



Università di Parma

Dipartimento di Ingegneria e Architettura

Introduzione all'Intelligenza Artificiale

Big Data & Business Intelligence

A.A. 2022/2023

Corso di «Introduzione all'Intelligenza Artificiale»

Corso di «Big Data & Business Intelligence»

Data Science & modelli predittivi

Monica Mordonini (monica.mordonini@unipr.it)



DATA SCIENCE

I MODELLI PREDITTIVI



IL NOCCIOLO DELLA QUESTIONE: *TROVARE LE CORRELAZIONI FRA I DATI*

- ❑ I cinque passi fondamentali per la scienza dei dati sono i seguenti:
 1. Porre una domanda interessante.
 2. Ottenere i dati.
 3. Esplorare i dati.
 - 4. *Creare un modello per i dati.***
 5. Comunicare e presentare i risultati.

Il nocciolo della questione: *trovare le correlazioni fra i dati*

- ❑ Per affrontare in modo scientifico un'attività di analisi dei dati si deve avere ben presente la differenza fra avere dei dati e avere conoscenze utili tratte dai dati.
- ❑ Avere i dati è solo un passo per utilizzare con successo la scienza dei dati.
- ❑ La capacità di ottenere, ripulire e tracciare i dati aiuta a raccontare la storia che i dati intendono offrire, ma non ne può rivelare il senso.

Il nocciolo della questione: *trovare le correlazioni fra i dati*

- ❑ Per fare questo si devono trovare le relazioni esistenti fra delle caratteristiche quantitative
- ❑ I **coefficienti di correlazione** sono una misura quantitativa che descrive l'intensità dell'associazione/relazione fra due variabili (compresa tra -1 e 1).
- ❑ *La correlazione fra due insiemi di dati ci dice quanto si “muovono insieme”.*
 - Alterando uno possiamo prevedere come si comporterà l'altro ...
- ❑ Gli algoritmi di Machine Learning cercano e sfruttano questa relazioni tra i dati per offrire previsioni accurate

Il nocciolo della questione: *trovare le correlazioni fra i dati*

In generale, una correlazione tenterà di misurare una relazione lineare fra le variabili.

- una correlazione positiva significa che quando una variabile aumenta, anche l'altra tende ad aumentare;
- una correlazione negativa significa che quando una variabile aumenta, l'altra tende a diminuire
- minima correlazione è a 0

Se non viene individuata una correlazione in questo modo, ciò non significa che non esiste alcuna relazione fra le variabili, ma solo che non esiste una linea “best fit” per le linee.

➤ Potrebbe esistere una relazione non-lineare fra le due variabili

Inoltre queste correlazioni hanno un senso?

Il nocciolo della questione: *trovare le correlazioni fra i dati*

Esempio: Si può trovare una relazione fra il numero di amici su Facebook e la felicità? Cioè:

- ❑ esiste un'associazione positiva fra il numero di amici online e la felicità
- ❑ esiste un'associazione negativa fra di essi
- ❑ non esiste alcuna associazione fra le variabili (cambiando una, l'altra non cambia granché).
- ❑ Osserviamo la matrice di correlazione
 - Ottenuta da un dataset reale di pubblico dominio
- ❑ Si vede che esiste una debole correlazione negativa fra queste due variabili

	friends	happiness
friends	1.000000	-0.216199
happiness	-0.216199	1.000000

Ciò non significa necessariamente il livello di felicità decresca aumentando il numero di amici su Facebook. *Questa causalità deve essere ulteriormente indagata.*

È importante considerare che la causalità non è implicita nella correlazione

Il nocciolo della questione: *trovare le correlazioni fra i dati*

Oss.: Quando i grafici e le statistiche mentono, in realtà mentono le persone.

Uno dei modi più facili per ingannare consiste nel **confondere la correlazione e la causalità**

- La *correlazione* è una metrica quantitativa compresa fra -1 e 1 che misura come si spostano due variabili l'una rispetto all'altra e stabilisce il grado con il quale le variabili cambiano insieme
- La *causalità* è l'idea che una variabile influenzi un'altra e determini effettivamente il valore di un'altra

ES.: si osservi da un dataset di campioni sperimentali due variabili:

- il numero medio di ore passate quotidianamente davanti alla TV
- le prestazioni lavorative su una scala da 0 a 100.

Come ci si potrebbe aspettare questi due fattori presentano una correlazione negativa: -0.824 cioè aumentando il numero di ore quotidiane di TV, le prestazioni lavorative medie calano.

Il nocciolo della questione: *trovare le correlazioni fra i dati*

Più si guarda televisione, peggio si lavora: un rapporto di correlazione o causalità?

- ❑ Molto spesso capita che due variabili siano, sì, correlate, ma senza alcuna relazione di causalità fra di esse.
- ❑ Potrebbe entrare in gioco un fattore di confusione.

Questo significa che esiste “in agguato” una terza variabile non individuata e che funge da collegamento fra le due variabili.

- ❑ Nell'esempio della TV a prima vista i dati sembrano suggerire proprio che le ore trascorse davanti alla TV causino una riduzione della qualità delle prestazioni lavorative.
- ❑ Ma è possibile e più plausibile che i dati suggeriscano l'esistenza di un terzo fattore, magari le ore di sonno, in grado di rispondere a questa domanda.
 - Probabilmente, guardando più TV la sera decresce la quantità di ore dedicate al sonno, il che a sua volta limita le prestazioni lavorative. In questo caso il fattore di confusione è rappresentato dal numero di ore di sonno per notte.

Il nocciolo della questione: *trovare le correlazioni fra i dati*

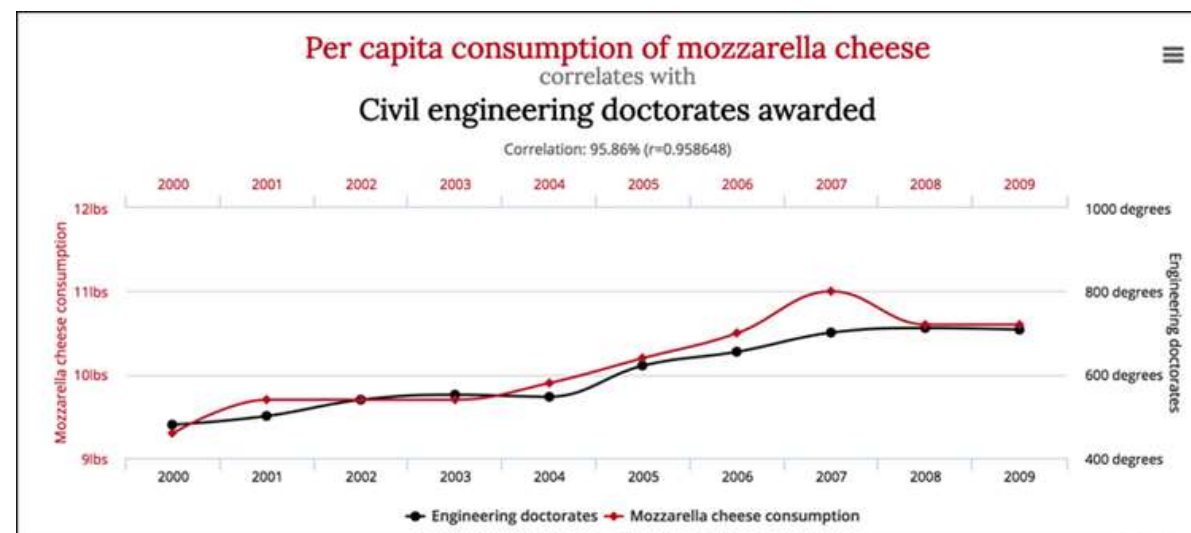
Correlazione o causalità?

In alcuni casi le variabili potrebbero non avere proprio nulla a che fare l'una con l'altra!

- Il tutto potrebbe essere semplicemente frutto di coincidenza.
- Vi sono molte variabili che sono correlate ma senza alcuna relazione di causalità.

Esempio: il consumo di mozzarelle determini il numero di lauree in ingegneria civile a livello mondiale

- I due andamenti sono molto simili, eppure non esiste un legame causale tra essi.



Il nocciolo della questione: *trovare le correlazioni fra i dati*

Attenzione al Paradosso di Simpson

- ✓ Fu descritto da G. U. Yule, in "Notes on the theory of association of attributes in Statistics", (Biometrika , 1903) e da E. H. Simpson, in "The interpretation of interaction in contingency tables" (Journal of the Royal Statistical Society, 1951)
- ✓ È alla base di frequenti errori nelle analisi statistiche nell'ambito delle scienze sociali e mediche, ma non solo.
- ***Il paradosso stabilisce che una correlazione fra due variabili può essere completamente invertita considerando fattori differenti.***
- Questo significa che anche se un grafico potrebbe mostrare una correlazione positiva, queste variabili possono diventare anti-correlate prendendo in considerazione un altro fattore (probabilmente un fattore di confusione).

Il nocciolo della questione: *trovare le correlazioni fra i dati*

Attenzione al Paradosso di Simpson

- ✓ Si ipotizzi una situazione nella quale, a parità di età, la percentuale di disoccupati tra i diplomati o i laureati sia la metà rispetto alla popolazione di chi non ha conseguito il diploma.
- ✓ Si consideri anche il fatto che, per motivi storici, tra le generazioni più anziane i diplomati siano in numero molto minore e che, per motivi legati al mercato del lavoro, tra i giovani il tasso di disoccupazione sia più elevato che tra gli anziani.
- ✓ in entrambi i casi la disoccupazione è circa doppia tra i non diplomati, rispetto ai diplomati

Lavoratori	senza diploma	con diploma	Totale
Giovani	20	80	100
Anziani	120	30	150
Totale	140	110	250

Tasso di disoccupazione	senza diploma	con diploma
Giovani	30%	15%
Anziani	5%	3,33%

https://it.wikipedia.org/wiki/Paradosso_di_Simpson

Il nocciolo della questione: *trovare le correlazioni fra i dati*

Attenzione al Paradosso di Simpson

- ✓ Si può calcolare il numero di disoccupati
- Questi valori assoluti permettono ora di calcolare il tasso di disoccupazione per i non diplomati e per i diplomati senza tenere conto dell'età
- Si scopre così che tra i diplomati il tasso di disoccupazione invece che essere la metà è maggiore di un quarto che tra i non diplomati, proprio il contrario di quello che si era ipotizzato.
- Questo paradosso è dovuto al fatto che il tasso di disoccupazione è nettamente maggiore nel gruppo che ha una maggiore percentuale di diplomati;

Disoccupati	senza diploma	con diploma	Totale
Giovani	6	12	18
Anziani	6	1	7
Totale	12	13	25

Percentuale di disoccupati	
senza diploma	$12/140 = 8,6\%$
con diploma	$13/110 = 11,8\%$

- trascurare l'esistenza di due relazioni fondamentali
 1. disoccupazione e età,
 2. tra età e titolo di studio
- fa giungere a conclusioni errate

Il nocciolo della questione: *trovare le correlazioni fra i dati*

Allora che significa la correlazione?

- Il modo migliore e più corretto per ottenere la causalità passa, normalmente, attraverso degli esperimenti casuali.
- Occorre suddividere la popolazione in gruppi campionati casualmente e svolgere una verifica delle ipotesi per concludere, con un certo grado di sicurezza, che esiste una vera causalità fra le variabili.
- ✓ Attenzione alla raccolta dati
- ✓ Attenzione nel comprendere le applicazioni e come usare bene i test statistici e la programmazione
- ✓ Pazienza nel provare e riprovare ...

MACHINE LEARNING



Che cosa si intende con machine learning?

- ❑ Dare ai computer la capacità delle macchine di apprendere dai dati, ***senza ricevere regole esplicite*** da un programmatore (essere umano)
- ❑ Il machine learning si occupa della capacità di trarre dai dati determinati pattern (segnali), anche se i dati contengono errori (rumore).
- ❑ Negli algoritmi classici è l'uomo a specificare il modo in cui individuare la soluzione migliore in un sistema complesso e poi l'algoritmo va alla ricerca di queste soluzioni, spesso lavorando in modo più veloce ed efficiente di un essere umano.
- ❑ Tuttavia, qui il collo di bottiglia consiste nel fatto che è l'essere umano a dover specificare qual è la *soluzione migliore*.
- In machine learning, al modello non viene detto qual è la soluzione migliore; piuttosto riceve vari esempi del problema e gli viene chiesto di *decidere qual è la soluzione migliore*.

Che cosa si intende con machine learning?

ML (algoritmi predittivi) vs algoritmi classici

➤ Riconoscimento di un volto:

❑ Classico

- Nell'algoritmo viene inserito codice che *definisce* un volto come una forma tondeggiante, con due occhi, capelli, naso e così via.
- L'algoritmo ricercherà nella fotografia queste caratteristiche "cablate" e dirà se è stato in grado o meno di trovarle

❑ Predittivo

- Non viene mai detto che cos'è un volto: vengono solo forniti degli esempi (training set), alcuni con volti, altri senza.
- compito del modello di machine learning trovare la differenza. Una volta individuata la differenza, usa queste informazioni per accettare una nuova immagine e *predire* se contiene o meno un volto.

Che cosa si intende con machine learning?

Il machine learning non è perfetto

- ❑ Quasi nessun modello di machine learning tollera l'impiego di dati “sporchi”, con valori mancanti o valori categorici
 - I dati usati sono già stati pre-elaborati e ripuliti
 - ❑ Ogni riga di un dataset ripulito rappresenta una singola osservazione dell'ambiente che stiamo tentando di modellare.
 - Se l'obiettivo è quello di trovare le relazioni esistenti fra le variabili, allora si deve partire dal ***presupposto che fra queste variabili esista in effetti una relazione***
- ✓ **Gli algoritmi non sono in grado di comunicare che tale relazione, in realtà, non esiste**

Che cosa si intende con machine learning?

Il machine learning non è perfetto

- La macchina è molto abile, ma fatica a collocare le cose nel loro contesto
 - L'output è una serie di numeri e metriche che tentano di quantificare l'efficacia del modello.
 - **Compito dell'essere umano valutare queste metriche e comunicare i risultati**
- La maggior parte dei modelli di machine learning è sensibile alla rumorosità dei dati.
 - **Questo significa che i modelli si confondono quando si includono dati insensati.**
 - Es se si cercano relazioni fra dei dati economici mondiali e una delle colonne in input è l'adozione di cuccioli nella capitale, tali informazioni sono probabilmente irrilevanti ma confonderanno il modello

Che cosa si intende con machine learning?

Attenzione

- ❑ Il machine learning è uno degli strumenti a disposizione di un esperto di scienza dei dati.
 - Ma non è l'unico
- ❑ Collocato allo stesso livello dei test statistici (chi quadrato o test t) o degli utilizzi pratici del calcolo delle probabilità e della statistica per stimare i parametri della popolazione.
- ❑ Compito dell'esperto dei dati di riconoscere quando il machine learning è applicabile
 - *e, soprattutto, quando non lo è.*

Tipi di ML (algoritmi predittivi)

- la scelta dell'algoritmo ha impatto sulle modalità di preparazione dei dati
 - alcuni algoritmi richiedono che i dati siano preparati in un determinato formato (es. numerico o normalizzato)
- la scelta dell'algoritmo comporta una susseguente attività di sistemazione dei dati
 - Che però non riguarda la parte più importante della fase di preparazione, ovvero la costruzione e la scelta delle variabili
- Si possono classificare per
 - Scopo
 - Caratteristiche delle modalità di apprendimento e tecniche utilizzate:
 - tipi di dati/strutture organiche utilizzati (albero/grafico/rete neurale);
 - campo della matematica dal quale attinge maggiormente (statistica/calcolo delle probabilità);
 - livello di calcolo richiesto per l'addestramento (apprendimento profondo).

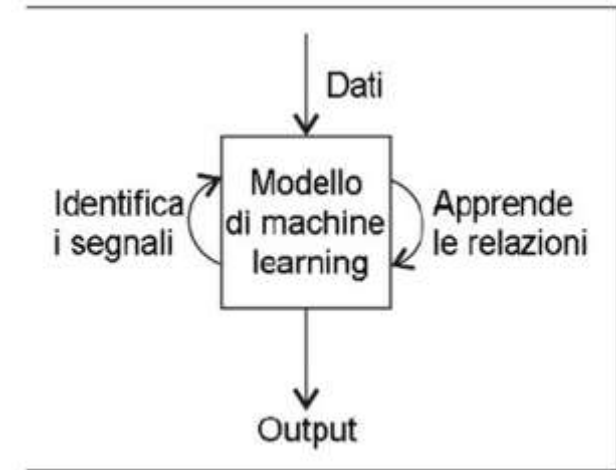


Figura 10.2 Funzionamento dei modelli di machine learning.

Classificazione degli algoritmi per scopo

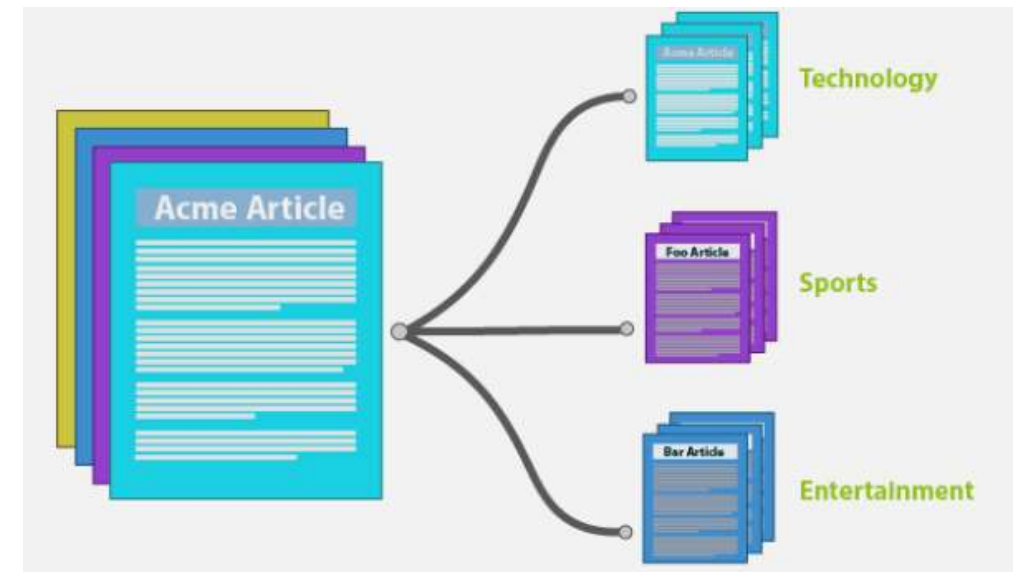
Permette di identificare quali algoritmi siano adatti ad un particolare problema predittivo

- ❑ Classificazione
- ❑ Regressione
- ❑ Clustering
- ❑ Association rules
- ❑ Serie temporali

Classificazione per scopo

□ Classificazione

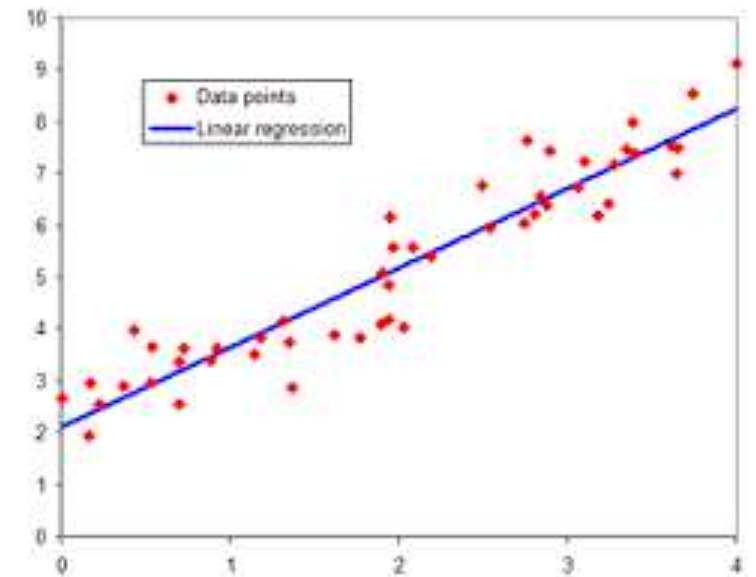
- Individuazione dell'appartenenza di un elemento ad una classe.
- L'output della classificazione è categorico e quindi può assumere un numero finito di possibili valori
- Agli algoritmi di classificazione appartengono i classificatori lineari (logistic regression, Naive Bayes, Perceptron, Support Vector Machine), gli alberi decisionali e le reti neurali



Classificazione per scopo

□ Regressione

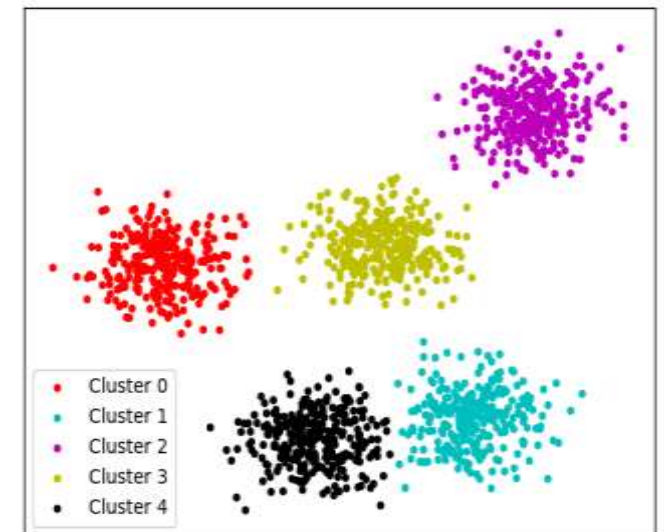
- l'output del modello è un valore numerico, che si vuol approssimare tramite una funzione dei dati di input.
- La variabile di output è continua e può assumere un numero infinito di valori
 - Es.: previsione del livello delle vendite in un periodo temporale futuro,
- Algoritmi che appartengono a questa categoria sono: la regressione lineare, la ridge regression, la lasso regression.



Classificazione per scopo

□ Clustering

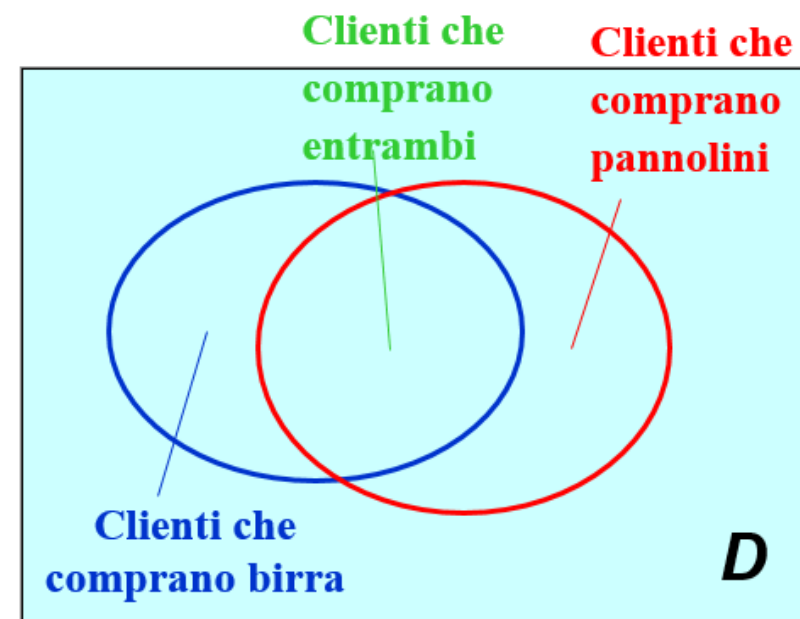
- il raggruppamento degli elementi di un dataset in gruppi (i cluster) utilizzando soltanto le informazioni contenute nei dati di input.
- Al contrario rispetto a quanto avviene nella classificazione, l'output del clustering (ovvero la categorizzazione) non è noto a priori.
- I raggruppamenti sono realizzati in base alla similarità dei punti.
 - Utile per esempio per la segmentazione della clientela in gruppi omogenei ma anche per l'identificazione di anomalie: frodi assicurative, uso fraudolento di carte di credito rubate, ecc.
- Gli algoritmi di clustering più utilizzati sono: k-means, k-medoids, DBSCAN, Hierarchical Clustering.



