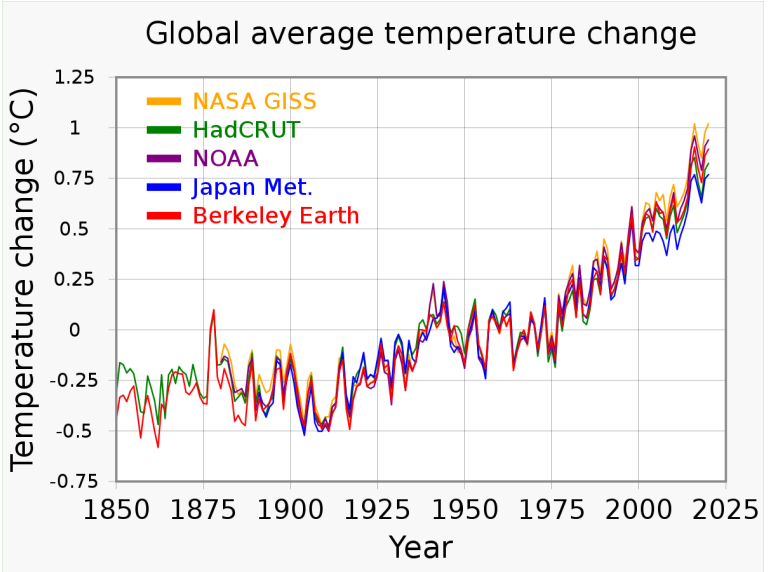


- "In che modo" le informazioni, derivate dai dati raccolti, sono rilevanti per i nostri obiettivi? “Come” i pezzi di queste informazioni sono collegati ad altri pezzi per aggiungere più significato e valore? E, **cosa più importante**, "come" possiamo applicare le informazioni per raggiungere il nostro obiettivo?

- Quando comprendiamo come applicare le informazioni per raggiungere i nostri obiettivi, le trasformiamo in **conoscenza**.

- Quando usiamo le conoscenze e le intuizioni acquisite dalle informazioni per prendere decisioni proattive, possiamo dire di aver raggiunto il gradino finale della piramide: la **saggezza**.

- La **saggezza** è il vertice della gerarchia e non è altro che conoscenza applicata per prendere la miglior decisione possibile!



- Quali sono le cause del riscaldamento globale? (**conoscenza)**

- Cosa possiamo fare per fermarlo? (**saggezza)**

Trasformare I dati al fine di ricavare conoscenza

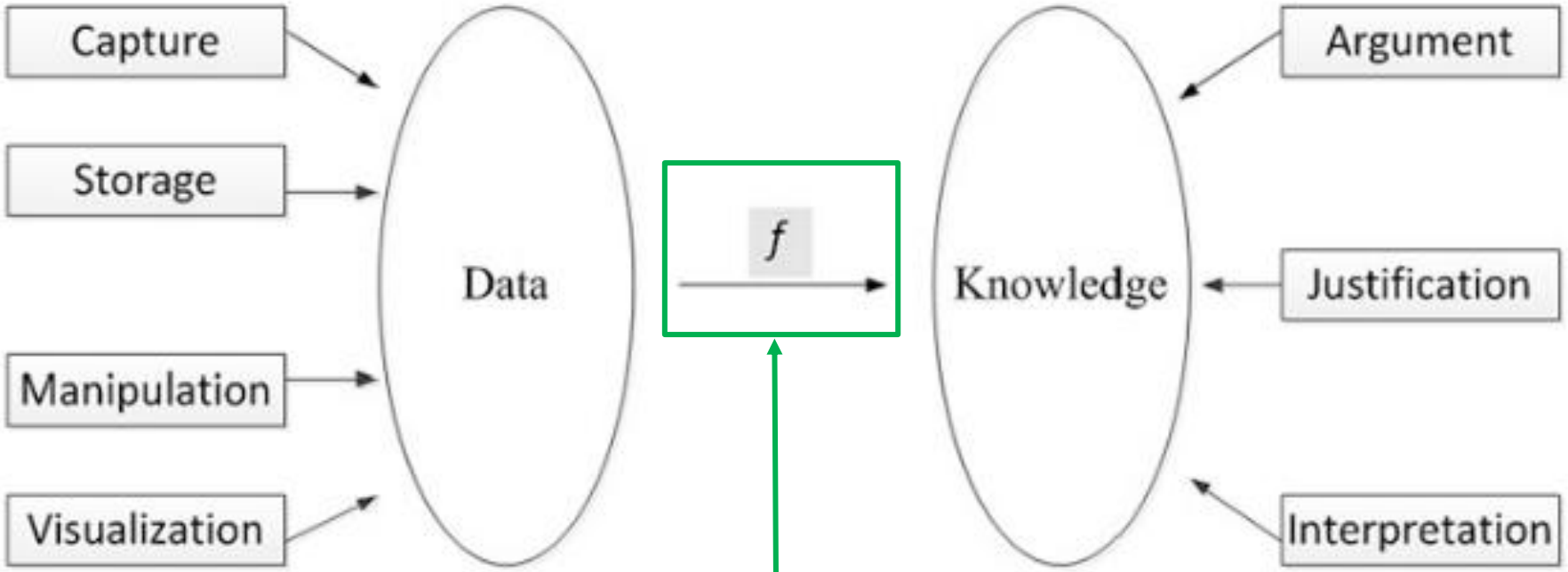
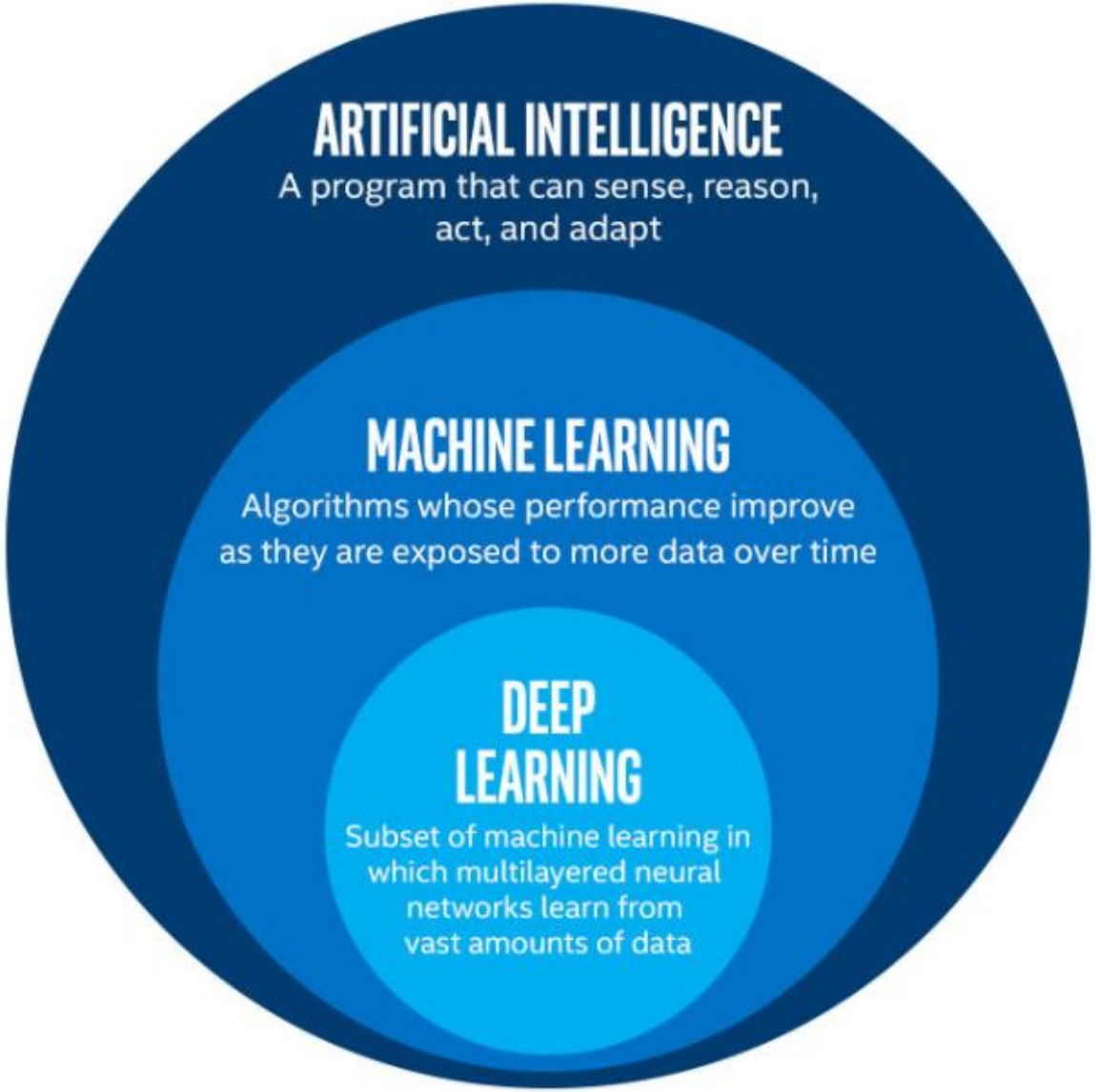


Fig. 1.1 Transformation of data into knowledge

Machine & Deep Learning

Artificial Intelligence (AI), Machine Learning (ML) e Deep Learning (DL)



https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55

Machine Learning – Un modo per risolvere problemi non classici

Problemi difficili da risolvere con algoritmi «classici»:

- **Problemi di regressione**;

- **Problemi di classificazione**;

- **Data Mining**.

Machine Learning – Caratteristiche

- Bassa conoscenza di informazioni (a priori);
- Dati con elevato numero di attributi (input features);
- Elevato volume di dati per il training;
- Adattabilità.

https://medium.com/@lokeshpara17/tiny-imagenet-using-pytorch-42a3f2ee3c9d

https://www.image-net.org/

Machine Learning – Learning Algorithm

- Definire un **modello** per il problema da risolvere: il modello dipende da un insieme di **parametri**;
- Definire una metrica per misurare le performance: una **misura di errore** per valutare il modello;
- Allenare il modello (aggiornare i parametri), per minimizzare l'errore, sui dati di training.

UN ALGORITMO DI LEARNING E' UN PROCESSO DI OTTIMIZZAZIONE

Machine Learning – Parametri ed iperparametri

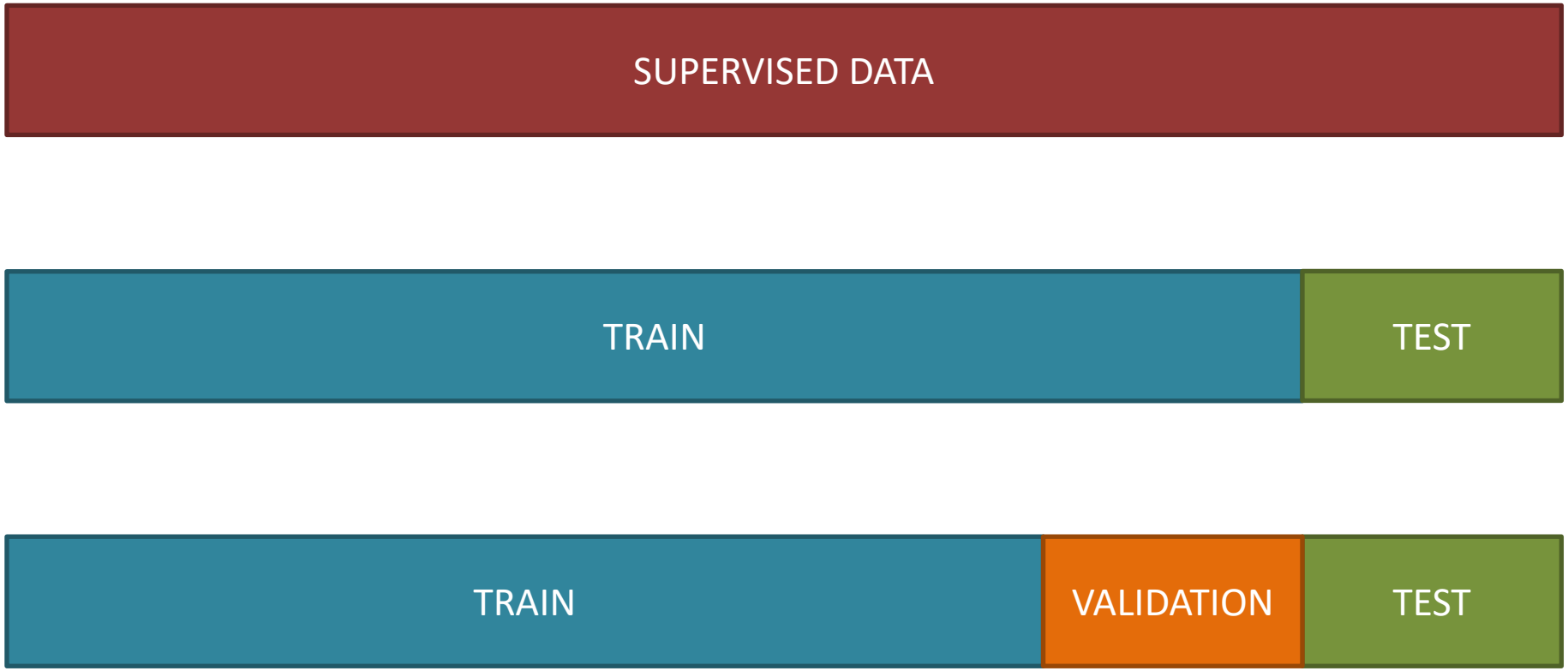
Parametri

Con parametri di un modello di machine e/o deep learning si intendono i parametri il cui valore cambia durante l'allenamento del modello. Il valore dei parametri cambia quindi in base ai dati del training set su «istruzione» dell'algoritmo utilizzato durante l'allenamento per minimizzare la funzione di errore.

