

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.*

Classificazione degli algoritmi per scopo

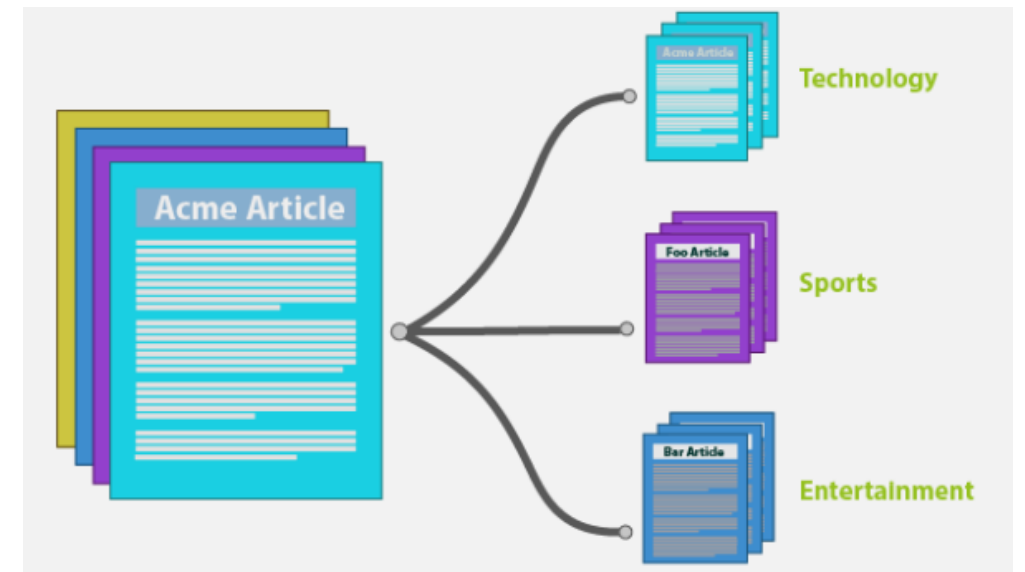
Permette di identificare quali algoritmi siano adatti ad un particolare problema predittivo

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

Classificazione per scopo

□ Classificazione

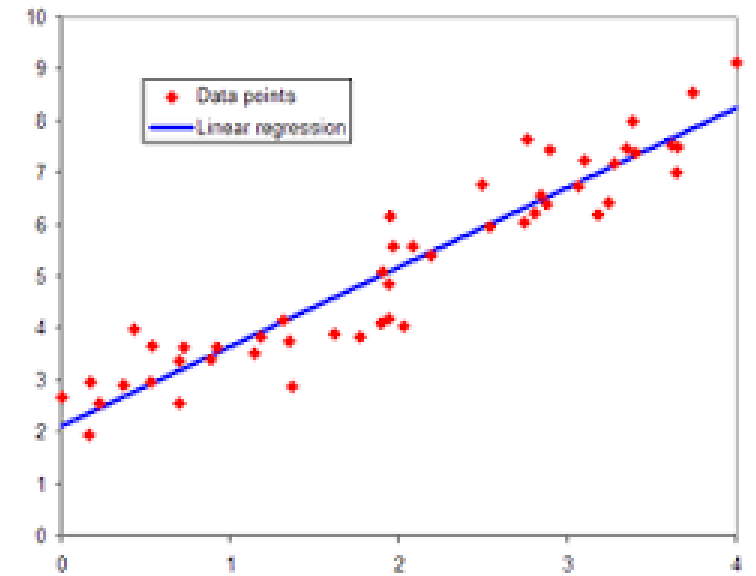
- Individuazione dell'appartenenza di un elemento ad una classe.
- L'output della classificazione è categorico e quindi può assumere un numero finito di possibili valori
- Agli algoritmi di classificazione appartengono i classificatori lineari (logistic regression, Naive Bayes, Perceptron, Support Vector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



Classificazione per scopo

□ Regressione

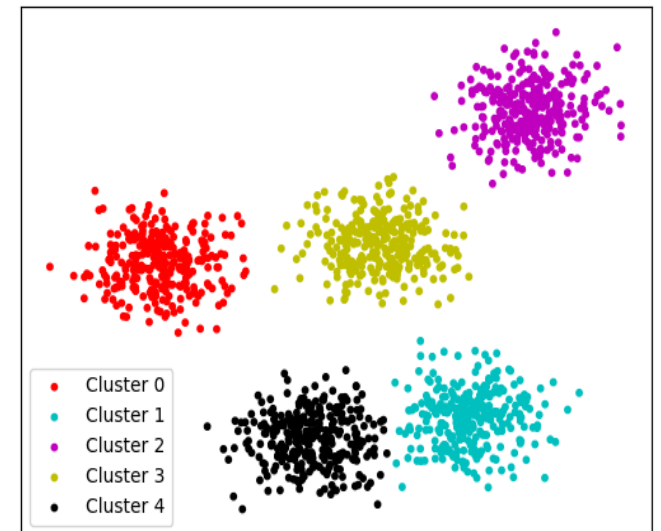
- L'output del modello è un valore numerico, che si vuol approssimare tramite una funzione dei dati di input.
- La variabile di output è continua e può assumere un numero infinito di valori
 - Es.: previsione del livello delle vendite in un periodo temporale futuro,
- Algoritmi che appartengono a questa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