Classificazione per scopo

□ Association rules

- Questi algoritmi sono utilizzati per estrarre regole che mettono in relazione gli elementi di un dataset.
- Sono utilizzati per recuperare gli insiemi di elementi che ricorrono frequentemente in un dataset
 - per esempio i prodotti che più spesso sono acquistati assieme
- Gli algoritmi : FP Growth FP=Frequent Pattern.

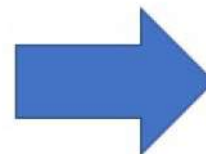


Classificazione per scopo

■ Serie temporali

- Algoritmi specifici per le serie temporali, che consentono di effettuare previsioni sullo andamento futuro di tali serie.
- Es, previsione andamento futuro delle vendite, utilizzando come input la serie storica delle vendite stesse.
- I modelli ARIMA (AutoRegressive Integrated Moving Average) e la decomposizione delle serie in trend, stagionalità e rumore sono due tra le tecniche presenti in questo campo.
- Alle serie temporali, se opportunamente adattate, possono essere applicati algoritmi di regressione o anche di classificazione

Tempo	Valore
t0	Val_t0
t1	Val_t1
t2	Val_t2
t3	Val_t3
t4	Val_t4
t5	Val_t5
t6	Val_t6
t7	Val_t7
t8	Val_t8
t9	Val_t9
t10	Val_t10
t11	??



V_Output	V1	V2	V3	V4	V5
Val_t0					
Val_t1	Val_t0				
Val_t2	Val_t1	Val_t0			
Val_t3	Val_t2	Val_t1	Val_t0		
Val_t4	Val_t3	Val_t2	Val_t1	Val_t0	
Val_t5	Val_t4	Val_t3	Val_t2	Val_t1	Val_t0
Val_t6	Val_t5	Val_t4	Val_t3	Val_t2	Val_t1
Val_t7	Val_t6	Val_t5	Val_t4	Val_t3	Val_t2
Val_t8	Val_t7	Val_t6	Val_t5	Val_t4	Val_t3
Val_t9	Val_t8	Val_t7	Val_t6	Val_t5	Val_t4
Val_t10	Val_t9	Val_t8	Val_t7	Val_t6	Val_t5
??	Val_t10	Val_t9	Val_t8	Val_t7	Val_t6

Figura 13.1: Serie storica trasformata in un dataset per le reti neurali.

- ✓ vantaggi rispetto a tecniche basate sull'auto regressione (cioè l'input = serie stessa) stanno nella possibilità di aggiungere variabili di input estranee alla serie stessa, ma che potrebbero includere informazioni in grado di migliorare le capacità predittive del modello.

Classificazione per modalità di apprendimento

- *Il procedimento con cui l'algoritmo impara dai dati di input è chiamato training.*
- ❑ Supervisionato
- ❑ Non supervisionato
- ❑ Semi-supervisionato

Classificazione per modalità di apprendimento

▣ Supervisionato

- Modelli di analisi predittiva => capacità di prevedere il futuro sulla base del passato
- Il machine learning con supervisione richiede l'impiego di un determinato tipo di dati: i *dati etichettati*.
 - Questo significa che dobbiamo addestrare il nostro modello fornendogli esempi storici etichettati con la risposta corretta
 - Es volto-non volto

Classificazione per modalità di apprendimento

□ Supervisionato

L'apprendimento con supervisione funziona usando delle parti dei dati per prevedere un'altra parte.

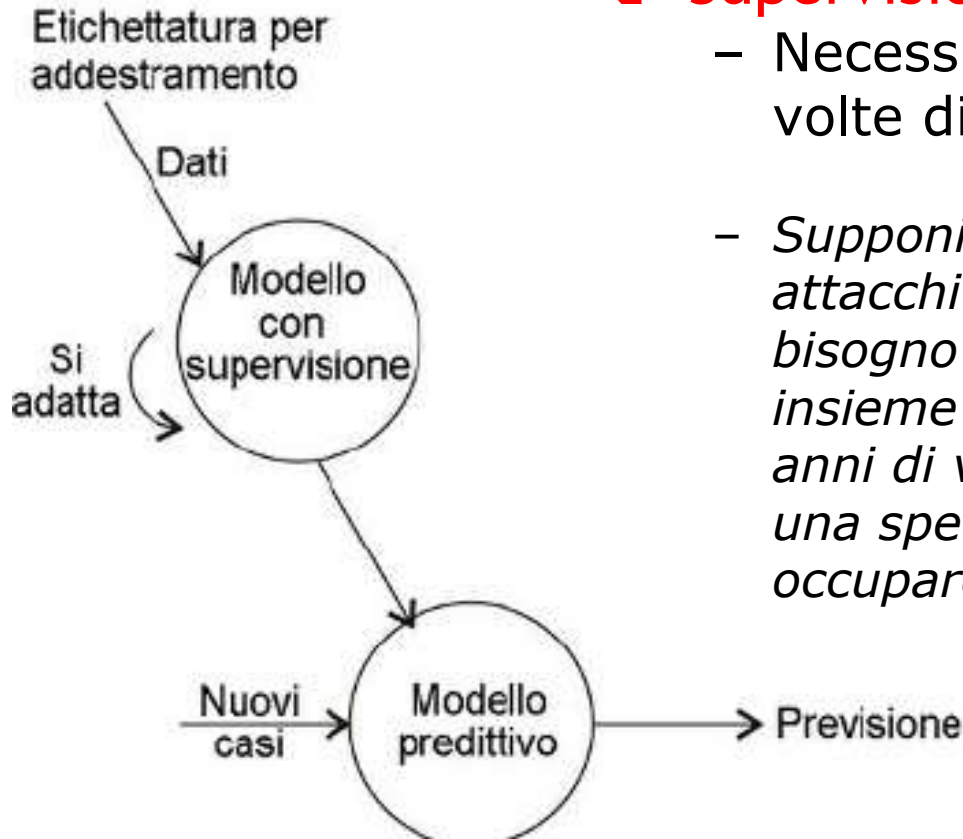
Si devono separare i dati in due parti:

1. I **predittori**, che sono le colonne che verranno usate per effettuare la previsione.
 - Sono chiamate anche caratteristiche, input, variabili o variabili indipendenti.
 2. La **risposta**, che è la colonna che vogliamo prevedere.
 - Questo è chiamato anche risultato, etichetta, target e variabile dipendente.
- L'apprendimento con supervisione tenta di trovare una relazione fra i predittori e la risposta per effettuare una previsione.
- L'idea è che in futuro si presentino dei dati osservati e potremo contare solo sui predittori

Classificazione per modalità di apprendimento

■ Supervisionato

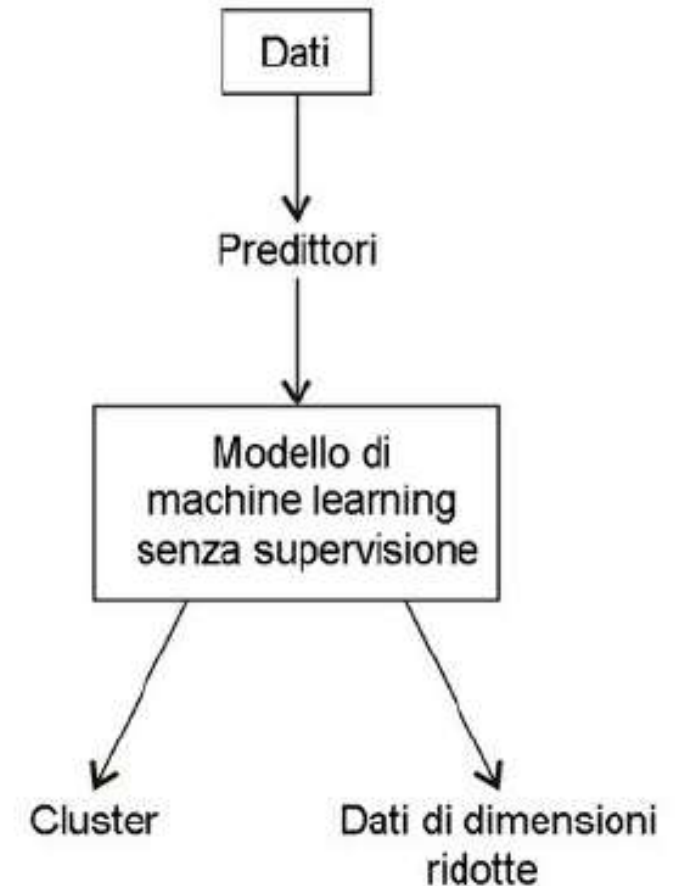
- Necessità di dati etichettati a volte difficili da reperire
- *Supponiamo di voler prevedere gli attacchi cardiaci: potremmo aver bisogno di migliaia di pazienti, insieme alle loro cartelle cliniche di anni di visite mediche e ottenerle è una specie di incubo per chi si deve occupare della raccolta dei dati.*



Classificazione per modalità di apprendimento

□ Non supervisionato

- non tentano di trovare una relazione fra i predittori e una specifica risposta e pertanto non vengono usati per effettuare previsioni di alcun tipo.
- Al contrario, vengono utilizzati per trovare nei dati forme di organizzazione e di rappresentazione precedentemente sconosciute
- Possono ridurre le dimensioni dei dati condensando insieme più variabili (riduzione dimensionale).
 - Un tipico esempio di questo tipo è la compressione dei file. La compressione sfrutta dei pattern presenti nei dati per rappresentare quegli stessi dati in un formato più compatto.
- Trovare dei gruppi di osservazioni che si comportano allo stesso modo e raggrupparli (*clustering*).



Classificazione per modalità di apprendimento

❑ Non supervisionato

- Vantaggi: non richiede dati etichettati
- Difetto: si perde ogni potere predittivo, perché la variabile di risposta contiene le informazioni per effettuare le previsioni e senza di essa il nostro modello non sarà in grado di eseguire alcun tipo di previsione.
- Difficile capire se si comporta correttamente

➤ *I modelli senza supervisione sono semplici suggerimenti per differenze e analogie, che richiederanno sempre un'interpretazione umana.*

Classificazione per modalità di apprendimento

□ Semi-supervisionato

- lavorano su un insieme di dati che solo in parte possiede già una classificazione.
- Solitamente, la parte di dati già classificata è una piccola percentuale del dataset, ma è sufficiente a rendere l'algoritmo più preciso.
- utile quando vi è grande disponibilità di dati non classificati ed il costo per classificarli manualmente è molto alto.
 - Es: i modelli generativi e gli algoritmi Self-Training.

PROBLEMATICHE COMUNI AGLI ALGORITMI DI ML

Problematiche comuni agli algoritmi di ML

➤ *Hyperparameter tuning*

- ❑ Parametri necessari all'algoritmo per funzionare,
- ❑ non ricavabili tramite i dati,
- ❑ ma impostati inizialmente dall'analista.
- ❑ Spesso, da essi dipende la bontà del modello predittivo.
- ❑ L'impostazione ottimale di questi iper-parametri non è semplice

Es.

- ❑ *il numero di layer e il numero di neuroni di input nelle reti neurali,*

Problematiche comuni

Tecniche per individuare i valori che massimizzano la capacità predittiva di un algoritmo

Grid Search (o parameter sweep)

- consiste nell'individuazione di un numero finito (e non molto grande) di possibili valori che ciascun parametro potrebbe assumere;
- poi si effettua il training dell'algoritmo utilizzando tutte le possibili combinazioni dei valori al fine di individuare quelli che massimizzano le metriche di valutazione.

il training di numerosi modelli è molto oneroso in termini di risorse di calcolo, tuttavia spesso si può parallelizzare , abbattendo così i tempi di calcolo.

Problematiche comuni

Tecniche per individuare i valori che massimizzano la capacità predittiva di un algoritmo

Random Search

- ❑ Con questo metodo i parametri da utilizzare sono estratti in modo casuale, creando uno spazio di ricerca più piccolo rispetto a quello del Grid Search, ma comunque sufficiente a individuare combinazioni ottimali di parametri.
- ❑ Spesso si utilizzano tecniche più sofisticate per l'estrazione di questo spazio quali la ricerca tramite algoritmi genetici

Problematiche comuni

➤ **Overfitting**

- ❑ accade quando un modello è troppo complesso ed eccessivamente adattato ai dati di training.
- ❑ La crescita della complessità del modello diminuisce l'errore predittivo sui dati del training set;
- ❑ all'aumentare della complessità del modello diminuisce anche l'errore sul test set,
- ***ma solo fino ad un certo punto:***
 - ❑ l'errore infatti ritorna a crescere superata una certa soglia di complessità.
 - ❑ La crescita di questo errore segnala che si sta entrando nel territorio dell'overfitting.

Problematiche comuni

➤ *Overfitting*

➤ *Bias*

rappresenta quanto, in media, le previsioni di un modello sono lontane dalla realtà,

➤ *Varianza*

indica di quanto le stime variano attorno alla media.

➤ *Underfitting*: in questo caso il modello è troppo semplice per poter avere in media una buona performance predittiva.

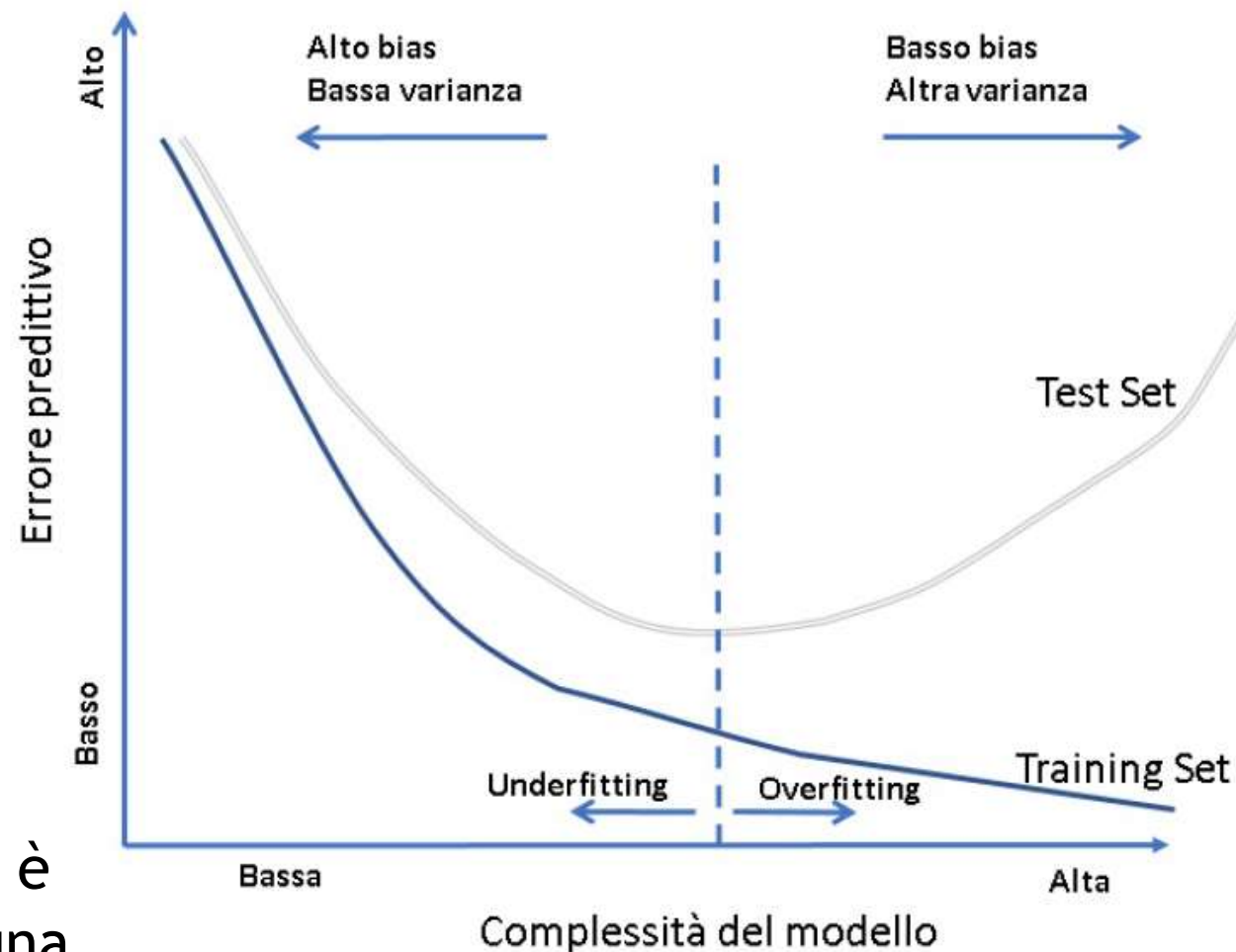


Figura 13.2: Grafico Complessità/Errore

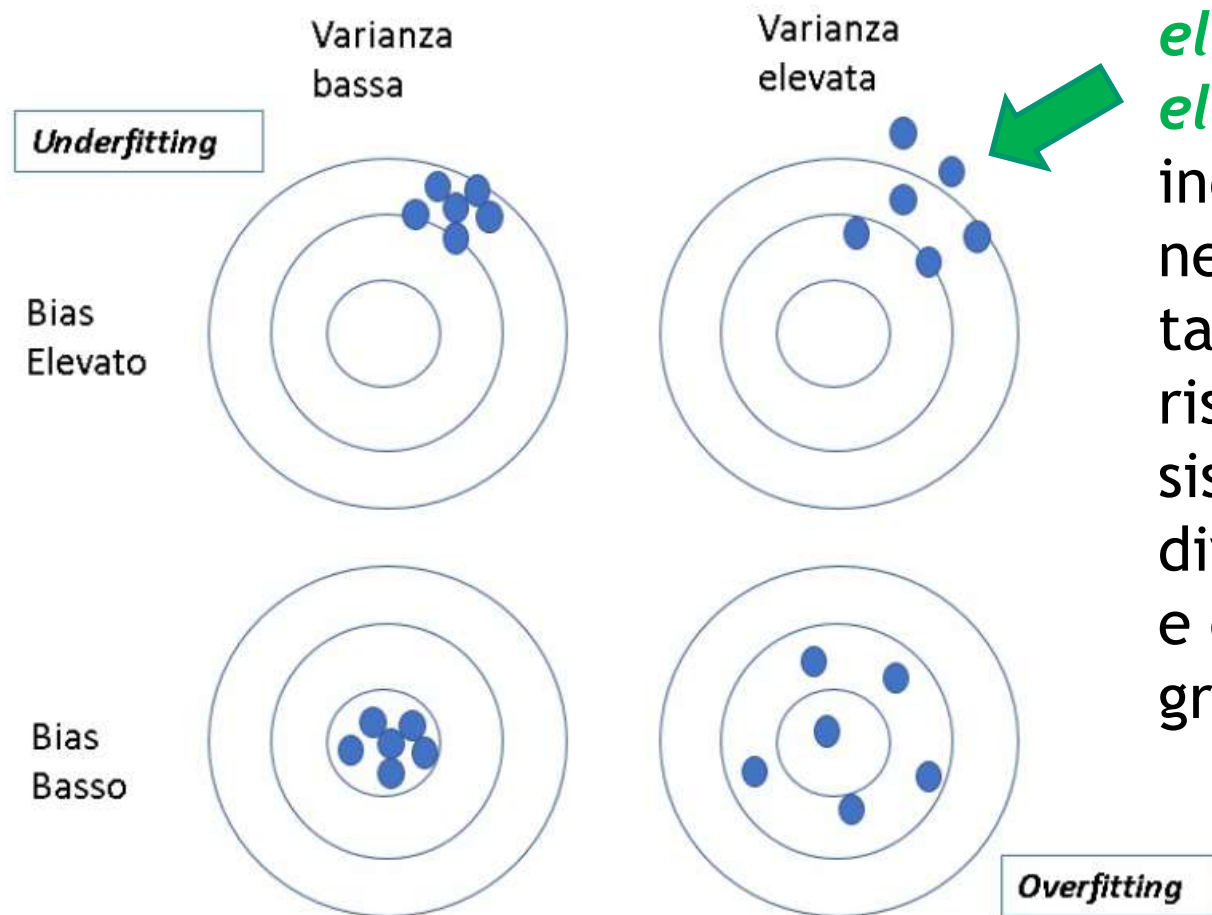
Problematiche comuni

➤ *Overfitting*

❑ *Bias*

❑ *Varianza*

*situazione
ottimale:*
basso bias
bassa varianza

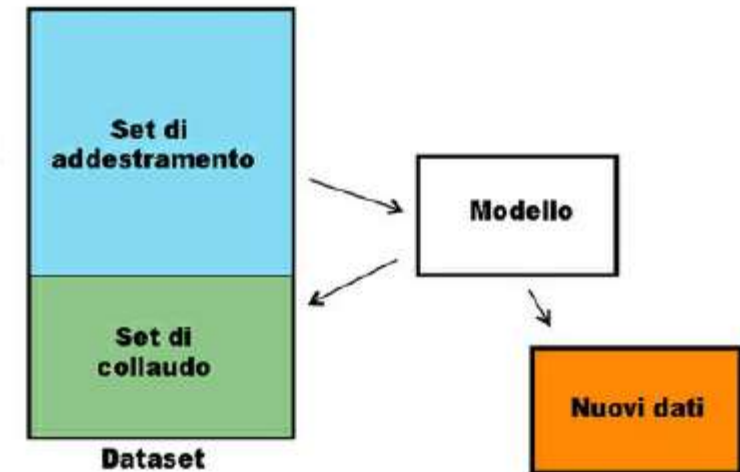


*elevato bias
elevata varianza*
indicano errori
nella modellazione
tali da produrre
risultati
sistematicamente
diversi dalla realtà
e con un elevato
grado di variabilità

Figura 13.3: Varianza / Bias.

Problematiche comuni

- **Utilizzo di un approccio addestramento/test**
 1. Suddividere il dataset in due parti
 2. Adattare il nostro modello con il set di addestramento e poi collaudarlo sull'insieme di test.
 3. Una volta che il nostro modello funziona abbastanza bene (sulla base delle nostre metriche), rivolgiamo l'attenzione del nostro modello sull'intero dataset.
 4. Il nostro modello attende nuovi dati che precedentemente nessuno aveva mai visto.
- L'obiettivo qui è quello di minimizzare gli errori extra-campione del nostro modello, ovvero gli errori che il nostro modello commette su dati che non ha mai visto prima (**Capacità di generalizzazione**)



GLI ALGORITMI DI ML

Algoritmi di classificazione



- La classificazione individua l'appartenenza ad una classe.
- Per esempio un modello potrebbe predire che il potenziale cliente 'X' risponderà sì ad un'offerta.
- Con la classificazione l'output predetto (la classe) è categorico ossia può assumere solo pochi possibili valori come: Sì, No, Alto, Medio, Basso...

Algoritmi di classificazione

✓ *Naive Bayes*

- L'algoritmo Naive Bayes si basa sulla determinazione della probabilità di un elemento di appartenere a una certa classe
- La tecnica si basa sul teorema di Bayes che definisce la probabilità condizionata (o *a posteriori*) di un evento rispetto ad un altro

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

$P(A|B)$ è la probabilità condizionata di A rispetto a B

$P(B|A)$ è la probabilità condizionata di B rispetto a A

$P(A)$ è la probabilità “a priori” di A, che non tiene conto di nessuna informazione circa B

$P(B)$ è la probabilità “a priori” di B che non tiene conto di nessuna informazione circa A

Algoritmi di classificazione

✓ *Naive Bayes*

- ❑ Le probabilità “*a priori*” possono *essere stimate* attraverso la frequenza campionaria, per quanto riguarda gli attributi discreti, mentre per gli attributi continui si assume che essi siano distribuiti secondo la distribuzione normale e si utilizza la funzione di densità per il calcolo delle probabilità.
- L'algoritmo *naïve bayesian classifier* assume che l'effetto di un attributo su una data classe è indipendente dai valori degli altri attributi.
- Questa assunzione, chiamata indipendenza condizionale delle classi, ha lo scopo di semplificare i calcoli e proprio per questo l'algoritmo prende il nome di “naïve”.
- Quando tale assunzione è vera nella realtà, l'accuratezza dell'algoritmo è paragonabile a quella dei decision tree e delle reti neurali.