Iper-parametri

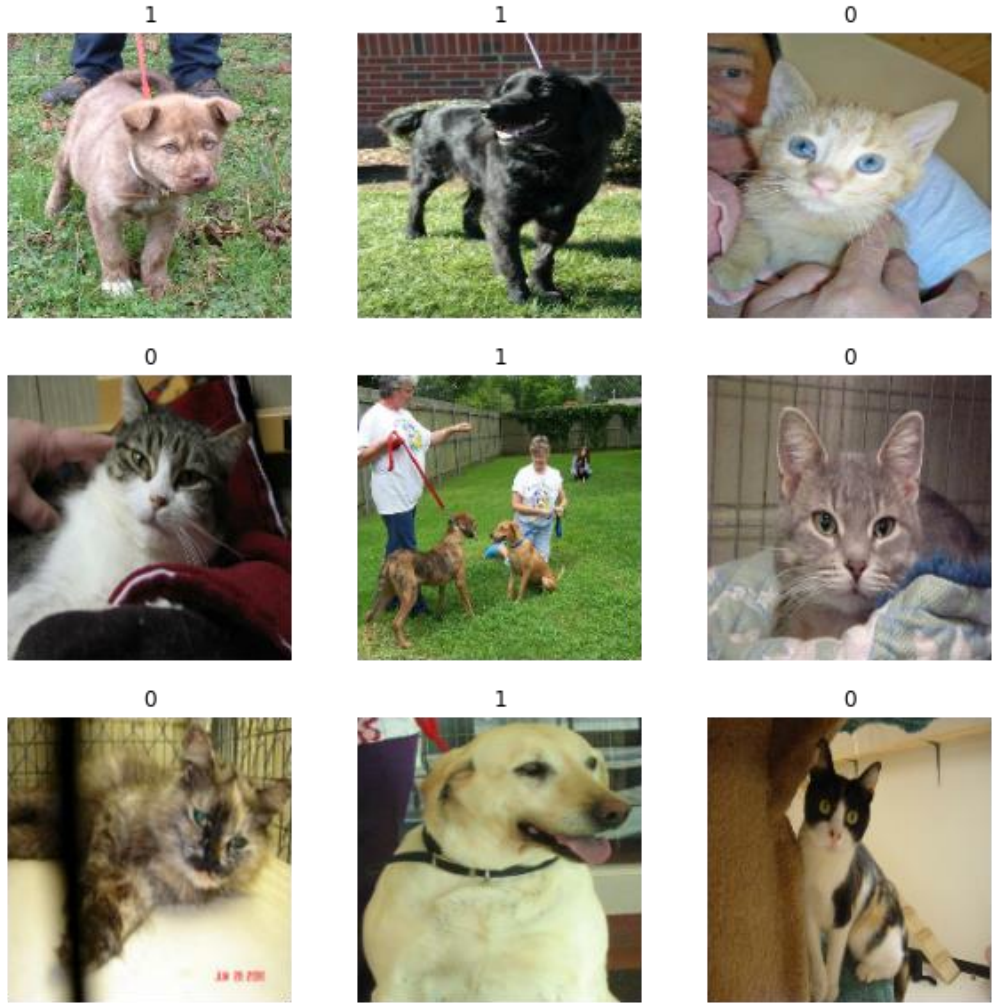
Con iper-parametri di un modello di machine e/o deep learning si intendono (se presenti) i parametri il cui valore viene deciso dal programmatore prima dell'allenamento del modello. Il programmatore può poi decidere di utilizzare parte dei dati che ha a disposizione (non appartenenti al training set) per formare un altro insieme di dati (validation set) ed utilizzarlo per ottimizzare la scelta degli iper-parametri.

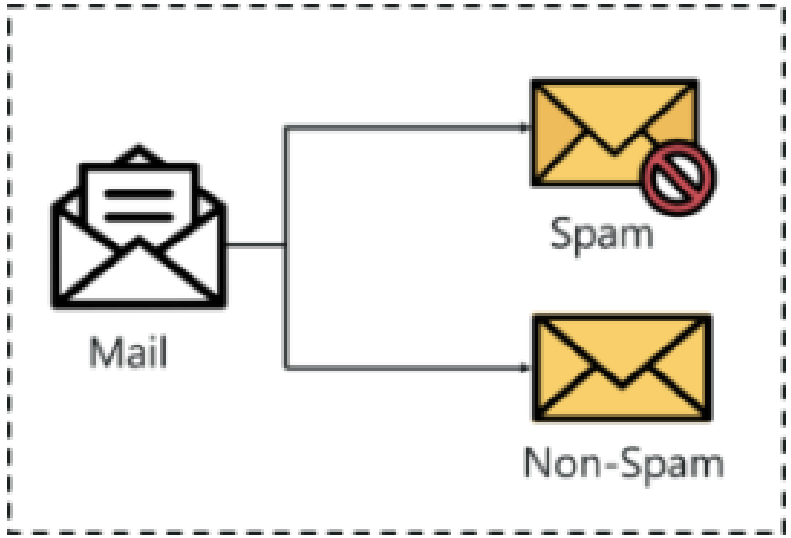
Machine Learning – Training, Validation e Test set



ATTENZIONE ALL'INTERSEZIONE TRA GLI INSIEMI: DEVE ESSERE VUOTA!

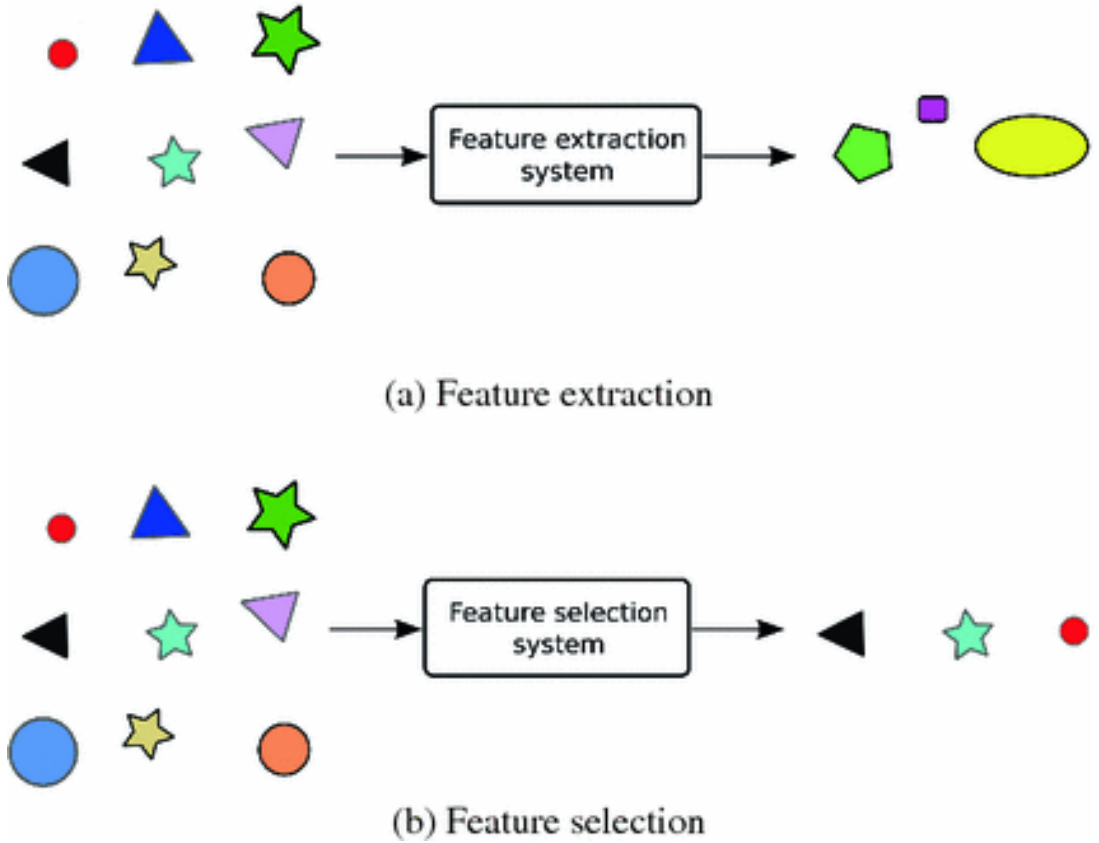
Machine Learning per la classificazione: Supervised learning





Machine Learning – Features

- Una qualsiasi informazione relativa ad un dato viene chiamata **feature**;
- Le features sono gli inputs in un processo/algoritmo di learning;
- Gli algoritmi di learning sono altamente sensibili alla scelta delle features;
- Scegliere delle «buone» features (**feature selection** e/o **feature extraction**) può essere molto difficile.

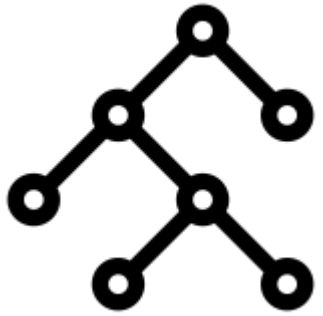


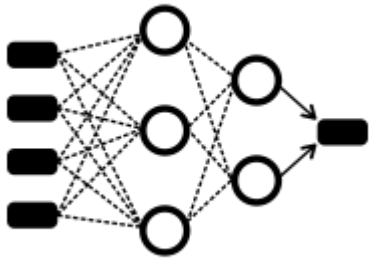
https://link.springer.com/chapter/10.1007/978-3-319-90080-3_2

ML supervisionato per la classificazione – Diverse tecniche

Diverse tecniche per definire i **modelli**:

- Modelli lineari;
- Modelli ad albero;
- Reti neurali;



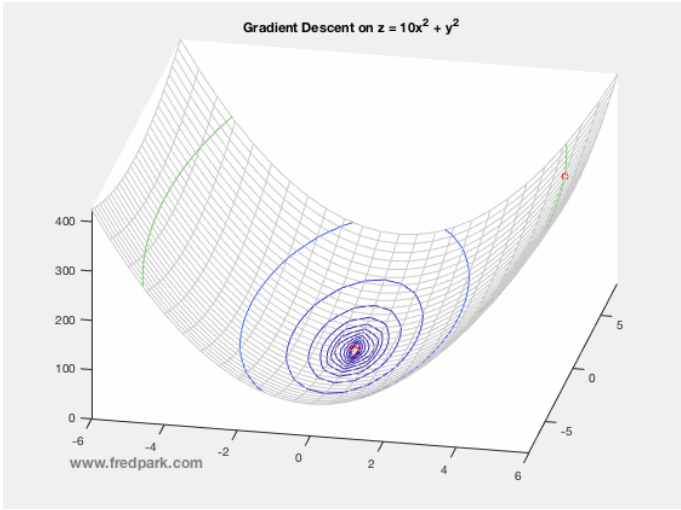


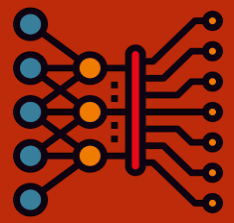
Diverse tecniche per definire **funzioni di errore**:

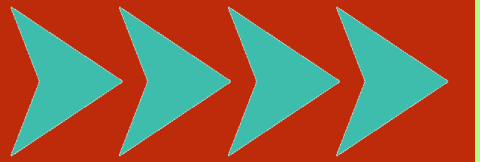
- Cross-entropy loss;
- Distanza cosenica;
- Funzione logistica;

Diverse tecniche per ottimizzare i modelli:

- Information gain;
- Conditional probability;
- Gradient descent;







Introduzione a ML per la Classificazione

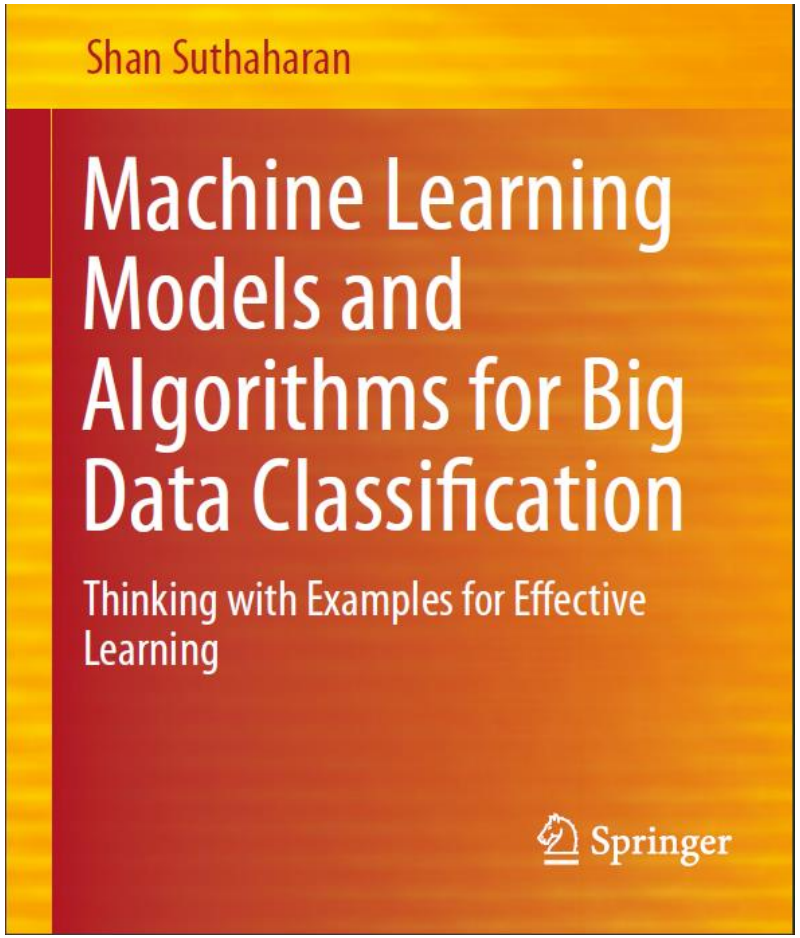
Lorenzo Stacchio

Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Approfondimenti



Capitolo 1 e 6

Perché ci riferiamo a modelli?

