Osservazioni sul Machine Learning

Federico Russo e Lorenzo Tondini

2020-2021

Prefazione

In questo articolo ci prefiggiamo l'obiettivo di affrontare i fondamenti matematici e informatici necessari ad una basilare comprensione del Machine Learning, un ambiente di ricerca molto promettente nato negli ultimi decenni del secolo scorso che potrebbe ampliare gli orizzonti dell'Intelligenza Artificiale, o IA, la disciplina che cerca di creare computer che imitino la mente umana o alcune sue facoltà. Utilizzato dalla maggior parte delle multinazionali di vario genere, il Machine Learning permette, tra le altre cose, di studiare i comportamenti dei consumatori, prevenire frodi, furti di dati e di identità. Per esempio, su di esso si basano i sistemi di controllo delle auto a guida autonoma, di filtrazione automatica delle mail di spam, di riconoscimento vocale e di identificazione di volti in immagini e video.

Come esplicitato nel titolo, questo testo non vuole essere esaustivo nei confronti dell'argomento trattato, ma è piuttosto una raccolta di informazioni essenziali grazie alla quale ci si può avvicinare a questa materia articolata e innovativa. Abbiamo inoltre scelto di approfondire alcuni aspetti matematici che a nostro avviso possono essere facilmente collegati al programma di quinta superiore del liceo scientifico, i cui contenuti sono prerequisiti al fine della comprensione di questo articolo. Per la sezione informatica, invece, è sufficiente una conoscenza dei concetti di algoritmo e strutture di controllo della programmazione, in quanto abbiamo affrontato l'argomento da un punto di vista meramente teorico ed esplicativo, tralasciando l'analisi del codice sorgente.

Un'applicazione concreta del Machine Learning, da noi ideata e realizzata, segue i capitoli dedicati alla teoria matematica e informatica, in modo tale che chiunque abbia letto le sezioni precedenti possa comprenderla pienamente. Essa consiste nella predizione dei prezzi degli immobili di uno specifico quartiere a partire da alcuni dati fondamentali quali la superficie, il numero di locali e di bagni. A causa di un fattore soggettivo nell'attribuzione del prezzo reale agli immobili, i risultati del programma contengono inevitabilmente un certo errore, che può essere tuttavia minimizzato mediante alcuni strumenti di calcolo e analisi dei dati.

Il nostro auspicio è che questo articolo possa risultare utile agli studenti liceali più interessati che, pur non possedendo le conoscenze matematiche necessarie, vorrebbero affacciarsi al mondo del Machine Learning, superando le difficoltà dovute all'elevato numero di informazioni che si trovano sulla rete.

Capitolo 1

La Matematica del Machine Learning

Come accade per la maggior parte delle discipline scientifiche, la matematica è alla base dell'Intelligenza Artificiale, e ne regola lo sviluppo. Al fine di comprendere e studiare i complessi aspetti del *Machine Learning*, sono necessarie alcune conoscenze negli ambiti di analisi matematica, algebra lineare e statistica univariata e multivariata. Infatti, lo studio di queste discipline fornisce gli strumenti adeguati per rappresentare, valutare ed elaborare i dati in input, garantendo la produzione di output attendibili. Uno studio più approfondito di un qualsiasi settore dell'IA richiederebbe fondamenti matematici molto avanzati.

1.1 Algebra Lineare

L'algebra lineare è una sottobranca dell'algebra che si occupa dello studio di vettori, matrici, spazi vettoriali, trasformazioni lineari, autovettori e autovalori. Tuttavia, in questi paragrafi affronteremo solo i primi due argomenti citati, la cui conoscenza è fondamentale per familiarizzare con il Machine Learning, senza approfondirli più del necessario.

1.1.1 Matrici e Vettori

Siano $m, n \in \mathbb{N}^+$. Dati $m \times n$ elementi di un insieme numerico stabilito, la tabella che li ordina in m righe e n colonne viene detta *matrice*, e i numeri si chiamano *elementi* della matrice, e possono essere rappresentati ordinatamente tra parentesi. Per indicare le matrici si utilizzano solitamente le lettere maiuscole, per gli elementi vengono usate invece le corrispondenti lettere minuscole contrassegnate da due indici, solitamente i e j, il primo dei quali indica il numero della riga ed è tale che $1 \leq i \leq m$, e il secondo quello della colonna, ed è tale che $1 \leq j \leq n$.

Esempio 1.1

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

Due o più matrici vengono dette dello stesso tipo se hanno lo stesso numero m di righe e lo stesso numero n di colonne, mentre i loro elementi che hanno gli stessi indici sono elementi corrispondenti; l'insieme di matrici dello stesso tipo si indica con $M_{m \times n}$. Esistono inoltre i *vettori riga* e i *vettori colonna*, cioè matrici contenenti rispettivamente una sola riga ed una sola colonna.

Una matrice è detta nulla se tutti i suoi elementi sono uguali a 0, ed è indicata con il simbolo O_{mn}, o più generalmente O, tralasciandone le dimensioni. Si chiama matrice opposta di A la matrice $-A$, formata da elementi corrispondenti opposti a quelli di A.

Se il numero m di righe è uguale al numero n delle colonne, la matrice si dice *quadrata*, altrimenti rettangolare. L'ordine di una matrice quadrata $n \times n$ è il numero n, la sua diagonale principale è formata da tutti gli elementi che hanno i due indici uguali fra loro, ovvero quelli che vanno da a_{11} ad a_{nn}, e la diagonale secondaria ha invece come estremi a_{1n} e a_{n1}.

Definiamo le seguenti operazioni con le matrici:

- la somma tra le matrici A e B dello stesso tipo è una matrice $C = A + B$ formata da elementi $c_{ij} = a_{ij} + b_{ij}$;
- il prodotto di una matrice A per uno scalare h è una matrice $C = hA$ i cui elementi sono $c_{ij} = ha_{ij}$;
- il prodotto tra un vettore riga A e un vettore colonna B con lo stesso numero n di elementi è una matrice $C = AB$ formata da un solo elemento $c_{11} = \sum_{k=1}^{n} a_{1k}b_{k1}$.

La *trasposta* di una matrice A si indica con A^T e si ottiene scambiando ordinatamente le righe con le colonne; essa è una trasformazione involutoria, poiché la trasposta della trasposta della matrice A è la matrice stessa. Inoltre, se per una matrice quadrata B risulta $B = B^T$, essa è simmetrica rispetto alla diagonale principale.

Esempio 1.2

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Ad ogni matrice quadrata A a coefficienti reali è associato un numero reale, chiamato *determinante*, che si indica generalmente con $det(A)$. Se A è di primo ordine, esso corrisponde all'unico elemento della matrice, ossia $det(A) = a_{11}$; se si tratta invece di un matrice di secondo ordine, il determinante è dato dalla differenza tra i prodotti degli estremi della diagonale principale e quelli della diagonale secondaria. È possibile calcolare il determinante anche di matrici quadrate di ordine $n > 2$, tuttavia il procedimento risulta più complicato, e non sarà trattato in questa sezione.

Esempio 1.3

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$det(A) = 1 \cdot 4 - 2 \cdot 3 = -2$$

1.2 Analisi

Fondamentali sono le nozioni di derivata parziale e gradiente. Per definirle, è necessario estendere il concetto di funzione: essa, per definizione, associa ad ogni elemento del dominio uno ed un solo elemento del codominio, tuttavia l'insieme di partenza non è necessariamente l'insieme $\mathbb{R}$. Si chiama funzione reale a più variabili reali una funzione del tipo $f : \mathbb{R}^n \to \mathbb{R}$, dove $\mathbb{R}^n$ è il prodotto cartesiano

di $\mathbb{R}$ con sé stesso n volte e $n \in \mathbb{N}$. In questo caso, gli elementi dell'insieme di partenza sono *n-uple ordinate*, e gli elementi dell'insieme di arrivo sono numeri reali.

Per esempio, una funzione $\varphi : \mathbb{R}^2 \to \mathbb{R}$ è una funzione a due variabili indipendenti, ha equazione $z = \varphi(x, y)$ e può essere rappresentata in un sistema di riferimento cartesiano tridimensionale $Oxyz$. Ad ogni coppia ordinata (x, y) è associato il corrispondente valore reale z, detto *quota*.

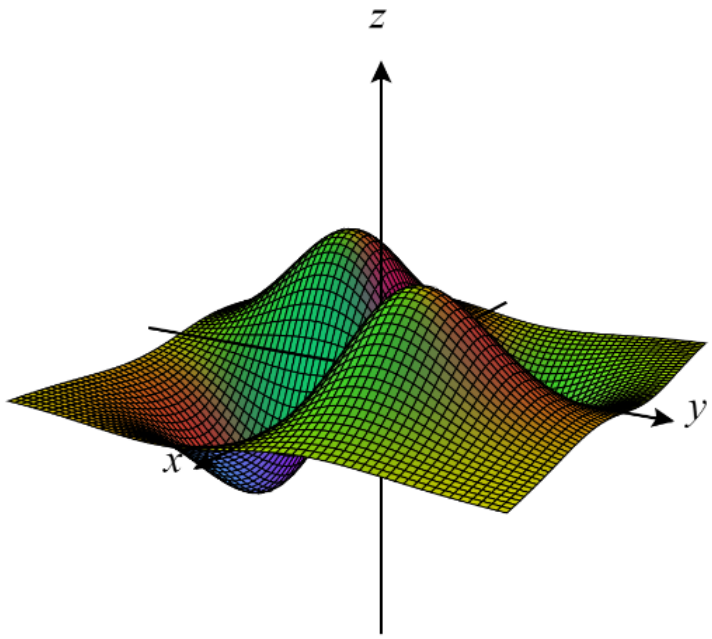


Grafico della funzione a due variabili indipendenti di equazione $z = \frac{5xy}{e^{x^2+y^2}}$.

1.2.1 Derivate Parziali

Nel caso di una funzione a due o più variabili indipendenti, è possibile estendere il concetto di derivata in modo tale che essa esprima la variazione della variabile dipendente rispetto a ciascuna di quelle indipendenti. Sia $f : \mathbb{R}^2 \to \mathbb{R}$ di legge $z = f(x, y)$ una funzione nello spazio. Si definisce *derivata parziale* rispetto ad x, e si indica con $\frac{\partial f(x,y)}{\partial x}$ oppure $f'_x(x, y)$, la derivata della funzione calcolata fissando il valore di y. Più formalmente, la derivata parziale di f rispetto ad x è definita con il seguente limite, se esiste ed è finito:

$$\frac{\partial f(x, y)}{\partial x} = \lim_{h \to 0} \frac{f(x + h, y) - f(x, y)}{h}.$$

Analogamente, si definisce derivata parziale di f rispetto ad y, e si indica con $\frac{\partial f(x,y)}{\partial y}$ oppure $f'_y(x, y)$, la derivata della funzione calcolata fissando il valore di x, ossia il limite, se esiste ed è finito:

$$\frac{\partial f(x, y)}{\partial y} = \lim_{h \to 0} \frac{f(x, y + h) - f(x, y)}{h}.$$

I precedenti limiti definiscono le derivate parziali di funzioni a due variabili indipendenti. Tuttavia è possibile definire le derivate parziali di una funzione $g : \mathbb{R}^m \to \mathbb{R}^n$ con un qualsiasi numero di

variabili indipendenti. Nelle applicazioni al Machine Learning, spesso è necessario calcolare derivate parziali di funzioni con un numero elevato di variabili indipendenti.

