**Machine Learning**

Varie aree Statistica, I.A, Data Science, Data Mining e Neuro Scienze contribuiscono alla realizzazione di algoritmi per l’apprendimento automatico (machine learning).

Data la grande mole di dati di chi oggi disponiamo, utilizzando algoritmi di autoapprendimento nel campo del machine learning, siamo in grado di trasformare questi dati in conoscenza. Si possono utilizzare questi algoritmi per individuare schemi nei dati ed eseguire previsioni sugli eventi futuri.

Nella seconda metà del ventesimo secolo, il ML si è evoluto come branca dell’Intelligenza Artificiale. Piuttosto che richiedere lka presenza una per individuare manualmente delle regole e costruire dei modelli per l’analisi di grandi quantità di dati, il ML offre un’alternativa più efficiente per catturare la conoscenza insita nei dati, modelli previsionali e poi prendere decisioni guidate dai dati.

Big Data: Enormi quantità di dati che non possono essere analizzati con metodologie classiche

Hardware: Gli algoritmi di ML sono degli algoritmi che eseguono calcoli matematici più o meno complessi e che quindi richiedono una potenza computazionale minima

Abbiamo tre metodi di apprendimento, tre tipi diversi di Machine Learning:

* **Apprendimento Supervisionato**

Alberi decisionali, Reti Neurali, SVM

Sono note le classi dei pattern utilizzati per l’addestramento. il training set è etichettato.

Si basa sul concetto di addestramento.

All’algoritmo viene dato in input il set di dati da analizzare sia i risultati che ci aspettiamo ci restituisca. In questo modo sarà l’algoritmo ad addestrarsi. In questo modo passando dei nuovi dati l’algoritmo sarà in grado di creare le regole interne per determinare un risultato.

I dati in input ed output non so come sono correlati.

Consiste nel trarre un modello a partire da dati di addestramento etichettati, i quali ci consentono di effettuare previsioni relative a dati non disponibili o futuri. Nell’insieme di campioni i segnali di output desiderati (le etichette) sono già noti.

Un compito di apprendimento supervisionato, sulla base di etichette di classi discrete, è chiamato anche compito di classificazione. Un’altra sottocategoria di apprendimento con supervisione e la regressione dove il segnale risultante è un valore continuo.

Si possono risolvere due tipi di problemi:

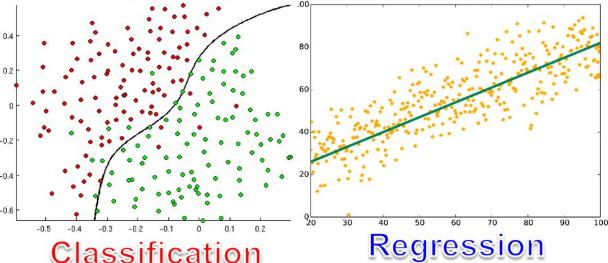
**Regressione**: Ho un Input continuo ed output continuo. Assegna un valore continuo ad un pattern.

La previsione di risultati continui, chiamata anche analisi di regressione. Nell’analisi di regressione, abbiamo un certo numero di variabili predittive (descrittive) ed una variabile target continua (risultato): cerchiamo di trovare una relazione fra queste variabili, tale che ci consenta di prevedere un risultato.

Nell’immagine è illustrata la regressione lineare, date delle variabili predittive (asse x) e variabili risposta (asse y), tracciamo una linea retta attraverso questi dati in modo da minimizzare la distanza (distanza quadratica media) fra i punti del campione e la linea. Possiamo utilizzare il punto d’intersezione e la pendenza che abbiamo appreso da questi dati per prevedere la variabile target per nuovi dati.

**Classificazione**: Ho un input discreto output discreto

La classificazione è una sottocategoria dell’apprendimento con supervisione, dove l’obiettivo è quello di prevedere le etichette di categoria delle classi per le nuove istanze, sulla base delle osservazioni compiute nel passato. Queste etichette sono valori discreti, non ordinati, che possono essere considerati come appartenenti ad un gruppo delle istanze.