- Il Machine learning, riguarda l'esplorazione e lo sviluppo di modelli matematici e algoritmi per imparare dai dati.

- Spesso si incentra sulla classificazione, che viene implementata **modellando** una **funzione (parametrica) di mappatura ottimale** tra il dominio dei dati e il set di classi noti tra cui vogliamo discriminare.

- Questa mappatura potrebbe essere una funzione parametrizzata o dei processi parametrizzati che apprendono le caratteristiche di un sistema dai dati in input.

- Il termine algoritmo viene spesso confuso nel contesto dell'apprendimento automatico.

- Infatti, è la modellazione che può consistere in diversi algoritmi di apprendimento per la derivazione di un modello;

- L'algoritmo di apprendimento viene utilizzato per addestrare, convalidare, e testare il modello utilizzando un dato set di dati per trovare un valore ottimale per i parametri, convalidarlo e valutarne le prestazioni.

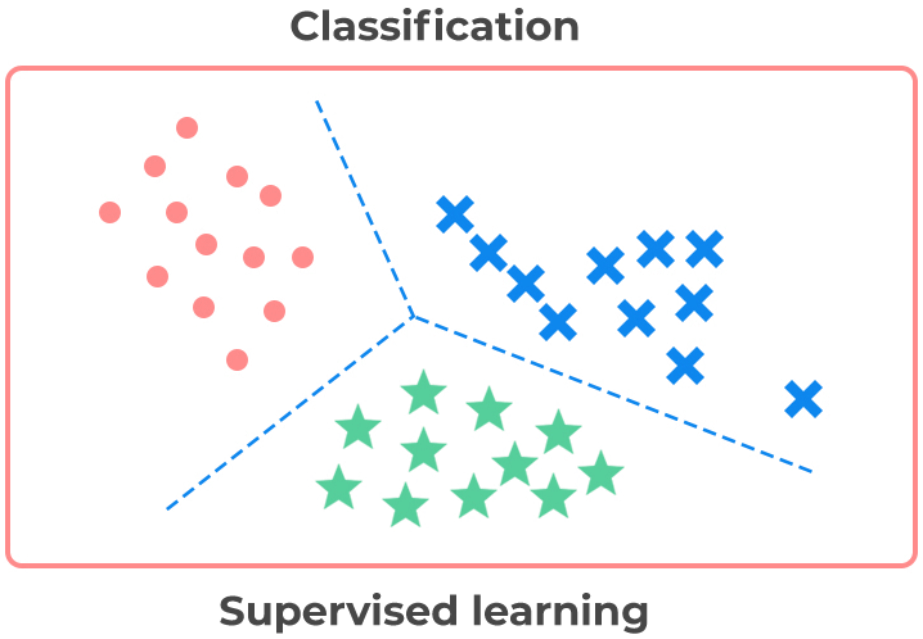
Machine Learning per la classificazione – Supervisionato vs non supervisionato

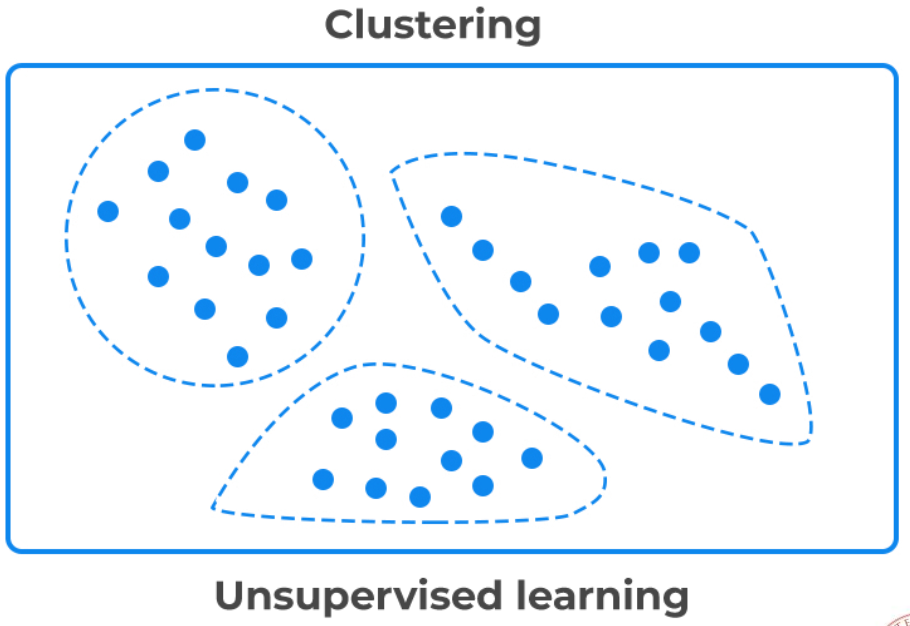
SUPERVISED LEARNING – Inputs + outputs (labels)

- **Classificazione**.

UNSUPERVISED LEARNING – Input (no labels)

- **Clustering;**
- **Feature extraction;**



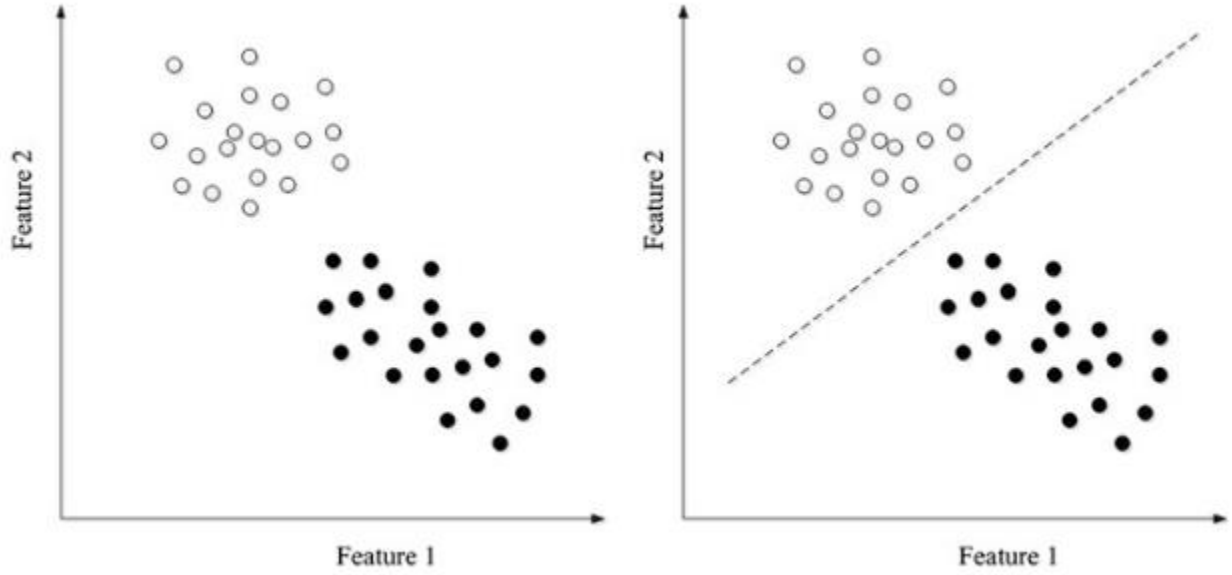


Classificazione (supervisionata)

- Quando affrontiamo problemi di classificazione, assumiamo di avere a disposizione **dati etichettati** (con classi) per generare delle regole (attraverso il training) per capire cosa dire di un certo dato in input, anche quando non l'abbiamo mai visto!

- Supponiamo ora di dover trovare una funzione matematica per separare I punti bianchi dai punti neri;

- Possiamo modellare una funzione parametrica di classificazione (lineare) che impari la configurazione ottimale dai dati stessi!



- In generale, un processo di classificazione, suppone di avere dati appartenenti ad un certo dominio D formato dalla relazione R_I, dove I indica il numero di features a disposizione!

- Se assumiamo che ci siano n classi, allora la funzione da modellare assume la seguente forma:

$$f : R^I \Rightarrow \{0, 1, 2, \ldots, n\}$$

- Ci sono diversi modelli di Machine learning per imparare questa tipologia di funzioni:
  - Alberi di decisione e Foreste random;
  - Funzioni logistiche univariate e multi-variate;
  - K-nearest neighbour;
  - Deep Learning (che è un campo a sè);

Clustering (classificazione non supervisionata)

- Quando affrontiamo problemi di clustering, assumiamo di avere a disposizione **dati non etichettati**, da cui possiamo ricavare delle regole approssimate per etichettare nuovi dati.

- Vediamo ora un esempio, in cui tutti I punti sono bianchi. Nonostante non abbiamo una conoscenza a priori di quali punti siano appartenenti a quali classi, possiamo chiaramente identificare dei pattern geometrici che ci indicano, in questo caso, che siamo in presenza di due gruppi (cioè cluster) per distinti!

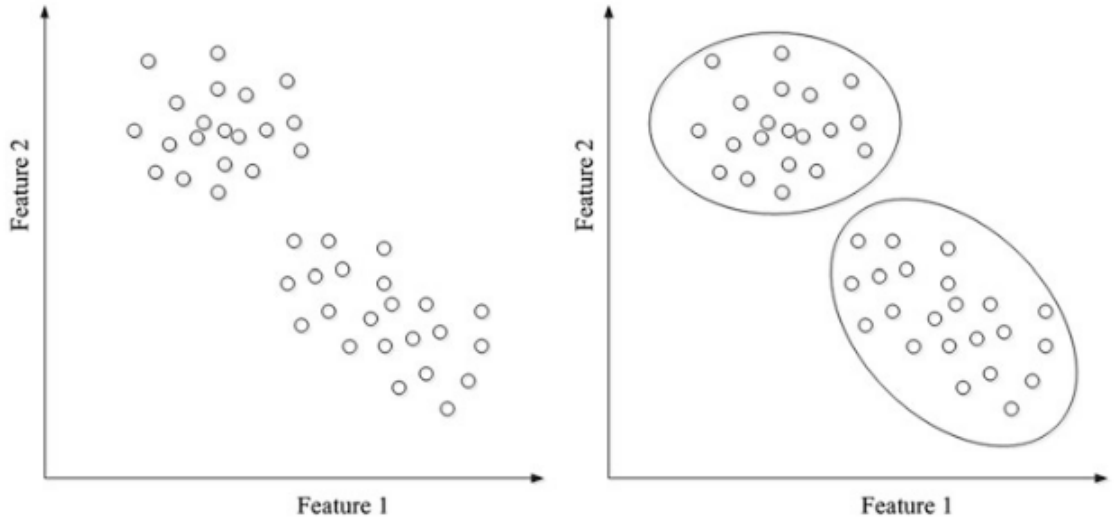


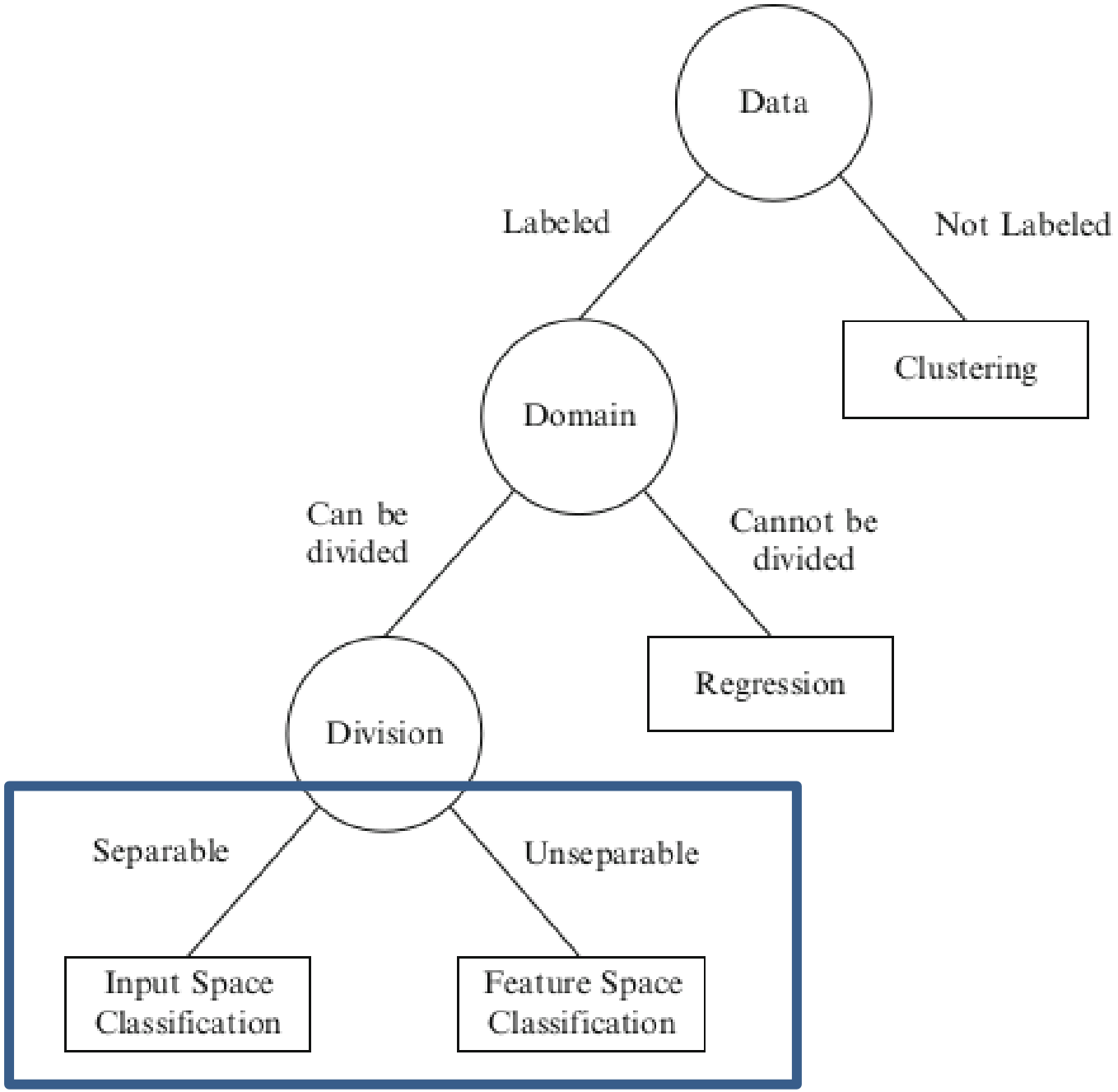
Fig. 1.4 Clustering is defined

- Come prima, un processo di classificazione, suppone di avere dati appartenenti ad un certo dominio D formato dalla relazione R_I, dove I indica il numero di features a disposizione!