Algoritmi di classificazione

✓ *Naive Bayes*

- L'algoritmo determina la classe di appartenenza in base alle probabilità condizionali per tutte le classi in base agli attributi dei vari elementi.
 - La classificazione corretta si ha quando la probabilità condizionale di una certa classe C rispetto agli attributi è massima.
- L'algoritmo possiede i seguenti punti di forza:
 1. Lavora bene in caso di "rumore" in una parte dati.
 2. Tende a non considerare gli attributi irrilevanti.
 3. Il training del modello è molto più semplice rispetto ad altri algoritmi.
- *Il rovescio della medaglia è rappresentato dall'assunzione dell'indipendenza degli attributi, che può non essere presente nella realtà*
 - *Questo limite è comunque superabile attraverso l'uso di tecniche di accorpamento di variabili di input*

Algoritmi di classificazione



- **Bayes con un esempio:** Problema di Monty Hall
- ❑ Si partecipa a un gioco a premi, in cui si può scegliere fra tre porte: dietro una di esse c'è un'automobile, dietro le altre, una capra.
- ❑ Si scelga una porta, la numero 1, e il conduttore del gioco a premi, che sa cosa si nasconde dietro ciascuna porta, ne apre un'altra, la 3, rivelando una capra.
- ❑ Quindi domanda: "Vorresti scegliere la numero 2?"
- ❑ E' conveniente cambiare la scelta originale? *Qual è la migliore strategia per il giocatore?*
- ❑ *Le possibilità di vittoria aumentano per il giocatore se cambia la propria scelta? La risposta è sì*

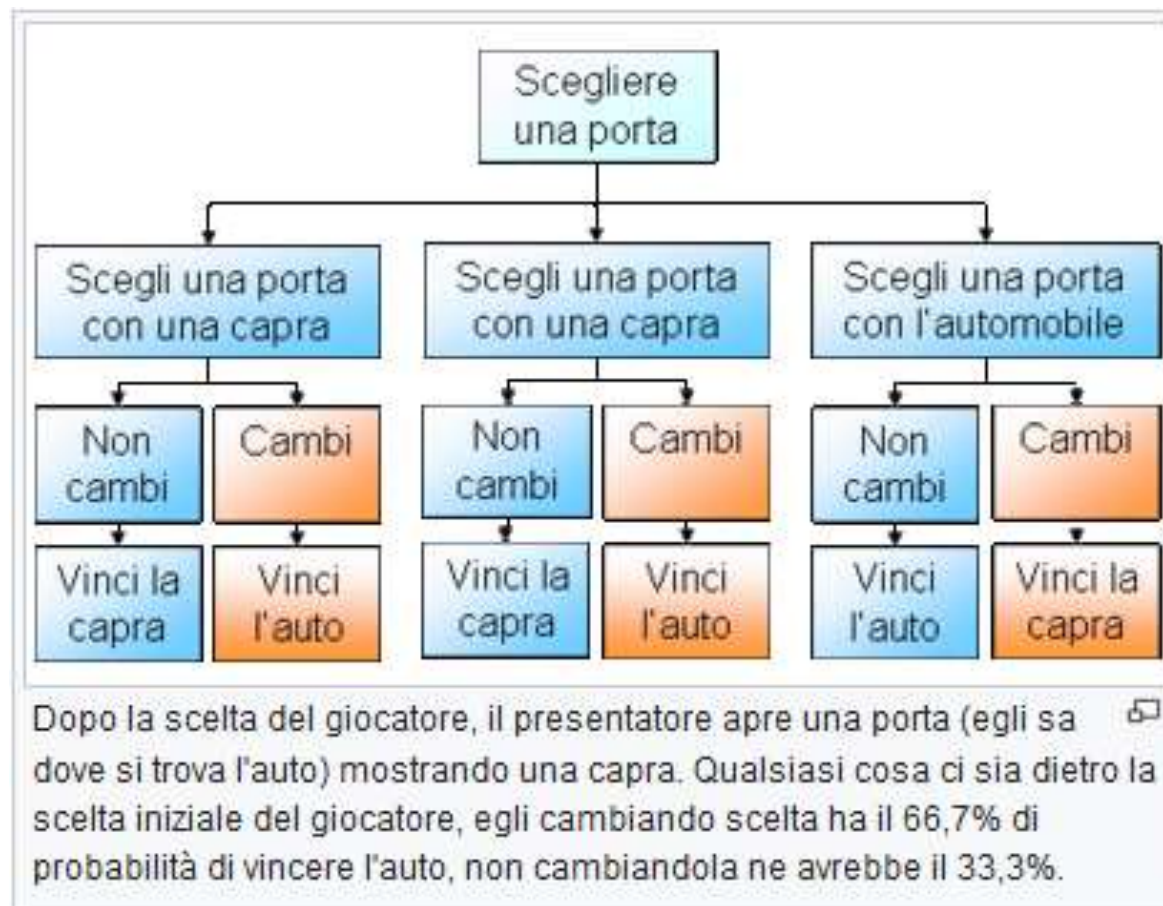
https://it.wikipedia.org/wiki/Problema_di_Monty_Hall

Algoritmi di classificazione

Nel momento in cui quest'informazione del presentatore arriva al concorrente, quest'ultimo ha una sola possibilità di approfittarne, cioè CAMBIARE la carta scelta:

in questo modo, a fronte del gioco, è come se gli fosse data la possibilità di scegliere 2 carte su 3 per tentare di vincere l'auto, da cui la probabilità dei due terzi (66 per cento).

Se non facesse il cambio, rimarrebbe invariabilmente "bloccato" sulla probabilità di un terzo (33 per cento) della sua giocata iniziale, avvenuta fatalmente PRIMA di avere l'informazione CERTA di dove si trovasse una delle due capre in gioco.



Attenzione è sempre una probabilità mai una certezza

Algoritmi di classificazione

✓ Alberi decisionali (*Decision Trees*)

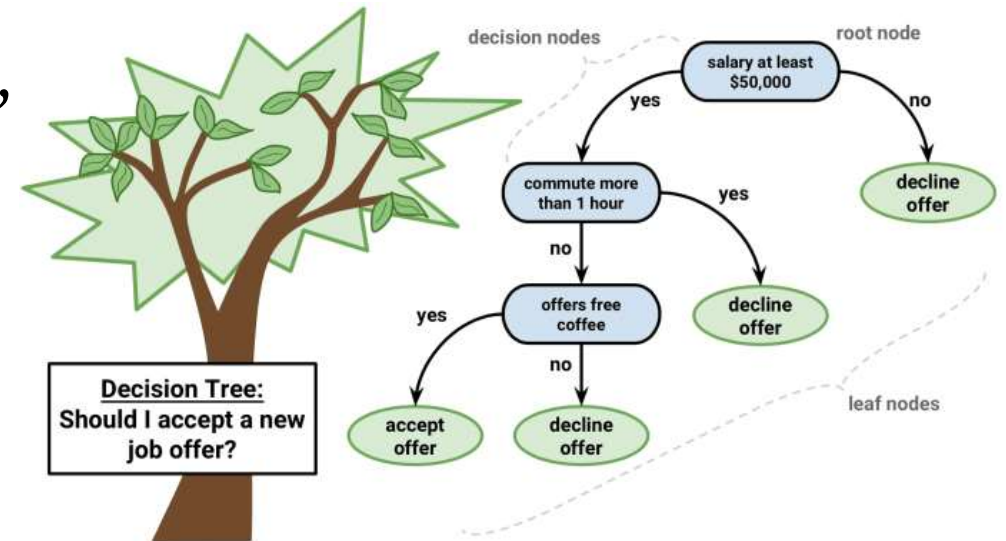
- Sono uno dei metodi di calcolo principalmente utilizzati nel data mining.
- in particolare per determinare a quale categoria appartiene un elemento, in base al valore dei suoi attributi noti.
- La tecnica su cui si basano è flessibile e permette di adattarli a numerose situazioni; inoltre il loro output è molto chiaro, dato che è rappresentato (anche visivamente) sotto forma di albero .
- Vi sono tre categorie di alberi decisionali:
 1. I ***classification tree***, utilizzati per determinare l'appartenenza degli elementi di un insieme a classi diverse.
 2. I ***regression trees***, utilizzati per previsioni relative a un numero reale (per esempio il valore di un indice azionario).
 3. I ***classification & regression trees***, che uniscono le due tipologie appena descritte

Algoritmi di classificazione



✓ Alberi decisionali (Decision Trees)

- Un albero decisionale è una struttura a grafo che include un nodo radice, da cui partono dei rami che arrivano a dei nodi figli. I nodi terminali sono detti «nodi foglia».
- 1. Ogni nodo interno denota un test su un attributo,
- 2. Ogni ramo indica il risultato di un test
- 3. Ogni nodo foglia contiene un'etichetta di classe.
- 4. Il nodo più in alto nell'albero è il nodo radice.



Algoritmi di classificazione

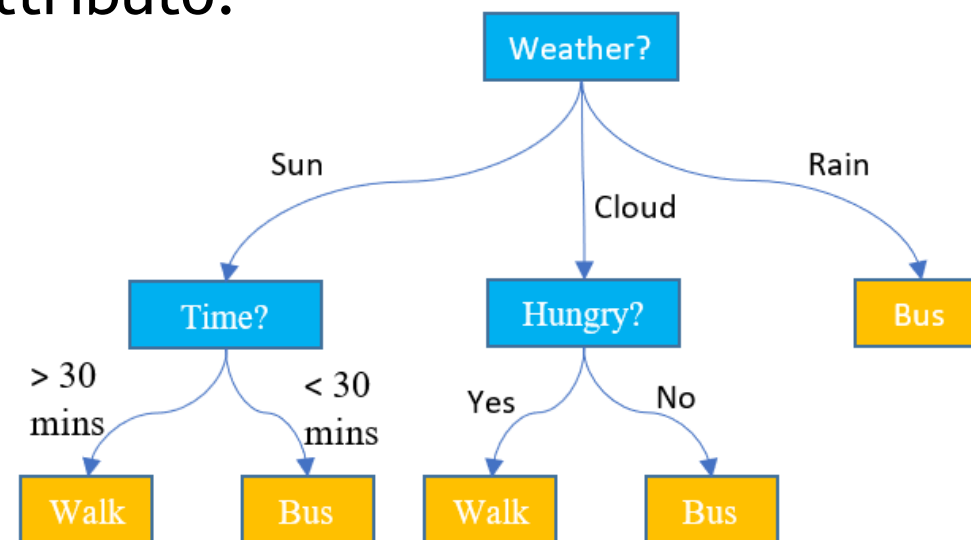
✓ Alberi decisionali (Decision Trees)

- Ogni nodo interno rappresenta un test su un attributo:

- Weather
- Time
- Hungry

- Ogni nodo foglia rappresenta una classe

- Walk
- Bus

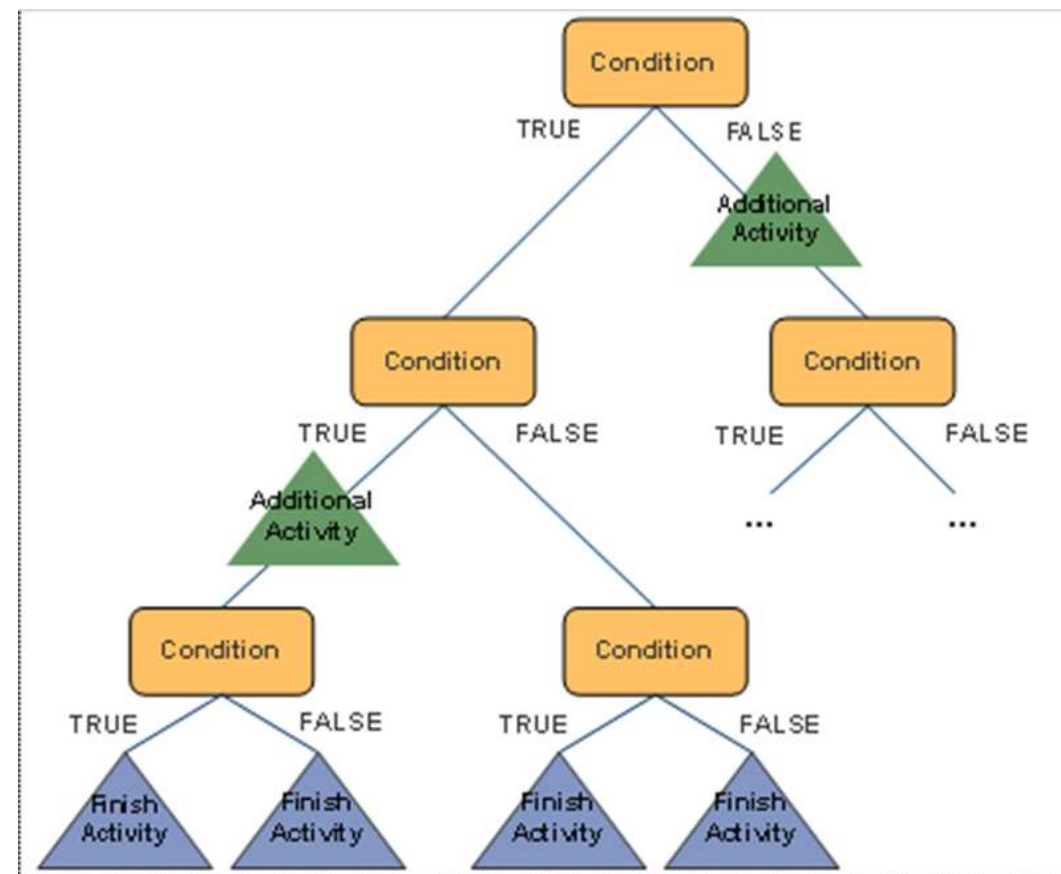


i nodi foglia rappresentano le classificazioni e le ramificazioni l'insieme delle proprietà che portano a quelle classificazioni. Di conseguenza ogni nodo interno risulta essere una macro-classe costituita dall'unione delle classi associate ai suoi nodi figli.

Algoritmi di classificazione

✓ Alberi decisionali (Decision Trees)

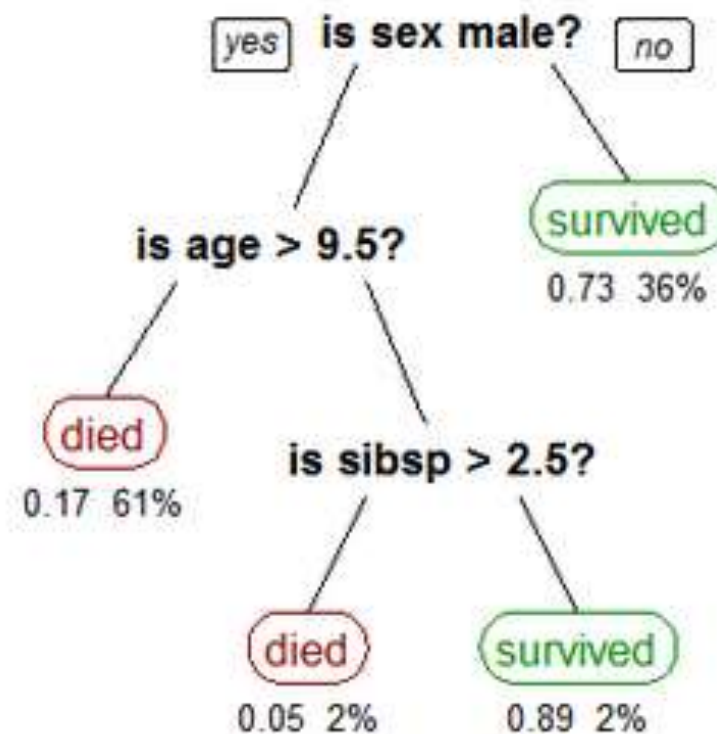
- ❑ I vantaggi di avere un albero decisionale sono i seguenti :
- ❑ Non richiede alcuna conoscenza di dominio.
- ❑ È facile da capire.
- ❑ Le fasi di apprendimento e classificazione di un albero decisionale sono semplici e veloci.



Algoritmi di classificazione

✓ Alberi decisionali (Decision Trees)

Un albero che mostra la sopravvivenza dei passeggeri sul Titanic ("sibsp" è il numero di coniugi o fratelli a bordo). Le figure sotto le foglie mostrano la probabilità di sopravvivenza e la percentuale di osservazioni nella foglia. Riassumendo: le tue possibilità di sopravvivenza erano buone se tu fossi (i) una femmina o (ii) un maschio più giovane di 9,5 anni con meno di 2,5 fratelli.



https://en.wikipedia.org/wiki/Decision_tree_learning

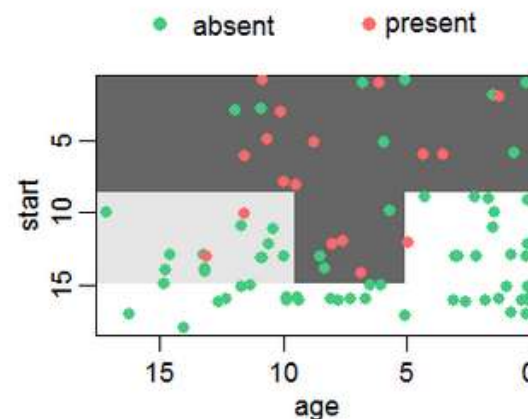
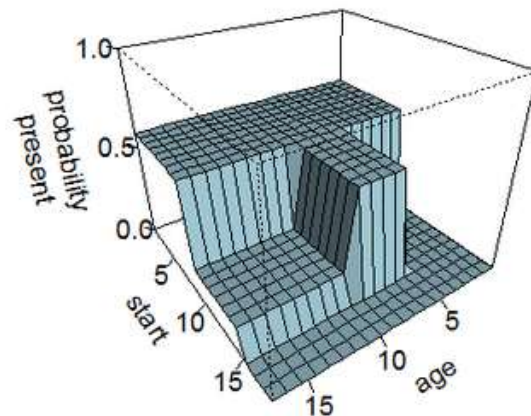
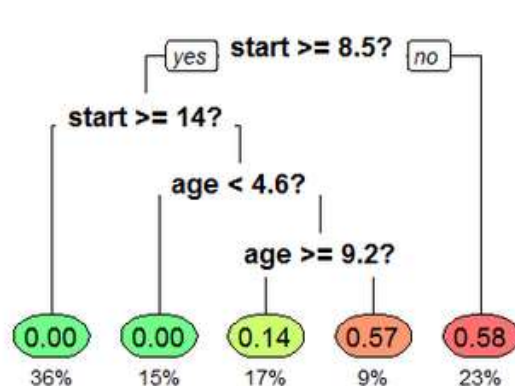
Algoritmi di classificazione

✓ Alberi decisionali (Decision Trees)

Un esempio di albero che stima la probabilità di cifosi dopo l'intervento chirurgico, data l'età del paziente e la vertebra in cui è stato avviato l'intervento chirurgico.

Lo stesso albero è mostrato in tre modi diversi. A sinistra: Le foglie colorate mostrano la probabilità di cifosi dopo l'intervento chirurgico e la percentuale di pazienti nella foglia. In centro: L'albero come trama prospettica. A destra Veduta aerea della trama centrale. La probabilità di cifosi dopo l'intervento chirurgico è più elevata nelle aree più scure.

(Nota: il trattamento della cifosi è notevolmente migliorato da quando è stata raccolta questa piccola serie di dati)



https://en.wikipedia.org/wiki/Decision_tree_learning

✓ Alberi decisionali (*Decision Trees*)

- l'algoritmo si fonda sul concetto di *entropia della teoria dell'informazione*.
- Poniamo di avere un insieme campione T di elementi e di voler trovare una regola di suddivisione degli elementi in un numero k di classi C_k in base agli attributi degli elementi.
- L'unica informazione che abbiamo è la classe di appartenenza degli elementi del campione.
- Lo scopo è di trovare una regola di classificazione per i nuovi elementi

Algoritmi di classificazione

✓ Alberi decisionali (*Decision Trees*)

- L'informazione di un certo sotto insieme I è definita dalla seguente equazione da:

$$Info(I) = - \sum_{i=1}^k \left(\frac{freq(C_i, I)}{count(I)} \times \log_2 \left(\frac{freq(C_i, I)}{count(I)} \right) \right)$$

dove

- $freq(C_i, I)$ è il numero di elementi della classe C_i presenti in I
- $count(I)$ è il numero totale di elementi di I

Algoritmi di classificazione

✓ Alberi decisionali (Decision Trees)

- Una volta che l'insieme T è stato partizionato in n sottoinsiemi secondo i valori di un attributo x possiamo calcolare l'informazione totale attraverso:

$$Info_x(T) = - \sum_{i=1}^n \left(\frac{count(T_i)}{count(T)} \times Info(T) \right)$$

- A questo punto possiamo calcolare il guadagno di informazione (*information gain*) che otteniamo dalla ripartizione in sottoinsiemi, attraverso la formula:

$$Guadagno(X) = Info(T) - Info_x(T)$$

✓ *Alberi decisionali (Decision Trees)*

- ❑ Ora, per ciascun attributo del nostro dataset, dobbiamo trovare quella ripartizione, basata sui valori dell'attributo, per la quale il guadagno è massimo.
- ❑ La ripartizione che massimizza il guadagno corrisponde al primo livello dell'albero, sotto l'insieme totale.
- ❑ A questo punto si ripete il processo per ciascuno dei sottoinsiemi che, al loro interno, presentano elementi appartenenti a classi diverse.
- ❑ Il processo si ferma quando i sottoinsiemi contengono elementi solo di una classe, oppure quando il continuare con la suddivisione non porta miglioramenti dell'accuratezza.

Algoritmi di classificazione

✓ Alberi decisionali (Decision Trees)

- ❑ Gli algoritmi degli alberi decisionali includono *tecniche di pruning* che hanno lo scopo di limitare la crescita dell'albero
- ❑ Il pruning evita che l'albero decisionale si adatti troppo ai dati di training, causando l'overfitting del modello.
 - Tale situazione è assolutamente da evitare, dato che fa venire meno la capacità del modello di generalizzare, abbassandone quindi le performance predittive.
- ❑ I metodi per il pruning sono divisi in due categorie:
 - ❑ 1. Pre-pruning
 - ❑ 2. Post pruning

Algoritmi di classificazione

✓ Alberi decisionali (*Decision Trees*)

- **Pre-pruning**, che ricomprensde le tecniche applicate in fase di costruzione dell'albero, per evitare che esso, in via preventiva, possa crescere oltre un certo livello.
 - Es, un metodo consiste nello specificare il numero minimo di elementi di un nodo, al di sotto del quale non è più possibile effettuare una suddivisione.
 - Tuttavia, non è semplice trovare una soglia che eviti l'overfitting e nel contempo garantisca un buon livello di precisione del modello.
- **Post-pruning**, che avviene dopo la costruzione dell'albero ove si sostituisce un sotto albero (sub-tree) con il nodo padre di tale sotto albero e valutando l'errore predittivo che si ha trattando tale nodo come foglia.
 - un metodo comunemente utilizzato è il *Cost Complexity Pruning*, tramite il quale si effettua la sostituzione se l'errore del nodo foglia è minore o uguale all'errore del sub-tree a cui è aggiunto un fattore di costo legato alla complessità del sotto albero.

Algoritmi di classificazione

✓ Alberi decisionali (*Decision Trees*)

- ❑ Principale vantaggio: la produzione di regole come parte dell'output dell'algoritmo, consentendo una facile interpretazione dei risultati.
- ❑ Inoltre possono gestire feature con relazioni lineari e non lineari con l'output.