Dal punto di vista geometrico, la derivata parziale $f'_x(x, y)$ rappresenta la funzione del coefficiente angolare della tangente alla curva ottenuta fissando il valore di y; ossia dato un punto di coordinate (x_0, y_0) del dominio di f, essa indica la pendenza della tangente a $f(x, y_0)$ nel punto x_0. Analoghe considerazioni valgono per la derivata calcolata rispetto ad y.

Per il calcolo delle derivate parziali, valgono le stesse regole di derivazione del calcolo differenziale a una variabile indipendente.

Esempio 1.4 Vediamo un esempio: data la funzione di equazione $f(x, y) = x^2 e^y - y^3$, le sue derivate parziali rispetto ad x e y sono rispettivamente:

$$\frac{\partial f(x, y)}{\partial x} = 2xe^y,$$

$$\frac{\partial f(x, y)}{\partial y} = x^2 e^y - 3y^2.$$

Anche nel calcolo differenziale a più variabili sono definite le derivate successive. In questo caso, però, ogni derivata parziale prima può essere derivata nuovamente rispetto a ciascuna delle variabili indipendenti della funzione. Per questo motivo, la derivata seconda di una funzione a due variabili indipendenti può essere:

- pura rispetto ad x se è derivata due volte rispetto alla variabile x, e si indica con $f''_{xx}(x, y)$ oppure con $\frac{\partial^2 f(x,y)}{\partial x^2}$;
- pura rispetto ad y se è derivata due volte rispetto alla variabile y, e si indica con $f''_{yy}(x, y)$ oppure con $\frac{\partial^2 f(x,y)}{\partial y^2}$;
- mista rispetto ad x se è derivata la prima volta rispetto ad y e la seconda rispetto ad x, e si indica con $f''_{yx}(x, y)$ oppure con $\frac{\partial^2 f}{\partial y \partial x}$;
- mista rispetto ad y se è derivata la prima volta rispetto ad x e la seconda rispetto ad y, e si indica con $f''_{xy}(x, y)$ oppure con $\frac{\partial^2 f}{\partial x \partial y}$.

Inoltre, per il teorema di Schwartz, se la funzione ammette derivate seconde miste continue, risulta sempre che:

$$\frac{\partial^2 f(x, y)}{\partial x \partial y} = \frac{\partial^2 f(x, y)}{\partial y \partial x}.$$

Le derivate parziali di secondo ordine sono anche rappresentate utilizzando una particolare matrice chiamata *hessiana*, nel seguente modo:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix},$$

e, poiché la matrice è di ordine 2, il suo determinante è:

$$det(H) = \frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 f}{\partial x \partial y} \cdot \frac{\partial^2 f}{\partial y \partial x}.$$

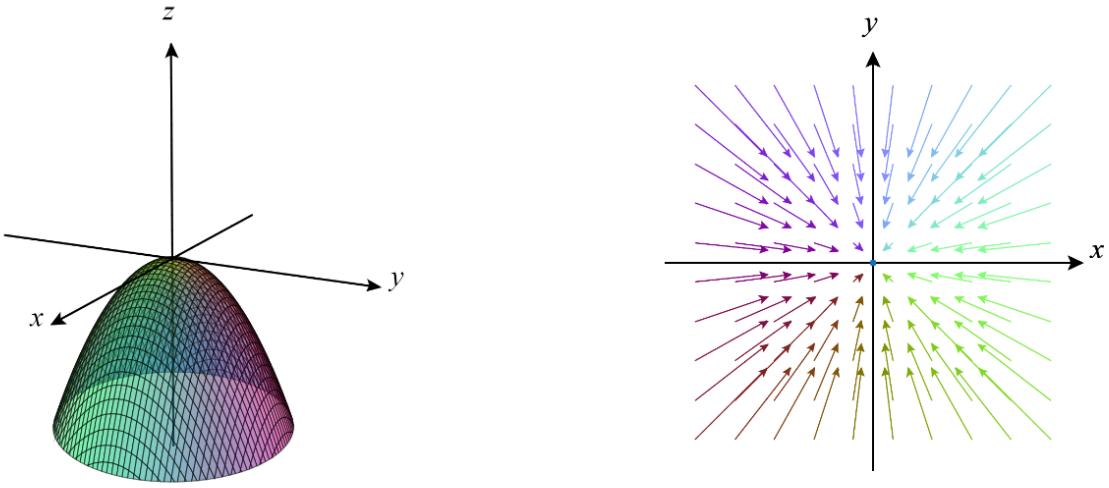
1.2.2 Gradiente

Si chiama *gradiente* di una funzione a due o più variabili indipendenti il vettore avente per componenti le derivate parziali della funzione rispetto a ciascuna delle sue variabili indipendenti, e si indica con ∇f oppure con *grad* f. Per una funzione $f : \mathbb{R}^2 \to \mathbb{R}$, risulta quindi che:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = \frac{\partial f}{\partial x}\hat{\mathbf{x}} + \frac{\partial f}{\partial y}\hat{\mathbf{y}},$$

dove $\hat{\mathbf{x}}$ e $\hat{\mathbf{y}}$ sono i versori lungo gli assi cartesiani del piano xy.

L'operatore gradiente associa a ciascun punto dello spazio un vettore, che esprime la variazione della funzione f rispetto ai diversi parametri da cui è composta, pertanto esso è un campo vettoriale. Il gradiente permette di determinare le direzioni di massima crescita e di massima discesa di una funzione differenziabile.



Grafici della funzione di equazione $z = -(x^2 + y^2)$ e del suo gradiente di componenti $(-2x, -2y)$.

È possibile estendere il teorema di Fermat sui punti stazionari al caso di una funzione a più variabili indipendenti: si ottiene che la condizione necessaria affinché un punto del dominio sia un punto critico, che nel caso di funzioni a due variabili indipendenti può essere un minimo relativo, un massimo relativo o un punto di sella, è che il gradiente della funzione calcolato in quel punto si annulli. Tuttavia, analogamente al caso di funzioni ad una variabile indipendente, questa condizione necessaria non permette di distinguere i punti critici, ovvero di capire di quale tipo si tratti; per farlo è necessario calcolare il determinante della matrice hessiana, chiamato *hessiano*, considerando i valori delle derivate parziali seconde nel punto critico. In particolare, per una funzione $f : \mathbb{R}^2 \to \mathbb{R}$, data la sua hessiana H con determinante $det(H)$, se nel punto (x_0, y_0) del dominio risulta:

- $det(H) > 0$ e $f''_{xx}(x_0, y_0) > 0$, allora (x_0, y_0), è un punto di minimo relativo;
- $det(H) > 0$ e $f''_{xx}(x_0, y_0) < 0$, allora (x_0, y_0), è un punto di massimo relativo;
- $det(H) < 0$, allora (x_0, y_0) è un punto di sella.

Inoltre, per una generica funzione $\varphi : \mathbb{R}^n \to \mathbb{R}$, il gradiente ne rappresenta la migliore approssimazione lineare della funzione in un qualsiasi punto $x_0 \in \mathbb{R}^n$, analogamente allo sviluppo in serie di Taylor di una funzione in più variabili, secondo la formula:

$$\varphi(x) \approx \varphi(x_0) + (\nabla\varphi)_{x_0} \cdot (x - x_0),$$

in cui $(\nabla\varphi)_{x_0}$ indica il gradiente di φ calcolato in x_0.

Discesa del Gradiente

Sia $f : \mathbb{R}^n \to \mathbb{R}$ una funzione a più variabili. Si definiscono direzione di crescita e direzione di discesa i vettori n-dimensionali orientati, rispettivamente, verso un massimo locale e verso un minimo locale. L'uso del gradiente è molto frequente nei problemi di ottimizzazione matematica, poiché esso indica la direzione di massima crescita della funzione, mentre il suo opposto ne indica la direzione di massima discesa.

Il metodo di *discesa del gradiente*, sviluppato dall'illustre matematico Cauchy, consente proprio di determinare gli estremanti della funzione mediante un algoritmo ricorsivo che prevede il calcolo del gradiente, per determinare la direzione da seguire, e la scelta di un certo passo η reale e positivo, da cui dipende la velocità di convergenza alla soluzione. Il passo η può essere costante o variare per ogni ripetizione, perciò si distinguono metodo stazionario e metodo dinamico.

1.3 Statistica

Alla base di tutti i progetti di Machine Learning, segnatamente nel campo dei *Big Data*, vi è la raccolta dei dati, che devono essere organizzati in modo funzionale in tabelle composte, chiamate *dataset*, le quali contengono le unità statistiche sulle righe, mentre i relativi caratteri, qualitativi e quantitativi, sulle colonne. Successivamente i dati possono essere trattati con i metodi statistici classici, al fine di pulire i dataset, calcolando gli indici di posizione e dispersione delle variabili, oppure di trovare correlazioni tra due o più caratteri.

Nonostante i concetti di statistica univariata e bivariata siano prerequisiti, provvediamo innanzitutto ad un ripasso di quelli fondamentali, tra cui gli indici di dispersione e i coefficienti di regressione, finalizzato alla comprensione dei successivi argomenti, tra cui alcuni elementi di statistica multivariata, che si occupa di fare rilevazioni contemporanee su più di due caratteri di una popolazione al fine di analizzarne le dipendenze.

Nei seguenti paragrafi indicheremo la media aritmetica di un insieme di dati con la lettera μ, le distribuzioni statistiche con le lettere maiuscole X e Y e le loro variabili con le lettere minuscole x e y contrassegnate da un pedice ne che indica la posizione.