- Se assumiamo di **poter estrarre** n^ classi, allora la funzione da modellare assume la seguente forma:

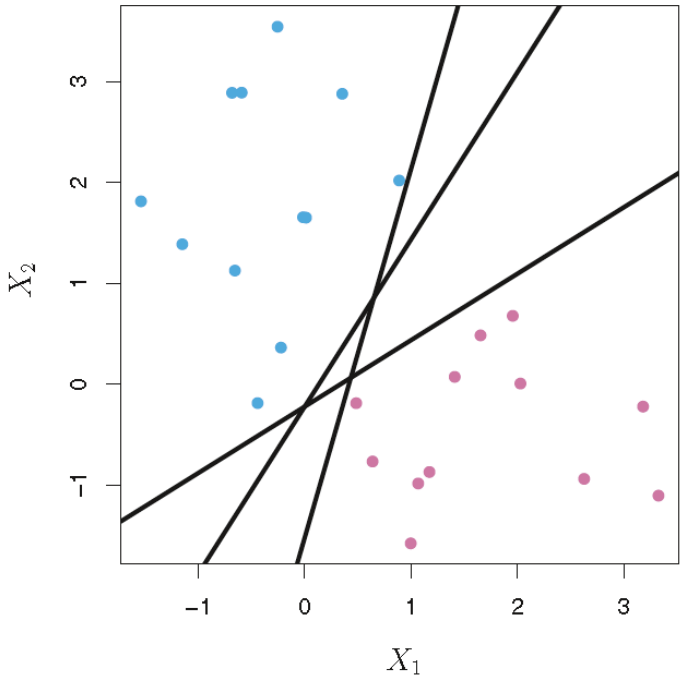
$$\hat{f} : R^I \Rightarrow \{0, 1, 2 \ldots, \hat{n}\}$$

- Ci sono diversi modelli di Machine learning per imparare questa tipologia di funzioni:
  - K-means clustering;
  - Hierarchical clustering.

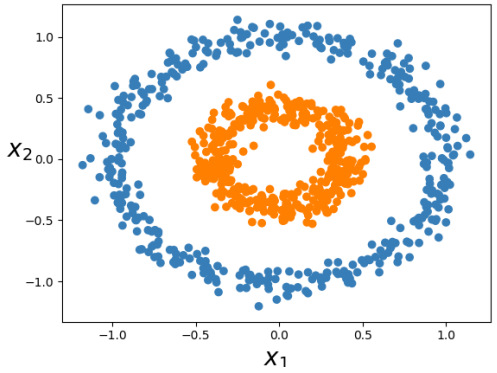


Dominio dei dati separabile vs non separabile

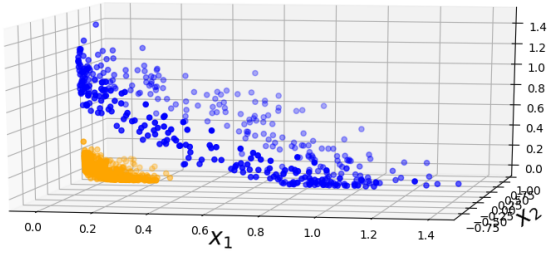
Spazio separabile



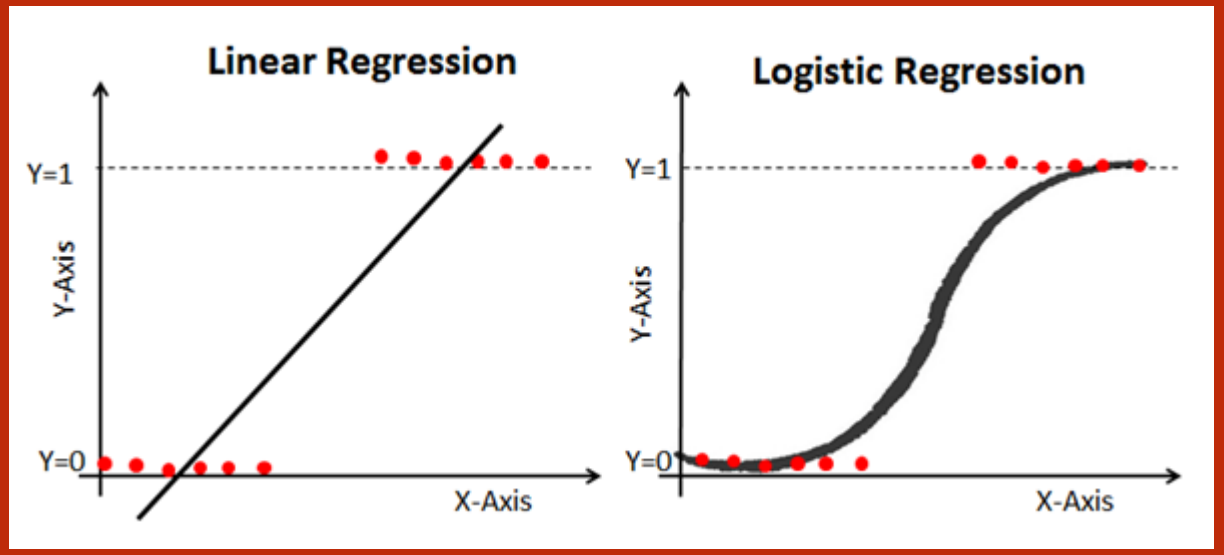
Spazio non separabile



Space transformation



https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f



Regressione Logistica

Lorenzo Stacchio

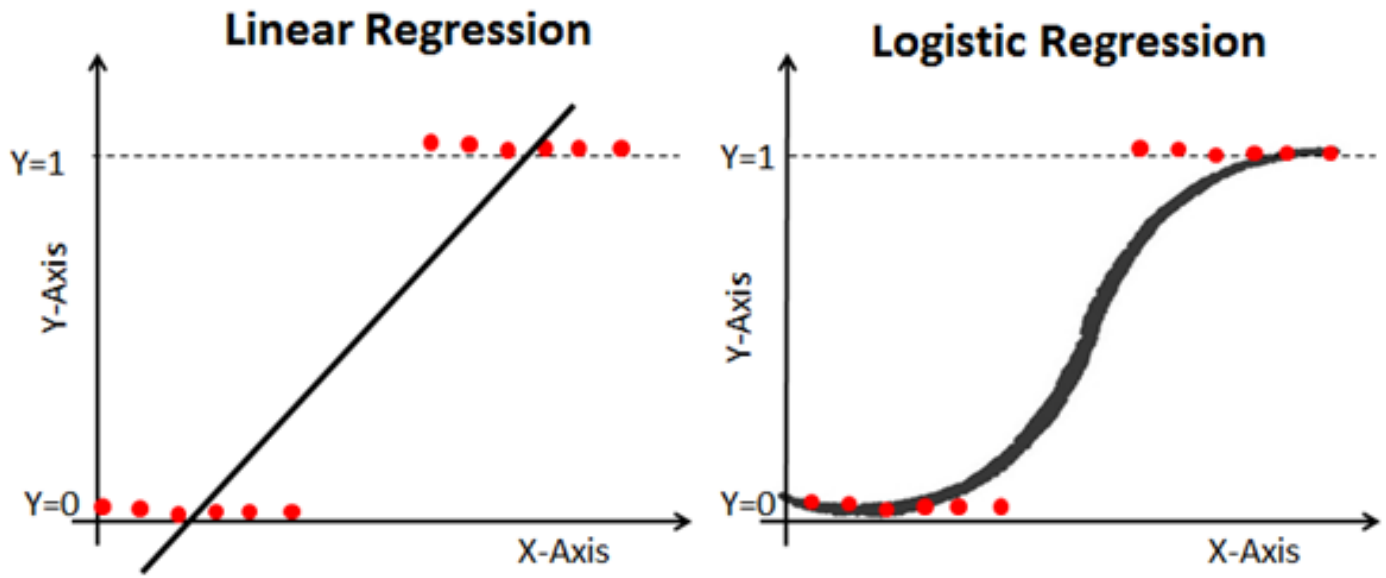
Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Regressione logistica: il metodo naive per la classificazione binaria

- I modelli di classificazione nascono per fornire risposte discrete in base a qualunque tipo di input;
- Esistono vari modelli di classificazione, il primo modello, che consente solamente di effettuare una **classificazione binaria** (es. spam/ non spam) è la **regressione logistica**;
- Infatti, la regressione lineare classica, non è adatta a discriminare tra due classi in quanto il modello si ottimizza su un dominio di **dati continuo** e non discreto!



- Possiamo riferirci inizialmente alla regressione logistica come una classica equazione, dove Y rappresenta la risposte discreta (es. vero/falso, 0/1);

- Y è in relazione con il dominio di dati X con una produttoria con il parametro a.

$$Y = aX$$

- Tuttavia, in questa equazione, **X può essere continuo o discreto** mentre **Y è discreto**;

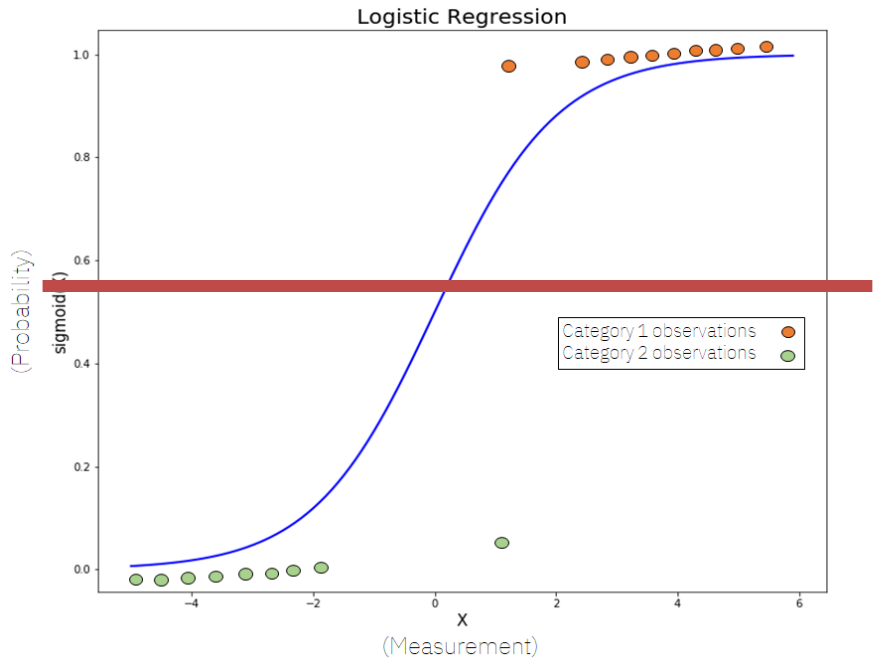
- In senso geometrico, questa relazione forma un effetto scala rendendo difficile adattare un modello matematico ai dati;

- Pertanto, dobbiamo definire una nuova equazione **intermedia ed infine applicare quello che viene definite come valore di soglia** per tirare fuori il valore discrete finale (vero/falso).

- La regressione logistica fitta non un modello di retta, ma una funzione cosiddetta **logistic (o sigmoide)**;

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

- Questa funzione mappa un qualunque valore, in un valore compreso tra 0 e 1, ossia una probabilità! In base al valore di threshold (0.5), la probabilità viene convertita nei sui valori estremi (0 e 1, spam e non spam, vero e falso).



- Cos'è la x nella formula?

$$x = \theta \times feature + b$$

$$sigmoid(x) = \frac{1}{1 + e^{-(\theta \times feature + b)}}$$

- Cosa vi ricorda questa formula?

- Che cosa possiamo ricavare dal fatto che questa sia esattamente la funzione di una retta? In realtà i parametri vengono ottimizzati su uno step intermedio continuo, che viene poi mappato in un valore di probabilità per classificare il nostro esempio in due classi!

- Come la addestriamo? Con metodi iterativi di ottimizzazione come la discesa del gradiente o con metodi probabilistici come la maximum likelihood estimation.

- Qual è la funzione di perdita che dobbiamo minimizzare?

Binary cross-entropy

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

M è il numero di esempi del set considerato

Cost è la funzione di loss

H è il nostro modello, theta sono i parametri

Features in input al modello

Classe

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1 - y)log(h_\theta(x))$$

Y è la classe, quando y è 1 solo questo termine è attivo

H è il nostro modello, theta sono i parametri

Y è la classe, quando y è 1 solo questo termine è attivo

Regressione logistica multi-variata cosa succede se siamo in presenza di più di una feature?