*Classificazione binaria*: l’algoritmo d’apprendimento impara un insieme di regole con lo scopo di distinguere fra due possibili classi.

*Classificazione multiclasse*: un dataset d’apprendimento che è costituito da più esempi di scrittura a mano di ciascuna lettera dell’alfabeto, fornendo un nuovo carattere scritto a amano, il nostro modello predittivo sarà in grado di prevedere con una certa precisione la lettera corretta dell’alfabeto. Il compito di classificazione consiste nell’assegnare alle istanze etichette di categoria non ordinate.

Classificazione: assegna una classe a un pattern. Necessario apprendere una funzione capace di eseguire il mapping dallo spazio dei pattern allo spazio delle classi. Si usa spesso anche il termine riconoscimento. Nel caso di 2 sole classi si usa il termine binary classification, con più di due classi multi-class classification.

Classe: insieme di pattern aventi proprietà comuni. Es. i diversi modi in cui può essere scritto a mano libera il carattere A. Il concetto di classe è semantico e dipende strettamente dall’applicazione: - 21 classi per il riconoscimento di lettere dell’alfabeto - 2 classi per distinguere le lettere dell’alfabeto italiano da quello cirillico.

* **Apprendimento Non** **Supervisionato**

Clustering, Regole di associazione.   
Non sono note le classi dei pattern utilizzati per l’addestramento. il training set non è etichettato.

Abbiamo una grande mole di dati l’algoritmo troverà dei pattern, delle relazioni…

Nell’apprendimento con supervisione, conosciamo in anticipo la risposta corretta quando descriviamo il nostro modello, mentre nell’apprendimento di rinforzamento definiamo una misura, o ricompensa, per le specifiche azioni messe in atto dall’agente. Nell’apprendimento senza supervisione, al contrario, abbiamo a che fare con dati non etichettati o dati dalla struttura ignota. Utilizzando tecniche di apprendimento senza supervisione siamo in grado di osservare la struttura dei nostri dati, per estrarre da essi informazioni cariche di significato senza però poter contare sulla guida né di una variabile nota relativa al risultato, né una funzione di ricompensa.

**Clustering:** Ricerca di sottogruppi tramite il clustering. Il clustering è una tecnica esplorativa di analisi dei dati che ci consente di organizzare una serie di informazioni all’interno di gruppi significativi detti cluster senza avere alcuna precedente conoscenza delle appartenenze a tali gruppi. Un gruppo di oggetti che condividono un certo grado di similarità, ma che sono più dissimilari rispetto agli oggetti presenti negli altri cluster.

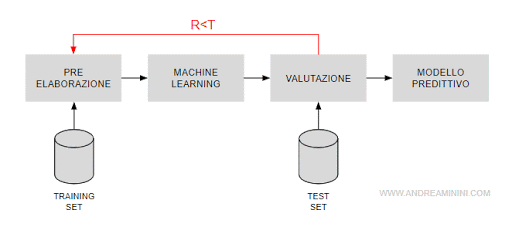
Il clustering è un’ottima tecnica per la strutturazione dell’informazione e per individuare relazioni significative nei dati.

Spesso nemmeno il numero di cluster è noto a priori I cluster individuati nell’apprendimento possono essere poi utilizzati come classi.

* **Apprendimento con** **rinforzo**
* il training set è etichettato parzialmente.

L’algoritmo impara dagli errori. Non partiamo dai dati ma (es. algoritmo della scacchiera impara dagli errori non dalle giocate degli altri). L’obiettivo è quello di sviluppare un sistema (agente) che migliori le proprie prestazioni sulla base delle interazioni con l’ambiente, le informazioni relative allo stato corrente dell’ambiente includono anche un cosiddetto segnale di ricompensa (reward), questo feedback non p l’etichetta o il valore corretto di verità, ma una misura della qualità con cui l’azione è stata misurata da una funzione di ricompensa.