Svantaggi:

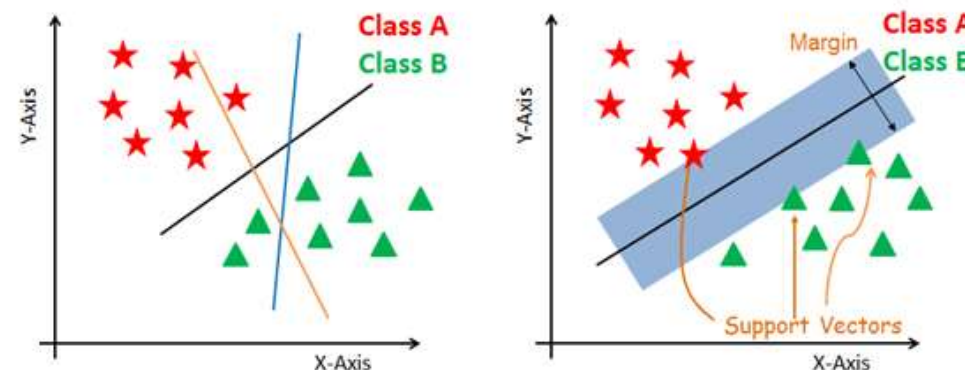
- ❑ Sono instabili, il che significa che un piccolo cambiamento nei dati può portare a un grande cambiamento nella struttura dell'albero decisionale ottimale.
- ❑ Sono spesso relativamente imprecisi. Molti altri predittori funzionano meglio con dati simili.
 - **Questo può essere risolto rimpiazzando un singolo albero decisionale con una foresta casuale di alberi decisionali (random forest), ma un random forest non è facile da interpretare come un singolo decision tree.**

Algoritmi di classificazione

✓ Support Vector Machines

- L'algoritmo SVM effettua la classificazione tramite la costruzione di iperpiani in grado di separare in modo ottimale due classi
- Una riga del dataset di input che descrive un certo elemento è chiamato vettore (vector) ed è costituito da tutte le feature che si ottengono dagli attributi originali tramite l'applicazione di una funzione di mapping.

I vettori che si trovano vicino all'iperpiano sono chiamati vettori di supporto (support vectors).

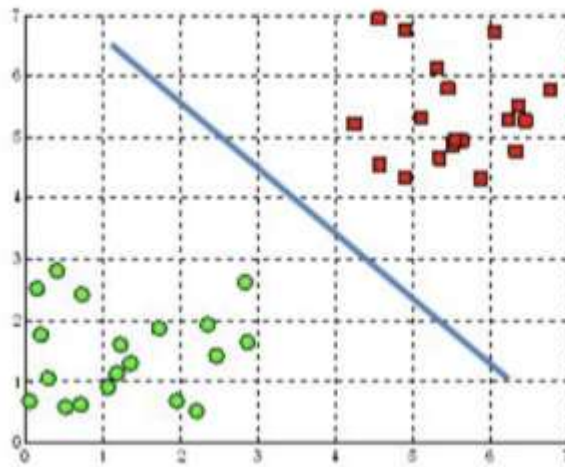


Algoritmi di classificazione

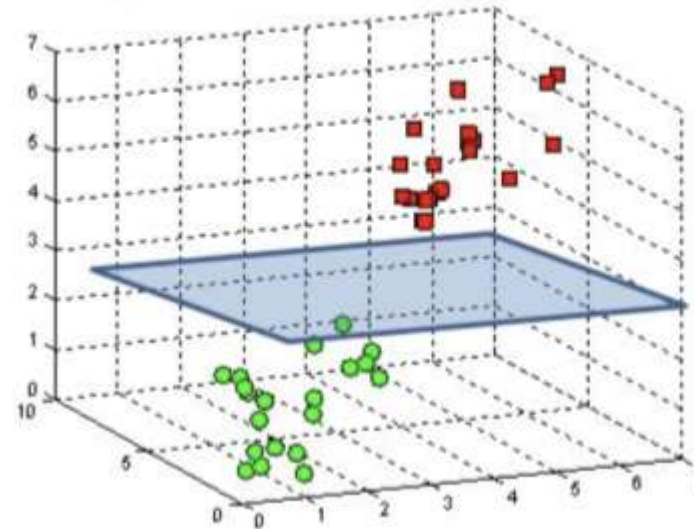
✓ *Support Vector Machines*

- Inizialmente consideriamo un mapping che non introduce alcuna trasformazione: Il compito dell'algoritmo è quello di separare insiemi di vettori utilizzando un iperpiano ottimale.

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane



Algoritmi di classificazione



✓ Support Vector Machines

- Vi sono infinite rette che possono separare i punti appartenenti alle due classi.
- La parte destra dell'immagine mostra una separazione non ottimale, mentre l'altra mostra la miglior separazione.
 - Si ottiene massimizzando la distanza tra la linea che separa (riga continua) e le linee tratteggiate, costruite in modo da essere adiacenti ai vettori di supporto.
- Inoltre le linee devono essere create in modo che non vi siano punti compresi tra le due linee tratteggiate. La distanza è chiamata *margin* (*margin*).

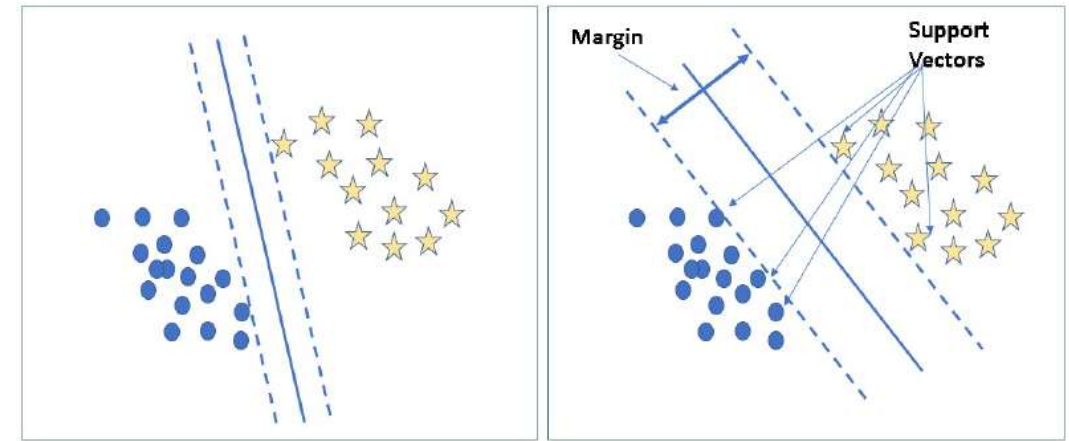


Figura 13.7: Separazione delle classi con SVM.

Algoritmi di classificazione



✓ Support Vector Machines

- È possibile che non si riesca a separare perfettamente le classi.
- In questo caso, nell'equazione del margine si tiene conto di un valore che rappresenta la distanza tra un punto e il margine che tocca i vettori di supporto della stessa classe a cui appartiene il punto che non è possibile separare (tecnica di soft-margin).

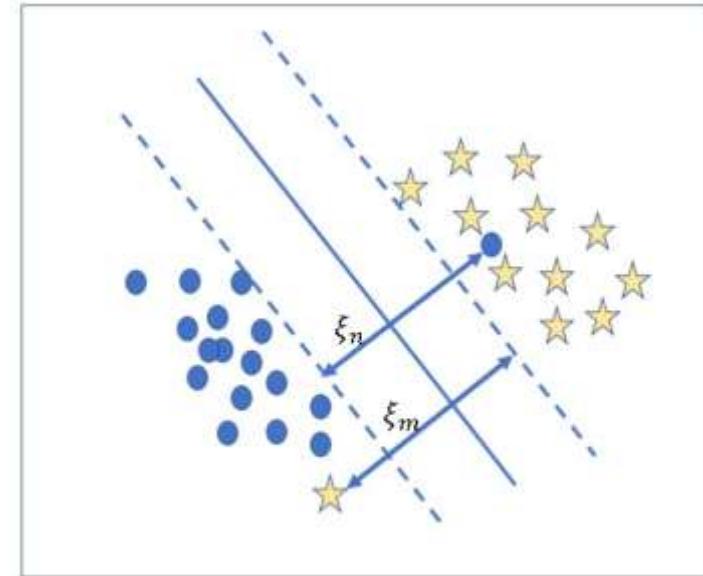


Figura 13.8: Classi non completamente separabili.

- *Tuttavia, anche la tecnica del soft margin, può non riuscire a fornire una separazione accettabile tra le due classi.*
- Esiste una soluzione anche a problemi che in apparenza non sono linearmente separabili. Essa consiste nel mappare i dati in uno spazio (**feature space**) con più dimensioni rispetto a quello di partenza.

Algoritmi di classificazione

✓ *Support Vector Machines : feature space*

- La Figura mostra un insieme di punti non separabili se visti in uno spazio a due dimensioni,
- ma che lo diventano se si applica una trasformazione polinomiale che mappi i punti bidimensionali (x_1, x_2) in uno spazio a tre dimensioni (z_1, z_2, z_3) .

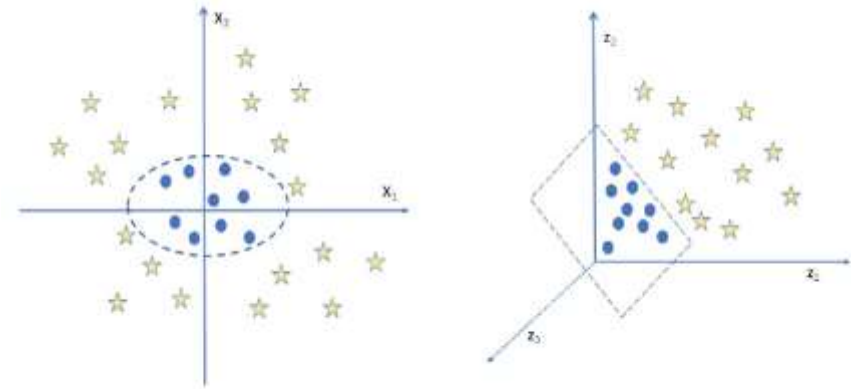


Figura 13.9: Separazione in uno spazio dimensionale più alto.

- *Questo approccio è però impraticabile per un numero elevato di dimensioni.*
- *Si dimostra che è possibile ottenere i benefici della mappatura in uno spazio dimensionale superiore, senza realmente operarla e quindi senza incorrere in problemi di calcolo.*
- Questo grazie all'utilizzo di una *funzione K* è *chiamata funzione kernel* e un processo chiamato **kernel trick** grazie al quale è possibile utilizzare la funzione kernel al posto del «reale» mapping delle feature

Algoritmi di classificazione

✓ Support Vector Machines :funzioni kernel

1. Kernel lineare, che corrisponde all'utilizzo delle feature senza un mapping in uno spazio dimensionale più elevato. → $K(x_1, x_2) = x_1^T x_2 + c$
2. Kernel polinomiale, che può risolvere problemi della slide precedente → $K(x_1, x_2) = (\alpha x_1^T x_2 + c)^d$
3. Kernel Gaussiano (o radial basis kernel) e similari utile per lo stesso tipo di problema
 - Se sigma è troppo grande il comportamento sarà simile al kernel lineare,
 - se troppo piccolo, la funzione sarà troppo sensibile al rumore nei dati di training.→ $K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$
4. Kernel Sigmoidale (o Hyperbolic Tangent Kernel, o MLP Kernel)
 - una SVM che utilizza il kernel sigmoide è equivalente a una rete neurale di tipo perceptron con soli due livelli.→ $K(x_1, x_2) = \tanh(\alpha x_1^T x_2 + c)$

Algoritmi di classificazione



✓ Support Vector Machines

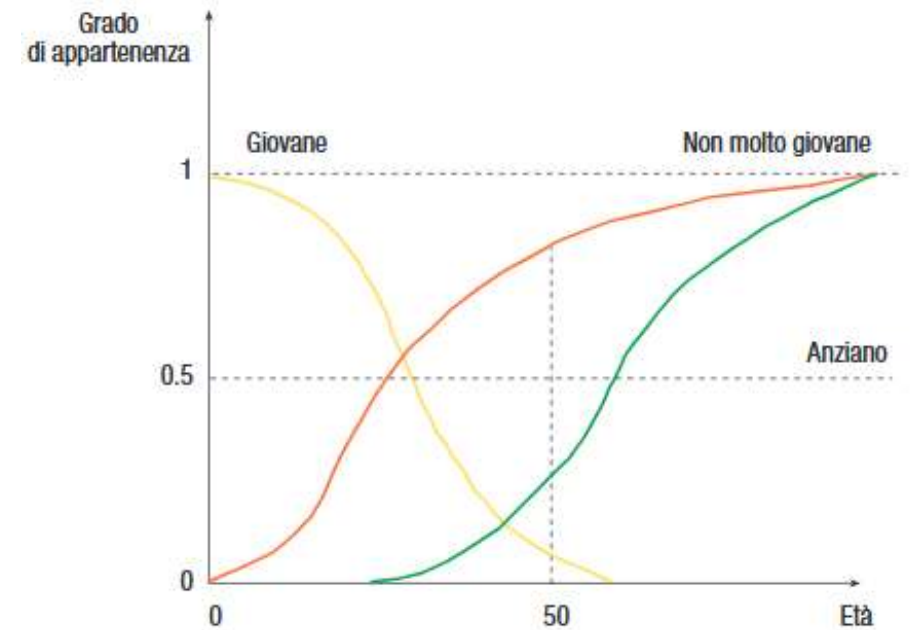
- ❑ La scelta del kernel più adatto e dei parametri delle funzioni **non** sono semplici da realizzare.
 - Una tecnica possibile consiste nel provare più funzioni e un set di parametri valutando i risultati e scegliendo la combinazione più performante.
- ❑ I vantaggi delle SVM:
 - capacità attraverso il kernel trick, di agire su problemi dove le classi non sono linearmente separabili.
 - lavorano bene in spazi dimensionali elevati.
 - Sono molto efficaci quando le classi sono sufficientemente separate (linearmente o non linearmente),
- ❑ E svantaggi
 - hanno qualche problema se i margini non sono ben definiti e vi sono troppe sovrapposizioni.
 - Inoltre non producono in maniera diretta la probabilità di appartenenza ad una data classe, che quindi deve essere ricavata utilizzando calcoli abbastanza onerosi.

Algoritmi di classificazione



✓ Fuzzy rules based systems

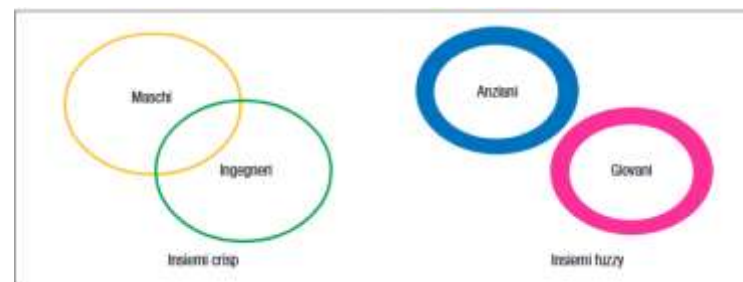
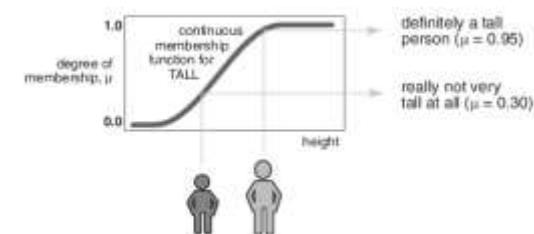
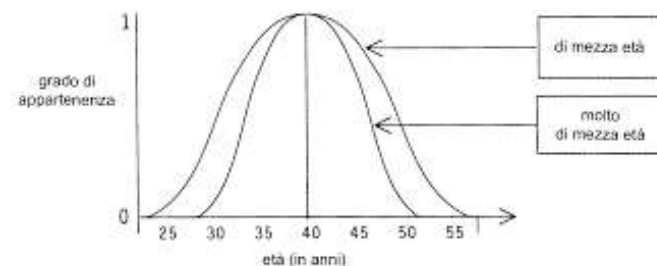
- I Fuzzy Rule Based Systems rappresentano un'applicazione della logica fuzzy a problemi di analisi predittiva.
- La logica binaria, che costituisce la base per il funzionamento dei computer, prevede che i predicati possano assumere solamente due stati: vero e falso.
- Tuttavia tale semplificazione risulta spesso imprecisa e non aderente alla realtà, la quale contempla numerose sfaccettature: non esistono infatti soltanto il bianco ed il nero, ma esistono numerose sfumature che si pongono tra questi due valori estremi.
- La logica fuzzy tiene in considerazione proprio queste sfumature, discostandosi così dalla logica binaria



Algoritmi di classificazione

✓ Fuzzy rules based systems

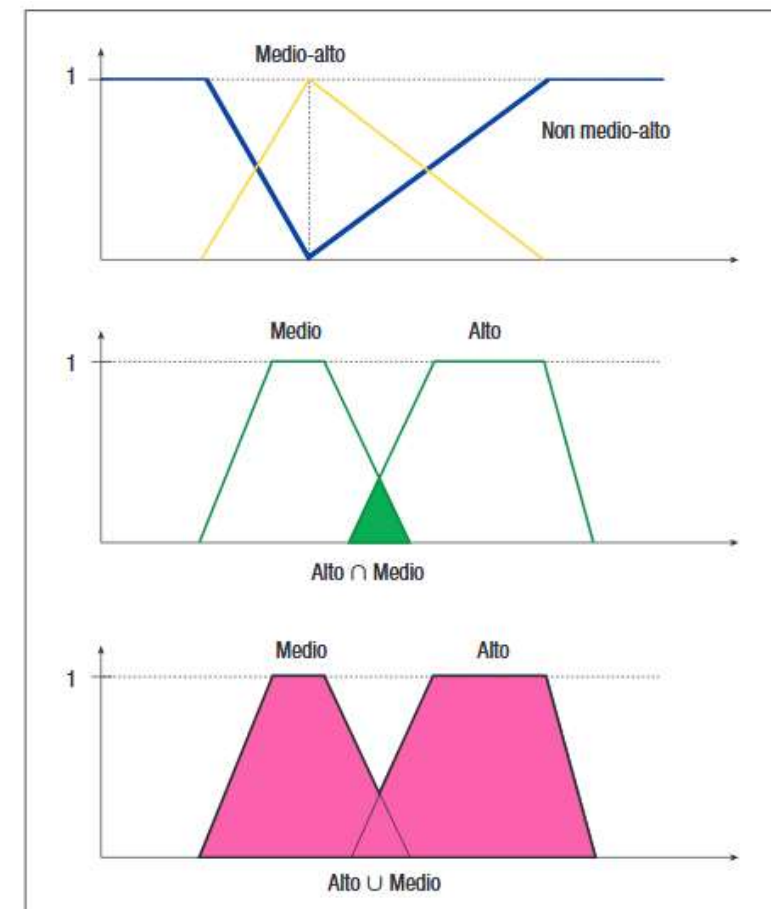
- Nella logica tradizionale un elemento appartiene o non appartiene ad un determinato insieme, mentre nella logica fuzzy, un dato elemento appartiene ad un insieme fuzzy con un grado di verità che può assumere infiniti valori nell'intervallo $[0,1]$.
- Il grado di verità (o di appartenenza ad un insieme fuzzy) è definito da una funzione di appartenenza (membership).



Algoritmi di classificazione

✓ Fuzzy rules based systems

- ❑ Variabili fuzzy come variabili linguistiche
 - ❑ Es la temperatura può essere descritta da una variabile fuzzy che possiede i valori “linguistici” freddo e caldo.
 - ❑ Ciascun elemento avrà un grado di appartenenza a ciascun valore (insieme fuzzy), per esempio 0.90 caldo e 0.10 freddo.
- In modo formale:
- ❑ x il nome della variabile (es. temperatura);
 - ❑ T l'insieme di definizione della variabile (es. $[0^\circ, 50^\circ]$);
 - ❑ F_i i fuzzy set definiti per la variabile (ovvero i valori linguistici, che nel caso delle temperature potrebbero essere molto freddo, freddo, tiepido, caldo, molto caldo);
 - ❑ $A_i(x)$ le funzioni di appartenenza a ciascun set, che possono avere varie forme: triangolari, trapezoidali, sigmoidali, gaussiane.).

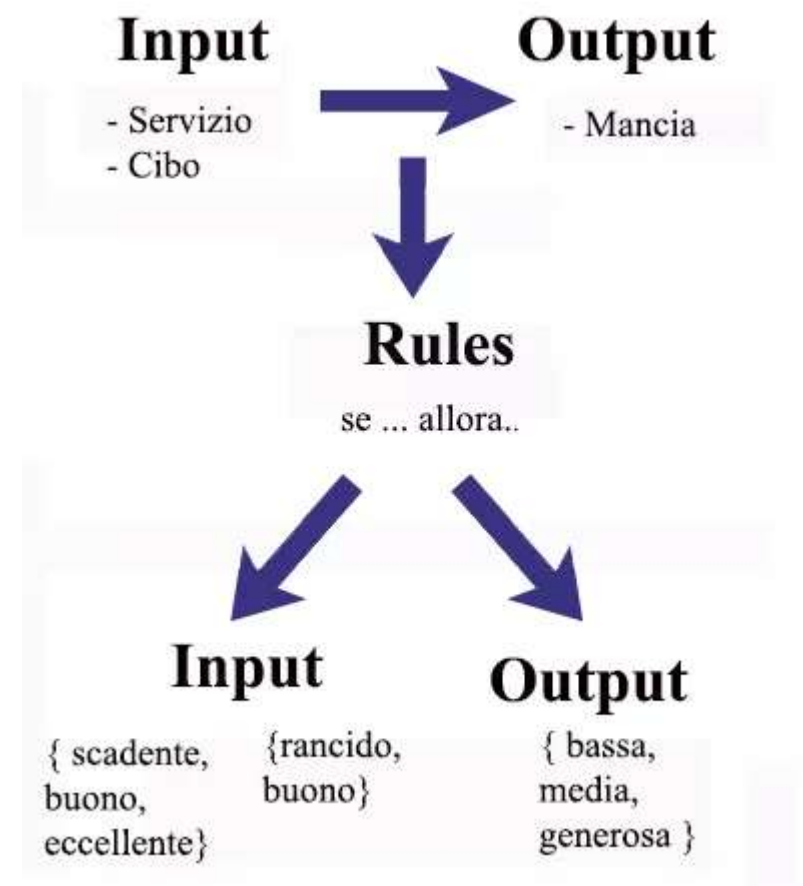


Algoritmi di classificazione

✓ Fuzzy rules based systems

□ Sistema fuzzy

1. La fuzzificazione: in questa fase le grandezze sono trasformate in base alle funzioni di appartenenza.
2. L'applicazione di regole
 - L'applicazione delle regole determina il valore dell'uscita a fronte della combinazione degli input. Le regole sono costituite da un insieme di proposizione IF...THEN.
3. La defuzzificazione: il valore di uscita che deriva dall'applicazione delle regole fuzzy va convertito in un valore deterministico attraverso un processo chiamato defuzzificazione.
 - Uno dei più semplici da comprendere è quello basato sulla media dei massimi: il valore di uscita è ottenuto come media aritmetica dei valori per i quali è massima l'altezza del fuzzy set determinato dalle regole.



➤ *Fuzzy rules based systems*

- ❑ L'utilizzo dei sistemi fuzzy per la classificazione e la regressione prevede due fasi:
 - La fase di apprendimento, nella quale, attraverso i dati di training si identifica la struttura, ovvero l'insieme delle regole. In seguito sono ottimizzati i parametri delle funzioni di appartenenza.
 - La fase predittiva, nella quale si applicano, ai nuovi dati, le tre fasi tipiche dei sistemi fuzzy ovvero: fuzzificazione, applicazione delle regole, defuzzificazione.