Classificazione per scopo

□ Clustering

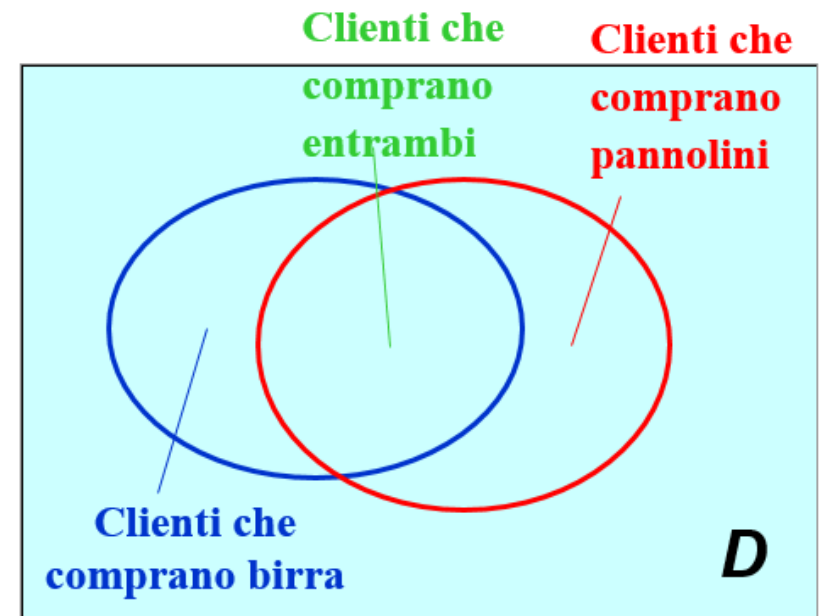
- il raggruppamento degli elementi di un dataset in gruppi (i cluster) utilizzando soltanto le informazioni contenute nei dati di input.
- Al contrario rispetto a quanto avviene nella classificazione, l'output del clustering (ovvero la categorizzazione) non è noto a priori.
- I raggruppamenti sono realizzati 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.

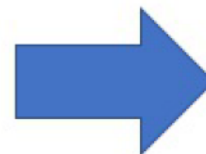


Classificazione per scopo

■ 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.

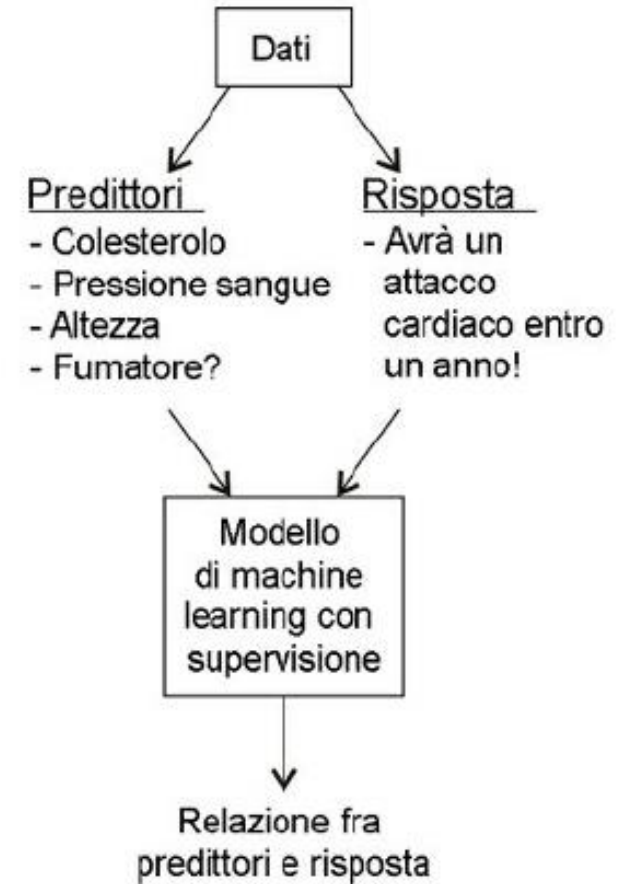
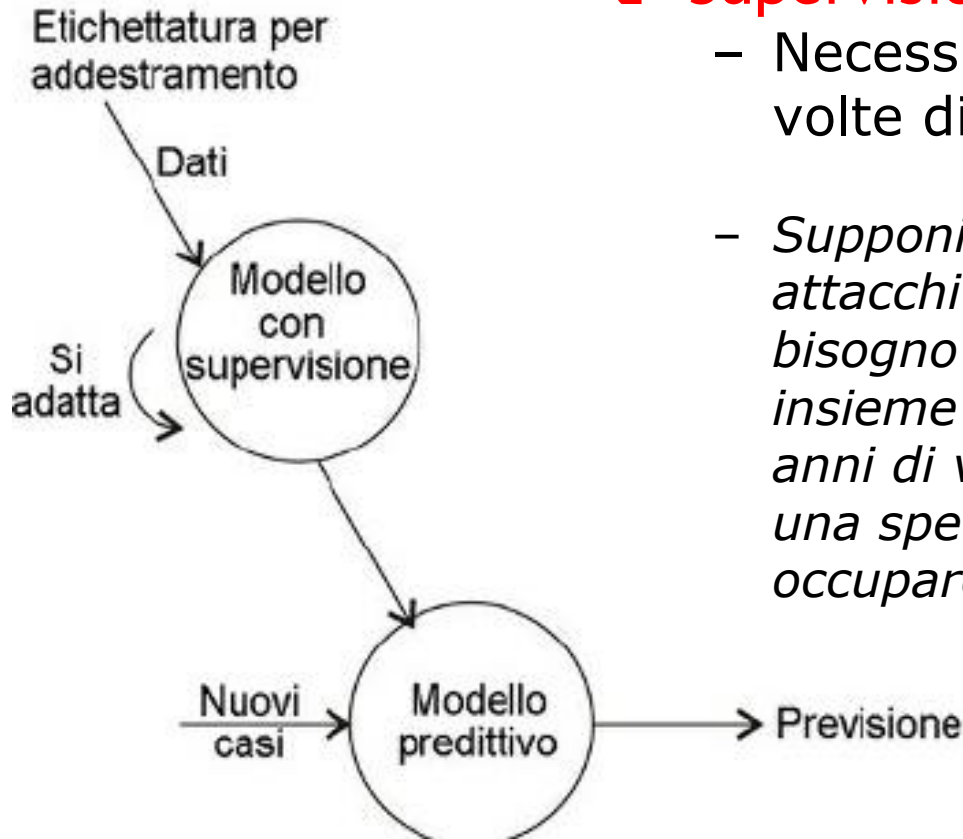
Si devono separare i dati in due parti:

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

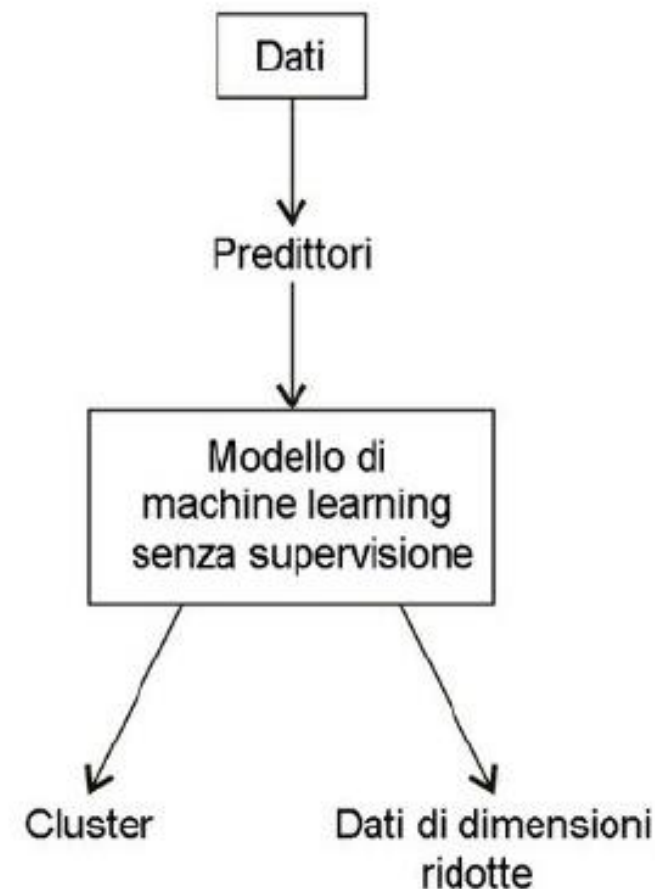
- Necessità di dati etichettati a volte difficili da reperire
- *Supponiamo di voler prevedere gli attacchi cardiaci: potremmo aver bisogno di migliaia di pazienti, insieme alle loro cartelle cliniche di anni di visite mediche e ottenerle è una specie di incubo per chi si deve occupare della raccolta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

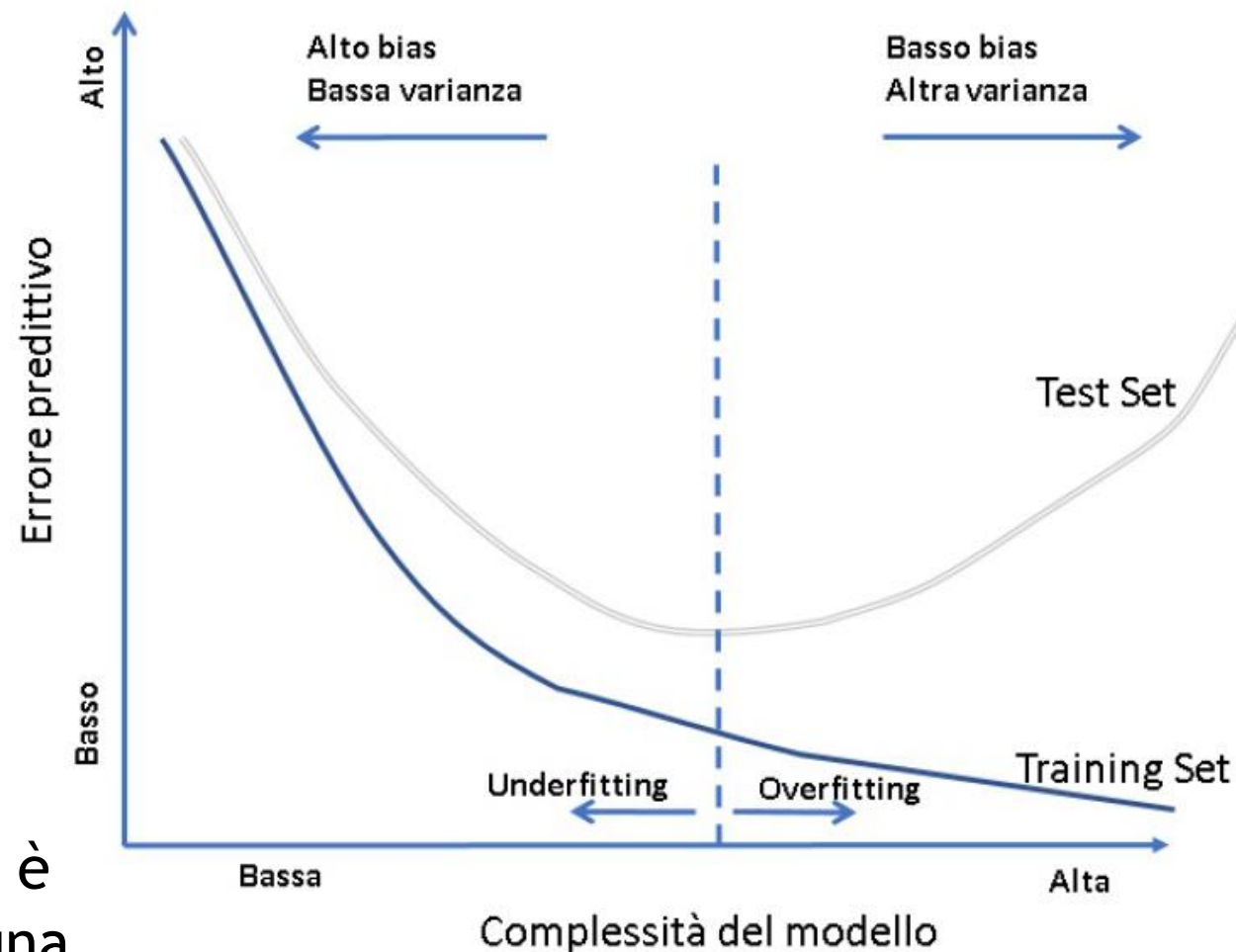
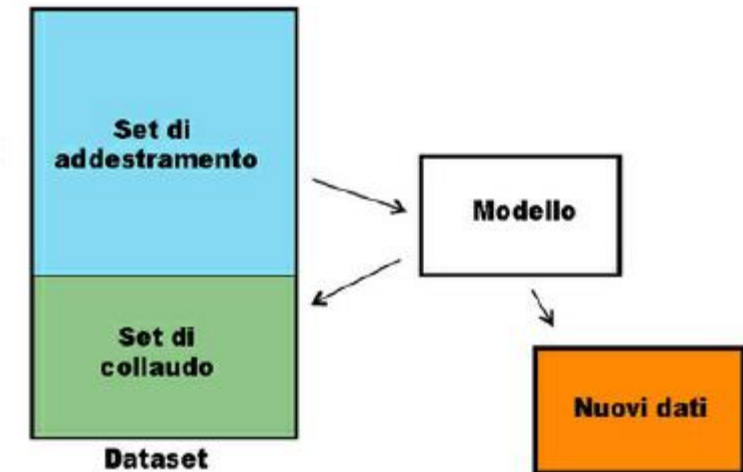


Figura 13.2: Grafico Complessità/Errore

Problematiche comuni

- **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

(*ALCUNI...*)

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
- 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”.