1.3.1 Indici di Dispersione

La considerazione della sola media aritmetica comporta una notevole perdita di informazioni rispetto ai dati, perciò introduciamo alcuni indici di dispersione. In un insieme di dati analizzati rispetto ad un carattere, la *deviazione standard*, o *scarto quadratico medio*, misura la dispersione dei dati dalla media μ. L'unità di misura della deviazione standard è uguale a quella dei dati osservati. Infatti, essa è data dalla media quadratica degli scarti dalla media aritmetica:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}.$$

Il radicando dell'espressione della deviazione standard prende il nome di *varianza* e si indica con il simbolo σ^2. Come la deviazione standard, essa è una misura del grado di variabilità di una distribuzione statistica, anche se viene utilizzata più raramente a causa della sua unità di misura, che è pari al quadrato di quella dei dati. Inoltre, il numeratore della varianza, ovvero la somma dei quadrati degli scarti dalla media, è chiamato *devianza*.

Il *coefficiente di variazione* è il rapporto tra la deviazione standard e la media aritmetica. Esso non dipende dall'unità di misura con cui il carattere è espresso; ciò permette di operare confronti tra grandezze dimensionalmente diverse. Viene indicato con $C.V.$ e risulta:

$$C.V. = \frac{\sigma}{\mu}.$$

1.3.2 Ridimensionamento dei Dati

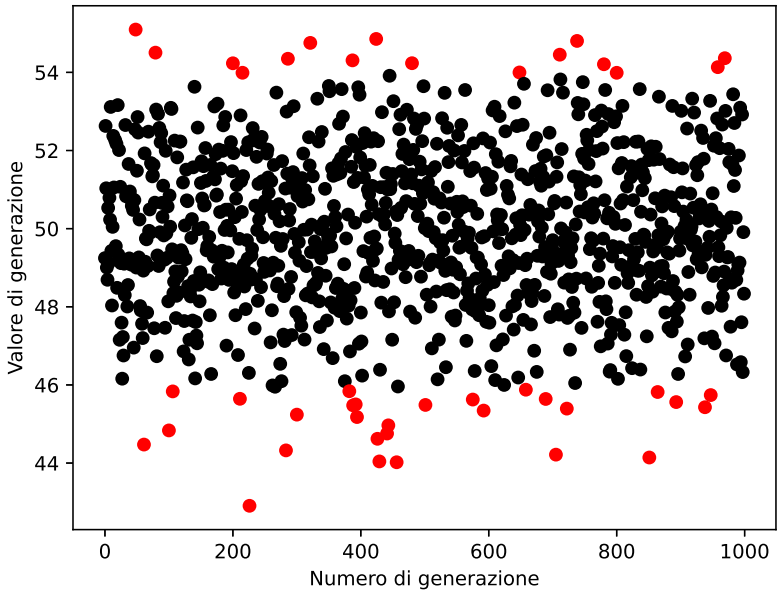
Prima di trattare un dataset, è spesso conveniente procedere col ridimensionamento dei suoi dati, un procedimento particolarmente utile per confrontare variabili con ordini di grandezza diversi, identificare valori anomali chiamati *outliers*, velocizzare l'ottimizzazione da parte degli algoritmi di apprendimento. Esistono essenzialmente due processi statistici che permettono di farlo: standardizzazione e normalizzazione.

Standardizzazione

La standardizzazione consente di ricondurre la distribuzione di una variabile ad una distribuzione standardizzata, ossia dotata di media e varianza pari a due valori fissati; nella maggior parte dei casi, questi valori sono, rispettivamente, 0 e 1. Per fare ciò, è necessario sostituire al valore della variabile x_i quello del suo corrispondente *Z-score*, che si calcola:

$$z_i(x_i) = \frac{x_i - \mu}{\sigma}.$$

Il modulo dello Z-score indica fondamentalmente la distanza, in unità di deviazioni standard, tra la variabile x_i e la media aritmetica della distribuzione X. Esso ci permette di stabilire se un punto è un outlier rispetto agli altri: per farlo è sufficiente definire una soglia dello Z-score oltre alla quale il valore è considerato tale. Chiaramente, il numero di punti che vengono considerati outliers cresce al decrescere del valore che identifica la soglia, poiché il criterio di selezione diventa più limitativo.



I punti sono stati generati con una distribuzione normale con $\mu = 50$, $\sigma^2 = 2$ e soglia dello Z-Score pari a 2. Gli outliers sono indicati con il colore rosso.

Normalizzazione

Grazie alla normalizzazione è possibile limitare l'escursione di un insieme di valori entro un intervallo $[a; b]$, che cambia in base alla natura dei dati, ma che solitamente ha per estremi -1 e 1, oppure 0 e 1. La formula per normalizzare un valore x_i della distribuzione X è:

$$x'_i = a + \frac{(x_i - min(X))(b - a)}{max(X) - min(X)}.$$

1.3.3 Correlazione

La *covarianza* esprime la correlazione statistica, o dipendenza, fra due distibuzioni di variabili X e Y, si indica con σ_{XY} e si ottiene calcolando la media dei prodotti degli scarti, ovvero:

$$\sigma_{XY} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y).$$

In base al segno di σ_{XY} si distinguono i casi:

- $\sigma_{XY} > 0$, all'aumentare di una variabile, aumenta in media anche l'altra;
- $\sigma_{XY} < 0$, all'aumentare di una variabile, diminuisce in media l'altra;
- $\sigma_{XY} = 0$, non c'è dipendenza lineare fra le due variabili.

Nell'utilizzo della covarianza si possono riscontrare principalmente due criticità: essa risente dell'ordine di grandezza delle distribuzioni su cui è calcolata e non possiede la stessa unità di misura dei dati originari. Per questo motivo, ad essa viene affiancata un'altro indice statistico, chiamato *coefficiente di correlazione lineare di Bravais-Pearson*, indicato con la lettera r. Esso è un numero puro compreso tra -1 e 1, ragione per cui non possiede nessuno dei due problemi della covarianza, con la quale è sempre concorde. Esso è definito come segue:

$$r = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}.$$

A seconda del suo valore, si deduce che:

- se $r = -1$, si tratta di correlazione lineare indiretta perfetta;
- se $-1 < r < 0$, la correlazione lineare è inversa: all'aumentare di una variabile diminuisce in media l'altra;
- se $r = 0$, non c'è correlazione lineare, come nel caso in cui $\sigma_{XY} = 0$;
- se $0 < r < 1$, la correlazione lineare è diretta: all'aumentare di una variabile aumenta in media anche l'altra;
- se $r = 1$, si tratta di correlazione lineare diretta perfetta.

In diverse situazioni del Machine Learning può diventare necessario confrontare i coefficienti di correlazione di n variabili contemporaneamente, con $n > 2$; per farlo vengono adottate matrici quadrate simmetriche $n \times n$ chiamate *matrici di correlazione*.

Esiste un altro indice, chiamato *coefficiente di determinazione* e indicato con R^2, che è calcolato elevando al quadrato l'indice di correlazione. A differenza di quest'ultimo, l'indice R^2 non indica la direzione della relazione lineare; esso varia infatti solo tra 0 e 1, dove 0 indica che non c'è correlazione lineare mentre 1 denota una correlazione lineare perfetta diretta o perfetta inversa. Esso è dato dal rapporto fra la devianza del modello di regressione e quella dei dati osservati:

$$R^2 = \frac{\sum_{i=1}^{n} (\bar{y}_i - \mu)^2}{\sum_{i=1}^{n} (y_i - \mu)^2},$$

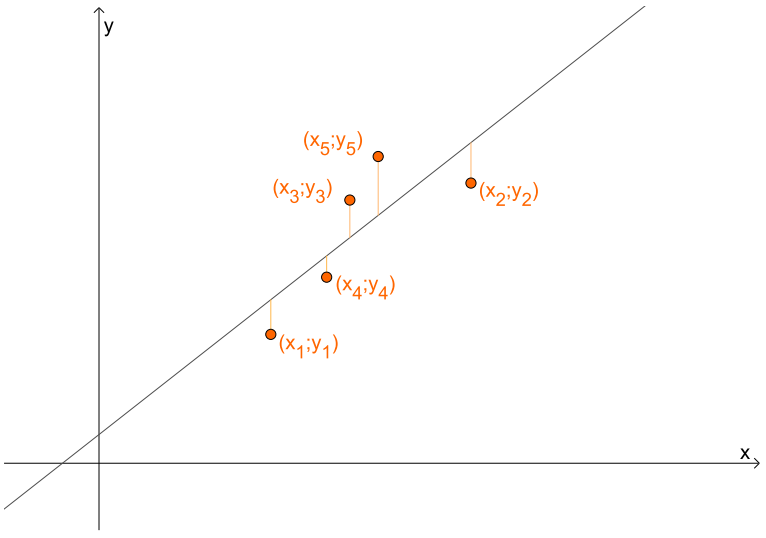
dove $\bar{y}_i$ e y_i sono, rispettivamente, l'i-esimo valore predetto dal modello e l'i-esimo dato osservato. Nel caso di una regressione lineare multipla, l'indice R^2 viene sostituito da un valore corretto, indicato con $\bar{R}^2$, che tiene conto del fatto che, aumentando il numero di variabili indipendenti, dette anche *regressori*, l'indice R^2 non può mai diminuire. Esso è dato dalla formula:

$$\bar{R}^2 = 1 - \frac{n-1}{n-k-1} \cdot \frac{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}{\sum_{i=1}^{n} (y_i - \mu)^2},$$

dove k è il numero dei regressori.

1.3.4 Regressione Lineare

Una metodologia utilizzata dalla statistica multivariata è la *regressione lineare*, che analizza l'influenza che uno o più caratteri esercitano su altri attraverso lo studio della loro correlazione. Nel caso in cui si considerino solamente due variabili x, indipendente, e y, dipendente, il modello prende il nome di *regressione lineare semplice*; se, come sovente accade nel Machine Learning, il numero di variabili indipendenti è maggiore di uno, si parla di *regressione lineare multipla*; se anche il numero di variabili dipendenti è maggiore di uno, la regressione è detta *multivariata*.



Regressione lineare su cinque punti di cui sono rappresentati i *residui*, ovvero gli scarti dalla retta di regressione.

Lo scopo della regressione lineare è quello di trovare una funzione, chiamata *interpolante lineare*, che permetta di rappresentare su un grafico la relazione fra i diversi caratteri. Nel caso di una regressione semplice, la sua forma si può esprimere come l'equazione della retta:

$$y_i = \beta_0 + \beta_1 x_i,$$

dove β_0 rappresenta l'intercetta e β_1 il coefficiente angolare. Valori di β_1 prossimi a zero indicano una scarsa correlazione, mentre valori elevati in modulo indicano una forte correlazione.

Per procedere con il *fitting* della retta di regressione, cioè il processo che individua la migliore interpolazione dei dati, è necessario stabilire una *funzione di costo* con cui calcolare l'errore del modello, in modo tale da poter scegliere la migliore fra le infinite rette di cui si dispone. La funzione di costo può essere, ad esempio, la sommatoria di tutti gli scarti quadratici tra i valori predetti e quelli da predire.