- Cos'è la x nella formula?

$$x = \theta \times feature + \theta_2 \times feature_2 + \theta_3 \times feature_3 \ldots + \theta_n \times feature_n + b$$

Let's code!

Boomer vs Non boomer

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Lorenzo Stacchio

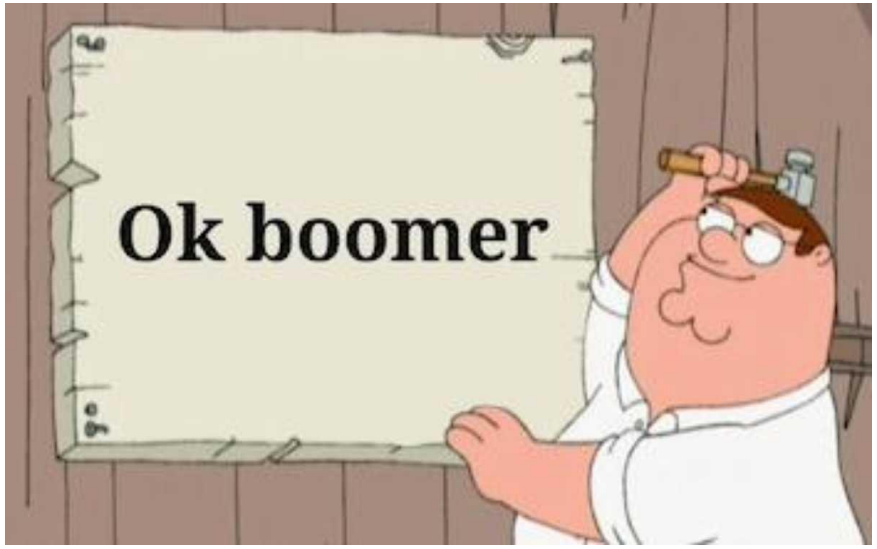
Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

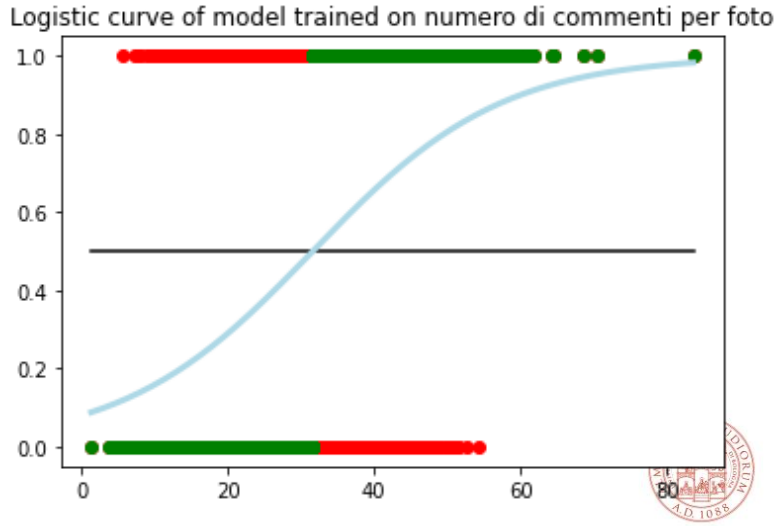
Regressione logistica per classificare un boomer da un non boomer

- Per divertirci assieme, ho creato un dataset sintetico ipotizzando l'esistenza di feature che fossero o meno discriminative per distinguere un boomer in un generico social network;

- Lo stesso fatto di creare un dataset sintetico per riconoscere i boomer è considerabile una roba da boomer? Probabilmente si.



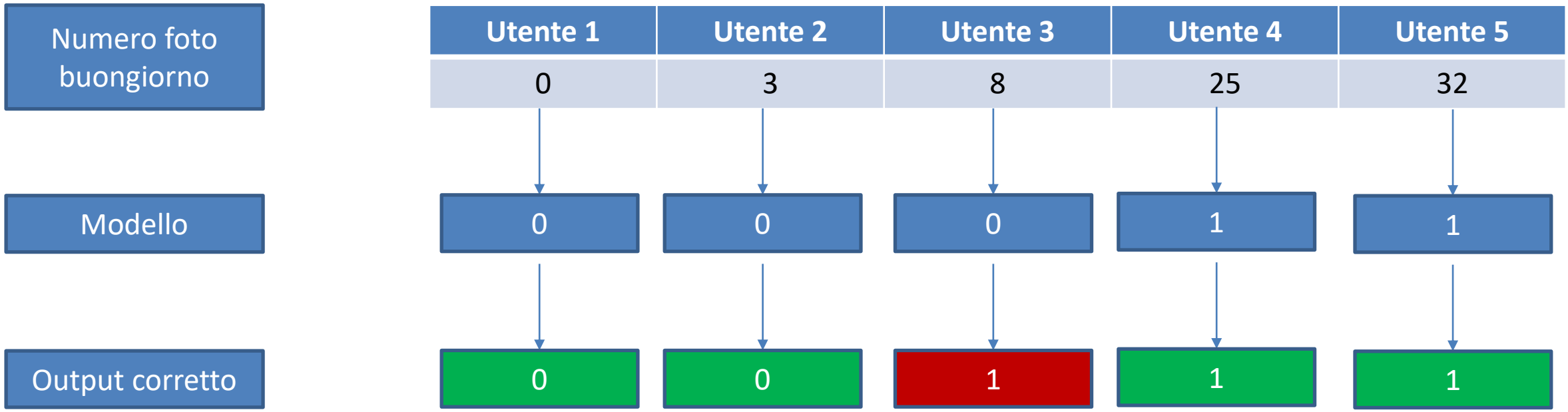
	numero di foto di buongiorno	numero di like per foto	numero di commenti per foto	boomer
0	21.511552	26.594268	31.419886	1
1	45.363636	26.056700	28.765644	1
2	72.151067	22.812552	29.106758	1
3	65.726706	18.886585	35.833150	1
4	45.304122	16.103766	41.662449	1
...	...	...	...	...
9995	3.844096	26.192649	28.039216	0
9996	3.484075	65.829633	44.351132	0
9997	4.183896	25.429293	23.193981	0
9998	2.533684	58.349155	22.775239	0
9999	2.296024	46.910793	17.531536	0



Misurare le performance di un classificatore: accuracy

- L'accuratezza è una metrica per valutare i modelli di classificazione.
- Informalmente, l'accuratezza è la frazione di previsioni che il nostro modello ha ottenuto correttamente. Formalmente, l'accuratezza ha la seguente definizione:

$$Accuracy = \frac{Numero\,di\,predizioni\,corrette}{Numero\,totale\,di\,predizioni}$$



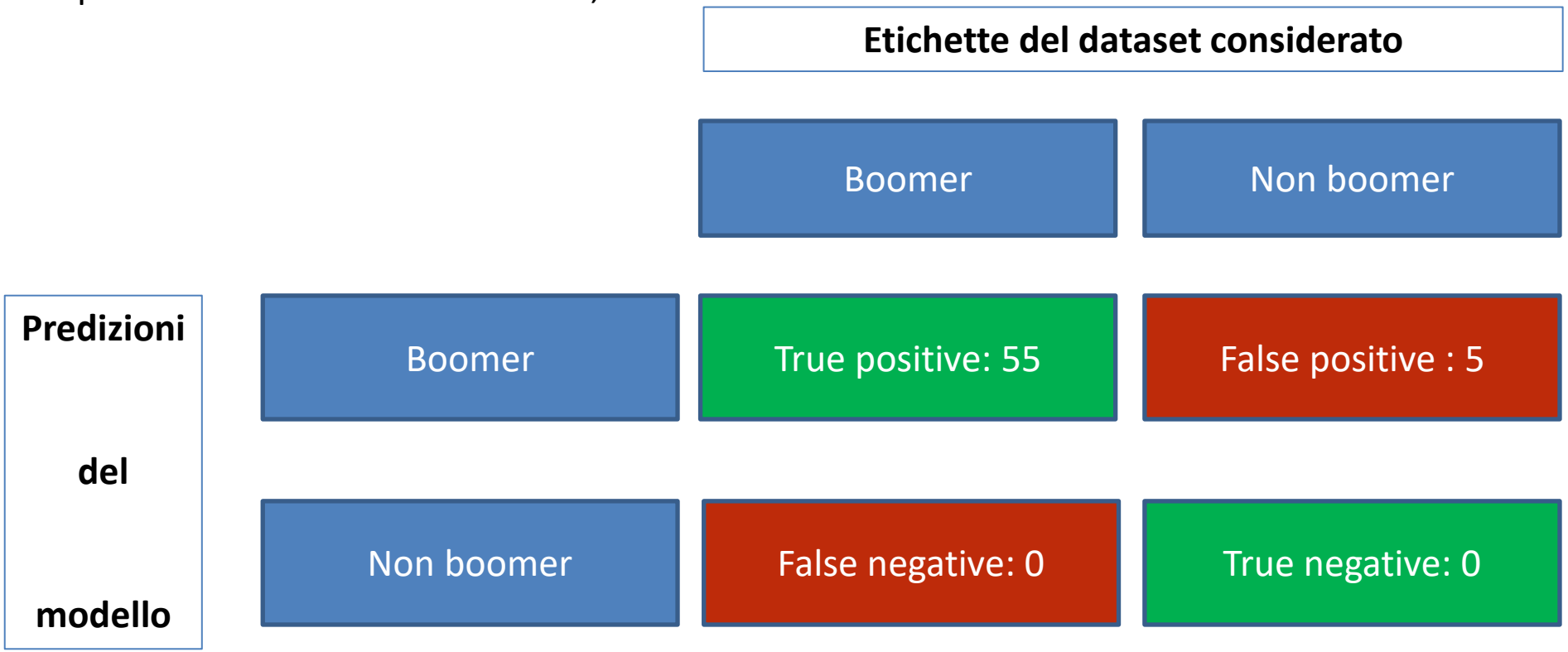
- Il nostro modello ha azzeccato 4 predizioni su 5, per cui abbiamo una accuratezza di 4/5 = 0,8, ossia 80%;
- L'accuratezza ha però dei limiti e funziona in maniera ottimale solo nella misura in cui ci troviamo in una situazione di dataset con classi bilanciate;
- Infatti, l'accuratezza da sola non racconta la giusta storia sulle performance del nostro modello con un set di dati con classi sbilanciate, cerchiamo di capire questo concetto con un esempio;

True Positives, True Negatives, False Positives e False Negatives

- Un **vero positivo (true positive)** è un risultato in cui il modello prevede correttamente la classe positiva. Allo stesso modo, **un vero negativo (true negative)** è un risultato in cui il modello prevede correttamente la classe negativa.

- Un **falso positivo(false positive)** è un risultato in cui il modello prevede in modo errato la classe positiva. E un **falso negativo (false negative)** è un risultato in cui il modello prevede in modo errato la classe negativa.

- Nel nostro caso la classi positiva corrisponde all'essere boomer e la classe negativa al non esserlo! Supponiamo di avere 55 esempi di boomer e 5 di non boomer;



- Considerando l’ultimo esempio, l’accuracy sarebbe corrisposta ad un valore molto alto, pari a 55 / 60 = 0.92, ossia 92%

- Tuttavia, vediamo che c’è un chiaro problema: tutti gli errori da parte del classificatore sono falsi positivi;

- Tutti gli esempi della classe **Non boomer** sono stati classificati come **Non Boomer** dal classificatore;

- Non avremmo potuto apprezzare questo fenomeno guardando solo l’accuracy, e soprattutto non avremmo potuto correre ai ripari per risolvere il problema!

- Secondo voi qual è il problema? Il classificatore si è chiaramente overfittato sulla classe **Boomer**!

Precision, Recall and F1-score

- In questi casi di sbilanciamento è bene trovare delle metriche più profonde della semplice accuratezza!
- A questo scopo esistono **precision e recall**!
- **Precision: Quale proporzione di classificazioni positive era effettivamente corretta?**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Nel nostro esempio:
  - **TP = 55;**
  - **FP = 5;**
  - **Precision = 55/60 = 0.92**

- **Recall: Quale proporzione di corretti positivi è stata identificata correttamente?**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Nel nostro esempio:
  - **TP = 55;**
  - **FN = 0;**
  - **Recall = 55/55 =1**

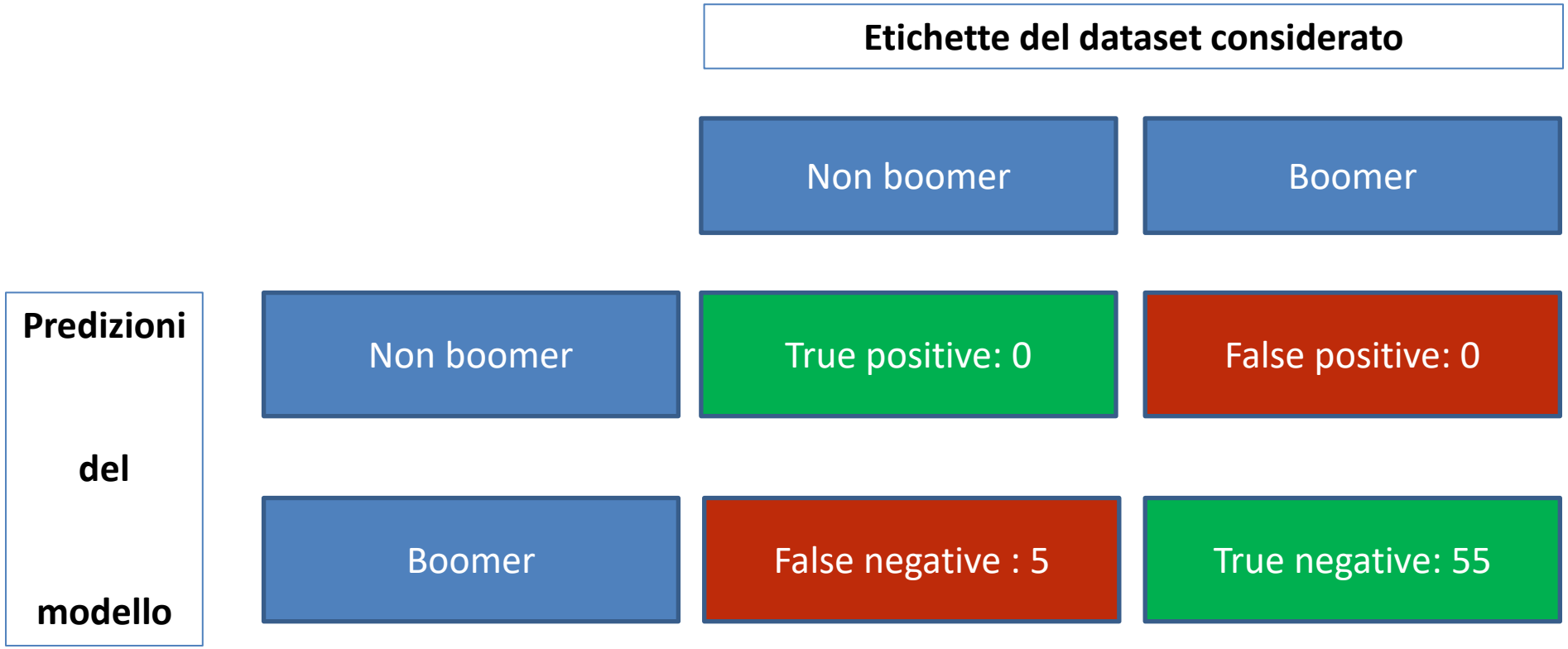
- **F1: Media armonica tra precision e recall (da usare al posto dell'accuracy)**

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

- Nel nostro esempio:
  - **precision = 0.92;**
  - **recall = 1;**
  - **F1 = 0.96**

Momento, momento, momento … Perchè questi risultati vanno così bene?