Algoritmi di regressione



- ✓ **La regressione predice un valore numerico specifico.**
- ✓ Determinano il valore di una variabile continua in base alle feature di input.
- Ad esempio un modello potrebbe predire che il cliente X ci porterà un profitto di Y lire nel corso di un determinato periodo di tempo.
- Le variabili in uscita possono assumere un numero illimitato (o comunque una grande quantità) di valori.
- Spesso queste variabili in uscita sono indicate come continue anche se talvolta non lo sono nel senso matematico del termine (ad esempio l'età di una persona)

Algoritmi di regressione



✓ *Regressione Lineare:*

- ❑ assume che la relazione tra la variabile target e le variabili di input sia lineare.
- ❑ La relazione comprende anche una variabile di errore, ovvero una variabile casuale non rilevata, che aggiunge rumore alla relazione lineare.
- Si assume che:
 - ❑ l'errore abbia una distribuzione normale con media 0 e varianza costante
 - e quindi non cambi al variare dei valori delle feature di input (omoschedasticità).
 - ❑ Vi sia una indipendenza degli errori e l'assenza di multi collinearità delle variabili di input
 - cioè la presenza di due o più variabili di input tra loro correlate

Algoritmi di regressione

✓ *Regressione Lineare:*

- Sia
 - y il vettore che rappresenta la variabile target,
 - X la matrice delle feature,
 - β il vettore dei parametri da stimare,
 - Epsilon variabile casuale dell'errore
- La stima dei parametri avviene con il metodo dei minimi quadrati (RSS - Residual Sum of Squares) che minimizza la somma al quadrato delle differenze tra il valore reale e il valore stimato
- La formula chiusa che si ottiene per la stima di b , sempre in notazione

$$y = X\beta + \varepsilon$$

$$RSS = \|y - X\hat{\beta}\|^2$$

$$\hat{\beta} = (X^T X)^{-1} (X^T y)$$

Algoritmi di regressione

✓ *Regressione logistica*

- La regressione logistica fa parte dei modelli lineari generalizzati, ovvero di quei modelli che prevedono:
 1. Una combinazione lineare delle feature di input.
 2. Una distribuzione esponenziale per la variabile di output (normale, Poisson, binomiale, gamma,...).
 3. Una link function che lega la media della distribuzione di output alla combinazione delle feature di input

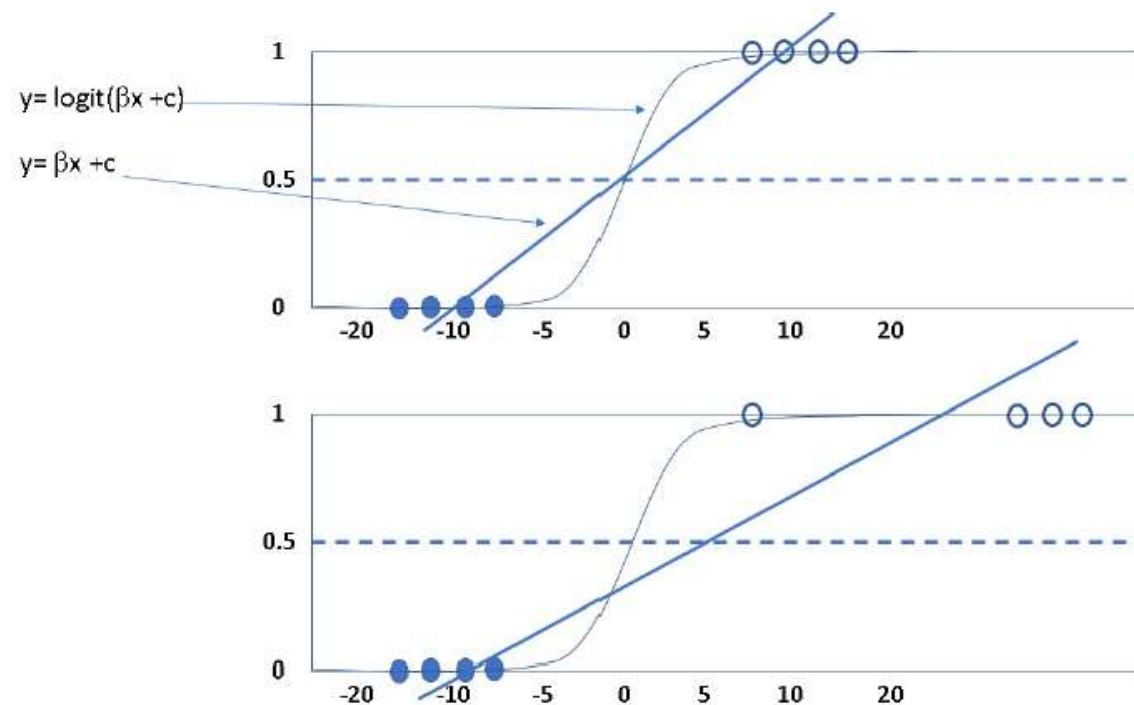


Figura 13.4: Esempio di regressione lineare e logistica.

✓ si vede come la regressione logistica non sia influenzata da valori estremi, proprio per la natura della funzione logit

- ❑ Da rivedere
- ❑ Regressione logistica

Algoritmi semi supervisionati

- ❑ Si pongono a metà strada tra gli algoritmi supervisionati e quelli non supervisionati
 - lavorano su un dataset per i quali alcuni elementi hanno la variabile di output valorizzata, mentre altri, di solito la maggioranza, non ha alcun valore nella variabile di output.
- ❑ Caso frequente di un problema di classificazione in cui vi sono pochi punti su cui si conosce la classe (*label*)
 - Ciò può essere dovuto al costo elevato per una classificazione manuale dei campioni
- ❑ Al fine di sfruttare i dati senza label, lavorano sotto determinate ipotesi:
 - Una di esse consiste nel considerare della stessa classe i punti che sono tra loro vicini.
 - Un'altra ipotesi considera i dati dello stesso cluster come appartenenti alla stessa classe (si sfruttano gli algoritmi di clustering per raggruppare gli elementi ed attribuirvi la classe).
 - Infine, per rendere più semplici i calcoli si assume che i dati possano essere approssimati su una varietà (o manifold) con dimensionalità di molto inferiore rispetto all'originale.

Algoritmi semi supervisionati

- Esistono numerose *tecniche semi supervisionate* raggruppabili in alcune categorie tra cui
 1. self-training,
 2. co-training *generative models* (modelli basati sul clustering),

Algoritmi semi supervisionati

- ❑ *tecniche semi supervisionate* : **self-training**
- ❑ consiste nell'utilizzare un algoritmo di classificazione (Naïve Bayes, ecc.) per attribuire una label ai dati non classificati.
- ❑ Poi si estraggono dai dati classificati i ***k elementi con score più elevato***
 - presentano la probabilità più alta di appartenenza alla classe attribuita dall'algoritmo
- ❑ ***L'insieme dei dati classificati è quindi aumentato*** rispetto al passaggio precedente e viene utilizzato per il training del classificatore.
- ❑ Questo processo si può poi iterare.
- Vantaggio: estremamente semplice e utilizza algoritmi di classificazione comuni.
- Attenzione: gli errori di classificazione commessi nei primi cicli si amplificano, visto che nei cicli successivi sono visti come dati di training e quindi corretti.

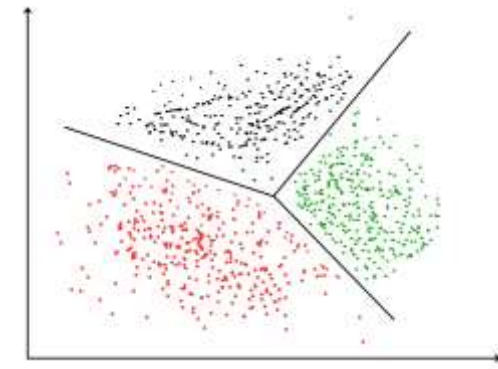
Algoritmi semi supervisionati

- ❑ **tecniche semi supervisionate : co-training**
- ❑ presuppone l'esistenza di due “viste” dei dati che siano tra loro indipendenti.
 - ES: per i contenuti di pagine web : immagini contenute nelle pagine e il testo delle pagine,
- ❑ Si esegue il training di due classificatori C1 e C2 sui due dataset D1 e D2
- ❑ Poi si classificano i dati senza label utilizzando C1 e C2 e si aggiungono a D1 i k elementi classificati da C2, che presentano lo score più elevato e a D2 i k elementi classificati da C1
- ❑ Il procedimento si può iterare
- Il cotraining è meno sensibile, rispetto al self-training, agli errori di classificazione.
- Tuttavia si assume che esistano due set di feature condizionalmente indipendenti data la classe e che ciascuno dei due set sia sufficiente per il training di un classificatore

Una variante effettua il training di diversi classificatori sugli stessi dati; allo step successivo sono aggiunti gli elementi che sono classificati allo stesso modo dalla maggioranza dei classificatori.

Algoritmi di clustering

- ❑ ***Tipico problema non supervisionato***
- ❑ E' un insieme di metodi per raggruppare oggetti in classi omogenee.
- ❑ Un cluster è un insieme di oggetti che presentano tra loro delle similarità, ma che, per contro, presentano dissimilarità con oggetti in altri cluster.
- ❑ L'input di un algoritmo di clustering è costituito da un campione di elementi,
- ❑ l'output è dato da un certo numero di cluster in cui gli elementi del campione sono suddivisi in base a una misura di similarità



- ***Quando usare l'apprendimento senza supervisione***
 - Quando la variabile di risposta non è del tutto chiara. Non vi è nulla che stiamo tentando esplicitamente di prevedere o correlare con le altre variabili.
 - Per estrarre dai dati una struttura laddove tale struttura o schema non sembra esistere
 - Quando viene usato un concetto senza supervisione chiamato estrazione delle caratteristiche. L'estrazione delle caratteristiche è un processo che consiste nel creare nuove caratteristiche a partire da quelle esistenti. Queste nuove caratteristiche possono essere perfino più efficaci di quelle originali.
 - E utilizzate in un successivo modello con supervisione

Algoritmi di clustering

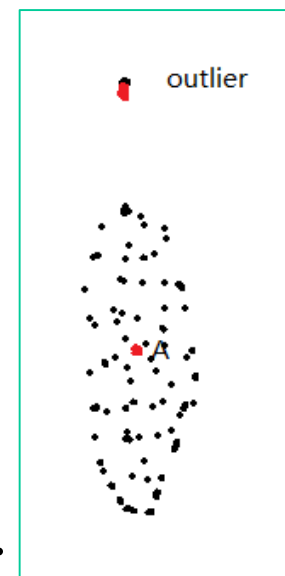
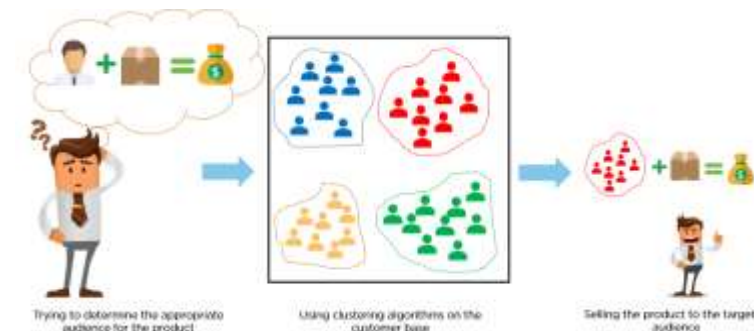
La cluster analysis è ampiamente utilizzata

- ❑ Ricerche di mercato.
- ❑ Riconoscimento di pattern.
- ❑ Raggruppamento di clienti in base ai comportamenti d'acquisto
- ❑ Posizionamento dei prodotti.
- ❑ Analisi dei social network, per il riconoscimento di community di utenti.
- ❑ Identificazione degli **outliers**.

➤ Gli outliers sono valori anomali che presentano grandi differenze con tutti gli altri elementi di un dataset.

La loro identificazione può essere interessante per due scopi:

- ❑ l'eliminazione di questi valori anomali, che potrebbero essere causati da errori,
- ❑ l'isolamento di questi casi che magari rivestono una certa importanza per il business.



Algoritmi di clustering

Si dividono in due categorie principali:

- ❑ Algoritmi di clustering gerarchico.
 - organizzano i dati in sequenze nidificate di gruppi che potremmo rappresentare in una struttura ad albero
- ❑ Algoritmi di clustering partizionale.
 - Gli algoritmi di clustering partizionale, invece, determinano il partizionamento dei dati in cluster, in modo da ridurre il più possibile la dispersione all'interno del singolo cluster, viceversa, di aumentare la dispersione tra i cluster
 - più adatti a dataset molto grandi

➤ *K-means*

Algoritmi di clustering

Quando è possibile clusterizzare?

un algoritmo di clustering, applicato ad un dataset, fornirà in ogni caso una suddivisione in cluster, che però potrebbero avere una scarsa validità dal punto di vista dell'utilizzo pratico

Bisogna procedere alla:

- Verifica della presenza nel dataset di cluster significativi.

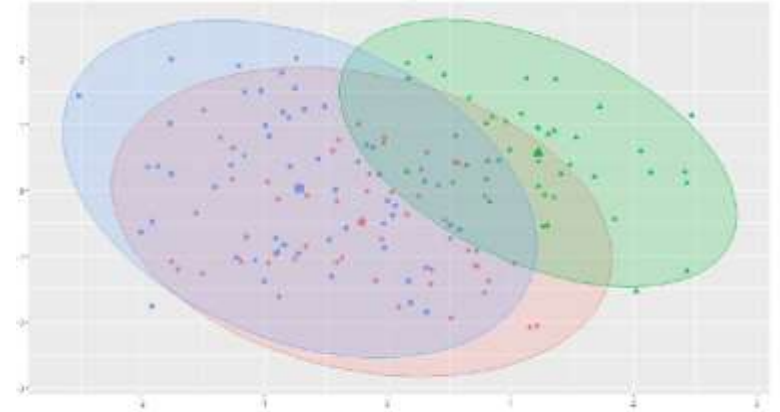


Figura 13.13: Clustering su dati casuali.

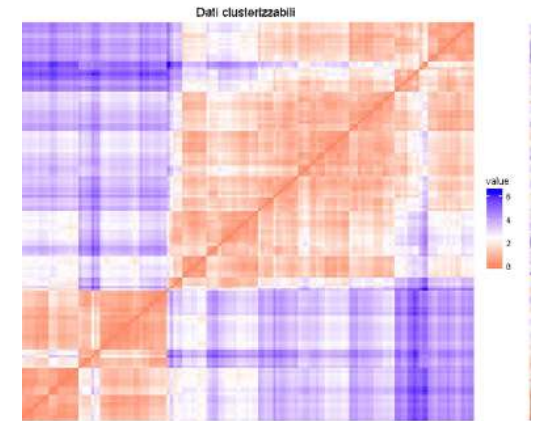


Individuazione di cluster senza senso a cui cioè non possibile dare una interpretazione

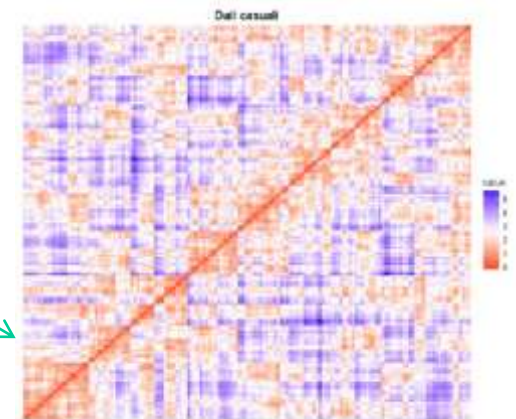
Algoritmi di clustering



Dati clusterizzabili



Dati non clusterizzabili



Misurare la propensione al clustering del dataset

1. Valutare la clusterizzazione con algoritmi diversi di clustering
2. Utilizzare degli indici statistici che stimano quanto i dati sono distribuiti in modo uniforme e che quindi i dati siano posizionati nello spazio dimensionale in modo casuale
 - applicabili prevalentemente con una ridotta dimensionalità
 - *Uno di questi è la statistica di Hopkins*
 - Se la statistica di Hopkins si avvicina al valore 0.5 significa che i dati sono uniformemente distribuiti e quindi la creazione di cluster **non è realizzabile**
3. Calcolare la matrice di dissimilarità usando la distanza euclidea (o una qualsiasi altra misura di distanza). La matrice è riordinata in modo da posizionare vicini gli oggetti simili e plottata su un grafo

Algoritmi di clustering

Le misure di distanza

➤ Implementano il concetto di similarità

□ Distanza Euclidea.

- Dati due punti $P = (p_1, p_2, p_3, \dots, p_k)$ e $Q = (q_1, q_2, q_3, \dots, q_k)$ è data da

$$d = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

□ Distanza Manhattan.

- La distanza tra due punti P_1 e P_2 con coordinate rispettivamente (x_1, y_1) e (x_2, y_2) è data dalla somma del valore assoluto delle differenze tra le coordinate:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Algoritmi di clustering

Le misure di distanza

➤ Implementano il concetto di similarità

- **Distanza della correlazione di Pearson.**

- Data da 1 meno il valore di correlazione.

- **Distanza del coseno (o di Eisen).**

- È un caso speciale della distanza della correlazione di Pearson, nella quale sostituiamo la media che entra nella formula, con il valore 0 ottenendo così:

$$d = 1 - \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

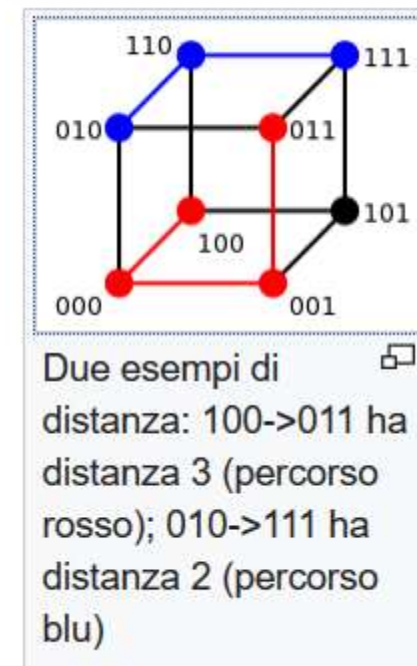
Algoritmi di clustering

Le misure di distanza

➤ Implementano il concetto di similarità

- ❑ **Distanza di Hamming.**
- ❑ Si utilizza come misura di similarità per le stringhe e, per stringhe di ugual lunghezza è definita come il numero di posizioni nelle quali i simboli corrispondenti sono diversi.
- ❑ Quindi misura il numero di sostituzioni necessarie per convertire una stringa nell'altra.

- La distanza di Hamming tra 10**1**1**1**01 e 10**0**1**0**01 è 2.
- La distanza di Hamming tra 2**1**4**3**896 e 2**2**3**3**796 è 3.



Le misure di distanza

- *è importante scegliere la misura di distanza più adeguata, poiché essa influenza molto il risultato del clustering*
- Per esempio le misure basate sulla correlazione considerano due punti vicini se sono molto correlati, anche se la loro distanza euclidea è grande.
- Quindi se ciò che interessa è raggruppare gli elementi in base al loro profilo e non in base alla magnitudine dei valori, le misure di correlazione sono le più adatte;
- per contro distanze come quella euclidea sono utili quando si vuol tener conto dell'entità dei valori.

Algoritmi di clustering



Algoritmo k-means

1. Definiamo il numero k di cluster desiderati.
2. Partizioniamo l'insieme in K cluster, assegnando a ciascuno di essi degli elementi scelti a caso.
3. Calcoliamo i centroidi di ciascun cluster k con la formula in alto
4. Calcoliamo la distanza degli elementi del cluster dal centroide, ottenendo un errore quadratico, con la formula in basso
5. A questo punto si riassegnano gli elementi del campione in base al più vicino centroide.
6. Si ripetono i passaggi 2, 3, 4 e 5 finché il valore minimo dell'errore totale non è raggiunto, oppure finché i membri dei cluster non si stabilizzano, oppure finché non si raggiunge un numero massimo di iterazioni, predefinito.

$$M_k = 1/n_k \times \sum_{i=1}^{n_k} x_{ik}$$

Dove:

M_k è il vettore delle medie, o centroide per il cluster k

n_k è il numero di elementi del cluster k

x_{ik} è l' i -esimo elemento del cluster

$$e_k^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2$$

dove:

e_k^2 è l'errore quadratico per il cluster k

x_{ik} è l' i -esimo elemento del cluster

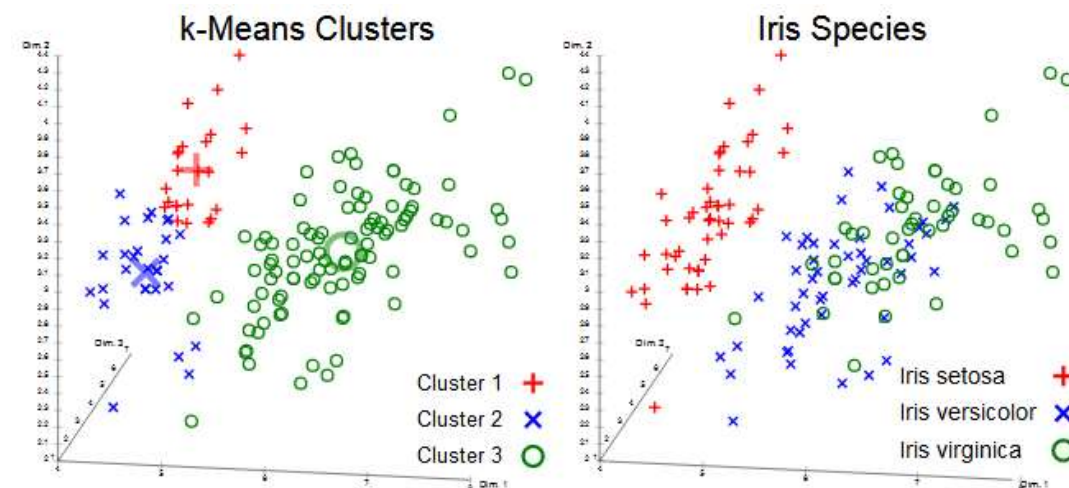
n_k è il numero di elementi del cluster k

M_k è il vettore delle medie, o centroide per il cluster k

Algoritmi di clustering

Algoritmo k-means

- L'algoritmo k-means è sensibile all'allocazione iniziale dei valori e, in base ad essa, potrebbe convergere a un minimo locale della funzione d'errore e non al minimo assoluto.
- Sempre a causa dell'allocazione iniziale casuale, i risultati del k-means potrebbero essere diversi ad ogni esecuzione sugli stessi dati.
- Inoltre è molto sensibile al rumore nei dati e alla presenza di outliers.
- Per far fronte al problema dell'allocazione iniziale, si può creare un certo numero di clusterizzazioni, valutando quale sia la migliore



[Iris flower data set](https://es.wikipedia.org/wiki/Archivo:Iris_Flowers_Clustering_kMeans.svg), clustered using [k means](#) (left) and true species in the data set (right). Note that k-means is non-deterministic, so results vary. Cluster means are visualized using larger, semi-transparent markers.

The visualization was generated using [ELKI](#).