Capitolo 2

L'Informatica del Machine Learning

Secondo la programmazione tradizionale, un programma deve contenere un numero di istruzioni ordinate e precise che vengono eseguite dal computer al fine di generare un output a partire da un input; in questo modo, la macchina non ha alcun grado di libertà, in quanto essa si limita ad eseguire pedissequamente le istruzioni fornite dal suo programmatore. Il Machine Learning rappresenta una variazione rispetto a questo approccio, in quanto la macchina, a cui viene presentato un insieme di dati, apprende autonomamente il modo con cui trattare tali dati al fine di risolvere il problema proposto dal programmatore. Tale apprendimento è generalizzato, ovvero deve poter essere sfruttato anche su nuovi dati non precedentemente considerati.

Questa tecnica di programmazione possiede numerosi vantaggi, come la grande capacità di adattamento e miglioramento e la quantità di interventi del programmatore che, rispetto alla programmazione tradizionale, è notevolmente ridotta grazie all'autonomia del sistema. Tuttavia, i meccanismi con cui un processo di Machine Learning giunge ai suoi risultati possono talvolta essere di difficile comprensione e il numero di dati richiesto da tali processi è decisamente ingente.

Per definire più formalmente il caratteristico modo di procedere del Machine Learning, si consideri un insieme di m esempi, ciascuno associato alla coppia ordinata di vettori (x_i, y_i) con $i \in \mathbb{N}$ e $1 \leq i \leq m$. I vettori x_i e y_i contengono, rispettivamente, i dati di input e di output, e sono legati dalla funzione obiettivo f non nota. Lo scopo del Machine Learning è quello di trovare la funzione che meglio approssimi f, ovvero che permetta di determinare gli output a partire dai vettori input degli esempi appartenenti all'insieme, ma anche di nuovi input.

Un semplice esempio permette di comprendere più a fondo la differenza tra programmazione tradizionale e Machine Learning. Si supponga di voler elaborare un algoritmo che restituisce un output y moltiplicando per 2 il valore dell'input x. Dunque la funzione f che lega input e output è espressa dalla legge $f(x) = 2x$. La programmazione tradizionale prevede che la funzione f sia nota e che venga dunque scritto un codice per cui ad una variabile y viene assegnato il valore $2x$. Nella programmazione con apprendimento automatico, invece, all'algoritmo non viene direttamente fornita la legge della funzione f, poiché esso deve calcolarla autonomamente. Dunque, esso prova ad associare agli input i relativi output mediante una funzione g da lui ideata, in modo tale che quest'ultima riproduca la legge di f. Come si può facilmente intuire, per problemi di questo tipo è sconveniente l'adozione del Machine Learning; tuttavia, con l'aumentare del numero di variabili di input e della complessità del problema, questo approccio risulta di gran lunga migliore.

2.1 Tipologie di Soluzioni

Il Machine Learning è uno degli ambiti dell'informatica con il numero più elevato di applicazioni

nella vita reale; di conseguenza, l'insieme di tutti gli algoritmi presenta una grande eterogeneità al suo interno. In particolare, in base al loro output, i problemi possono essere divisi in tre categorie: di classificazione, di regressione e di clustering.

Classificazione

La *classificazione* prevede che ogni input faccia parte di una particolare classe, o categoria, e lo scopo del sistema classificatore è appunto quello di associare ad ogni input la corretta classe di appartenenza. L'output della classificazione è una variabile categorica che può essere nominale oppure ordinale, a seconda che si riferisca al nome o al numero della classe. La classificazione può essere binaria o multi-classe in base al fatto che le classi da assegnare, ovvero i possibili valori che può assumere l'output, siano due o più di due. Un esempio di classificatore è il sistema che permette di riconoscere, e dunque distinguere, gli oggetti in un'immagine, oppure quello che si occupa della filtrazione dei messaggi spam.

Regressione

La *regressione* è lo stesso processo che viene utilizzato in statistica e ha per output una variabile quantitativa, in particolare un numero reale. Sistemi che si basano sulla regressione sono, per esempio, quelli che analizzano le serie storiche dei valori delle azioni borsistiche. Essa può essere di tipo non lineare se la relazione fra le variabili di input e output viene espressa per mezzo di una curva diversa da una retta.

Clustering

Il *clustering* consiste, invece, nel raggruppamento di elementi di un insieme di dati in classi omogenee, chiamate *clusters*, che devono differenziarsi il più possibile dalle altre dello stesso insieme. Un elemento che distingue il clustering dalla classificazione e dalla regressione è che in questo caso non vengono forniti i valori di output, e sono quindi sfruttate solamente le caratteristiche intrinseche degli input; in questo senso, è simile ad una classificazione privata dell'output. La variabile di output può rappresentare la classe di appartenenza di un dato input oppure la sua distanza dal centro del cluster al quale esso appartiene.

2.2 Modalità di Apprendimento

L'acquisizione dei dati è la prima fase del Machine Learning; essa può avvenire seguendo diversi tipi di procedure, come la rilevazione statistica, il monitoraggio di un processo o il web scraping, ovvero l'estrapolazione di dati a partire dal codice sorgente di una pagina web. Dopo esser stati raccolti, i dati vengono organizzati in un dataset e vengono selezionate le diverse variabili da trattare come input o come output. Affinché il modello di predizione sia il più preciso possibile, è necessario che la quantità di dati sia ragguardevole e che questi siano significativamente inerenti al problema trattato.

Tuttavia, non tutti gli algoritmi lavorano utilizzando sia l'input sia l'output, come esemplificato dalla differenza tra clustering e classificazione. Ne consegue che le modalità con cui la macchina impara sono molteplici; in particolare, l'apprendimento può essere: supervisionato, non supervisionato, semi-supervisionato, che combina i primi due, con rinforzo e con trasferimento. Gli esempi trattati in questo articolo sono tutti riconducibili al primo tipo.

Apprendimento Supervisionato

L'*apprendimento supervisionato* consiste nella ricerca di un modello generale che permetta di associare agli input i relativi output nel modo migliore possibile. Per generare tale modello, il dataset viene diviso in due sezioni, chiamate *training* e *test*, che ne occupano rispettivamente circa il 70%-80% e il 20%-30%. È fondamentale che esse siano omogenee, ovvero che contengano una distribuzione simile di esempi per ogni classe di appartenenza; è questo il motivo per cui vengono stabilite casualmente. Alle due parti del dataset corrispondono due diversi momenti della modellazione: inizialmente, il training lavora sui dati di input e di output e, mediante algoritmi di ottimizzazione, determina i parametri che minimizzano l'errore sulla predizione dell'output. In seguito, il test elabora previsioni su nuovi input, basandosi sui parametri precedentemente calcolati nel training; tali previsioni vengono poi confrontate con i valori reali per calcolare l'errore del modello.

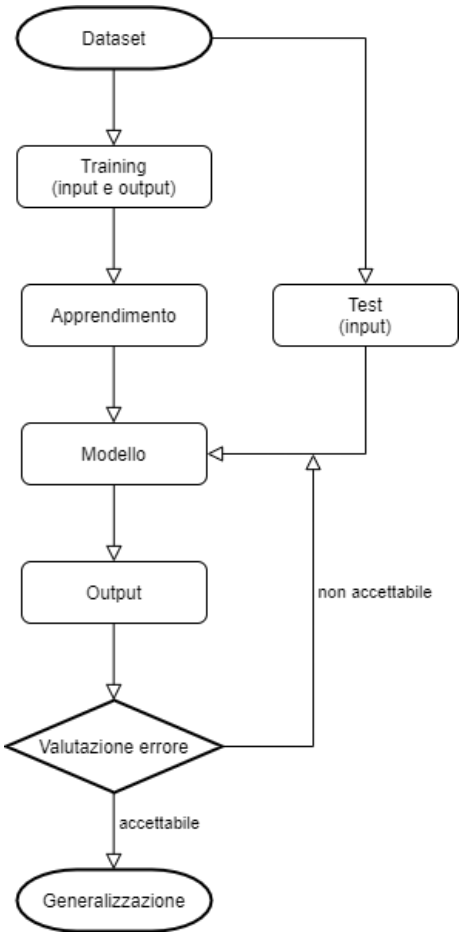
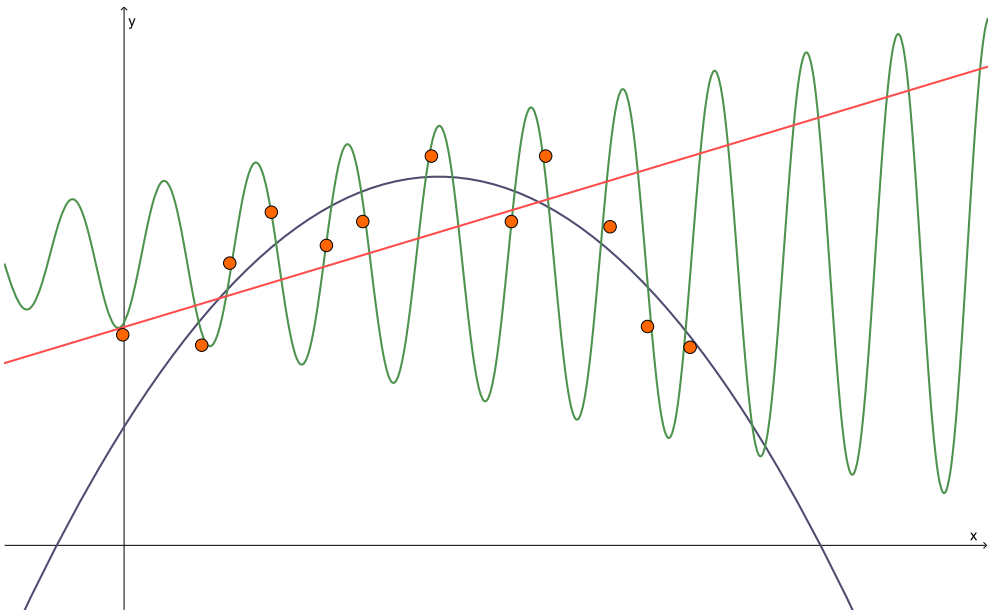


Diagramma di flusso che illustra il processo train-test utilizzato nell'apprendimento supervisionato.

Questo modello può riscontrare due principali criticità, che derivano rispettivamente da un difetto e da un eccesso della fase di training: l'*underfitting* e l'*overfitting*. Nel primo caso, il modello è troppo semplice e approssimativo rispetto ai dati da trattare, e quindi non riuscirà a prevedere adeguatamente né nella fase di training né in quella di test. Nel secondo caso, al contrario, esso è complesso e troppo specifico nei confronti dei dati di training, che vengono appresi ottimamente, ma ha poca capacità di adattarsi, e quindi di generalizzare, sui dati di test; le cause possono essere una selezione di variabili di input non rappresentative oppure il fatto che il training considerato sia troppo piccolo rispetto al dataset.