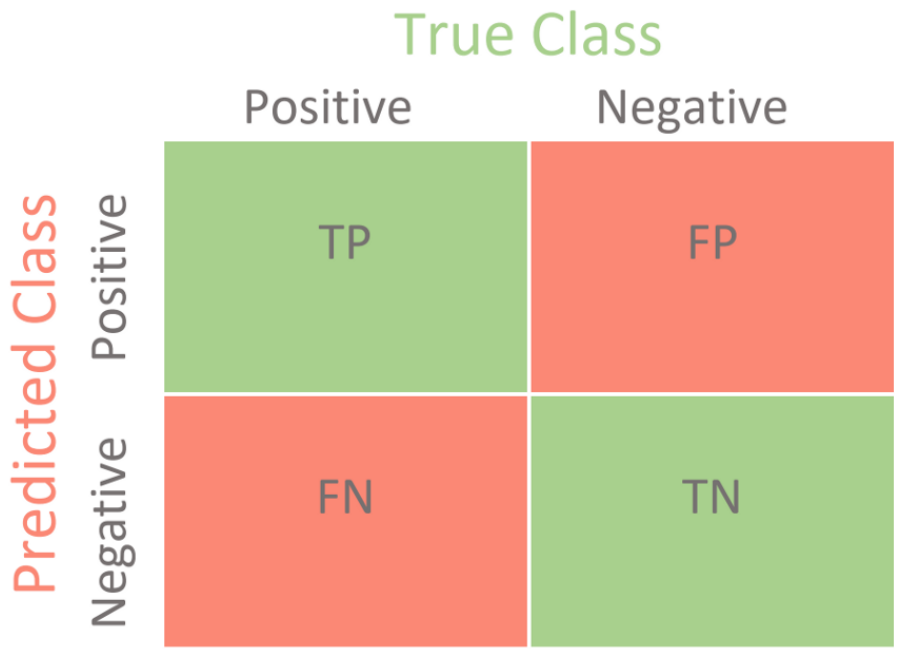
- Perchè abbiamo considerato come classe positiva la classe dei boomer!
- Cosa succederebbe se considerassimo come classe positiva quella dei non boomer?

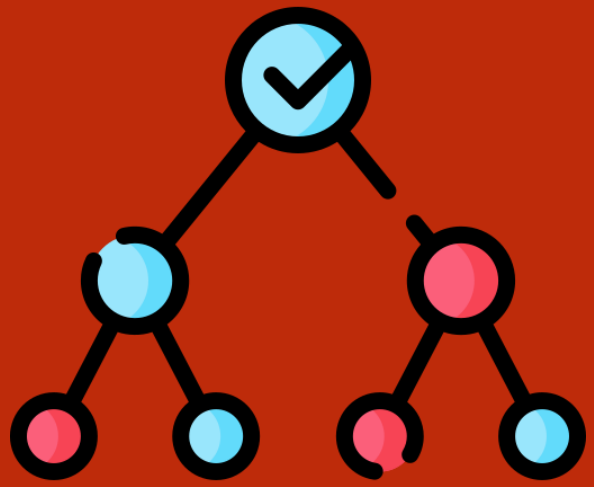


- **Precision = 0 / 0 è impossibile, si considera come 0;**
- **Recall = 0/5 = 0;**
- **F1- score = 0**

Confusion matrix

- La matrice di confusione è un modo tabellare per visualizzare le prestazioni del modello di previsione.
- Ogni voce della matrice di confusione denota il numero di previsioni fatte dal modello in cui ha classificato le classi correttamente o in modo errato.
- **Si basa sui concetti che abbiamo appena visto!**





Alberi di decisione

Lorenzo Stacchio

Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Alberi di decisione

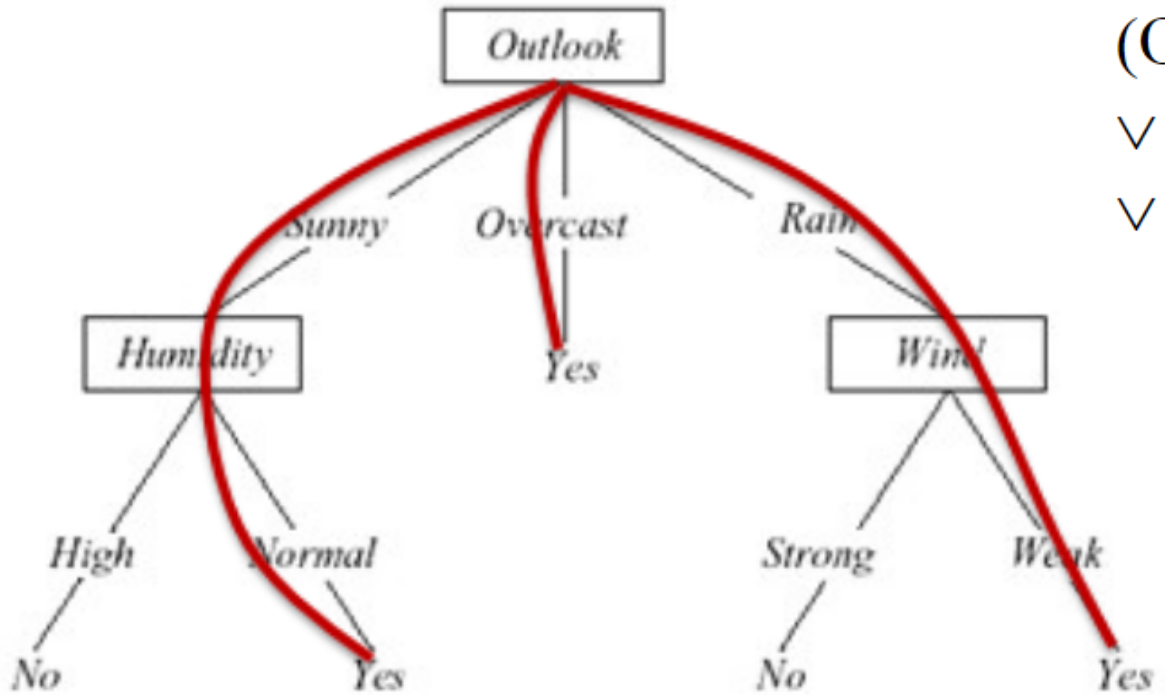
- Nel mondo del Machine Learning, **gli alberi di decisione** sono una **sorta di modelli non parametrici**, che possono essere utilizzati sia per la classificazione che per la regressione;

- Questi modelli sono perciò **flessibili poiché** non aumentano il numero di parametri quando aggiungiamo più feature (se li costruiamo correttamente) e possono produrre una previsione categorica (come se una pianta è di un certo tipo o no) o una previsione numerica (come il prezzo di una casa);

- Sono costruiti utilizzando due tipi di elementi: nodi e rami. Ad ogni **nodo**, viene valutata una delle **feature** dei nostri dati per suddividere le osservazioni nel processo di addestramento o per fare in modo che un punto dati specifico segua un determinato percorso quando si effettua una previsione.

- In pratica, per ogni feature, viene effettuato un test booleano per capire quale percorso dovremo intraprendere per fornire una corretta classificazione!

PlayTennis?



$$(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal})$$
$$\vee\ (\text{Outlook} = \text{Overcast})$$
$$\vee\ (\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})$$

Fonte

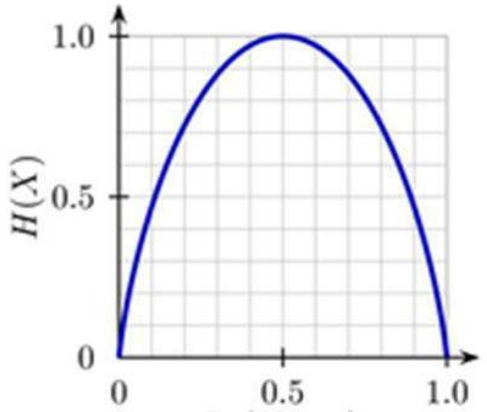
Come costruire un albero di decisione? Partiamo dai dati

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Come scegliere l'ordine con il quale valutare le feature? L'entropia è sempre la risposta!

- L'entropia dell'informazione o entropia di Shannon quantifica la quantità di incertezza (o sorpresa) coinvolta nel valore di una variabile casuale;
- Matematicamente, è definite come la sommatoria dei prodotti tra le probabilità di un evento ed il logaritmo della probabilità inversa, per tutti gli eventi associate a una variabile causale;
- Il suo significato nell'albero decisionale ci permette di stimare l'impurità o l'eterogeneità della variabile che stiamo considerando;

$$\text{Entropy} = \sum p(x) \log\left(\frac{1}{p(x)}\right)$$



Probabilità associata ad un certo evento

Logaritmo dell'inversa della probabilità di quell'evento

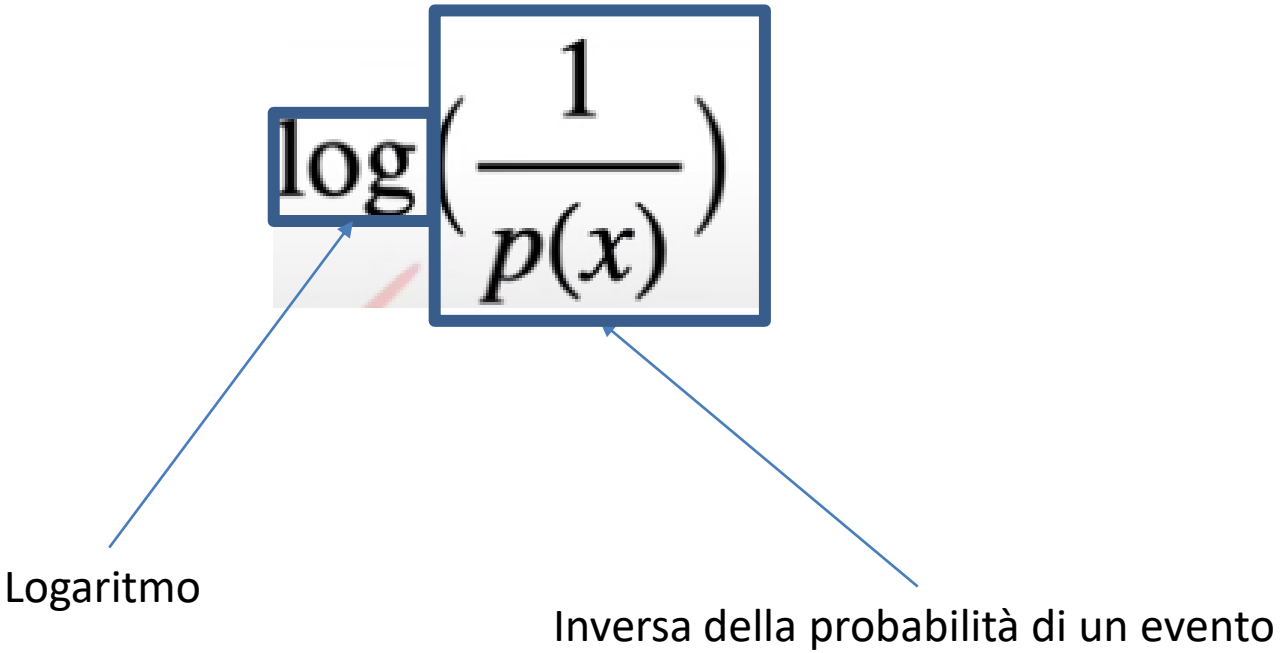
Stima della sorpresa nell'intera variabile causale