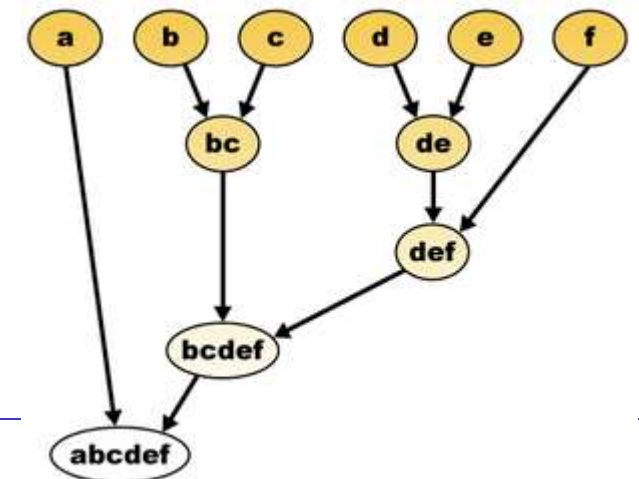
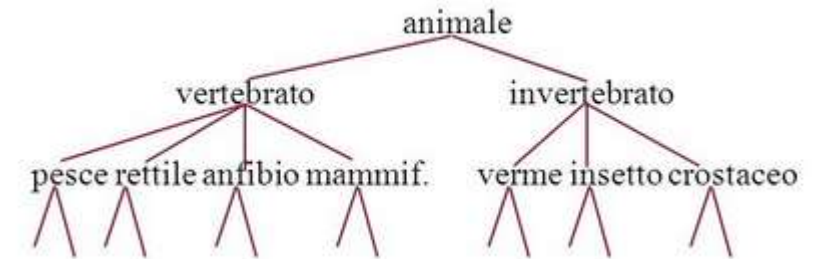
https://es.wikipedia.org/wiki/Archivo:Iris_Flowers_Clustering_kMeans.svg

Algoritmi di clustering



Clustering gerarchico

- è un approccio di clustering che mira a costruire una gerarchia di cluster. Le strategie per il clustering gerarchico sono tipicamente di due tipi:
 - **Divisivo**: si tratta di un approccio "*top down*" (dall'alto verso il basso) in cui tutti gli elementi si trovano inizialmente in un singolo cluster che viene via via suddiviso ricorsivamente in sotto-cluster.
 - **Agglomerativo**: si tratta di un approccio "*bottom up*" (dal basso verso l'alto) in cui si parte dall'inserimento di ciascun elemento in un cluster differente e si procede quindi all'accorpamento graduale di cluster a due a due.
- Il risultato di un clustering gerarchico è rappresentato in un **dendrogramma**

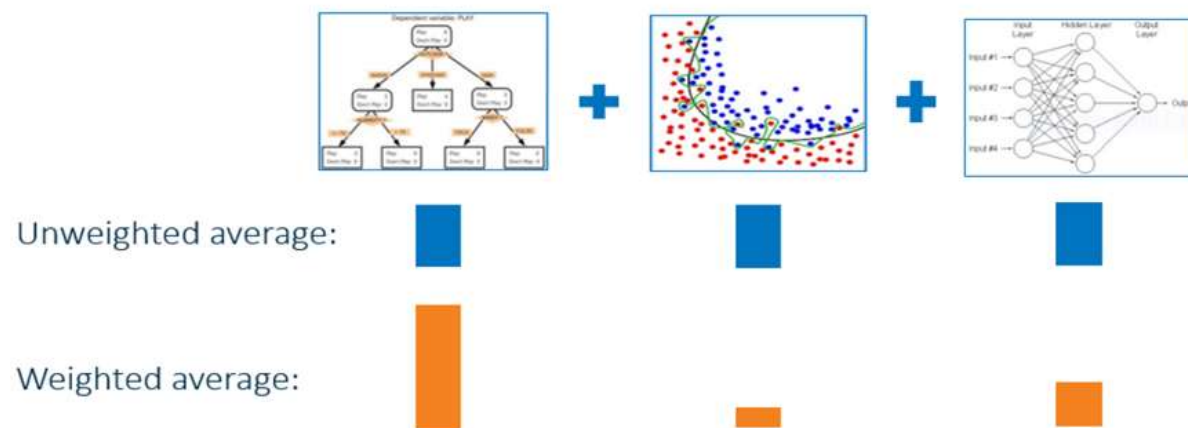
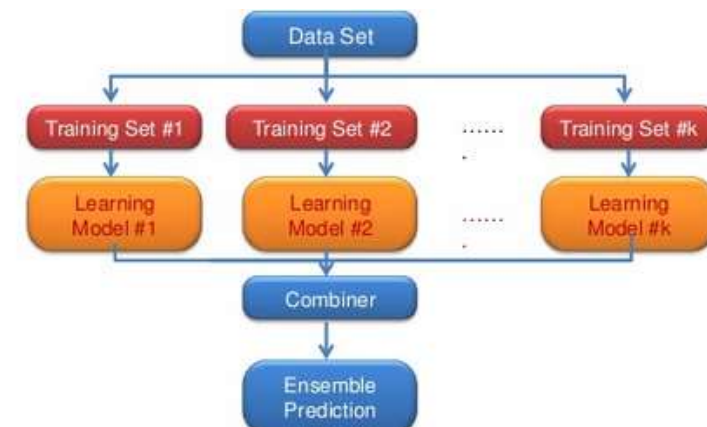


Model ensembles

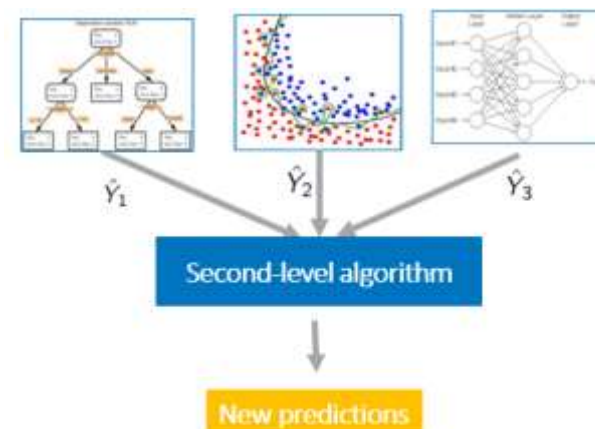
- Il concetto di model ensemble riguarda *la creazione di un insieme di modelli che lavorano assieme, fornendo un risultato combinato in grado di migliorare le performance predittive in termini di bias e varianza.*
 - Il training di ciascuno dei modelli può, a seconda delle tecniche utilizzate, essere eseguito sugli stessi dati per ciascun modello, oppure su campioni casuali, che quindi saranno differenti per ognuno dei modelli.
 - Anche la composizione dei risultati finalizzata ad ottenere l'output finale è ottenuta in modo diverso a seconda del tipo di ensemble usato.
- A fronte del miglioramento anche sensibile delle performance predittive, va detto che esiste un aspetto negativo legato alle tecniche di ensemble e dovuto alla perdita delle proprietà esplicative che certi algoritmi posseggono.
 - ES:, sappiamo che gli alberi decisionali producono regole, che accompagnano la prediction; tali regole in un ensemble di alberi non sono estraibili, visto che in questo caso il risultato è una sintesi dei vari modelli (una media, o il risultato di un meccanismo di votazione)

Model ensembles

- Un **supermodello** viene prodotto mediante:
- Regressione**: si calcola la media delle previsioni di ogni modello.
- Classificazione**: si va a votazione e si sceglie la previsione più comune o si calcola la media delle probabilità previste.



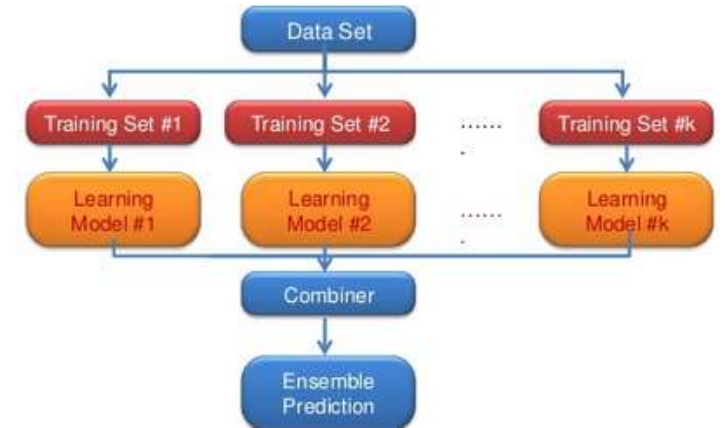
Averaging predictions to form ensemble models.



Model stacking uses a second-level algorithm to estimate prediction weights in the ensemble model.

Model ensembles

- Perché l'ensembling possa funzionare correttamente, i modelli devono avere le seguenti caratteristiche.
- **Accuratezza:** ogni modello deve come minimo comportarsi meglio del modello nullo (*nb è il modello completamente a caso*).
- **Indipendenza:** le previsioni di un modello non sono influenzate dal processo di previsione di un altro modello.
- Se ci sono una certa quantità di modelli appropriati, il caso particolare in cui un modello sbaglia probabilmente non influenzerà gli altri modelli,
- pertanto tali errori verranno ignorati quando i modelli verranno combinati fra loro.

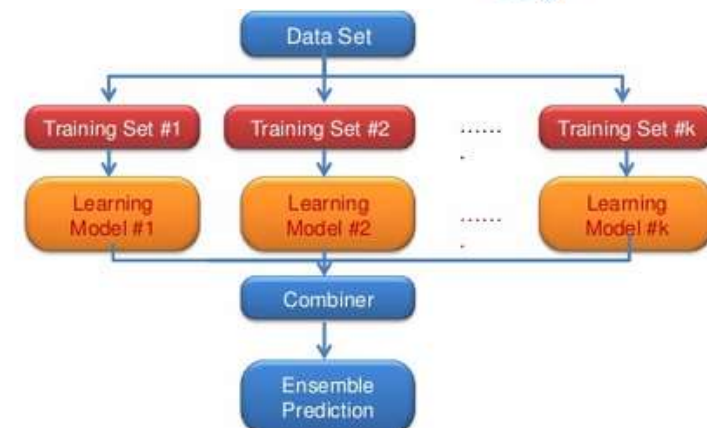


Model ensembles

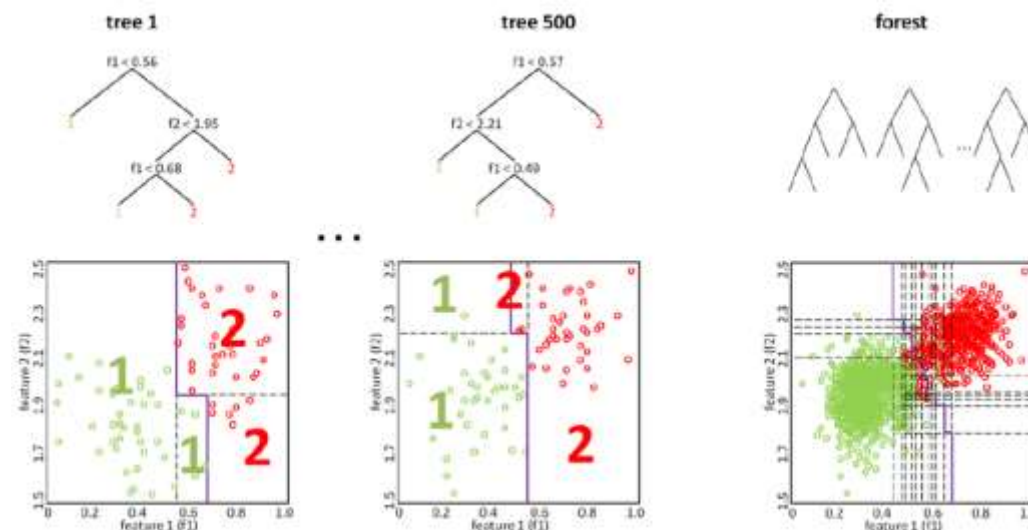


I metodi di ensembling sono sostanzialmente due:

- si assemblano manualmente i singoli modelli scrivendo una grande quantità di codice;
- si usa un modello che esegua automaticamente l'ensembling.



Un esempio è l'algoritmo di **Random Forest** che combina più alberi decisionali



Model ensembles

- ❑ Gli alberi decisionali tendono ad avere un bias contenuto e un'elevata varianza.
 - Dato qualsiasi dataset, l'albero può continuare a porre domande (prendere decisioni) finché non è in grado di distinguere ogni singolo esempio presente nel dataset.
 - Potrebbe continuare a porre domande su domande finché non vi sarà un unico esempio in ogni foglia (nodo terminale).
- ❑ L'albero esagera, cresce troppo in profondità e finisce per memorizzare, semplicemente, ogni singolo dettaglio del nostro set di addestramento.
- ❑ Tuttavia, **ripartendo**, l'albero potrebbe, potenzialmente, porre domande differenti e crescere comunque molto, in profondità.
- ❑ Questo significa che vi sono molti possibili alberi in grado di distinguere tutti gli elementi, il che porta a un'elevata varianza.
- **Un albero è incapace di eseguire una buona generalizzazione.**
- ❑ Per ridurre la varianza di un singolo albero, possiamo limitare il numero di domande che un albero può porre o possiamo creare una versione d'insieme di singoli alberi decisionali: le foreste casuali => random forest

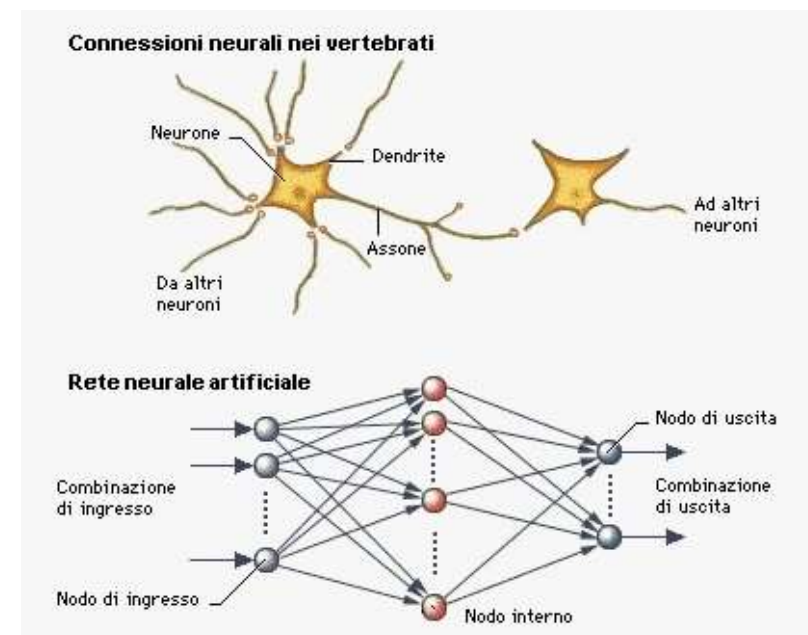
- Il random forest utilizza una **tecnica di Bagging** (bootstrapping o bootstrap sampling):
 - Prevede M estrazioni casuali di N elementi a partire dallo stesso training set. N è il numero di elementi dell'intero training set e le estrazioni avvengono con ripetizione, ovvero con la possibilità che un elemento possa essere estratto più volte
 - Per ciascuna delle M estrazioni è eseguito il training di un modello, utilizzando un algoritmo di classificazione. Il valore finale prodotto dall'ensemble è determinato per votazione: ciò significa che per un dato elemento x del dataset, la classe di appartenenza è quella predetta dalla maggioranza dei modelli (in caso di parità la classe è determinata in modo arbitrario)
 - Il bagging rende più stabile l'algoritmo di classificazione, riducendo la varianza dei risultati,
 - Ma se le variabili presentano valori poco frequenti alcuni di essi potrebbero non essere mai selezionati negli M campioni

Come funziona il bagging per gli alberi decisionali?

1. Lasciar crescere B alberi usando i campioni di bootstrap tratti dai dati di addestramento.
 2. Addestrare ogni albero sul suo campione di bootstrap ed effettuare le previsioni.
 3. Combinare le previsioni:
 - calcolare la media delle previsioni (per gli alberi di regressione);
 - Prendere il voto (per gli alberi di classificazione).
- Il numero di alberi solitamente tra 1000 e 5000 e il numero di nodi di ciascun albero garantisce che tutte le features siano estratte.
 - Il campionamento fa sì che la struttura degli alberi sia sempre diversa e ciò implica che si tengano in considerazione anche features che sono rilevanti anche solo a livello locale.
 - I rischi di overfitting sono comunque limitati, poiché la randomizzazione regolarizza l'impatto di ciascuna feature.
 - Infine il sampling riduce il costo di costruzione di un gran numero di alberi.

Reti neurali

- Le reti neurali (Neural Network, NN) sono modelli di calcolo «ispirate» dal modo di funzionare del cervello umano.
- Così come il nostro cervello è formato da neuroni interconnessi da legami chiamati sinapsi, le NN sono costituite da unità di calcolo (o neuroni artificiali) e da connessioni.
- Le NN possono essere rappresentate come grafi i cui nodi sono i neuroni e i cui archi sono le interconnessioni.



Le reti neurali si basano sul fatto che non sono solo una struttura complessa, ma anche flessibile. Il che significa:

- sono in grado di stimare funzioni di qualsiasi forma
- possono adattarsi e cambiare letteralmente la propria struttura interna sulla base dell'ambiente in cui operano.

Reti neurali

- ❑ Ciascuno dei nodi rappresenta un'unità di calcolo adattiva, poiché il proprio output dipende da parametri modificabili.
- ❑ I neuroni artificiali, infatti, sono in grado di variare i propri parametri di calcolo sulla base dei dati di training:
 - una NN ottiene, con il processo di apprendimento, un insieme di parametri ottimali che rappresentano la conoscenza del problema analizzato.
- ❑ Le reti neurali, grazie alla loro flessibilità, si adattano a numerosi tipi di problemi, quali, per esempio:
 - Analisi di marketing e di promozioni.
 - Stima di fluttuazioni del mercato finanziario.
 - Analisi di processi di produzione e industriali.
 - Diagnosi mediche.
 - Text mining.
 - ...



Reti neurali

- ❑ *Riconoscimento di pattern*: questa è probabilmente l'applicazione più comune delle reti neurali. Alcuni esempi sono il riconoscimento della scrittura e l'elaborazione delle immagini (riconoscimento facciale).
- ❑ *Movimento di entità*: fra gli esempi vi sono le auto a guida automatica, la robotica e il movimento dei droni.
- ❑ *Rilevamento delle anomalie*: poiché le reti neurali sono abili a individuare i pattern, possono essere usate anche per riconoscere quando un punto dei dati non segue un pattern. Pensate a una rete neurale che esegue il monitoraggio del mercato delle azioni; dopo aver appreso il movimento naturale delle azioni, potrebbe avvertire quando accade qualcosa di insolito.

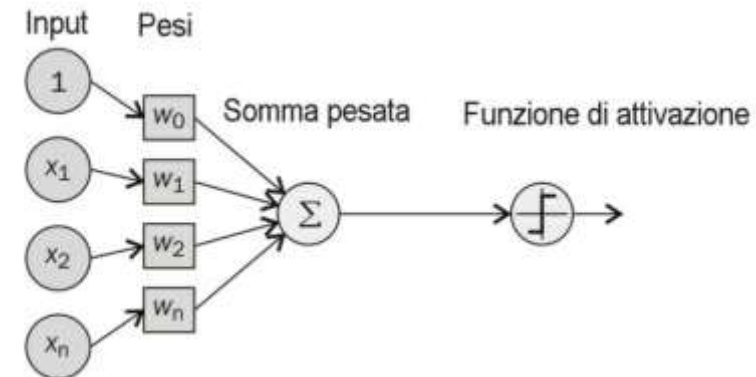


Reti neurali

Il singolo Perceptron:

- Un perceptron, rappresentato di seguito, accetta un certo input e produce in output un segnale.
- Questo segnale viene ottenuto combinando l'input con numerosi pesi e poi attraversando una funzione di attivazione
- Nel caso di semplici output binari, in genere si usa la funzione logistica (o sigmoide) che ha valori di uscita compresi fra 0 e 1:

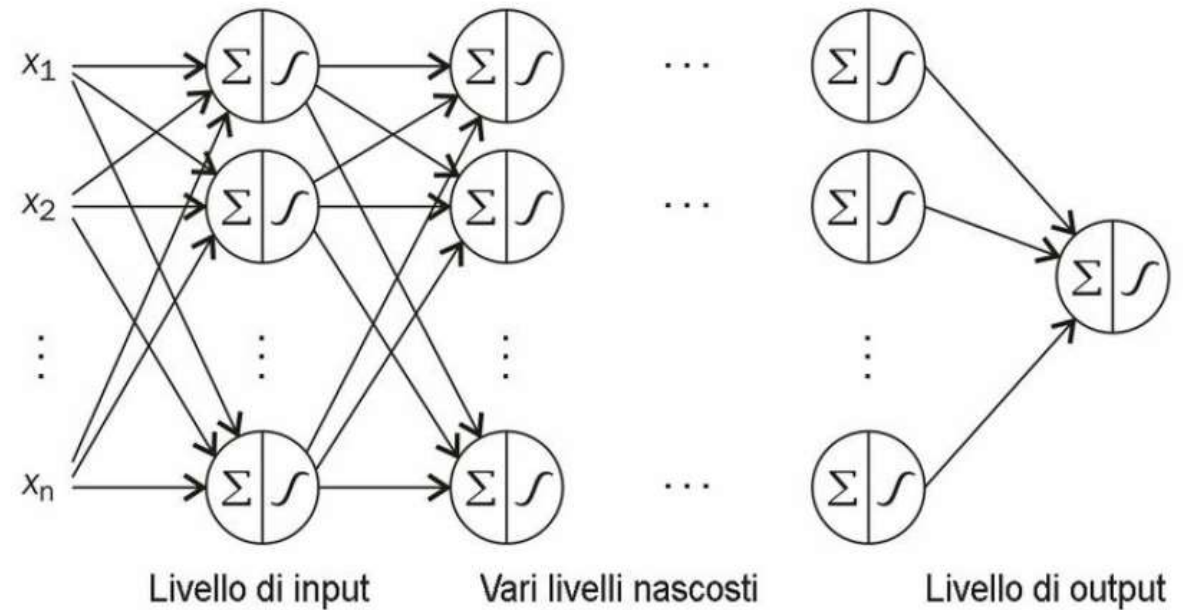
$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$



Reti neurali

La rete neurale MLP (Multilayer Perceptron):

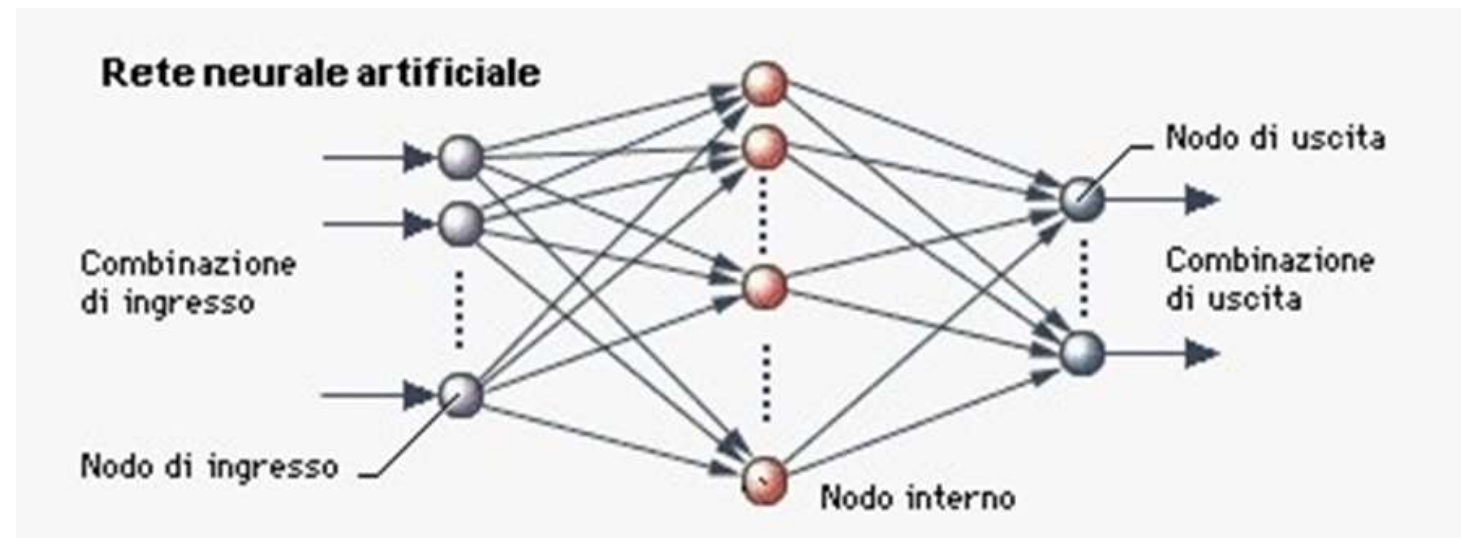
- ❑ Per creare una rete neurale, dobbiamo connettere fra loro più perceptron a formare una rete,
- *Un perceptron multilivello (MLP) è un grafo finito aciclico.*
- I nodi sono neuroni con funzione d'attivazione (in genere quella logistica)
- i collegamenti tra neuroni hanno un valore associato, detto peso sinaptico, che ha lo scopo di amplificare o ridurre l'importanza che un dato neurone ha all'interno della rete.