Tramite l’interazione con l’ambiente, un agente può quindi utilizzare l’apprendimento di rafforzamento per imparare una serie di azioni che massimizzano questa ricompensa tramite un approccio esplorativo di tipo trial-and-error.

****

**Pre – elaborazione: date una “forma” ai dati**

I dati sono un ingrediente fondamentale del machine learning, dove il comportamento degli algoritmi non è pre-programmato ma appreso dai dati stessi. Utilizzeremo spesso il termine Pattern per riferirci ai dati, ad esempio un pattern può essere un volto, un carattere scritto a mano, un’impronta digitale, un segnale sonoro, un frammento di testo, l’andamento di un titolo di borsa.

**Tipi di dati:**

* **Numerici**: valori associati a caratteristiche misurabili o conteggi. Tipicamente continui (ma anche discreti, es. interi), in ogni caso soggetti a ordinamento. Rappresentabili naturalmente come vettori numerici nello spazio multidimensionale. L’estrazione di caratteristiche da segnali (es., immagini, suoni) produce vettori numerici detti anche feature vectors. Es. Persona: [altezza, circonferenza toracica, circonferenza fianchi, lunghezza del piede].
* **Categorici**: valori associati a caratteristiche qualitative e alla presenza/assenza di una caratteristica (yes/no value). Non «semanticamente» mappabili in valori numerici. Es. Persona: [sesso, maggiorenne, colore occhi, gruppo sanguigno]. Talvolta soggetti a ordinamento (ordinali): es. temperatura ambiente: alta, media o bassa. Normalmente gestiti da sistemi a regole e alberi di classificazione. Molto utilizzati nell’ambito del data mining, spesso insieme a dati numerici (mixed).
* **Sequenze**: pattern sequenziali con relazioni spaziali o temporali. Es. uno stream audio (sequenza di suoni) corrispondente alla pronuncia di una parola, una frase (sequenza di parole) in linguaggio naturale, un video (sequenza di frame), l’andamento di un titolo di borsa (sequenza temporale del prezzo di chiusura). Spesso a lunghezza variabile. La posizione nella sequenza e le relazioni con predecessori e successori sono importanti. Critico trattare sequenze come pattern numerici. Allineamento spaziale/temporale, e «memoria» per tener conto del passato.

La pre-elaborazione dei dati è uno dei passi cruciali di qualsiasi applicazione di apprendimento automatico. Molti algoritmi d’apprendimento automatico richiedono anche che, per ottenere le massime prestazioni, le caratteristiche scelte adottino la stessa scala, il che spesso viene ottenuto trasformano le caratteristiche in un intervallo [0,1] oppure in una distribuzione normale standard con media 0 e varianza 1.

Bisogna portare il dataset sulla stessa scala perché se i valori dei dati sono sufficentemente differenti l'algoritmo potrebbe dare più importanza ai valori grandi non essendo non essndo più sufficientemente preciso.  
I modelli come gli alberi decisionali non soffrono di questo problema.  
  
Due metodi:

* NORMALIZZAZIONE PORTA I VALORI IN UN RANGE TRA 0 ed 1  
  dato un vettore X = [x1,x2,x3.....xn]  
  Applica la formula Xi norm = (Xi – Xmin) / (Xmax - Xmin) va fatto per ogni elemento dell'array da i che va da 0 ad n
* STANDARDIZZAZIONE PORTA LA DISTRIBUZIONE IN UNA DISTRIBUZIONE NORMALE  
  ovvero in una distribuzione di valori con media centrata a 0 e deviazione standard pari ad 1  
  per ogni valore del vettore applico la formula: X­i std = (Xi - Xmean) / Xsd  
  Conserva le informazioni dei valori che si discostano ma alcuni valori possono essere negativi.