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

✓ *Alberi decisionali (Decision Trees)*

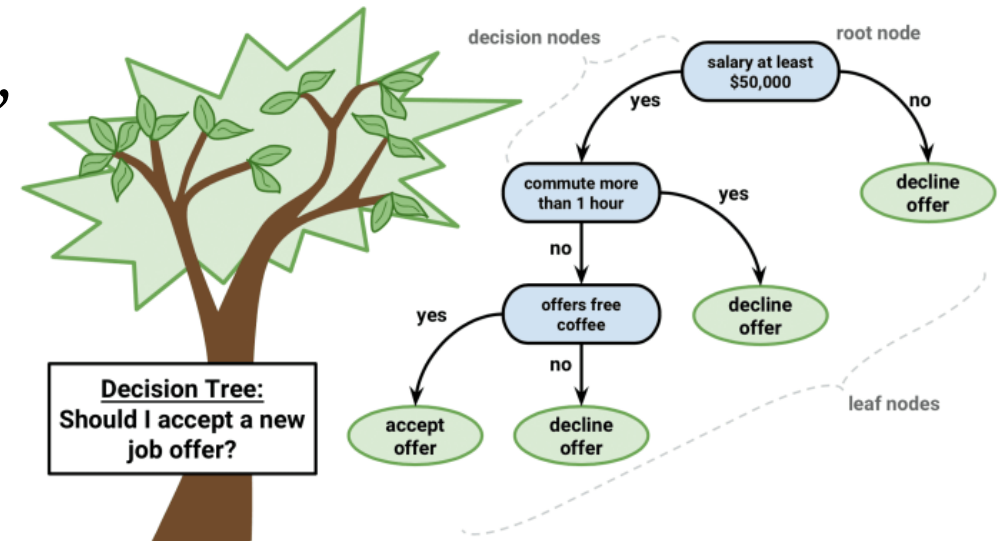
- ❑ Sono uno dei metodi di calcolo principalmente utilizzati nel data mining.
- in particolare per determinare a quale categoria appartiene un elemento, in base al valore dei suoi attributi noti.
- ❑ La tecnica su cui si basano è flessibile e permette di adattarli a numerose situazioni; inoltre il loro output è molto chiaro, dato che è rappresentato (anche visivamente) sotto forma di albero .

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

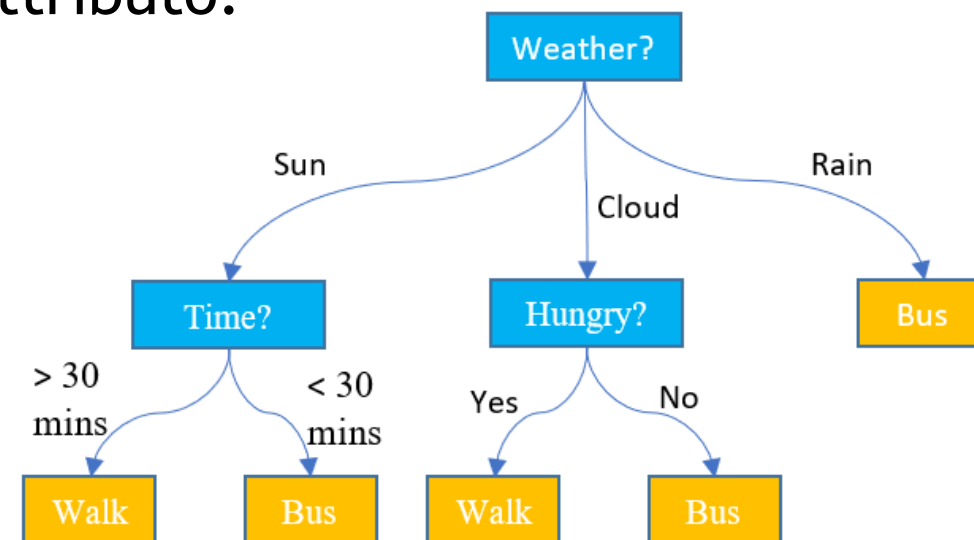
✓ Alberi decisionali (Decision Trees)

- Ogni nodo interno rappresenta un test su un attributo:

- Weather
- Time
- Hungry

- Ogni nodo foglia rappresenta una classe

- Walk
- Bus

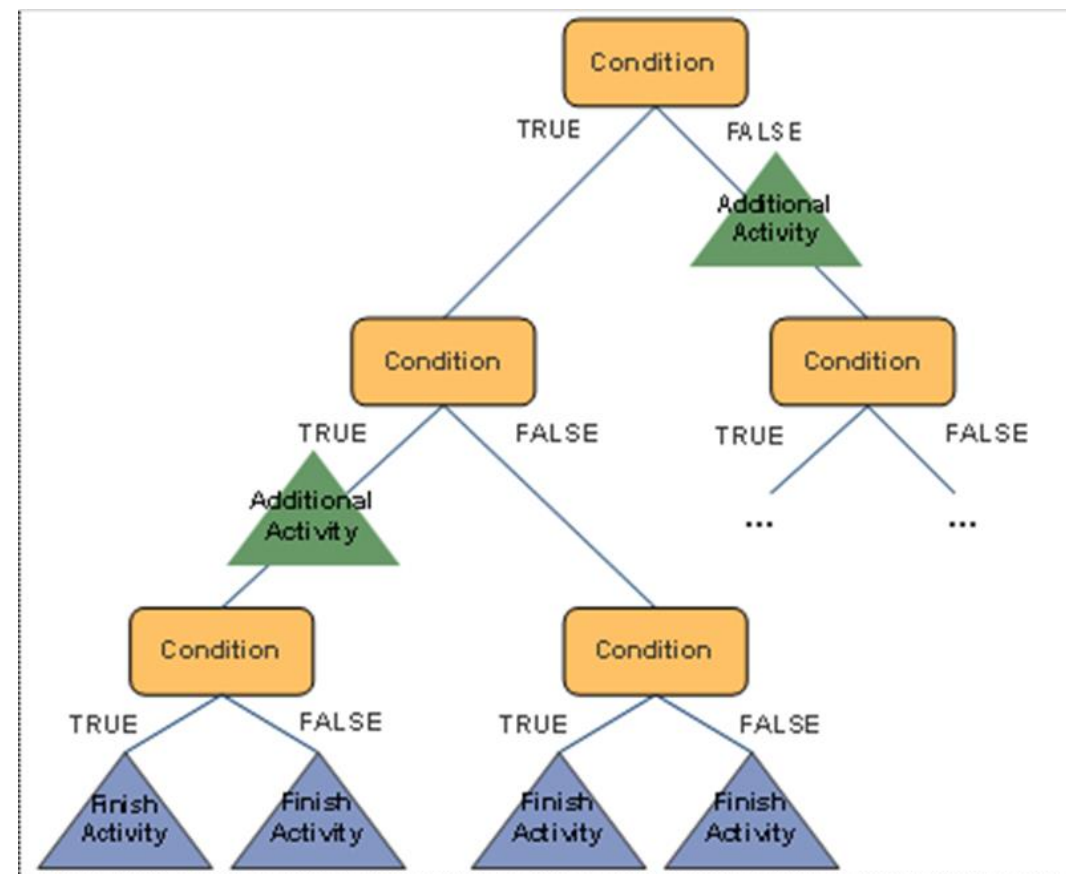


i nodi foglia rappresentano le classificazioni e le ramificazioni l'insieme delle proprietà che portano a quelle classificazioni. Di conseguenza ogni nodo interno risulta essere una macro-classe costituita dall'unione delle classi associate ai suoi nodi figli.

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)*

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 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
 - ❑ 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,
- ϵ variabile casuale dell'errore

$$y = X\beta + \epsilon$$

➤ 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

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

➤ La formula chiusa che si ottiene per la stima di β , sempre in notazione

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

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

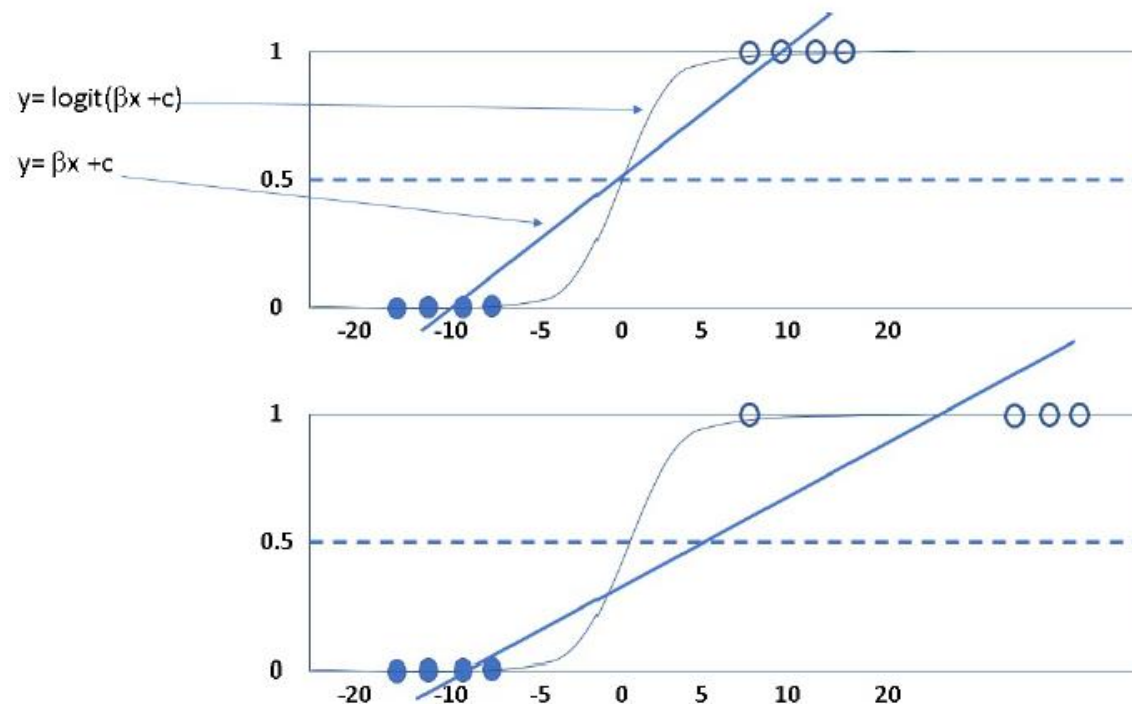
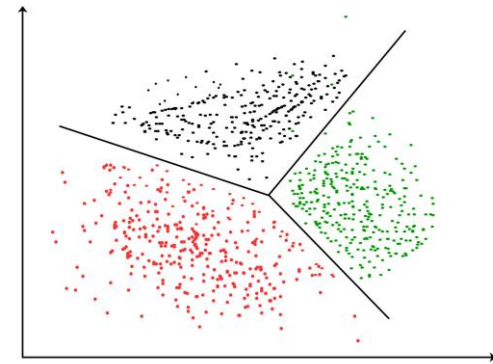


Figura 13.4: Esempio di regressione lineare e logistica.