Per comprendere più a fondo queste due problematiche è necessario introdurre i concetti di *bias*, che è la differenza tra il valore medio della previsione e il valore da predire, e *varianza*, che rappresenta la variabilità della previsione, e quindi la sensibilità del modello rispetto alla casualità dei dati. Modelli con un bias elevato e una varianza bassa segnalano la presenza di underfitting, mentre nel caso opposto, ovvero di bias basso e varianza elevata, il problema è l'overfitting. La soluzione più corretta consiste in un bilanciamento tra la semplicità dell'underfitting e la complessità dell'overfitting; un modello di questo tipo viene detto *well-fitted*.



Esempio grafico di un modello ideale (blu), di uno con underfitting (rosso) e di uno con overfitting (verde).

Apprendimento non Supervisionato

L'*apprendimento non supervisionato* prevede che gli esempi forniti contengano i soli dati di input, e che si sfruttino le caratteristiche intrinseche ad essi. La selezione e l'ordine dei dati che vengono esposti all'algoritmo riveste dunque una grande importanza.

2.3 Preprocessing

La frase "Garbage in, garbage out", molto utilizzata ai tempi dell'invenzione dei primi calcolatori e tradotta in "spazzatura che entra, spazzatura che esce", si riferisce all'impossibilità della macchina di produrre output di qualità, se i dati forniti in input non sono a loro volta di qualità. Nel campo del Machine Learning questo concetto viene tradotto in atto nella fase di *preprocessing*, o *pre-elaborazione dei dati*, in cui questi ultimi vengono puliti e trattati sotto diversi aspetti al fine di renderli ottimali per il tipo di modello che verrà successivamente selezionato. Tra le diverse operazioni preliminari che vengono svolte, vi sono: la strutturazione del dataset, il trattamento dei valori mancanti, chiamati anche missing values, dei valori duplicati e degli outliers, il ridimensionamento dei dati mediante standardizzazione o normalizzazione, la trasformazione di caratteri qualitativi, anche detti categorical values, in quantitativi e l'aumento dei dati a disposizione tramite tecniche di *data augmentation*.

2.3.1 Strutturazione

Prima di compiere questa serie di operazioni, è necessario assicurarsi che il dataset sia strutturato, ovvero che sia organizzato in una matrice contenente nelle colonne i tipi di variabili, o *features*, che descrivono i dati, e nelle righe i relativi valori per ogni osservazione, o esempio. Inizialmente, le immagini non si presentano sotto forma di dataset strutturati, e hanno quindi bisogno di essere trasformate in matrici nelle quali ogni valore rappresenta un pixel; se si tratta di immagini in bianco e nero, ogni pixel fa riferimento ai valori di una scala di grigio che varia da 0 (nero) a 255 (bianco), mentre nel caso di immagini a colori viene adottato il modello RGB (Red-Green-Blue), che si basa sulle gradazioni di rosso, verde e blu.

2.3.2 Bilanciamento

Se il problema in esame è di classificazione, è conveniente prestare attenzione al *bilanciamento* del dataset, ovvero al fatto che le classi di output siano distribuite omogeneamente al suo interno; qualora dovessero esserci degli squilibri, una corretta classificazione sarebbe ampiamente compromessa. Per risolvere questo problema, si profilano due soluzioni antitetiche: il *sottocampionamento*, una riduzione della quantità di esempi che si riferiscono alla classe più rappresentata, e il *sovracampionamento*, un aumento, tramite data augmentation, dei dati della classe con la frequenza minore.

2.3.3 Selezione e Riduzione delle Caratteristiche

Una delle fasi fondamentali del preprocessing consiste nella *selezione delle caratteristiche* dell'oggetto, che verranno successivamente prese in considerazione dal modello. Esse possono essere quantitative o qualitative, costanti o variabili, ma in ogni caso devono possedere un'ottima capacità di discriminazione, ovvero devono poter distinguere gli oggetti in classi omogenee al loro interno ma il più possibile diverse le une dalle altre, e inoltre devono essere indipendenti, cioè tra le features non deve esserci una forte correlazione, o *collinearità*. Nonostante possa sembrare intuitivo che un aumento del numero di features sia legato ad una maggiore correttezza del modello predittivo, esiste un numero ottimale di features che non bisogna superare, onde evitare che il training sia insufficiente e che quindi si verifichi una situazione di underfitting.

Per la selezione delle features si possono seguire due metodi: *feature selection*, che consiste nel sceglierne un sottoinsieme rispetto all'insieme originale, e *feature extraction*, nel quale diverse features vengono combinate e unite, perdendo la dimensione fisica reale dei dati. Entrambi gli approcci richiedono l'utilizzo di tecniche di riduzione della dimensionalità, come la *Principal Component Analysis*, che si basa sull'utilizzo dell'algebra lineare per selezionare le componenti principali che contengono maggiori informazioni sui dati, oppure la *Linear Discriminant Analysis*. Le features possono poi essere contenute in vettore riga di dimensione N, che viene rappresentato in uno spazio N-dimensionale.

2.3.4 Trasformazione dei Caratteri Qualitativi

Gli algoritmi adottati dal Machine Learning possono lavorare esclusivamente con quantità numeriche, dunque è necessario intervenire sul dataset al fine di trasformare tutti i caratteri qualitativi, come stringhe di testo o variabili booleane, in quantitativi. Per esempio, nel caso di una stringa, si può associare ciascuna lettera al relativo numero indicato dal codice ASCII, oppure la si può considerare come un riferimento ad una particolare categoria espressa da un valore numerico; in quest'ultimo caso, il carattere viene distinto in *categoriale* e *ordinale*, che, a differenza del primo, si riferisce a categorie ordinate. Esempi di caratteri categoriali sono i gruppi sanguigni e il sesso biologico, mentre

i giudizi scolastici e le categorie reddituali sono ordinali. In particolare, i metodi di codifica principali sono i seguenti:

- Label Encoding: assegna ad una variabile categorica un numero intero da 1 a n, dove n è il numero di valori distinti assunti dal carattere qualitativo;
- Ordinal Encoding: funziona similmente al Label Encoding, con la differenza che i caratteri qualitativi sono ordinali;
- One Hot Encoding: genera un vettore riga di lunghezza n, dove n è il numero di modalità distinte assunte dal carattere qualitativo ordinale. L'i-esimo elemento del vettore, con $0 < i \leq n$, è 1 se il carattere si esprime nell'i-esima modalità, è 0 altrimenti; dunque il vettore contiene solo un elemento di valore 1, mentre gli altri $n-1$ elementi valgono 0. Questo metodo evita gli equivoci derivanti dall'ordinamento numerico, ma la creazione di nuove colonne potrebbe rallentare il successivo apprendimento da parte del modello predittivo.

Esempio 2.1

Categoria	Rosso	Verde	Blu	Giallo
Giallo	0	0	0	1
Giallo	0	0	0	1
Rosso	1	0	0	0
Blu	0	0	1	0
Rosso	1	0	0	0
Verde	0	1	0	0

Altre tecniche più avanzate vengono utilizzate per l'elaborazione del linguaggio naturale, abbreviato dall'inglese *Natural Language Processing* in NLP; in genere esse consistono nella creazione di un vocabolario costituito dalle parole più significative del testo da analizzare.

2.3.5 Missing Values e Outliers

Se un'unità statistica possiede un missing value in una o più delle sue variabili, possono essere compiuti due tipi di operazioni. La prima è l'*imputazione*, ovvero la sostituzione del valore assente con un valore fisso o con un indice di posizione, come moda o mediana, calcolato sulla distribuzione del carattere corrispondente; in alternativa, si può procedere con l'eliminazione di tutta l'unità statistica, che porta a considerare solo quelle che non hanno missing values per una determinata variabile. Solitamente, se i caratteri contenenti valori mancanti sono più di uno, si applicano entrambe le operazioni, scegliendo di volta in volta in base al loro tipo.

Alcune variabili del dataset possono contenere dei valori anomali rispetto a quelli della loro distribuzione, ed essi vengono trovati mediante algoritmi di *outlier detection*, che possono basarsi, per esempio, sullo Z-score. Una volta individuati tutti gli esempi contenenti outliers, si calcola il loro rapporto rispetto alla totalità del dataset e si procede cancellando tali esempi oppure sostituendone i valori considerati outliers con altri.

2.3.6 Ridimensionamento dei Dati

Il ridimensionamento dei dati, o *feature scaling*, viene applicato alle sole variabili di input e permette di velocizzare notevolmente la successiva ottimizzazione dei modelli predittivi, che funzionano meglio quando i dati hanno una scala relativamente simile. Non esistono particolari regole empiriche in grado di stabilire se sia meglio procedere con il feature scaling mediante standardizzazione oppure normalizzazione; il metodo migliore è dunque sperimentare ambedue le tecniche e comparare i risultati ottenuti. Le tecniche a disposizione sono la standardizzazione, se il valore viene sostituito con il suo Z-score rispetto alla distribuzione della variabile cui fa riferimento, e la normalizzazione, che solitamente viene utilizzata per limitare l'escursione dei dati all'intervallo $[0; 1]$, oppure in modo che il massimo assoluto della distribuzione assuma valore 1. Quest'ultima è particolarmente sensibile agli outliers, e per questo motivo è adeguata quando la deviazione standard è molto piccola; la normalizzazione in base al massimo ha la peculiarità di non modificare i valori 0 della distribuzione.

2.4 Modelli del Machine Learning

Una volta completata la fase di preprocessing, occorre trovare un modello che traduca i dati di input in valori di output nel modo più corretto possibile. Esistono modelli utilizzati specificamente per i problemi di regressione, come la discesa stocastica del gradiente, o di classificazione, come il k-Nearest Neighbors, mentre altri, come gli alberi di decisione, possono essere adottati indifferentemente per la regressione e per la classificazione.

In particolare, i metodi di ottimizzazione dei modelli per la regressione si dividono in analitici e numerici; i primi usano delle equazioni le cui soluzioni sono definite, mentre i secondi trovano dei valori attraverso una ricorsione che si ferma quando l'approssimazione scende sotto una certa soglia.

I modelli vengono valutati durante la fase di training attraverso funzioni di costo, la cui derivazione indica come modificare i parametri per minimizzare l'errore del modello; un esempio di funzione di costo è l'*errore quadratico medio*, che viene calcolato tra gli output predetti dal modello e quelli attesi.