Reti neurali

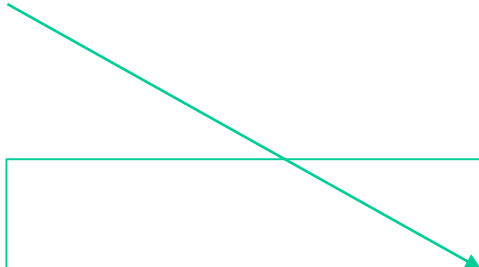
La rete neurale MLP (Multilayer Perceptron):

- ❑ Il layer di ingresso, formato dai neuroni di input, attraverso i quali sono forniti di dati.
- ❑ Uno o più layer intermedi (o nascosti) che eseguono elaborazioni dei dati.
- ❑ Un layer di output, che fornisce il risultato.



Reti neurali

- Ciascun neurone riceve in ingresso la somma pesata dei pesi sinaptici e dei valori di attivazione dei altri neuroni ad esso collegati.
- Il singolo neurone calcola il proprio valore di attivazione, trasmesso poi al livello successivo della rete, per mezzo di una funzione di attivazione


$$a_i = f\left(\sum_j w_{ij} \times a_j\right)$$

a_i = valore di attivazione del neurone i

f = funzione di attivazione

w_{ij} = peso sinaptico del j-esimo neurone collegato al neurone i

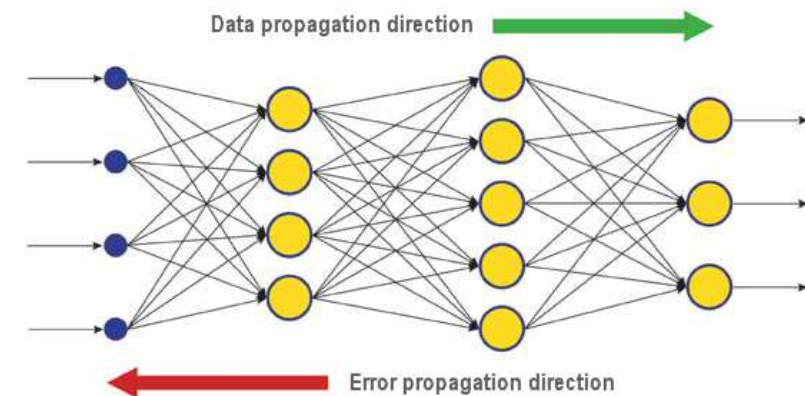
a_j = valore di output del neurone j-esimo

Reti neurali

- ❑ Mentre addestriamo il modello, aggiorniamo i pesi (inizialmente casuali) del modello in modo da ottenere la migliore previsione possibile.
- ❑ Se un'osservazione attraversa il modello e produce un output falso quando avrebbe dovuto essere vero, le funzioni logistiche dei singoli perceptron vengono leggermente modificate. Questa è chiamata propagazione all'indietro (*back-propagation*).
- ❑ Le reti neurali vengono normalmente addestrate in lotti: alla rete vengono forniti contemporaneamente vari punti dei dati di addestramento per più volte, e ogni volta l'algoritmo di back-propagation richiederà una modifica dei pesi interni della rete.
- Il processo è ripetuto finché l'errore totale sarà portato sotto a una soglia prestabilita

Reti neurali

1. Si utilizzano come valori di ingresso i dati del training set, si calcolano i valori di ciascun neurone per ogni layer fino ad ottenere il valore di output.
2. Si esegue il confronto tra il valore di output e il valore desiderato e si calcola l'errore totale.
3. Si esegue la back propagation calcolando i valori di delta da applicare ai pesi.
4. Si ripete il calcolo con i nuovi pesi e si ottiene il nuovo valore di output.
5. Si ripete il processo fino a portare l'errore al di sotto di una soglia prestabilita.



Reti neurali

- ❑ È facile immaginare che la rete può crescere molto in profondità e può avere molti livelli nascosti, che determinano la complessità della rete neurale. ????
- ❑ Quando le reti neurali crescono diventando molto profonde, entriamo nel campo dell'*apprendimento profondo (Deep Learning)*.
- ❑ Il grande vantaggio delle reti neurali profonde (reti formate da molti livelli) è il fatto che sono in grado di approssimare quasi ogni funzione e, teoricamente, possono apprendere le combinazioni ottimali di caratteristiche e poi usarle per ottenere il massimo potere predittivo possibile.
- ❑ Le reti neurali hanno però un grave difetto. Se lasciate operare, sviluppano un'elevatissima varianza:
 - basta rieseguire il modello e istanziare i pesi in modo differente per far sì che la rete si possa in modo molto differente
- ❑ Inoltre hanno bisogno di grandi capacità di calcolo

Reti neurali

Pregi e difetti

- ❑ Le reti neurali possono essere impiegate con dati soggetti a rumore o dove non esistono modelli analitici in grado di affrontare il problema
- ❑ I risultati ottenuti mediante le reti neurali sono efficienti ma possono richiedere una fase di training onerosa in termini di tempo di calcolo e di ampiezza del campione, soprattutto per trovare relazioni complesse tra i dati.
- ❑ Per modellare problemi complessi è possibile aggiungere layer di neuroni, teoricamente sono in grado di approssimare quasi ogni funzione e possono apprendere le combinazioni ottimali di caratteristiche
 - ma solo fino a un certo punto!
- ❑ Le reti neurali sono una black-box machine: non è possibile estrarre in modo semplice le regole di apprendimento che portano ad un determinato risultato di classificazione o regressione

Reti neurali

- ❑ *Il problema del vanishing gradient problem*
- ❑ con il meccanismo di back propagation, l'errore è propagato all'indietro in ciascun layer, uno alla volta;
- ❑ i pesi sono aggiustati in modo da ottimizzare l'errore dell'intera rete;
- ❑ se vi sono molti layer nascosti, man mano che si indietreggia l'errore scompare gradualmente, lasciando invariati i pesi dei layer più vicini all'input;
- Questo fa venir meno i vantaggi che si potrebbero avere aggiungendo layer a una rete Multi Layer Perceptron

Reti neurali

Deep Learning

- ❑ Quando le reti neurali crescono diventando molto profonde, entriamo nel campo dell'apprendimento profondo (Deep Learning);
 - Questi algoritmi cercano di affrontare il problema del vanishing gradient problem
- ❑ Il grande vantaggio delle reti neurali profonde è il fatto che sono in grado di approssimare quasi ogni funzione e, teoricamente, possono apprendere le combinazioni ottimali di caratteristiche e poi usarle per ottenere il massimo potere predittivo possibile.
- ❑ Hanno però un grave difetto.
 - Se lasciate operare, sviluppano un'elevatissima varianza: basta rieseguire il modello e istanziare i pesi in modo differente per far sì che la rete si possa in modo molto differente
- ❑ Inoltre hanno bisogno di grandi capacità di calcolo

Reti neurali

- ❑ ***Deep Learning: Strutture di base, gli auto-encoders***
- ❑ cioè una rete neurale feed-forward (cioè dove i dati fluiscono in una sola direzione, dall'input verso l'output), che ha lo scopo di apprendere una rappresentazione compressa del dataset di input.
- ❑ In questi algoritmi lo strato di input e lo strato di output sono uguali, visto che rappresentano gli stessi dati.
- ❑ I layer interni acquisiscono, con il training, la struttura dei dati e le feature.
- ❑ Il numero di neuroni nello strato nascosto è inferiore a quello dei neuroni di input e output, ***consentendo una rappresentazione compatta e compressa dei dati.***

Reti neurali

- ❑ ***Deep Learning: Strutture di base, gli auto-encoders***
- ❑ cioè una rete neurale feed-forward (cioè dove i dati fluiscono in una sola direzione, dall'input verso l'output), che ha lo scopo di apprendere una rappresentazione compressa del dataset di input.
- ❑ In questi algoritmi lo strato di input e lo strato di output sono uguali, visto che rappresentano gli stessi dati.
- ❑ I layer interni acquisiscono, con il training, la struttura dei dati e le feature.
- ❑ Il numero di neuroni nello strato nascosto è inferiore a quello dei neuroni di input e output, ***consentendo una rappresentazione compatta e compressa dei dati.***

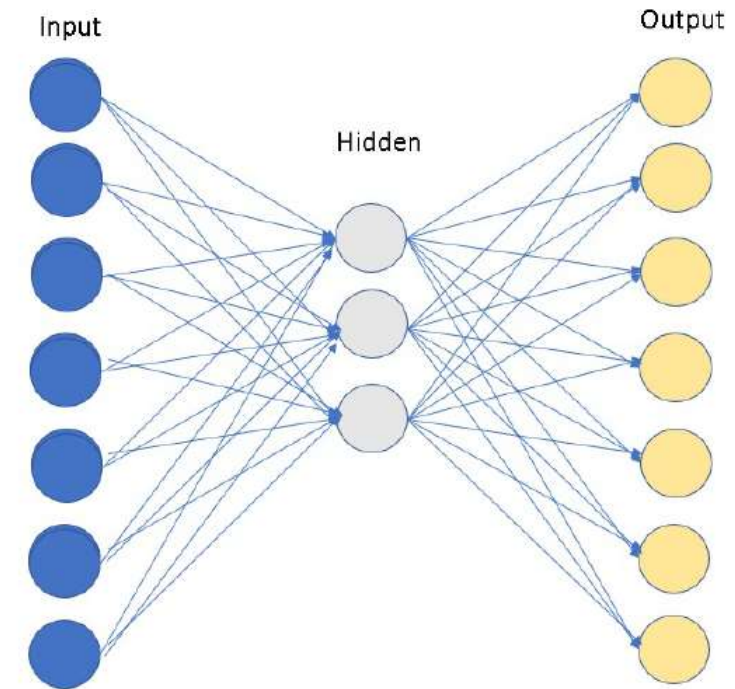
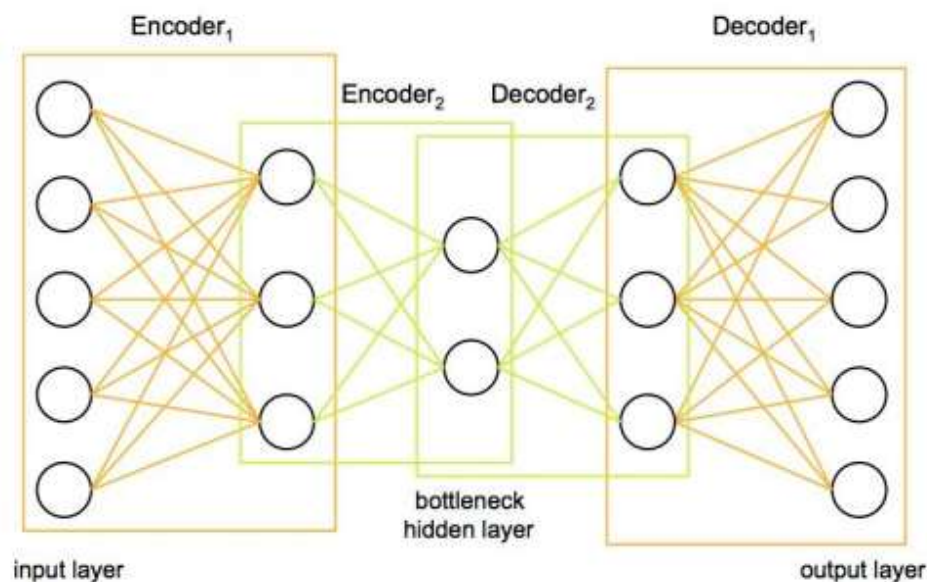


Figura 13.21: Schema di un auto encoder.

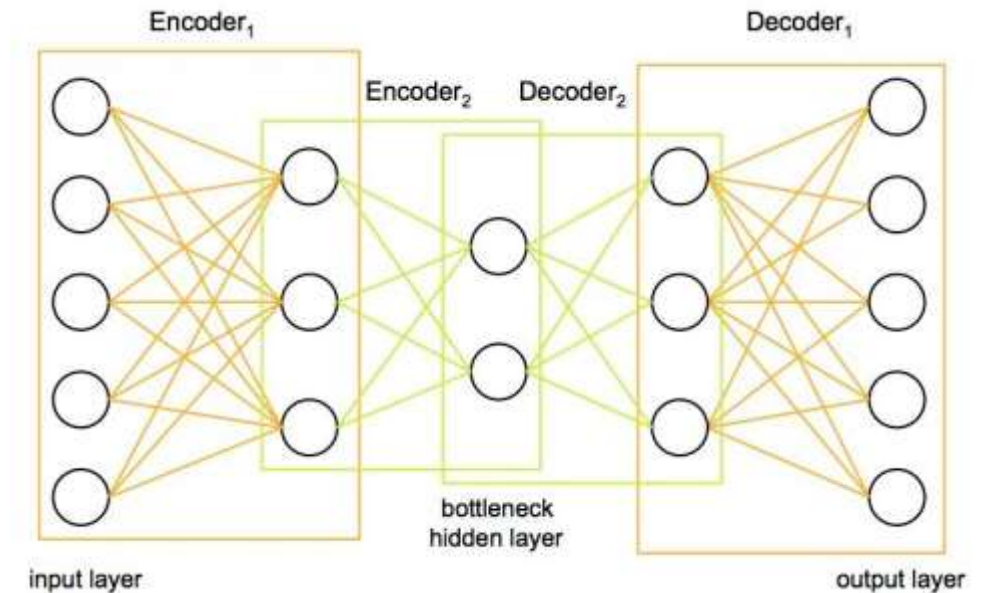
Reti neurali

- ❑ **Deep Learning: Stacked auto-encoders**
- ❑ Le strutture di base possono essere messe in serie una dopo l'altra creando delle reti complesse.
- ❑ È dimostrato che, grazie alla modalità con cui avviene il training di ogni layer (cioè in modo indipendente dagli altri layer), non si producono gli effetti negativi della back propagation, ovvero il vanishing gradient



Reti neurali

- ❑ **Deep Learning: Stacked auto-encoders**
 - il funzionamento è il seguente:
 1. Un primo auto-encoder utilizza i dati di input per il training. L'autoencoder è costituito da un layer di input (che rappresenta i dati originali, un hidden layer H1 e un output layer).
 2. Il secondo auto-encoder utilizza come layer di input l'hidden layer H1 del primo auto-encoder. In pratica l'output layer è utilizzato per il training, ma è sistematicamente ignorato nei passaggi successivi.
 3. Eventuali successivi auto-encoder lavorano allo stesso modo.

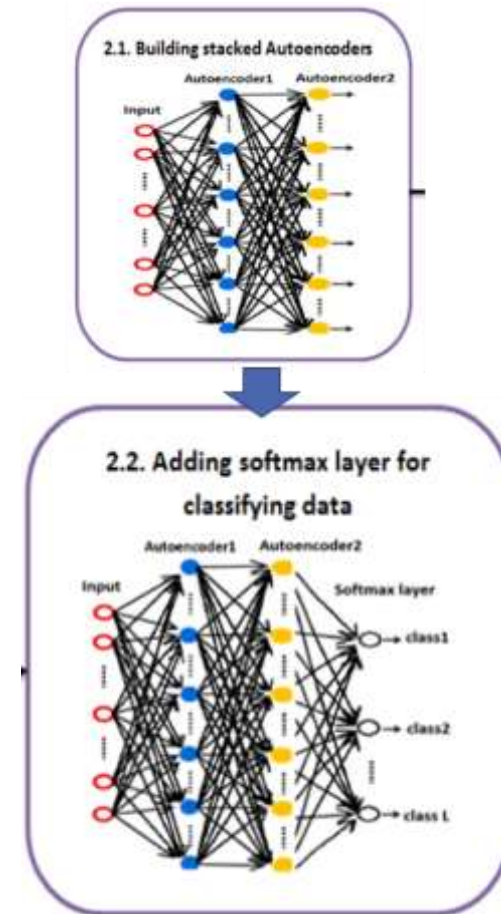


Quindi il training, cioè la determinazione dei pesi, avviene in modo indipendente per ciascun layer (layer-wise training).

Reti neurali

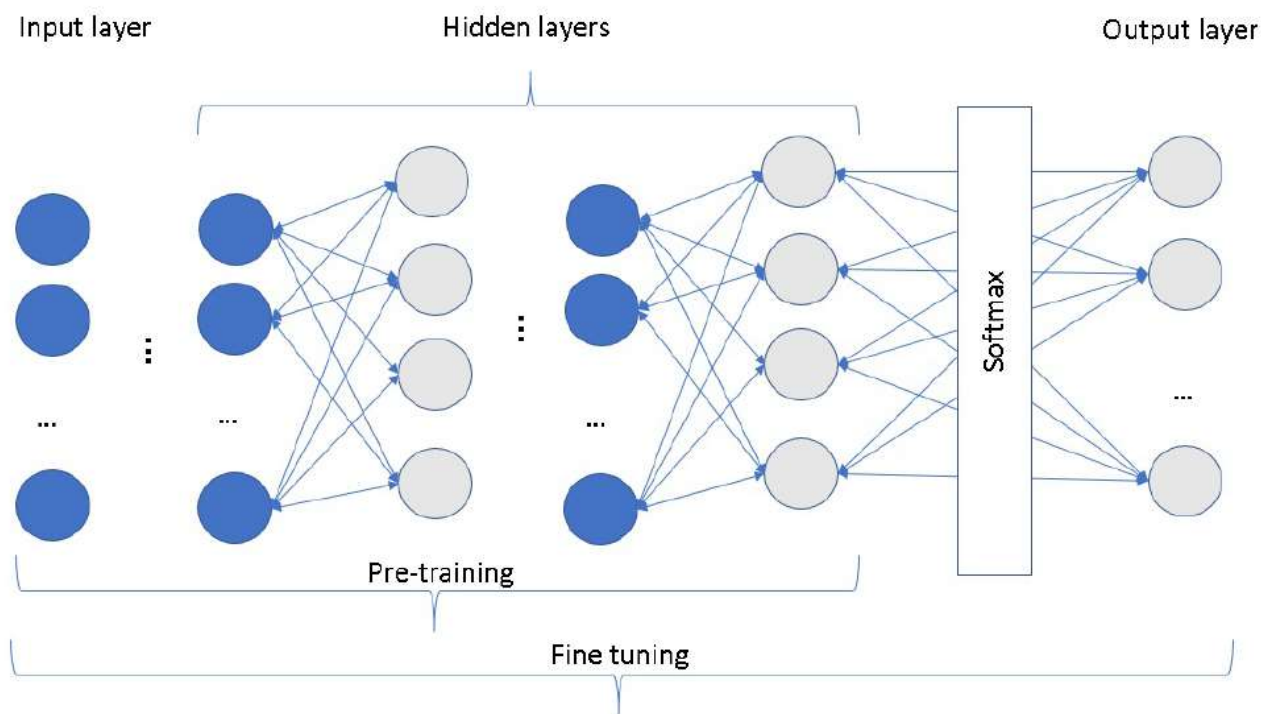
- ❑ **Deep Learning: Stacked auto-encoders**
- ❑ Arrivati a questo punto non vi è ancora alcuna connessione tra i dati di input e l'output finale (cioè le classi, per un problema di classificazione).
- ❑ Il passaggio finale che consente di realizzare la classificazione, consiste nell'aggiunta di uno o più layer *fully-connected* (ovvero dove tutti i nodi di input sono connessi a tutti i nodi di output).
- ❑ Il layer potrebbe essere costituito da :
 1. un classificatore classico come un random forest
 2. Oppure da una funzione softmax, che è una generalizzazione della funzione logistica in grado di schiacciare i numeri reali contenuti in un vettore x di K dimensioni in valori reali compresi tra 0 e 1

$$\text{softmax}(v)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}}, \text{ per } j = 1, \dots, K$$



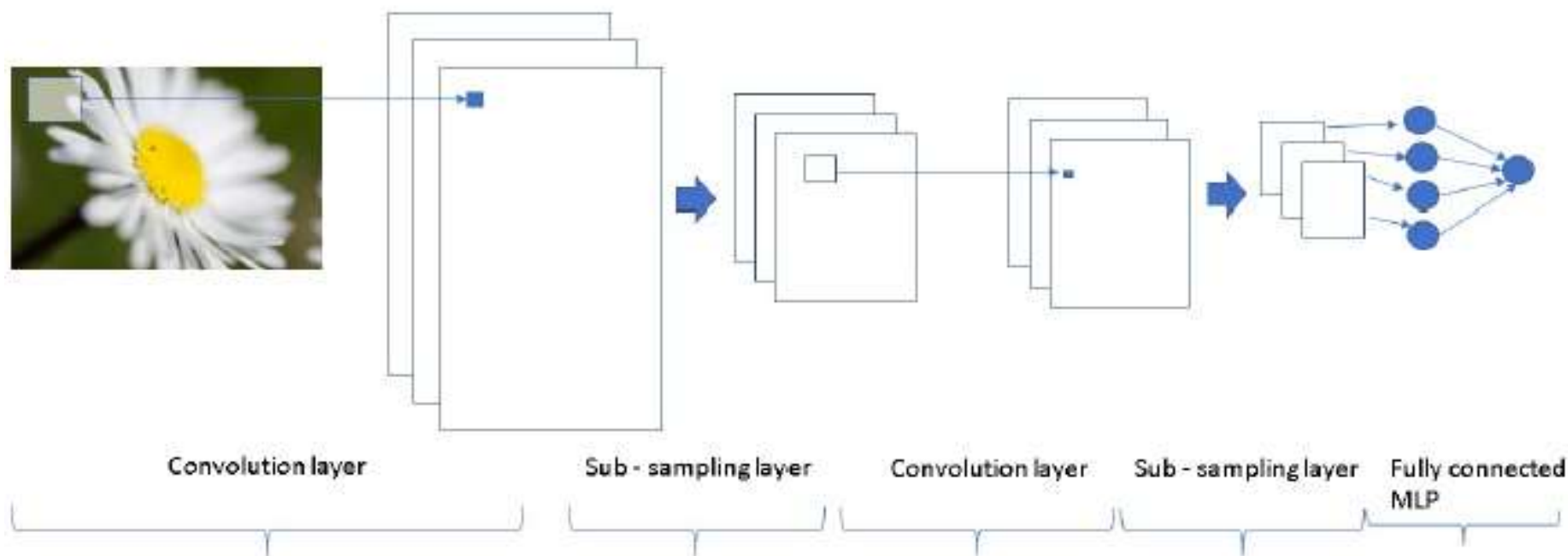
Reti neurali

- **Deep Learning:** Il passaggio finale con cui si aggiunge il classificatore e si utilizza la back propagation è detto ***fine-tuning***.



Reti neurali

- ❑ *Deep Learning: Convolutional Neural Networks (CNN)*
- sono utilizzate prevalentemente per il riconoscimento di immagini



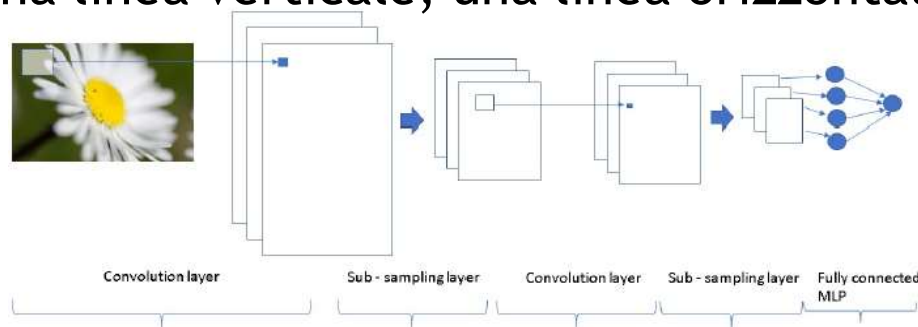
Reti neurali

Deep Learning: Convolutional Neural Networks (CNN)

La CNN, invece di esaminare tutta l'immagine, elabora un piccolo quadrato di pixel alla volta. Tale quadrato è via via spostato sull'immagine, con un certo passo, in modo da elaborarla tutta.