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

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

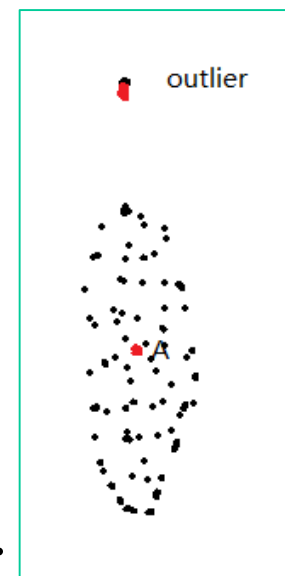
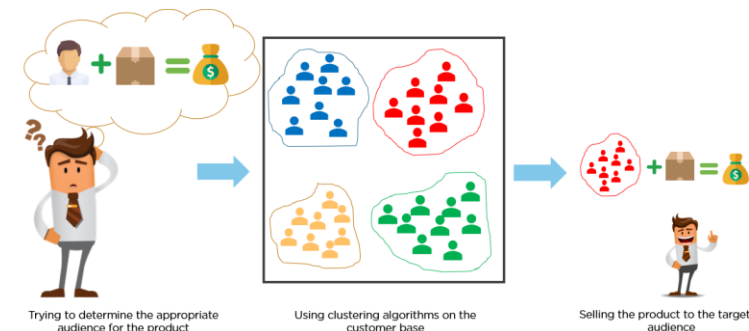
La cluster analysis è ampiamente utilizzata

- ❑ 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:

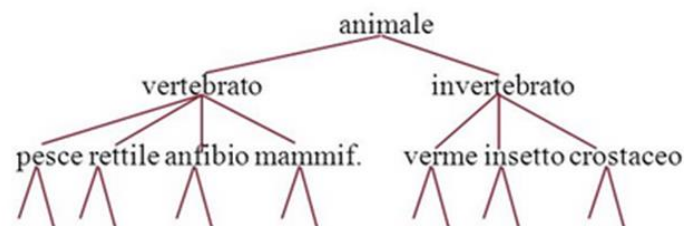
- ❑ l'eliminazione di questi valori anomali, che potrebbero essere causati da errori,
- ❑ l'isolamento di questi casi che magari rivestono una certa importanza per il business.



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



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

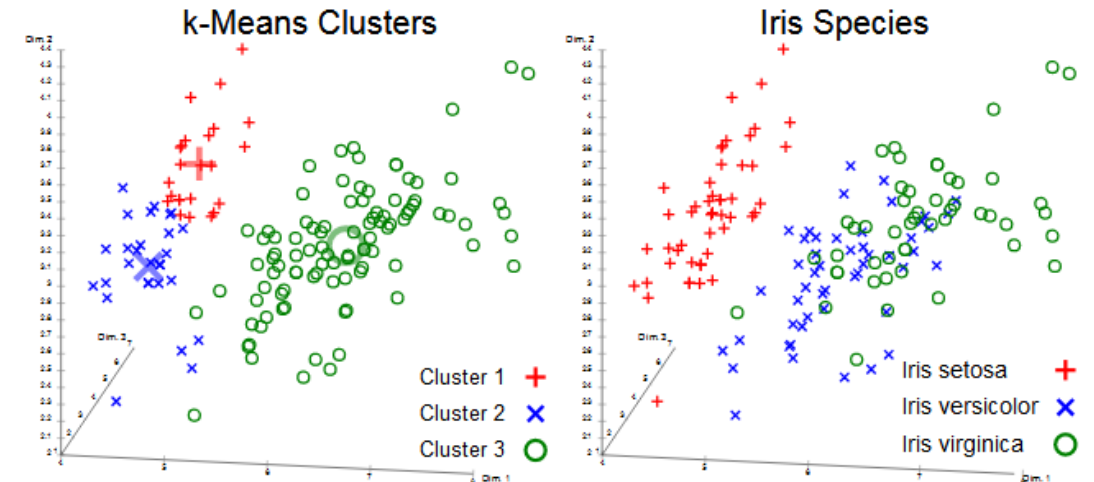
M_k è il vettore delle medie, o centroide per il cluster k

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.



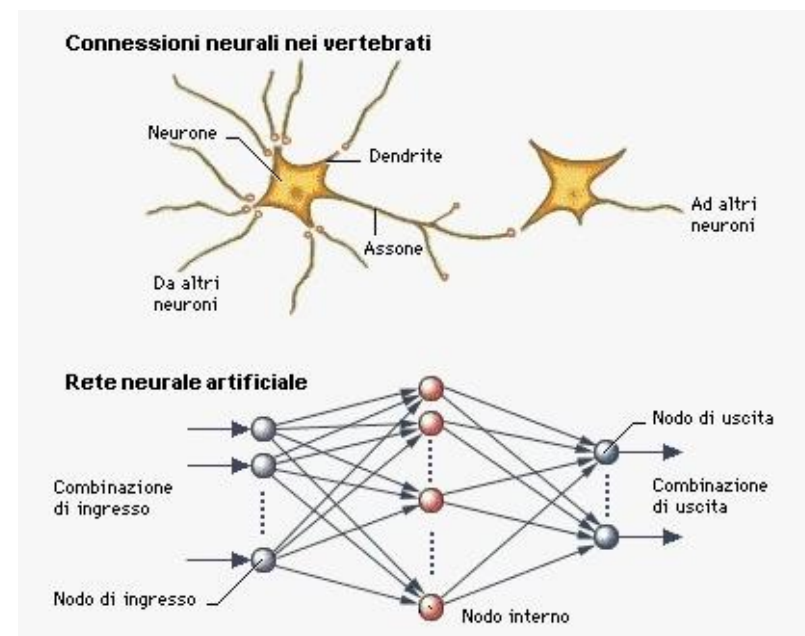
[Iris flower data set](#), clustered using [k means](#) (left) and true species in the data set (right). Note that k-means is non-deterministic, so results vary. Cluster means are visualized using larger, semi-transparent markers.

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

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

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.

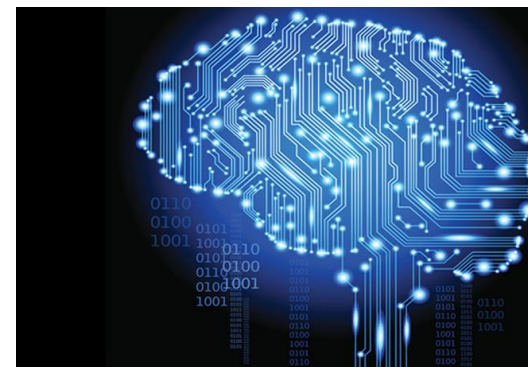


Le reti neurali si basano sul fatto che non sono solo una struttura complessa, ma anche flessibile. Il che significa:

- 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.
 - ...