2.4.1 Metodo dei Minimi Quadrati

Il *metodo dei minimi quadrati*, abbreviato dall'inglese *Ordinary Least Squares* in OLS, è uno degli algoritmi di ottimizzazione analitici più utilizzati nel Machine Learning per determinare il coefficiente angolare e l'intercetta di una retta di regressione, la cui equazione è

$$\bar{y_i}(m, b) = mx_i + b;$$

mentre la funzione di costo e da minimizzare, ovvero l'errore sulla totalità delle predizioni, è dato dalla devianza dei residui, ossia

$$e(m, b) = \sum_{i=1}^{n} (\bar{y_i}(m, b) - y_i)^2 = \sum_{i=1}^{n} (mx_i + b - y_i)^2,$$

di cui si può calcolare il gradiente:

$$\nabla e = \begin{bmatrix} e'_m \\ e'_b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} 2x_i(mx_i + b - y_i) \\ \sum_{i=1}^{n} 2(mx_i + b - y_i) \end{bmatrix}.$$

Per il teorema di Fermat una funzione derivabile ovunque ha un punto di minimo relativo solo se il gradiente della funzione in quel punto è uguale a 0, dunque è necessario calcolare le derivate parziali della funzione e porle tutte pari a 0. Nei successivi calcoli l'indice e l'intervallo della sommatoria verranno trascurati.

$$\nabla e = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{cases} \sum (mx_i^2 + bx_i - x_i y_i) = 0 \\ \sum (mx_i - y_i + b) = 0 \end{cases} \Leftrightarrow \begin{cases} \sum mx_i^2 + \sum bx_i - \sum x_i y_i = 0 \\ \sum mx_i - \sum y_i + \sum b = 0 \end{cases} \Leftrightarrow$$

$$\begin{cases} m\sum x_i^2 + b\sum x_i - \sum x_i y_i = 0 \\ m\sum x_i - \sum y_i + nb = 0 \end{cases} \Leftrightarrow \begin{cases} m\sum x_i^2 + b\sum x_i - \sum x_i y_i = 0 \\ b = \frac{\sum y_i - m\sum x_i}{n} \end{cases}$$

da cui, sostituendo opportunamente e svolgendo i calcoli, si ottengono i seguenti valori di m e b:

$$m = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2},$$

$$b = \frac{\sum y_i - m\sum x_i}{n}.$$

2.4.2 Discesa Stocastica del Gradiente

Il metodo OLS consente di trovare il valore esatto dei migliori coefficienti di regressione; tuttavia, applicando questa tecnica a casi più complessi, come una regressione lineare multipla, l'efficienza dell'algoritmo si riduce notevolmente: esso non è facilmente scalabile. Dunque, è conveniente usare un metodo di ottimizzazione numerico, la *discesa stocastica del gradiente*, abbreviata in SGD dall'inglese *Stochastic Gradient Descent*. Esso consente di determinare il minimo di una superficie di errore E attraverso successive modifiche dei pesi in base all'opposto del gradiente, $-\nabla E$, ovvero la direzione di massima discesa. Tuttavia, il gradiente non si annulla solo in corrispondenza del minimo assoluto, la posizione che ottimizza i parametri, ma anche negli eventuali punti di minimo locale e nei flessi.

Si consideri la regressione lineare multipla data dall'equazione

$$\bar{z_i} = \omega_0 + \omega_1 x_1 + \omega_2 x_2,$$

in cui i parametri $\omega_0, \omega_1, \omega_2$ sono componenti del vettore $\vec{\omega}$, e la relativa funzione di costo $E : \mathbb{R}^2 \to \mathbb{R}$, calcolata attraverso l'errore quadratico medio, di equazione

$$E(\vec{\omega}) = \frac{1}{n}\sum_{i=1}^{n} (z_i - \bar{z_i})^2 = \frac{1}{n}\sum_{i=1}^{n} (z_i - (\omega_0 + \omega_1 x_1 + \omega_2 x_2))^2.$$

Ad ogni passo, il vettore dei pesi varia nella direzione di massima discesa, in modo tale da minimizzare la superficie di errore E, seguendo la legge $\vec{\omega}_{n+1} = \vec{\omega}_n - \eta \nabla E(\vec{\omega})$, dove η è il passo della discesa del gradiente e n è l'indice della successione. Perciò risulta che

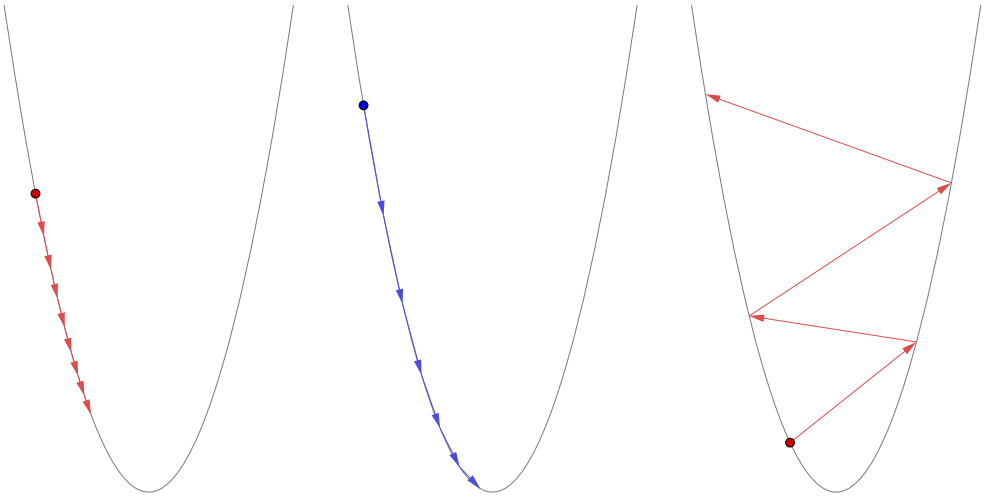
$$\Delta\vec{\omega} = \vec{\omega}_{n+1} - \vec{\omega}_n = -\eta \nabla E(\vec{\omega}),$$

e per ogni componente ω_i del vettore $\vec{\omega}$ si può scrivere

$$\Delta\omega_i = -\eta \frac{\partial E}{\partial \omega_i}.$$

Varianti della SGD

Poiché la discesa stocastica del gradiente è indispensabile per vari ambiti del Machine Learning, tra cui l'addestramento delle reti neurali, con il passare del tempo sono state elaborate numerose varianti, anche straordinariamente recenti, come nel caso di AdaGrad, pubblicato solamente nel 2011. Una caratteristica fondamentale di questi metodi è la presenza di più iperparametri da ottimizzare: tra questi, un esempio è proprio il già citato parametro η, il cui valore, se troppo grande, determina una scarsa capacità di convergenza e, se troppo piccolo, implica una convergenza molto lenta.



Nella prima figura il learning rate è troppo piccolo e il processo di ottimizzazione è eccesivamente lento, nella figura centrale esso ha un valore adeguato, nella terza figura il valore è troppo grande e il parametro non converge al minimo locale.

La variante più basilare dell'SGD prevede l'utilizzo di un tasso di apprendimento adattivo, o dinamico, cioè una funzione che decresce alla crescita del numero di iterazioni della discesa del gradiente. Si è verificato, tuttavia, che in molti ambiti, tra cui proprio i problemi di Machine Learning, questa estensione ha una debole capacità di generalizzazione, e che risultano più efficienti altri metodi.

Il momento, ad esempio, è una variante che considera tutte le iterazioni e aggiorna il parametro ω usando una combinazione lineare del gradiente e del parametro all'iterazione precedente, cioè:

$$\Delta\omega_{i+1} = \omega_{i+1} - \omega_i = \alpha\Delta\omega_i - \eta\nabla E(\omega_i) \Leftrightarrow \omega_{i+1} = \omega_i + \alpha\Delta\omega_i - \eta\nabla E(\omega_i),$$

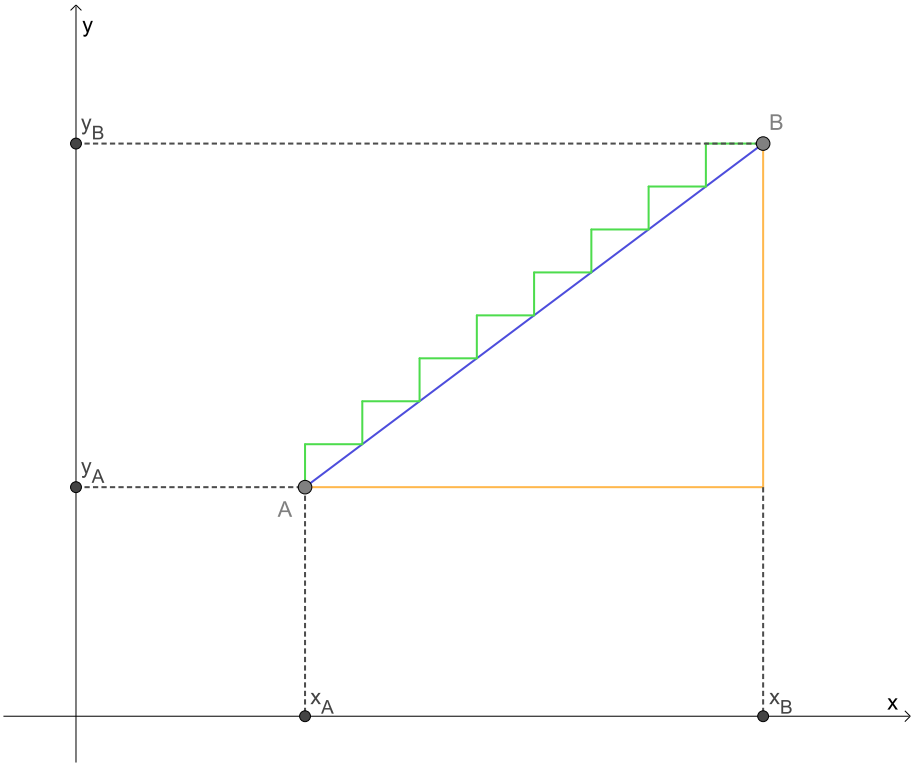
dove α è un iperparametro compreso tra 0 ed 1 che controlla il peso delle iterazioni passate. Un'estensione del momento è il modello NAG, che, correggendo l'espressione di ω_i in $\omega_i + \alpha\omega_i$, approssima la posizione futura dei parametri, ossia previene variazioni eccessive, migliorando di conseguenza i risultati del training.