- Supponiamo di considerare una variabile casuale (feature) che definisca due eventi: testa o croce;
- Supponiamo che questa moneta sia truccata che 9 volte su 10 ritorni testa;
- Avremo quindi la probabilità del 90% di avere testa e del 10% di avere croce;

$$Moneta = (testa, croce)$$

$$p(Moneta = testa) = 0.9 \qquad p(Moneta = croce) = 0.1$$

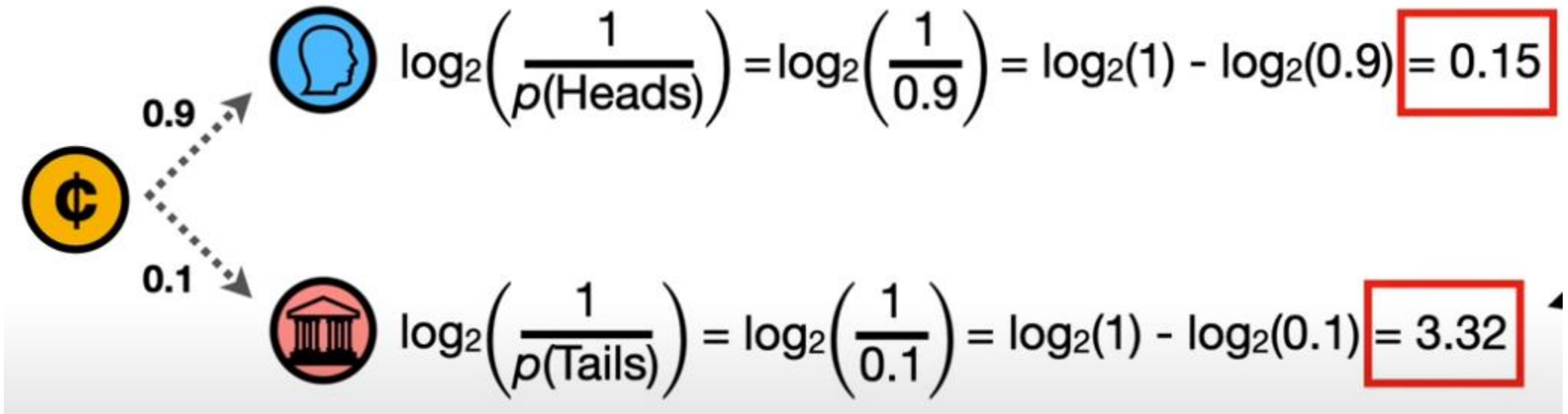
Qual è il grado di sorpresa dietro ad ogni evento?



- L'inversa della probabilità cerca di fornire un'informazione significativa della sopresa, intuitivamente, più la probabilità è alta, meno saremmo sopresi che quell'evento sia accaduto (e viceversa);

- Il logaritmo è utilizzato per gestire due problemi:
  - se p(x) fosse 0, l'operazione sarebbe impossibile ma sfruttando le proprietà dei logaritmi riusciamo a renderla calcolabile;
  - La funzione logaritmo ci fornisce una funzione più levigata, consentendoci di ricavare dei calcoli più precisi;
  - N.B., la base del logaritmo dipende dal numero di eventi definiti in x (in questo caso 2);

$$\log\left(\frac{1}{p(x)}\right) = \log(1) - \log(p(x))$$



https://www.youtube.com/watch?v=YtebGVx-Fxw

Probabilità associata ad un evento e concetto di «sorpresa»

$$Moneta = (testa, croce)$$

$$p(Moneta = testa) = 0.9$$

$$\log_2\left(\frac{1}{p(testa)}\right) = 0.15$$

$$p(Moneta = croce) = 0.1$$

$$\log_2\left(\frac{1}{p(croce)}\right) = 3.32$$

- A questo punto, vogliamo calcolare il livello di sorpresa per la variabile casuale nella sua interezza!
- Possiamo approssimare il livello di entropia, usando la formula del valore atteso di probabilità;

$$\sum_{x=1}^{X} x \times p(X = x)$$

$$(0.9 \times 0.15) + (0.1 \times 3.32) = 0.47$$

Entropia!

Probabilità di osservare la sorpresa

Valore della sorpresa

$$\sum_{x=1}^{X} x \times p(X = x)$$

$$\sum_{x=1}^{X} \log_2 \left(\frac{1}{p(x)} \right) \times p(x)$$

Entropia di una moneta non truccata

$$Moneta = (testa, croce)$$

$$p(Moneta = testa) = 0.5$$

$$\log_2\left(\frac{1}{0.5}\right) = 1$$

$$p(Moneta = croce) = 0.5$$

$$\log_2\left(\frac{1}{0.5}\right) = 1$$

$$entropy(moneta) = (0.5 \times 1) + (0.5 \times 1) = 1$$

- Abbiamo entropia massima!
- Se ci pensate è corretto: ad ogni lancio non sappiamo mai cosa aspettarci!

Torniamo ai nostri alberi

- L'entropia nell'albero decisionale ci permette di stimare l'impurità o l'eterogeneità della variabile che stiamo considerando;

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

- Outlook è una variabile casual con 3 eventi:
  - Sunny, probabilità 5/14
  - Overcast, probabilità 4/14
  - Rainy, probabilità 5/14

- Lo stesso ragionamento possiamo fare per le altre variabili… e possiamo chiaramente calcolarne l'entropia!

- Perchè calcolare l'entropia di queste variabili?

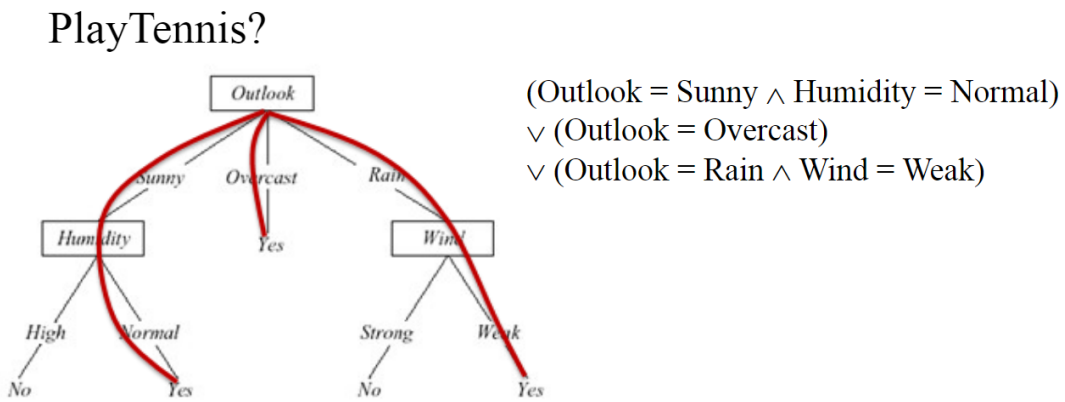
Information gain

- L'information gain o reduction in randomness è semplicemente la sottrazione tra l'entropia della variabile target (giocare a tennis) e l'entropia di una variabile feature (outlook, temperature, humidity, windy).

- **L'information gain è fondamentale**: ci dice quale variabile, tra le tante che stiamo considerando ci può far prendere più facilmente una decisione sul giocare a tennis o meno!

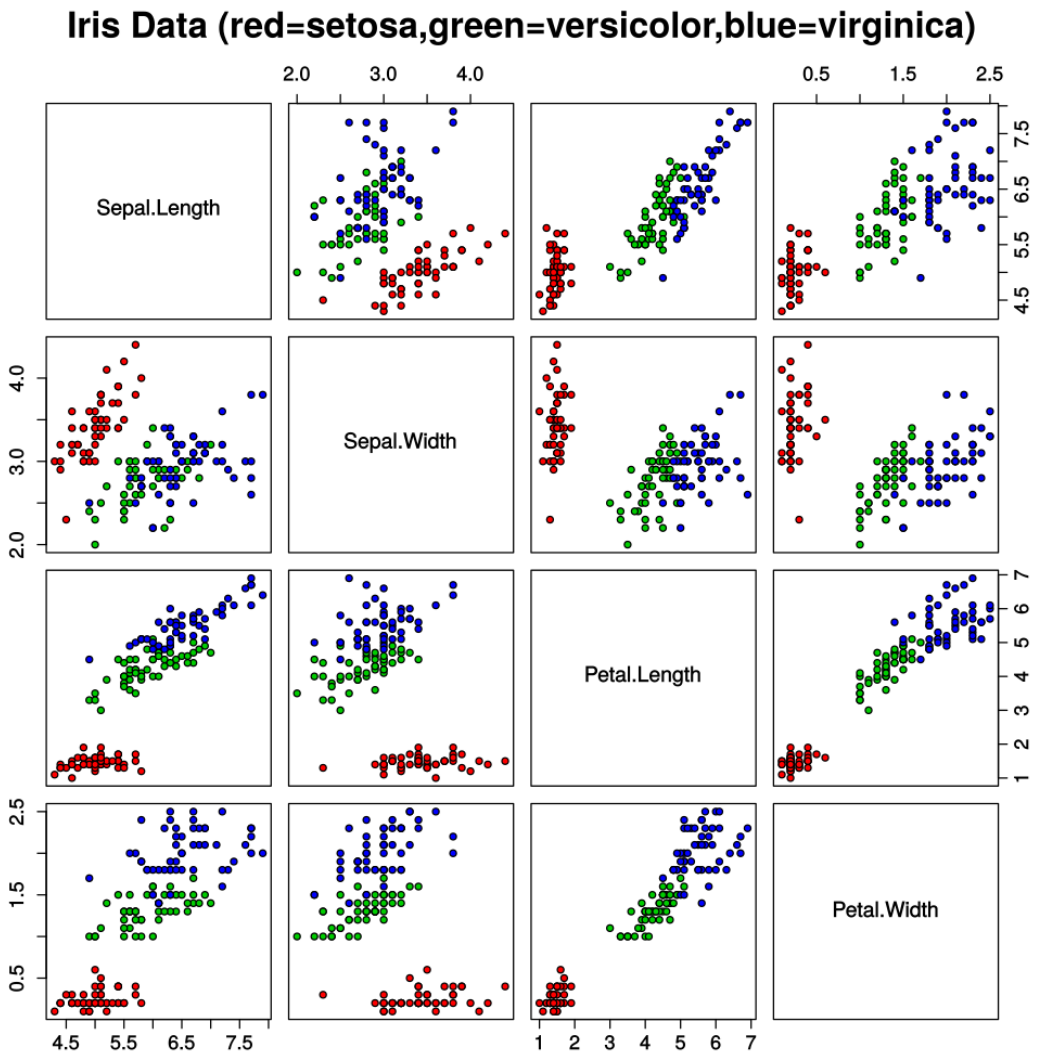
- Senza fare calcoli matematici, **pensiamo intuitivamente: se ho un alto grado di incertezza nel giocare o meno, le previsioni meteo mi danno un fattore che diminuisce la mia incertezza!**

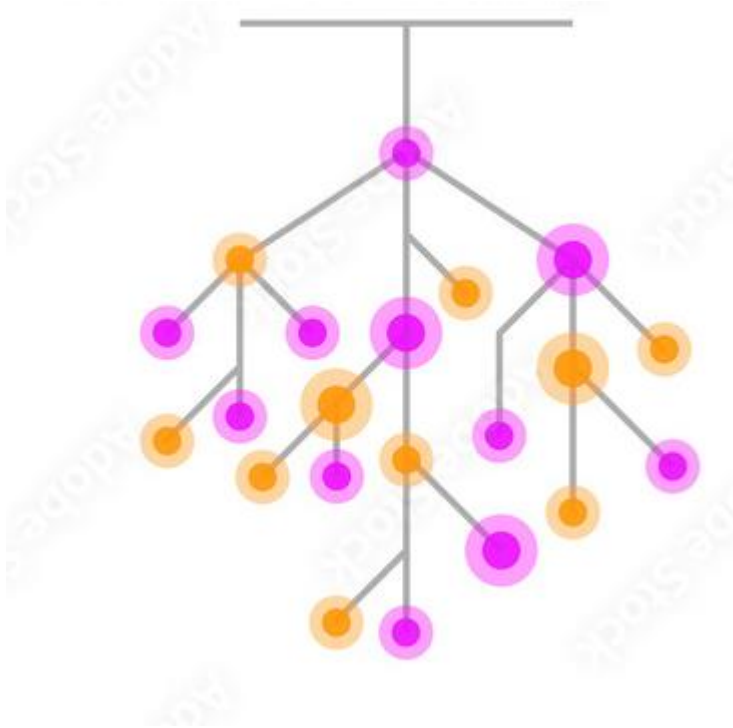
- **Una volta viste le previsioni meteo, chiaramente devo capire se l'umidità e il vento hanno valori corretti per permettere un buon gioco;**

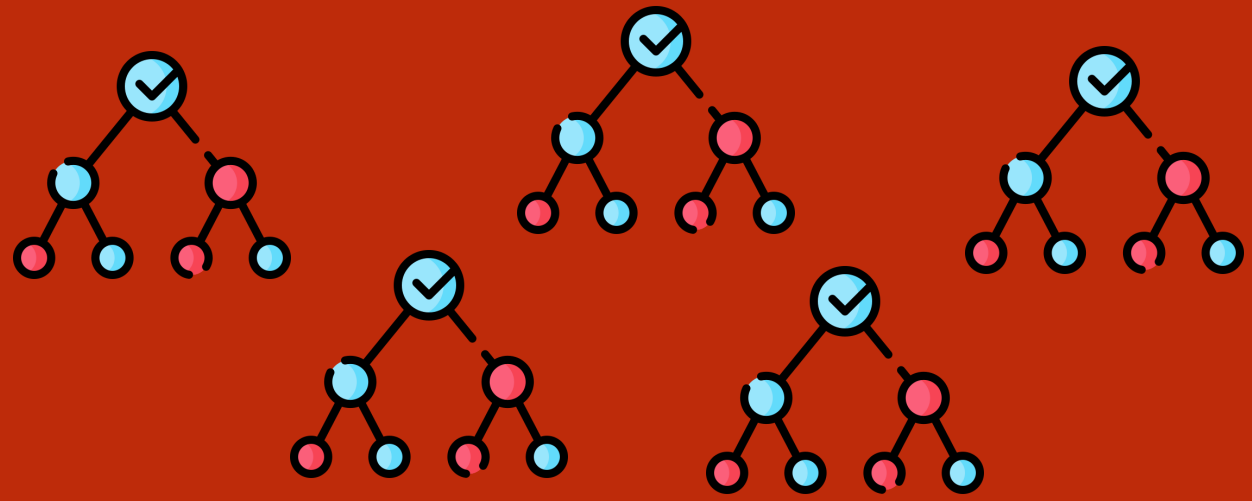


Let's code!