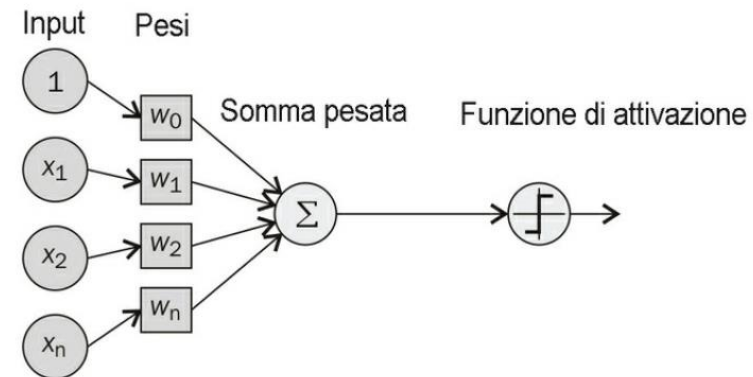


Reti neurali

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:

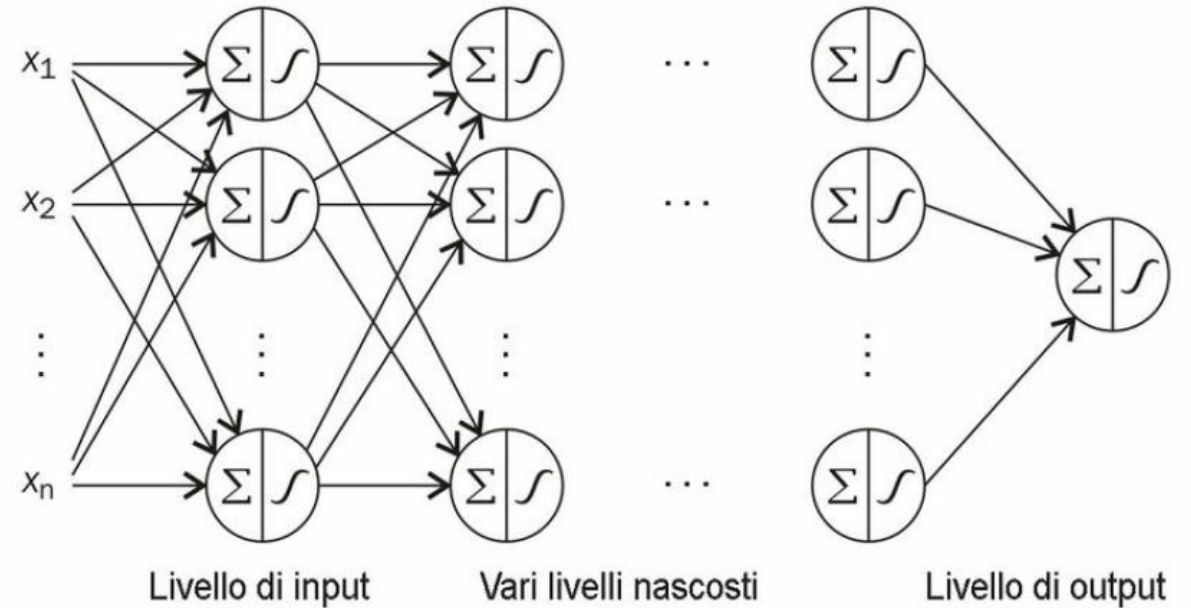
$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$



Reti neurali

La rete neurale MLP (Multilayer Perceptron):

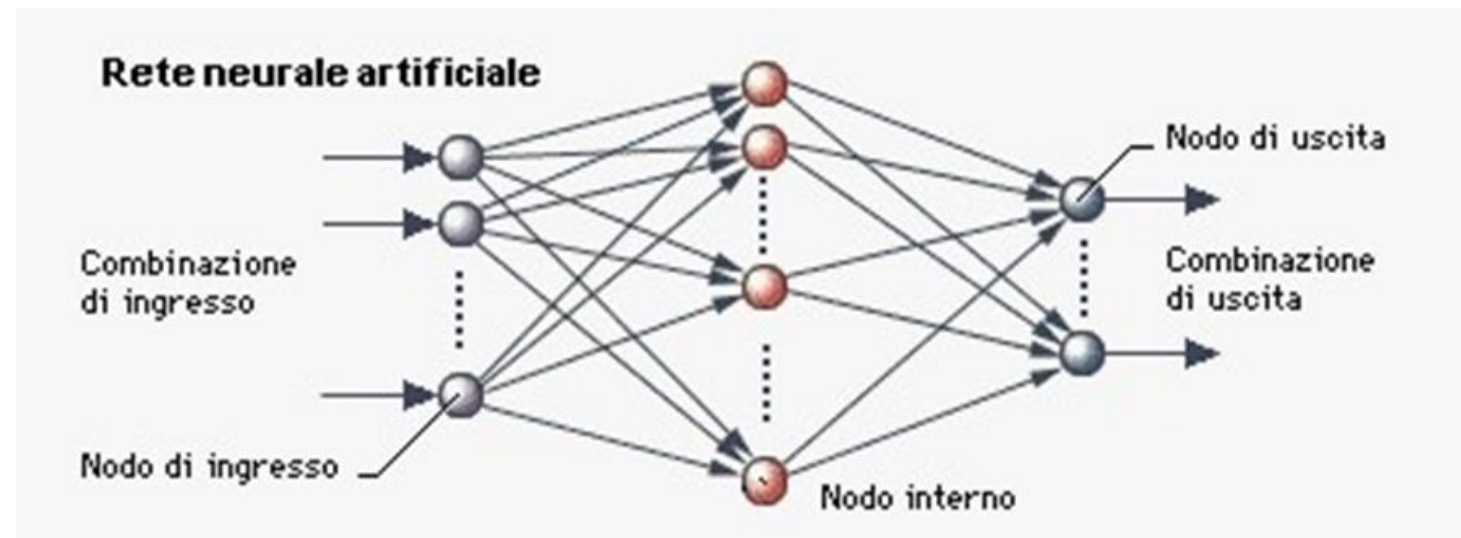
- ❑ Per creare una rete neurale, dobbiamo connettere fra loro più perceptron a formare una rete,
- *Un perceptron multilivello (MLP) è un grafo finito aciclico.*
- I nodi sono neuroni con funzione d'attivazione (in genere quella logistica)
- i collegamenti tra neuroni hanno un valore associato, detto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