Per determinare se il nostro algoritmo d’apprendimento non solo si comporta bene su un set di addestramento, ma segue generalizzazioni corrette sui nuovi dati, lo suddividiamo in modo casuale in due set distinti: set di addestramento / training ed il set di test.

Utilizziamo il set di addestramento per informare ed ottimizzare il modello di apprendimento automatico, mentre teniamo da parte fino all’ultimo il set di test, per valutare il modello finale.

**Addestramento e selezione di un modello predittivo**

Gli algoritmi possono essere addestrati in vario modo:

* **Batch**: l’addestramento è effettuato una sola volta su un training set dato. una volta terminato il training, il sistema passa in «working mode» e non è in grado di apprendere ulteriormente. Attualmente, la maggior parte dei sistemi di machine learning opera in questo modo.
* **Incrementale**: a seguito dell’addestramento iniziale, sono possibili ulteriori sessioni di addestramento

Rischio: Catastrofic Forgetting (il sistema dimentica quello che ha appreso in precedenza).

* **Naturale**: addestramento continuo (per tutta la vita) Addestramento attivo in working mode. Coesistenza di approccio supervisionato e non supervisionato. human-like learning involves an initial small amount of direct instruction (e.g. parental labeling of objects during childhood) combined with large amounts of subsequence unsupervised experience (e.g. self-interaction with objects)

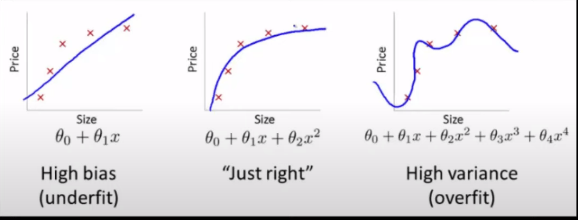
In genere, il comportamento di un algoritmo di Machine Learning è regolato da un set di parametri Θ (es. i pesi delle connessioni in una rete neurale). L’apprendimento consiste nel determinare il valore ottimo Θ\* di questi parametri.

Dato un training set 𝑇𝑟𝑎𝑖𝑛 e un insieme di parametri, la funzione obiettivo 𝑓 (𝑇𝑟𝑎𝑖𝑛, Θ) può indicare: l’ottimalità della soluzione da massimizzare: Θ\* = 𝑎𝑟𝑔𝑚𝑎𝑥Θ (𝑓 𝑇𝑟𝑎𝑖𝑛, Θ)  
oppure l’errore o perdita (loss-function) da minimizzare: Θ\* = 𝑎𝑟𝑔𝑚𝑖𝑛Θ 𝑓 (𝑇𝑟𝑎𝑖𝑛, Θ)  
𝑓 𝑇𝑟𝑎𝑖𝑛, Θ può essere ottimizzata.

È essenziale confrontare almeno un certo gruppo differente di algoritmi, in modo da addestrarli e selezionare poi il modello che offre le migliori prestazioni. Ma prima di poter confrontare modelli differenti, dobbiamo decidere le metriche da impiegare per misurare le prestazioni.

Una metrica comunemente utilizzata è l’accuratezza della classificazione, che è definita come la proporzione tra le istanze classificate correttamente.

Non dobbiamo modellare i dati, ma creare un modello che impari a generalizzare, se gli si da un nuovo dato l’output sarà corretto.

La soluzione a questo problema e dividere il set di dati in due set TrainSet e TestSet. Dati in input i dati del TrainSet l’algoritmo apprenderà su questi dati, ma quello che ci interessa è che abbia imparato anche sul TestSet ovvero se l’algoritmo ha generalizzato dev’essere in grado di funzionare su dati su chi non è stato mai allenato (capire se funziona). La potenza sta appunto nel poter apprendere da molti parametri.