2.4.3 Modello k-Nearest Neighbors

Un algoritmo utile per i metodi di classificazione è il *k-Nearest Neighbors*, che permette di valutare un nuovo elemento c all'interno di un dataset di n vettori, formato da elementi già suddivisi nelle loro classi. Gli elementi del dataset devono poter essere rappresentati in uno spazio m-dimensionale, dove m rappresenta il numero di caratteri dei dati; inoltre, all'interno di tale spazio deve essere definita una metrica, il metodo di calcolo della distanza tra due punti, grazie alla quale si può definire l'insieme U dei k punti più vicini a c. Poiché si presuppone che punti appartenenti alla stessa classe abbiano

caratteristiche simili, e di conseguenza si trovino relativamente vicini nello spazio, al punto c viene assegnata la classe C con più riscontri, o voti, nell'insieme U.

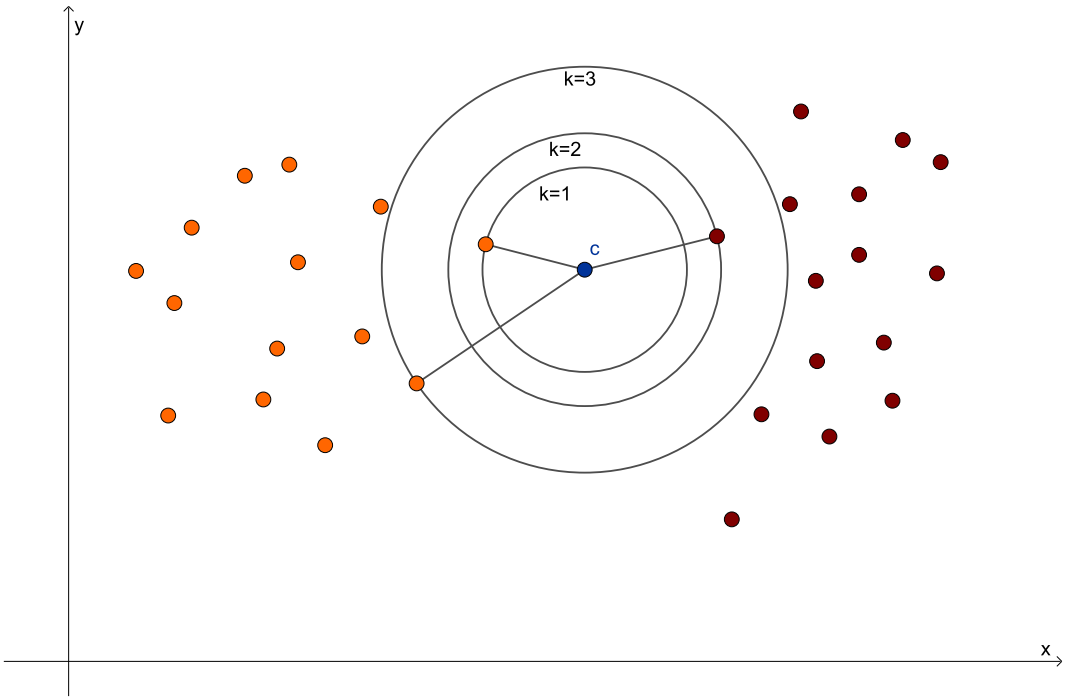
Le metriche utilizzate possono essere di vario genere, anche non vettoriali, in quanto è sufficiente un criterio per stabilire quanto due punti sono vicini. Le più comuni sono: quella euclidea, la metrica più semplice, che associa alla distanza tra due punti la lunghezza del segmento avente per estremi i punti stessi, quella di Manhattan, secondo cui la distanza tra i due punti è uguale alla somma del valore assoluto delle differenze delle loro coordinate, e quella di Minkowsky, una generalizzazione delle due precendenti.



La linea blu indica la distanza euclidea tra i due punti A e B, la linea spezzata verde ha la stessa lunghezza della spezzata arancione ed entrambe indicano la distanza di Manhattan.

La precisione del metodo dipende dall'iperparametro k, di cui si cerca il valore che ottimizzi il procedimento. Se k è molto piccolo, l'insieme U è formato da pochi elementi, ciascuno dei quali assume quindi molto valore rendendo il metodo eccessivamente sensibile; in particolare, se $k = 1$ si ottiene un diagramma di Voronoi, una tassellatura che delimita le regioni di spazio più vicine a ciascun elemento del dataset. Se k è molto grande, le stime risultano poco accurate. Per evitare questi due casi, a k si può provare ad assegnare un valore prossimo a $\sqrt{n}$. In un articolo, è stato dimostrato che, per un parametro k accettabile e un numero sufficientemente grande di dati, il k-NN ha al massimo un tasso di errore doppio rispetto a quello di un classificatore ottimale. Tuttavia si devono considerare alcuni aspetti che influiscono sull'efficacia del metodo.

Il k-NN ha molti vantaggi, tra cui la semplicità di costruzione, la velocità di calcolo e la necessità di ottimizzare unicamente il parametro k, ma perde rapidamente accuratezza al crescere del numero di caratteri, si confonde in caso di caratteri non significativi, che, sebbene non abbiano grande rilevanza, rendono simili due dati avvicinandoli nello spazio, e necessita di calcolare le distanze dell'insieme in input con ogni punto dello spazio.

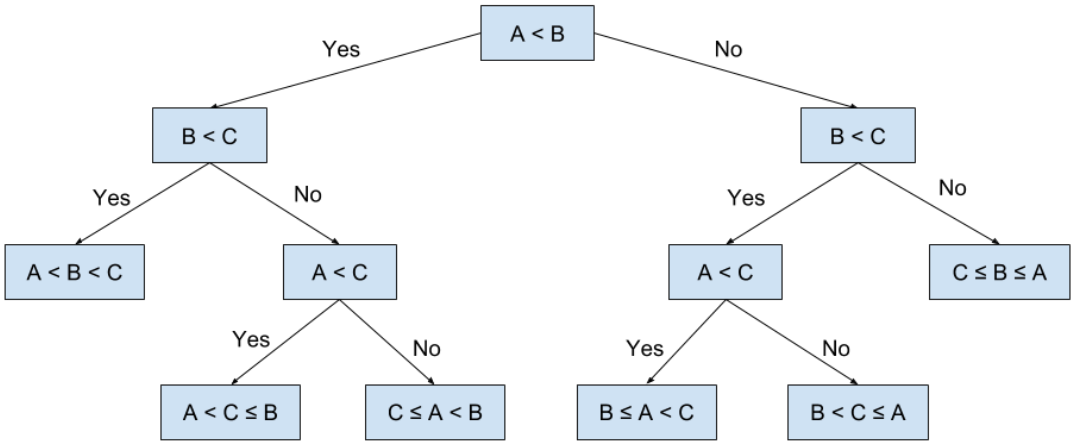


La figura mostra i punti più vicini a c per $k = 1$, $k = 2$, $k = 3$. In quest'ultimo caso, al punto c viene attribuita la classe dei punti arancioni.

2.4.4 Alberi di Decisione

Un *albero di decisione*, o *decisione tree*, è un modello che permette di prendere una decisione su un input in base ad una serie di domande sui valori assunti dalle sue caratteristiche. È costituito da un certo numero di nodi contenenti le informazioni, collegati tra loro da archi in modo tale da non formare cicli. Si distinguono tre tipi di nodi: quello da cui si sviluppano tutti gli altri si chiama radice e non ha archi entranti, quelli che presentano un solo arco entrante ed uno o più archi uscenti sono i nodi intermedi, mentre quelli terminali, privi di archi uscenti, sono detti foglie. Inoltre, i nodi sono distribuiti su diversi livelli; il numero massimo di nodi collocati sullo stesso livello è chiamato larghezza dell'albero. Un cammino, o path, è il percorso impiegato per spostarsi da un nodo ad un altro. La lunghezza di un cammino è data dalla quantità di livelli attraversati; il massimo della lunghezza dei cammini dalla radice alle foglie è detta altezza dell'albero.

Una struttura molto utile per la gestione dei dati è l'albero binario, in cui ciascun nodo ha al massimo due nodi discendenti. Essa infatti permette di decidere il percorso rispettando gli esiti di successive sequenze di selezione che agiscono sui valori delle variabili. Ciascun percorso possibile è caratterizzato da una serie di risposte alle domande poste nei nodi; le foglie dell'albero contengono gli output, cioè le risposte finali, ciascuna delle quali può essere la classe di appartenenza dell'esempio, nel caso di una classificazione, o i parametri della regressione.



Un esempio di classificazione di tre variabili mediante un albero di decisione.

Durante la fase di training, per massimizzare l'efficienza del processo, si devono porre nei vari nodi le domande che minimizzino i percorsi; per svolgere questa operazione è necessario un criterio che permetta un confronto tra di esse. Un nodo che contiene una condizione che è soddisfatta da tutti i dati o non è soddisfatta dal alcun dato è puro; tuttavia, nella maggior parte dei casi ciò non accade, perciò i nodi sono impuri. Per misurare e confrontare l'impurità dei nodi si usa il metodo Gini, che fornisce un valore strettamente compreso tra 0 e 1. La domanda che presenta il minimo indice ha un livello di impurità minore, ed è perciò ottimale.

Una tecnica utilizzata per incrementare ulteriormente la precisione del metodo è la foresta di alberi, o random forest, che consiste nella creazione di un certo numero di alberi ai quali vengono assegnati parametri distinti e che vengono allenati diversamente, generando percorsi differenti. L'output finale è il risultato che, al termine del processo, ha conquistato più voti.

2.5 Valutazione Complessiva del Modello

Per poter determinare la correttezza di un modello è necessario procedere alla sua valutazione, che avviene principalmente in tre fasi: durante il training, a seguito del training e durante l'utilizzo da parte dell'utente finale. È importante sottolineare come le prime due fasi siano profondamente diverse tra loro: mentre la prima, della quale si è già parlato nella sezione precedente, calcola l'errore sulla singola osservazione del training per procedere dunque all'ottimizzazione dei parametri, la seconda, invece, compie una valutazione complessiva sulle osservazioni del test, e calcola la capacità di generalizzazione del modello, che permette di scegliere se utilizzarlo o modificarlo.

Segue un elenco di alcune tra le principali metriche utilizzate per la valutazione complessiva dei modelli; oltre ad esse, è necessario stabilire quale debba essere l'errore tollerabile da non superare.

2.5.1 Classificazione

L'accuratezza misura il rapporto tra le classificazioni corrette e quelle eseguite; chiaramente, essa ha un valore compreso tra 0 e 1, il quale rappresenta una classificazione perfetta. Tuttavia, questa

metrica non è particolarmente indicativa qualora non vi sia grande eterogeneità tra le classi del dataset; per questo motivo essa viene affiancata dall'accuratezza media per classe.