Reti neurali

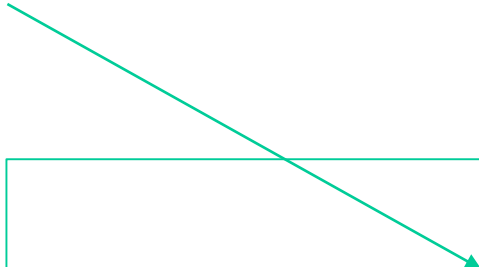
La rete neurale MLP (Multilayer Perceptron):

- ❑ 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

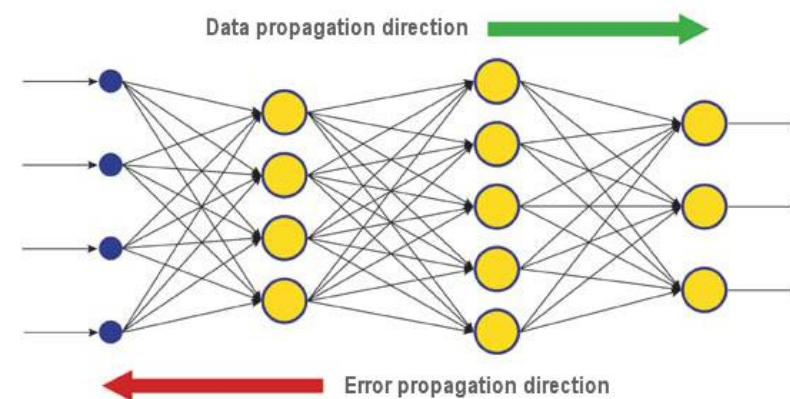
a_j = valore di output del neurone j-esimo

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*).
- 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

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

- ❑ 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 profonde 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

IL TEST E LA VALUTAZIONE DEI MODELLI PREDITTIVI

Il test e la valutazione dei modelli predittivi

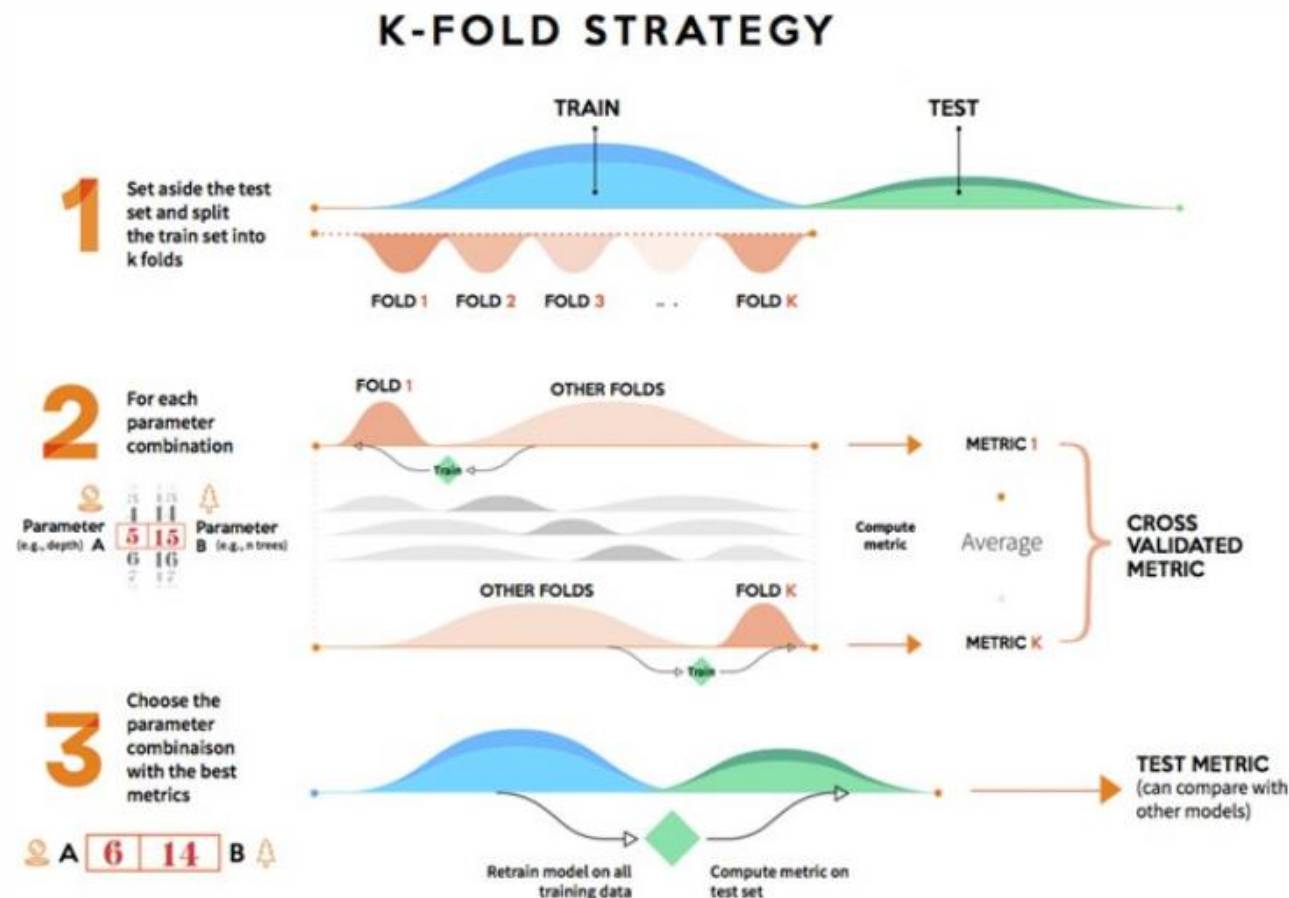
Le modalità di valutazione sono diverse a seconda del tipo di algoritmo

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

Valutazione dei modelli di classificazione

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 (test).
- Quindi si calcola una determinata metrica per ogni parte
- Alla fine si calcola o la media dei risultati



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 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).

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