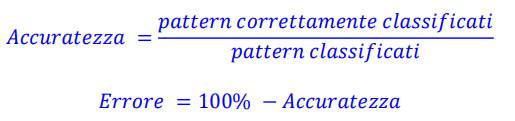
* Training Set (Train) è l’insieme di pattern su cui addestrare il sistema, trovando il valore ottimo per i parametri Θ.
* Validation Set (Valid) è l’insieme di pattern su cui tarare gli iperparametri H (ciclo esterno).
* Il Test Set (Test) è l’insieme di pattern su cui valutare le prestazioni finali del sistema. Sempre forte è la tentazione di tarare gli iperparametri sul test set, ma questo dovrebbe essere evitato, pena sovrastima delle prestazioni.

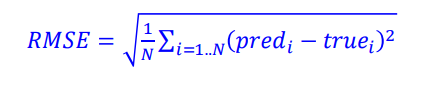
Partizionare i dati in dati in Training, Validation e Test, richiede in genere di generare (random) gli indici degli elementi da assegnare ai diversi insiemi per poi procedere alla suddivisione facendo attenzione a dividere nello stesso modo anche le etichette (in classificazione) o valori target (nella regressione).

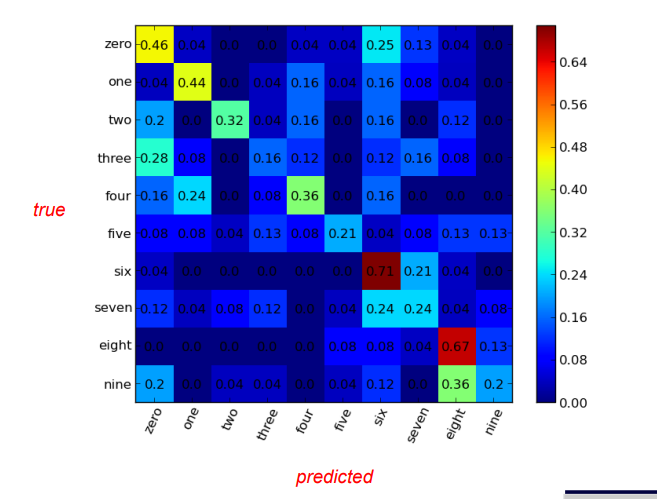
Dobbiamo dunque capire quale modello si comporta meglio sul dataset di test finale e sui dati reali non utilizzando però questo dataset di test per la scelta del modello ma per la valutazione finale del modello stesso. Quindi il modello deve comportarsi al meglio sul dataset di test ma senza però essere addestrato sul dataset di test bensì su quello di training.

Possono essere utilizzate varie tecniche di convalida incrociata nelle quali il dataset di addestramento viene ulteriormente suddiviso in sottoinsiemi di addestramento di convalida, in modo da stimare le prestazioni di generalizzazione del modello.

Ma come si valutano le prestazioni, e su quali dati?  
Una possibilità consiste nell’utilizzare direttamente la funzione obiettivo per quantificare le prestazioni.  
In genere però si preferisce una misura legata direttamente alla semantica del problema.

In un problema di **Classificazione**, l’accuratezza di classificazione [0…100%] è la percentuale di pattern correttamente classificati. L’errore di classificazione è il complemento.

Nei problemi di **Regressione**, si valuta in genere l’RMSE (Root Mean Squared Error) ovvero la radice della media dei quadrati degli scostamenti tra valore vero e valore predetto.

La **matrice di confusione** (confusion matrix) è molto utile nei problemi di classificazione (multiclasse) per capire come sono distribuiti gli errori. Nell’esempio un problema di classificazione digit (10 classi). Sulle righe le classi true sulle colonne le classi predicted. Una cella (r,c) riporta la percentuale di casi in cui il sistema ha predetto di classe c un pattern di classe vera r. Idealmente la matrice dovrebbe essere diagonale. Valori elevati (fuori diagonale) indicano concentrazioni di errori.