- A ciascuna porzione di immagine è applicata un filtro (o kernel) costituito da una matrice quadrata della stessa dimensione della porzione di pixel.
- Viene calcolato il prodotto scalare tra le matrici (per ciascun canale RGB)
- Gli scalari ottenuti formano 3 matrici (una per ogni canale RGB) chiamate activation map.
- Le tre matrici sono sommate.

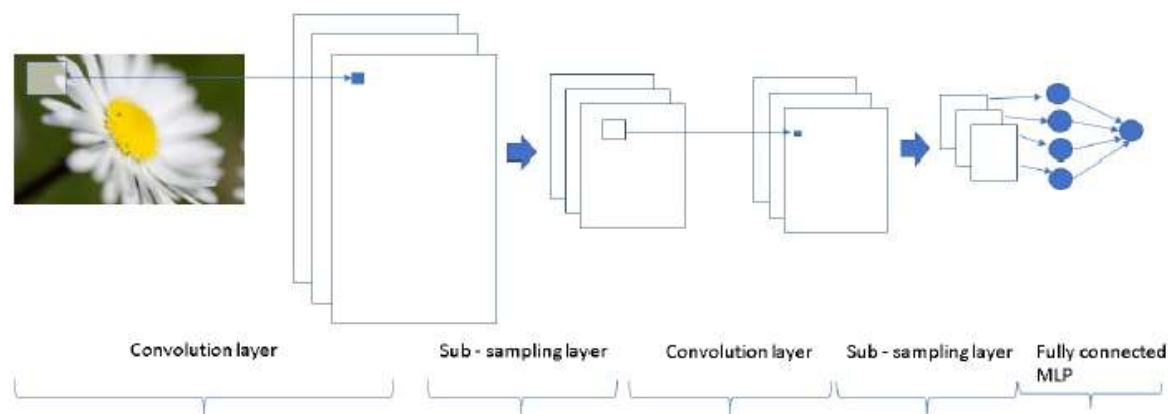
Il processo è ripetuto per ogni filtro che si vuol applicare all'immagine. Ogni filtro consente la ricerca di un pattern di pixel diverso (per esempio una linea verticale, una linea orizzontale, una linea obliqua, ecc.)



Reti neurali

Deep Learning: Convolutional Neural Networks (CNN)

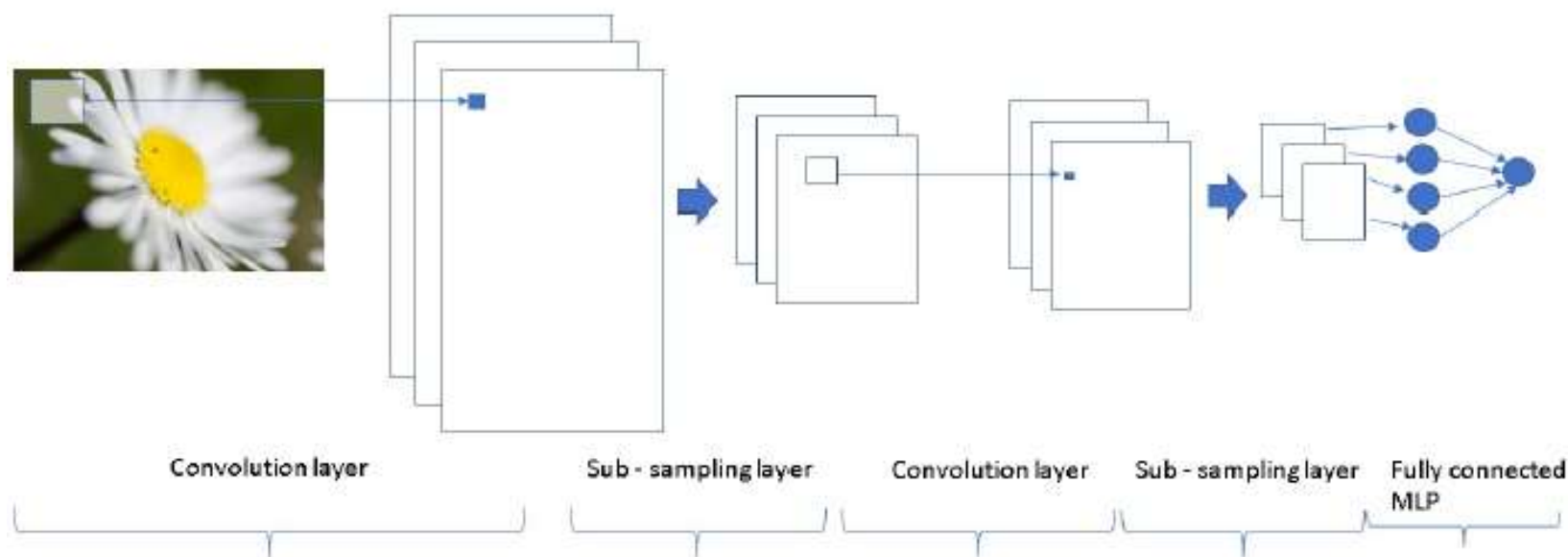
- Le immagini possono avere una risoluzione molto alta e per questo possono dar luogo a matrici molto grandi.
- Per ridurre la dimensione delle matrici è utilizzata una tecnica chiamata max pooling, che consiste nel suddividere una matrice in quadranti (per esempio 2 per 2) e calcolare il massimo dei valori contenuti in tale quadrante.
- Il massimo è mantenendo come output e sostituisce interamente il corrispondente quadrante.



Reti neurali

Deep Learning: Convolutional Neural Networks (CNN)

- La CNN è costituita da layer che effettuano le convoluzioni, intervallati da layer che effettuano il sub-sampling (cioè la riduzione della dimensione) e da un layer finale, fully connected che funge da classificatore.



IL TEST E LA VALUTAZIONE DEI MODELLI PREDITTIVI

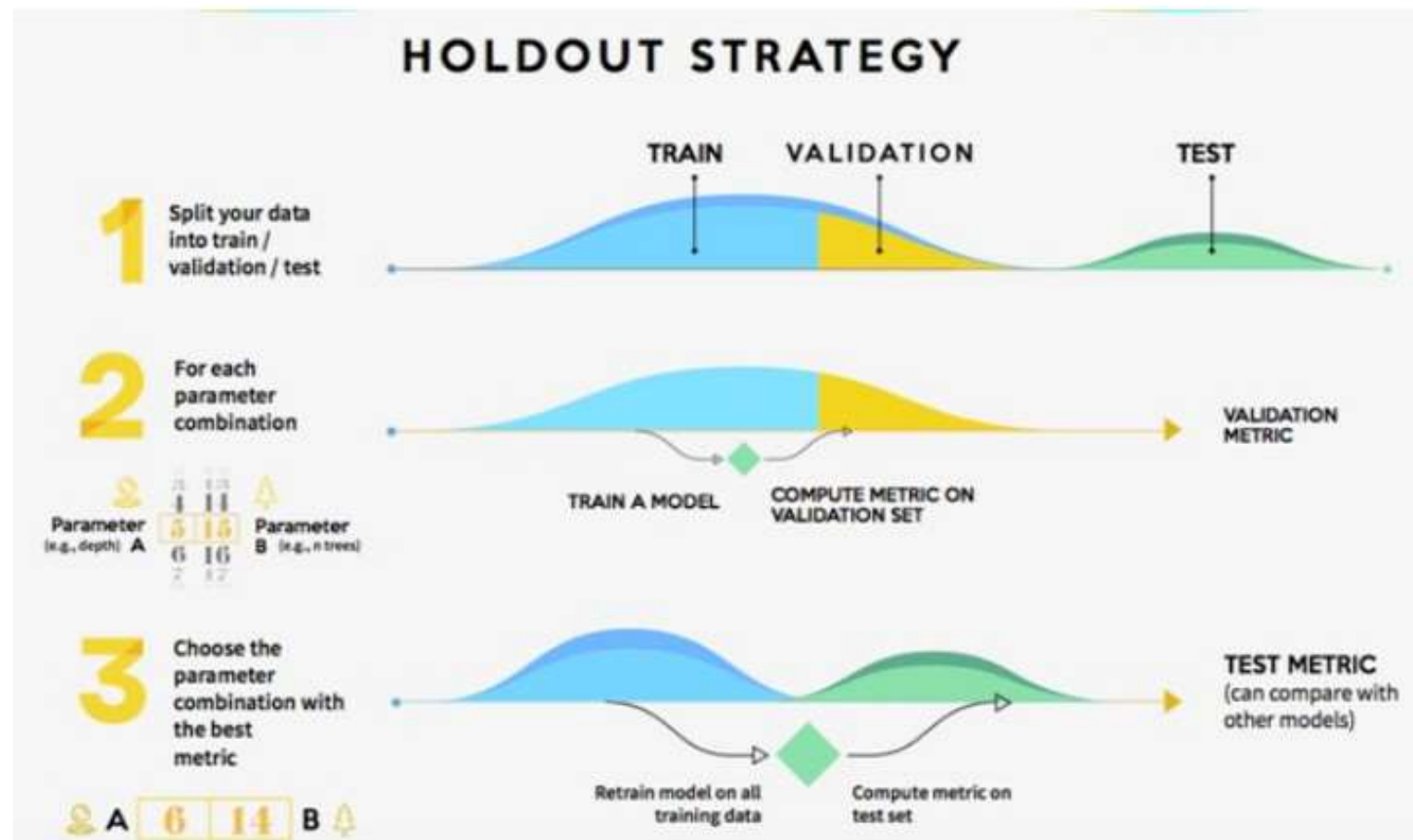
Il test e la valutazione dei modelli predittivi

Le modalità di valutazione sono diverse a seconda del tipo di algoritmo

- ❑ i modelli di classificazione,
- ❑ i modelli di regressione
- ❑ i modelli di clustering.

Valutazione dei modelli di classificazione

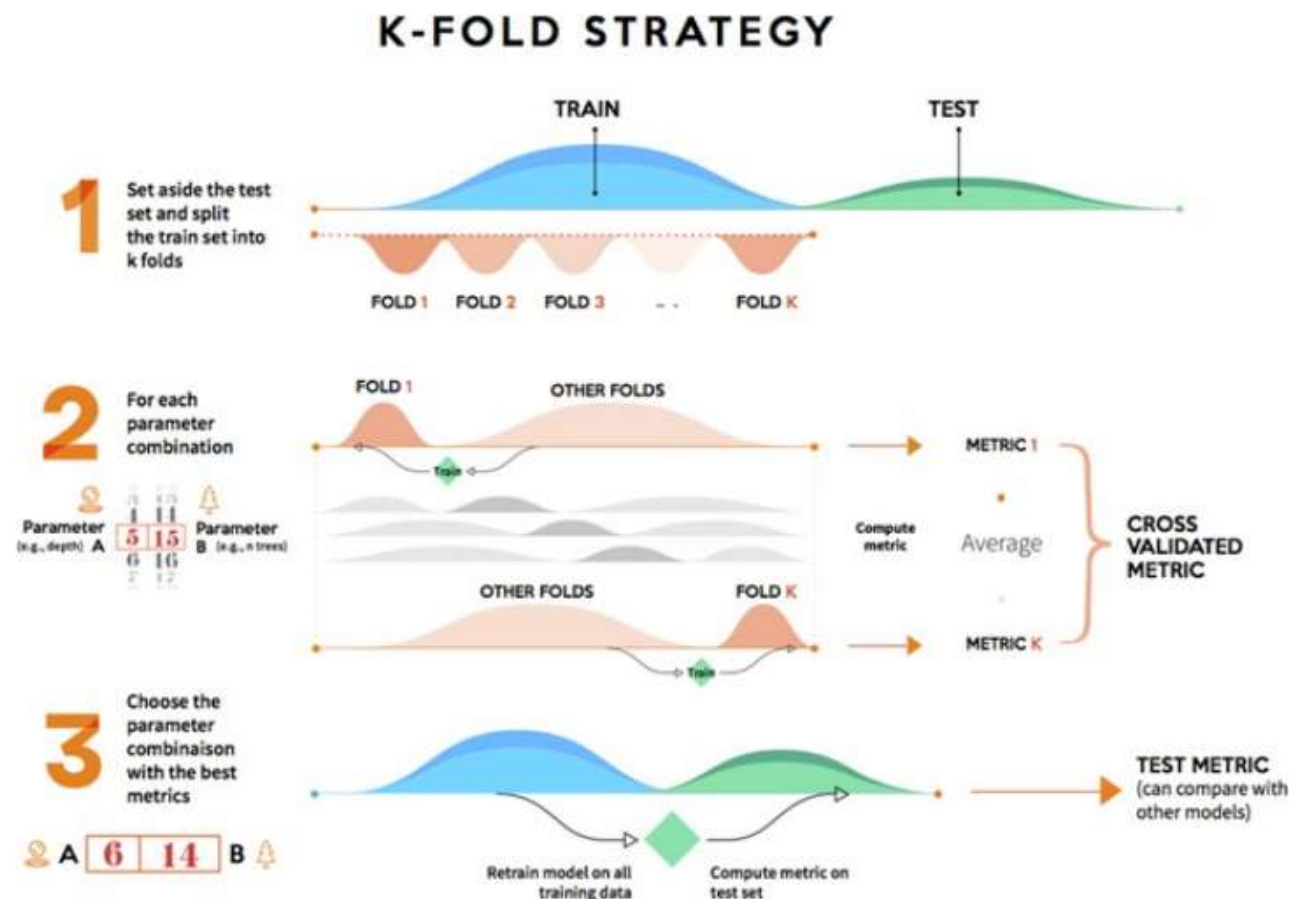
Hold-out e cross validation



Valutazione dei modelli di classificazione

Hold-out e cross validation

- Si prende un numero finito k di parti (fold) di uguali dimensioni.
- Per ogni subset si considera $k-1$ delle parti come set di addestramento e la parte rimanente come set di collaudo.
- Quindi si calcoliamo una determinata metrica per ogni parte. Alla fine si calcola o la media dei risultati



Valutazione dei modelli di classificazione

Cross validation

- È una stima più accurata dell'errore di previsione fuori campione rispetto a una singola suddivisione addestramento-collauda, perché svolge più suddivisioni addestramento-collauda indipendenti e poi calcola la media di tutti i risultati.
- Fa un uso molto più efficiente dei dati rispetto a una singola suddivisione addestramento-collauda, perché l'intero dataset viene usato per svolgere non una sola, ma più suddivisioni addestramento collauda.
- Ogni record del nostro dataset viene usato sia per l'addestramento sia per il collauda.
- Questo metodo costringe a valutare un compromesso fra efficienza e costo computazionale. Una convalida incrociata 10-fold è dieci volte più costosa dal punto di vista computazionale rispetto a una singola suddivisione addestramento-collauda.
- Questo metodo può essere usato per l'ottimizzazione dei parametri e la scelta del modello.

Valutazione dei modelli di classificazione

Cross Validation e matrice di confusione

- Dalla cross validation scaturiscono metriche di valutazione che consentono di effettuare la scelta dell'algoritmo o della parametrizzazione migliore
- La più utilizzata è la **matrice di confusione** che altro non è che una cross tabulazione delle classi reali e delle classi predette.

*Caso binario:
2 sole classi
Es : (positivo/negativo)*

		Prediction		
		1	0	
Classe reale	1	True Positive (TP)	False Negative (FN)	Totale classe positiva $P = TP + FN$
	0	False Positive (FP)	True Negative (TN)	Totale classe negativa $N = FP + TN$

Figura 14.2: Matrice di confusione.

Valutazione dei modelli di classificazione

Matrice di confusione

- Il quadrante (1,1), che indica quanti elementi della classe positiva sono stati correttamente individuati dall'algoritmo (veri positivi o True Positive o TP).
- Il quadrante (0,0), che indica quanti elementi della classe negativa sono stati correttamente individuati dall'algoritmo (veri negativi o True Negative o TN).
- Il quadrante (0,1), che indica quanti elementi della classe negativa reale sono stati posti dall'algoritmo nella classe positiva (falsi positivi o False Positive o FP).
- Il quadrante (1,0), che indica quanti elementi della classe positiva reale sono stati posti dall'algoritmo nella classe negativa (falsi negativi o False Negative o FN)

		Prediction		
		1	0	
Classe reale	1	True Positive (TP)	False Negative (FN)	Totale classe positiva $P = TP + FN$
	0	False Positive (FP)	True Negative (TN)	Totale classe negativa $N = FP + TN$

Figura 14.2: Matrice di confusione.

Valutazione dei modelli di classificazione

Matrice di confusione e metriche di valutazione del modello

- **Accuracy:** $\frac{(TP + TN)}{(P + N)}$
- **Precision (o Positive Predictive Value):** $\frac{TP}{(TP + FP)}$
- **Sensitivity (o Recall o True Positive Rate)** $\frac{TP}{(TP + FN)} = \frac{TP}{P}$
- **Specificity (o True Negative Rate)** $\frac{TN}{(TN + FP)} = \frac{TN}{N}$

Valutazione dei modelli di classificazione

Matrice di confusione e metriche di valutazione del modello

Attenzione

- L'accuracy è una misura che potrebbe essere fuorviante.
- Es.: se avessimo 100.000 istanze di volti non volti e solo 10 fossero *volti* (classe positiva), un modello che classificasse tutti i casi come non *volti* (classe negativa) avrebbe un'accuracy di $(0 + 99990)/100000 = 99.99\%$.
- Il modello valutato con questa metrica risulta quasi perfetto.
- Tuttavia non coglie ciò che davvero interessa, cioè i volti.
Utilizzando come metrica la *Sensitivity* (o *Recall*) avremmo un valore pari a $0/10 = 0\%$.

Valutazione dei modelli di classificazione

Matrice di confusione e metriche di valutazione del modello

Per ottimizzare l'algoritmo è bene impiegare la metrica più adeguata al problema e cioè:

- l'**accuracy** quando desideriamo che la maggior parte degli elementi siano correttamente classificati, indipendentemente dalla produzione di falsi positivi o falsi negativi.
- la **sensitivity** quando vogliamo massimizzare i True Positive senza però far crescere troppo i False Positive.
 - Vi sono dei casi in cui vi può essere un costo molto elevato collegato ai falsi positivi, perciò i modelli che riescono a predire molti veri positivi, ma nel farlo introducono nel risultato molti falsi positivi, avranno una valutazione bassa in termini di sensitivity.
- la **precision** quando vogliamo massimizzare i veri positivi e minimizzare i falsi negativi.
 - Siamo nella situazione in cui è prioritario classificare correttamente i veri positivi, anche al costo di creare un elevato numero di falsi positivi. Questo perché il costo dei falsi positivi è basso, mentre il costo dei falsi negativi è molto alto.
- la **specificity** quando occorre massimizzare il numero di veri negativi.

Valutazione dei modelli di classificazione

Matrice di confusione con classificazione non binaria

Esempio di matrice di confusione

		Predetti			Somma
		Gatto	Cane	Coniglio	
Reali	Gatto	5	2	0	7
	Cane	3	3	2	8
	Coniglio	0	1	11	12
Somma		8	6	13	27

- Si può notare che dei 7 gatti reali, il sistema ne ha classificati 2 come cani.
- Allo stesso modo si può notare come dei 12 conigli veri, solamente 1 è stato classificato erroneamente.
- Gli oggetti che sono stati classificati correttamente sono indicati sulla diagonale della matrice, per questo è immediato osservare dalla matrice se il classificatore ha commesso o no degli errori.

Esempio tratto da wikipedia

Valutazione dei modelli di classificazione

Matrice di confusione e F-measure

- **F-measure** è una misura dell'accuratezza di un test.
 - Si deriva dalla matrice di confusione.
- La misura tiene in considerazione precisione e recupero del test, dove la precisione è il numero di veri positivi diviso il numero di tutti i risultati positivi, mentre il recupero è il numero di veri positivi diviso il numero di tutti i test che sarebbero dovuti risultare positivi (ovvero veri positivi più falsi negativi).

$$F\ measure = \frac{2 \cdot sensitivity \cdot precision}{sensitivity + precision} = \frac{2|TP|}{2|TP| + |FP| + |FN|}$$

Valutazione dei modelli di classificazione

La curva ROC (Receiver Operating Characteristic)

- uno strumento messo a punto durante la seconda guerra mondiale dagli ingegneri che si occupavano dei radar per cercare di distinguere i segnali relativi a oggetti nemici dai segnali causati da stormi di uccelli.
- Per il calcolo delle curve ROC occorre che il modello produca come output, oltre alla previsione anche la probabilità di appartenenza alla classe positiva.
- Questo perché la curva ROC mostra tutti i possibili valori di falsi positivi e veri positivi che è possibile ottenere da un modello variando la soglia di probabilità che determina l'appartenenza alla classe positiva. Normalmente tale soglia è 0.5 e indica che se un elemento ha probabilità di appartenenza alla classe positiva minore o uguale allo 0.5, esso è classificato nella classe negativa.

Valutazione dei modelli di classificazione

La curva ROC (Receiver Operating Characteristic)

- Il **punto (0,0)** rappresenta una classificazione in cui non vi sono falsi positivi, ma nemmeno veri positivi.
- Il **punto (0,100)** indica una classificazione perfetta: 0 falsi positivi e 100% veri positivi.
- Il **punto (100,100)** è il risultato di una strategia in cui tutti gli elementi sono classificati come veri positivi: così facendo il tasso di falsi positivi è massimo.

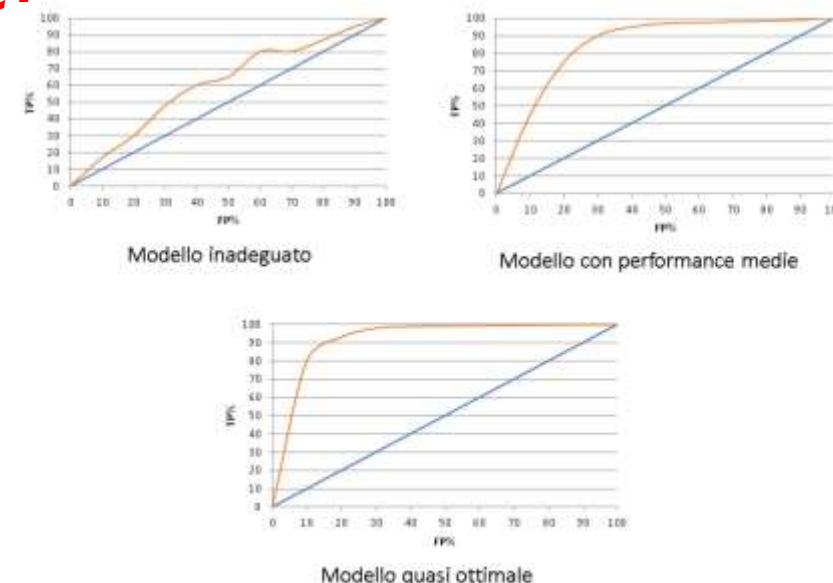


Figura 14.6: Esempi di tre curve ROC che descrivono modelli con performance differenti.

La retta diagonale che unisce i punti (0,0) e (100,100) rappresenta un classificatore completamente casuale: i nostri modelli dovranno per lo meno presentare una curva che stia sopra a quella del classificatore casuale.

Valutazione dei modelli di clustering

- La misura più comunemente utilizzata è lo **scarto quadratico medio** (SSE - Sum of Squared Error)
 - Per ogni punto l'errore è la distanza dal centroide del cluster a cui esso è assegnato

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

- x è un punto appartenente al cluster C_i e m_i è il rappresentante del cluster C_i
 - è possibile dimostrare che il centroide che minimizza SSE quando si utilizza come misura di prossimità la distanza euclidea è la media dei punti del cluster.

$$m_i = \sum_{x \in C_i} x$$

- Ovviamente il valore di SSE si riduce incrementando il numero dei cluster K
 - Un buon clustering con K ridotto può avere un valore di SSE più basso di un cattivo clustering con K più elevato