Un'altra metrica della classificazione è il *log-loss*, che viene usata quando l'output è costituito da una serie di valori, compresi tra 0 e 1, che esprimono la probabilità di appartenenza dell'esempio a ciascuna classe; tale appartenenza ha come come condizione il superamento di un valore soglia della probabilità, comunemente impostato a $0,5$.

2.5.2 Regressione

L'*errore assoluto medio*, abbreviato in MAE, è una metrica che non prende in considerazione la grandezza delle osservazioni, ma solamente gli errori assoluti ad esse relativi; per questo motivo, esso aumenta con la grandezza delle osservazioni su cui è calcolato. Inoltre, la funzione che calcola il MAE non è derivabile, in quanto presenta un punto angoloso e perciò non può essere utilizzata anche come funzione di costo.

Una metrica che viene invece adottata, oltre che per la valutazione del test, come funzione di costo, è l'errore quadratico medio, abbreviato in MSE. Nonostante esso sia una delle metriche più comuni, può portare facilmente ad una sottovalutazione o sopravvalutazione dell'errore, causati dal suo elevamento al quadrato. L'MSE può essere sostituito dall'RMSE, che è la sua radice quadrata; questa operazione, tra le altre cose, rende l'errore della stessa unità di misura del valore da predire. Il coefficiente di determinazione, di cui si è parlato nella sezione di statistica, è un'altra metrica utilizzata per valutare i modelli di regressione.

Capitolo 3

Una Applicazione

La ricerca sul Machine Learning e sui temi ad esso inerenti, come Intelligenza Artificiale e Deep Learning, è in continua evoluzione; attualmente questi ambiti sono tra i più promettenti nel campo dell'informatica applicata.

Uno dei problemi più comuni che viene trattato mediante tecniche di Machine Learning consiste nel calcolo del prezzo degli immobili a partire da alcune loro caratteristiche, quali il numero di locali e di bagni, la metratura e la disponibilità di uno o più posti auto; poiché il prezzo è un dato di tipo numerico qualitativo e non una classe, il problema è di regressione. Nonostante questo tipo di previsione sia resa difficoltosa dal fatto che ad attribuire i prezzi agli immobili sia una moltitudine di soggetti, è comunque possibile costruire un modello il cui errore sia tollerabile, come verrà esposto in seguito.

3.1 Dataset di Riferimento

Il dataset di riferimento è stato costruito attraverso tecniche di web scraping sul sito Immobiliare.it, all'interno del quale è possibile eseguire ricerche filtrate e ordinate degli immobili. In particolare, sono state considerate 1418 case tra i quartieri Bolognina, Barca, Santa Viola, Borgo Panigale, Noce-Pescarola, San Donato, Pilastro e Corticella della città metropolitana di Bologna. Ogni abitazione viene descritta dalle seguenti 11 caratteristiche: quartiere di appartenenza, superficie in m^2, numero di locali, bagni e posti auto, presenza di un balcone, di una cantina o del giardino, condizione generale dell'immobile, classe energetica e prezzo in euro. Da questo momento indicheremo le variabili con l'iniziale maiuscola, per una maggiore chiarezza.

Le prime 10 variabili sono di input, mentre l'ultima, Prezzo, è l'output che il programma dovrà predire. Esse sono sia quantitative sia qualitative, ed è dunque necessario effettuare delle trasformazioni su queste ultime; in tutti i casi, i caratteri si prestano per un ordinal encoding, poiché rappresentano dati ordinabili. La variabile Quartiere viene codificata in ordine crescente in base al rapporto medio tra prezzo e superficie delle case di ciascun quartiere. Le variabili Balcone, Cantina e Giardino vengono invece codificate come 0 o 1 in base alla presenza o all'assenza di tali ambienti. Nella tabelle seguenti è contenuta invece la codifica delle variabili che si riferiscono allo Stato generale dell'immobile ed alla sua Classe energetica.

Stato	Valore
Da ristrutturare	0
Buono o abitabile	0
Ottimo o ristrutturato	0,1
Nuovo o in costruzione	0,8

Quartiere	Valore
Borgo Panigale	2,4
Pilastro	3
San Donato	4
Santa Viola	4
Corticella	4
Barca	4,3
Noce - Pescarola	4,6
Bolognina	7
Aeroporto	2,4

Per quanto riguarda la pulizia dei dati, la prima operazione da compiere è rimuovere gli immobili che sono stati inseriti accidentalmente nella ricerca e che sono identificabili in quanto presentano delle categorie anormali nella variabile Quartiere; si procede analogamente per quelli con 4 o più bagni o con 3 o più posti auto. I valori oltre i quali queste variabili provocano la rimozione dell'immobile sono stati scelti arbitrariamente in modo da escludere casi straordinari.

Se un'immobile contiene invece missing values, esso viene eliminato indiscriminatamente; infatti, l'elevata quantità di esempi da cui è formato il dataset, che al termine di tutte le operazioni di preprocessing passa da 1418 a 671 immobili, garantisce un numero sufficiente di dati. Naturalmente, si procede anche all'eliminazione degli outliers, che vengono considerati tali se hanno Z-score maggiore di $1,5$ nelle variabili Superficie, Prezzo, o nel rapporto tra prezzo e superficie, il cui calcolo è successivo all'acquisizione dei dati.

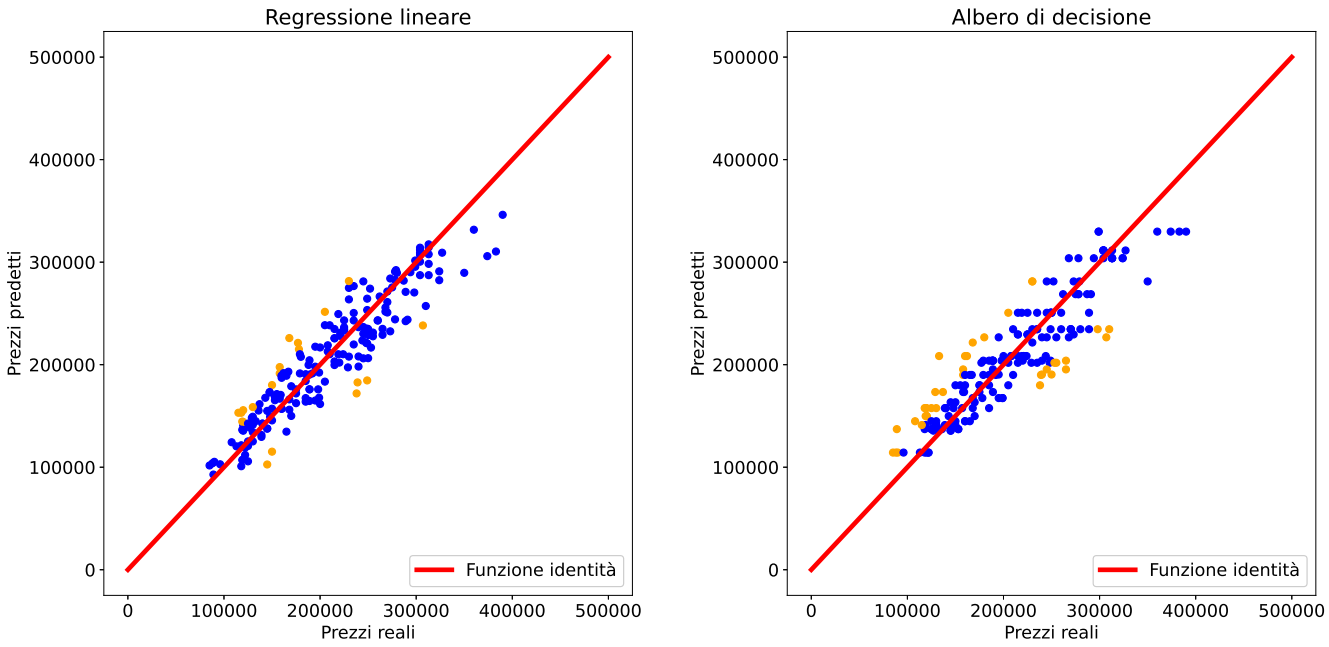
Un problema molto importante che si può presentare in un dataset di questo tipo è la collinearità tra caratteristiche in input; in particolare, le variabili Locali e Superficie hanno un indice di correlazione lineare di Pearson uguale a $0,76$, che indica dunque la superfluità del numero di locali. La tabella sottostante mostra, invece, la correlazione tra i dati in input e quelli in output; intuitivamente, Superficie è la variabile che influenza maggiormente il prezzo. Al contrario, Cantina e Balcone possono non essere considerate, dal momento che la loro correlazione con il prezzo è pressoché nulla.

Features	Quartiere	Superficie	Bagni	Balcone	Cantina	Garage	Giardino	Stato	Energia
Prezzo	0,26	0,86	0,71	0,01	0,08	0,22	0,31	0,36	0,39

Una volta ultimate tutte queste operazioni di pulizia dei dati, si procede al loro ridimensionamento tramite standardizzazione.

3.2 Risultati

Il linguaggio di programmazione impiegato è Python, di cui sono state usate varie librerie, tra le quali Beautiful Soup, per il web scraping, Pandas, per la gestione del dataset, e Scikit-learn, per le tecniche di Machine Learning. Diversi modelli sono stati utilizzati per risolvere questo problema, tra cui alcuni precedentemente spiegati, come la regressione lineare, con il metodo dei minimi quadrati, e gli alberi di decisione. Le metriche tenute in considerazione sono l'indice di Pearson, l'R^2 e l'errore medio percentuale, abbreviato in MAPE.



Ciascun punto dei grafici rappresenta un immobile dell'insieme di test; i punti arancioni indicano immobili con un MAPE maggiore del 20%.

I risultati ottenuti sono schematizzati nella tabella seguente.

	Regressione lineare	Albero di decisione
Pearson	0,94	0,90
R^2	0,89	0,82
MAPE	10%	11,5%

Come si può osservare, in entrambi i casi, l'indice di correlazione lineare aumenta significativamente rispetto a quello che le singole caratteristiche di input hanno con la variabile Prezzo, passando da un valore di $0,86$ a $0,94$.

Indice

Bibliografia

- Lanconelli E., *Lezioni di Analisi Matematica 2*, Pitagora Editore
- Marmo R., *Algoritmi per l'Intelligenza Artificiale*, Hoepli
- Abeasis S., *Elementi di Algebra Lineare*, Zanichelli
- Bergamini M., *Manuale Blu 2.0 di Matematica*, volume 3B, Zanichelli
- Bergamini M., *Fondamenti di Matematica*
- *Corso completo di Data Science e Machine Learning con Python*, Udemy
- Enciclopedia della Matematica Treccani
- Dizionario delle Scienze Fisiche Treccani
- Dipartimento di Economia dell'Università di Bologna