Un processo di classificazione green: classificare il dataset iris con un albero di decision







Random forest

Lorenzo Stacchio

Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

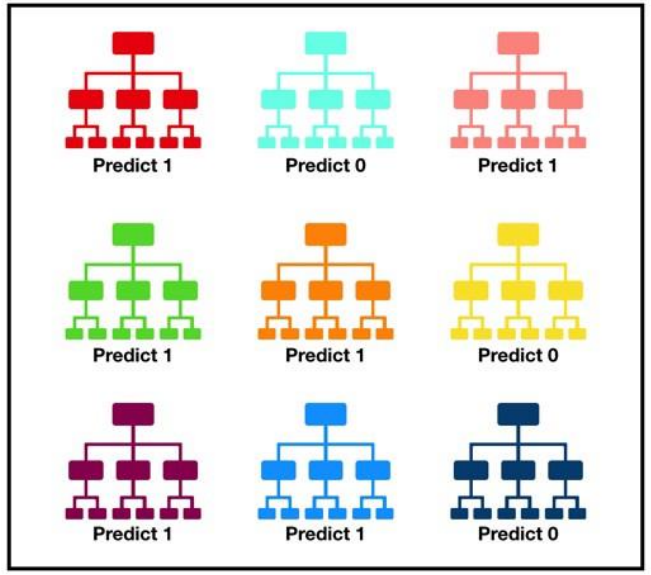
Foreste casuale (random forest)

• Una foresta casuale, come suggerisce il nome, consiste in un gran numero di alberi decisionali individuali che operano come quello che viene definito **ensemble**;

• Ogni singolo albero nella foresta casuale emette una previsione di classe e la classe con il maggior numero di voti diventa la previsione del nostro modello;

• Il concetto fondamentale alla base della foresta casuale è semplice ma potente: la saggezza delle folle.

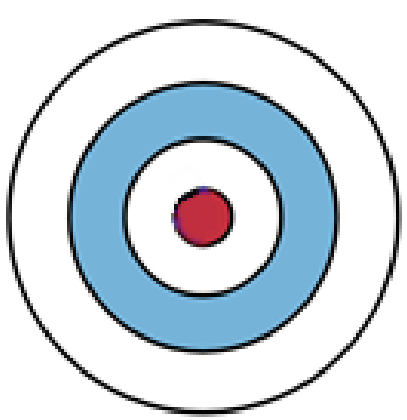
• In data science, il motivo per cui il modello di foresta casuale funziona così bene è che un gran numero di modelli relativamente non correlati che operano come un insieme supererà i singoli modelli costituenti (**average effect**).



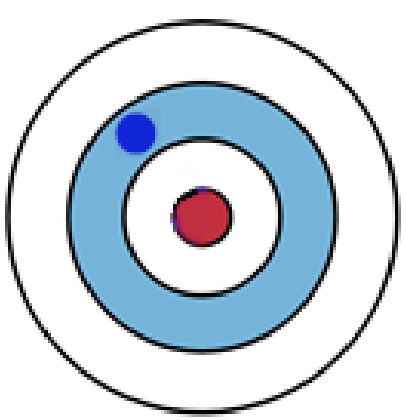
Tally: Six 1s and Three 0s
Prediction: 1

• Per ottenere questo effetto però, gli alberi non devono andare nella stessa direzione di scelta (alberi non correlati)!

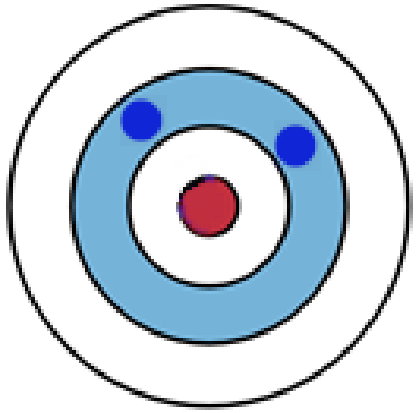
• In questo modo, alcuni alberi proteggeranno gli altri dagli errori!



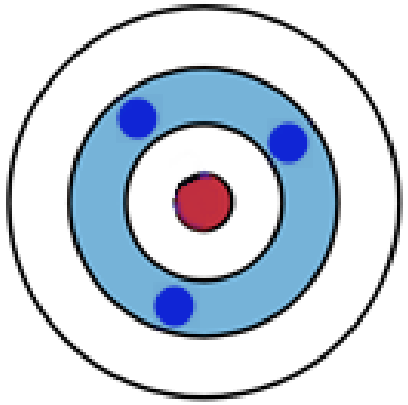
Piano della loss



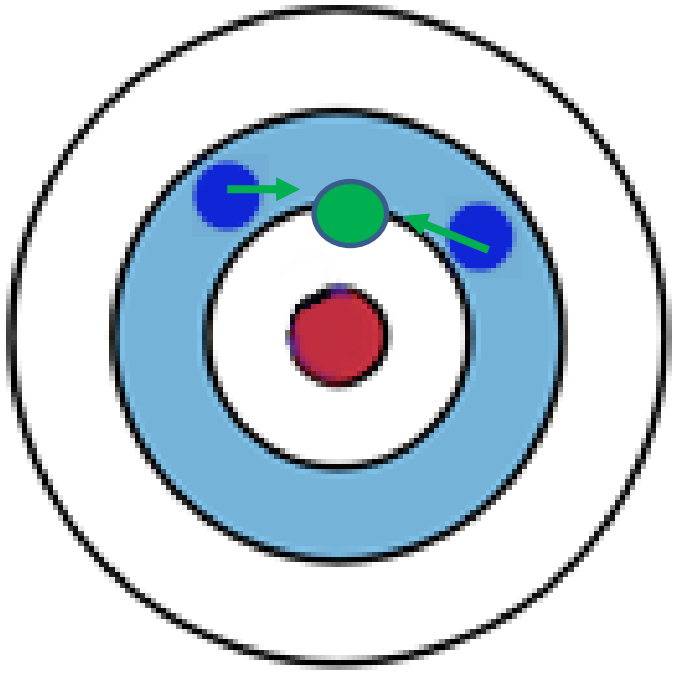
1° albero



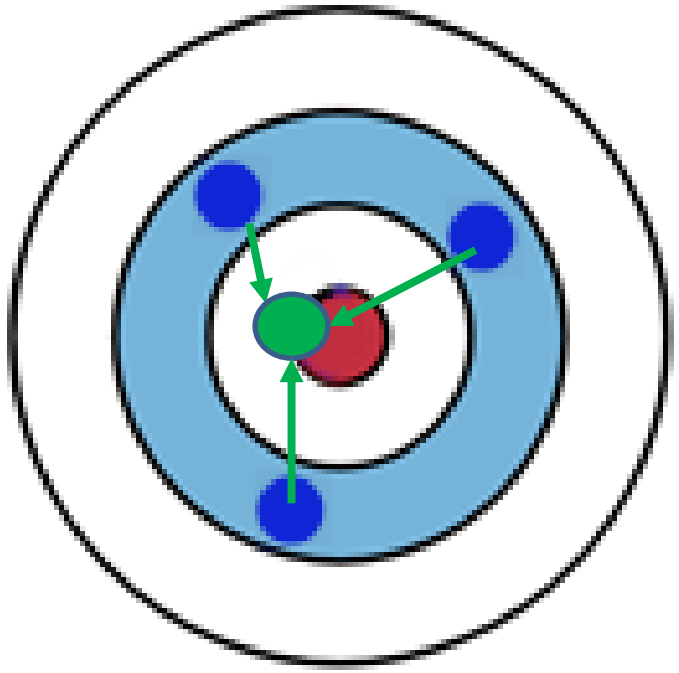
2 alberi scorrelati



3 alberi scorrelati



2 alberi scorrelati



3 alberi scorrelati

Come evitare alberi correlati? La tecnica Bagging

- Gli alberi delle decisioni sono molto sensibili ai dati su cui vengono addestrati: piccole modifiche al set di addestramento possono comportare strutture ad albero significativamente diverse;

- La random forest ne trae vantaggio, consentendo ad ogni singolo albero di campionare in modo casuale dal set di dati **con reinserimento**, risultando in alberi diversi (processo di bagging).

- Da notare che i vari alberi non suddividono il dataset in parti uguali l'una separata dall'altra, ma pescano semplicemente dei campioni diversi dallo stesso dataset!

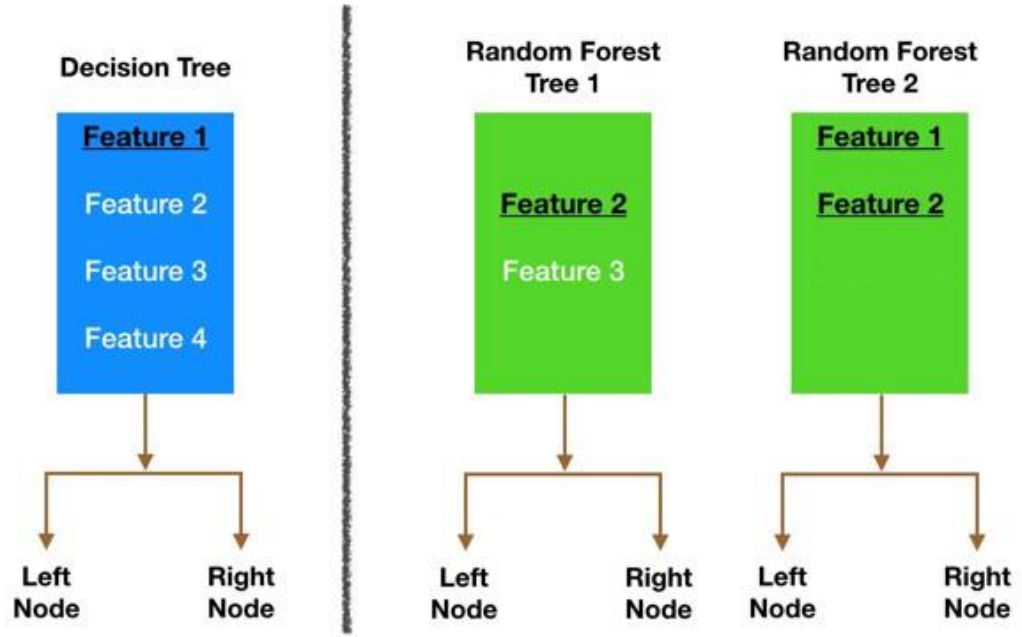
- Per semplicità, pensate a una scatola da cui ognuno di voi possa pescare un numero. Normalmente, una volta pescato, un numero non viene più ripetuto ma non qui, dove **ogni giocatore rimetterebbe il numero dentro la scatola, dando la possibilità al giocatore successivo di pescarlo!**

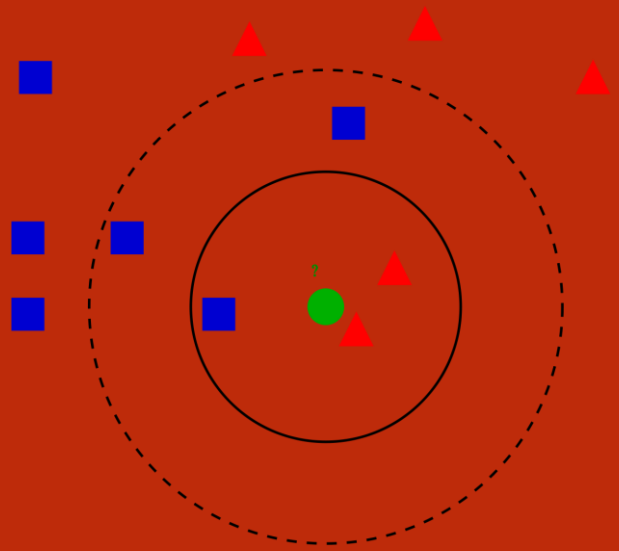
Come evitare alberi correlati? Feature randomness

- Normalmente, ogni albero di decisione ad ogni nodo analizzerebbe il valore di una certa feature per capire cosa fare;

- Tuttavia, noi vogliamo alberi che siano non solo scorrelati nei campioni ma anche nelle feature!

- Per questo, si decide un numero random di feature da assegnare ad ogni albero (il re-inserimento vale sempre);





Il mondo è pieno di classificatori

Lorenzo Stacchio

Studente di dottorato in Computer Science

Dipartimento di Scienze per la Qualità della Vita

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Il mondo è pieno di classificatori

- Ci sono moltissimi classificatori (supervisionati e non) basati su ML di cui non abbiamo parlato:
  - K-means;
  - KNN;
  - SVM;
  - Naive Bayes;
  - Multi-layer perceptron;

- C'è poi un'intera branca dedicata al deep learning, che è ora considerato lo stato dell'arte per la classificazione delle immagini;

- Tuttavia, molti di questi si basano su concetti di cui abbiamo parlato:
  - Probabilità;
  - Entropia;
  - Gradient descent;
  - …;

Python ha un'attiva comunità di sviluppatori Machine Learning

Big Data & Data Science

Deep Learning

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Lorenzo Stacchio

Dipartimento di Scienze per la Qualità della Vita

lorenzo.stacchio2@unibo.it