



Pedro Domingos

L'Algoritmo Definitivo

La macchina che impara da sola
e il futuro del nostro mondo

Scienze

Bollati Boringhieri

molta strada nell'epoca dei big data, nemmeno se la mole di dati non è poi così «big». Gran parte dei fenomeni che cercate di rappresentare con un modello dettagliato richiedono un approccio non lineare che solo il machine learning può offrirvi e che porta con sé una visione scientifica del mondo di tipo nuovo. Oggi tendiamo a parlare di *mutamento di paradigma* con facilità eccessiva, ma credo di non esagerare se dico che è proprio questo l'argomento del libro.

Se siete esperti di machine learning, gran parte del contenuto del libro vi sarà già familiare, ma potrete ugualmente trovarvi idee originali, curiosità storiche, esempi e analogie utili. Soprattutto, però, spero che il libro offra un punto di vista nuovo sull'argomento, o che addirittura vi stimoli a spingere le vostre riflessioni su strade nuove. I frutti sui rami più bassi sono sempre a portata di mano ed è giusto coglierli, ma non dovremmo mai perdere di vista gli obiettivi più ambiziosi che aspettano di essere raggiunti appena un po' più in là (a tale proposito, spero che perdonerete la licenza poetica che mi sono concesso utilizzando l'espressione «Algoritmo Definitivo» per parlare di un learner multiuso).

Se siete studenti - liceali ancora incerti sul percorso universitario, laureandi che si chiedono se dedicarsi alla ricerca, o professionisti affermati che stanno pensando a un'evoluzione della propria carriera - spero che il libro accenda in voi l'interesse per questo settore affascinante. Il mondo ha un bisogno disperato di esperti di machine learning: se deciderete di unirvi a noi, vi attendono momenti entusiasmanti e soddisfazioni materiali, e avrete un'opportunità irripetibile di contribuire al benessere della società. Se vi state già dedicando all'argomento, spero che il libro vi aiuti ad apprezzarne le caratteristiche; e se nel vostro viaggio dovesse capitare di imbattervi nell'Algoritmo Definitivo, vorrà dire che ho fatto bene a scriverlo.

Un'ultima cosa, ma non meno importante: se cercate qualcosa che vi stupisca, il machine learning è quello che fa per voi. Vi aspettiamo!

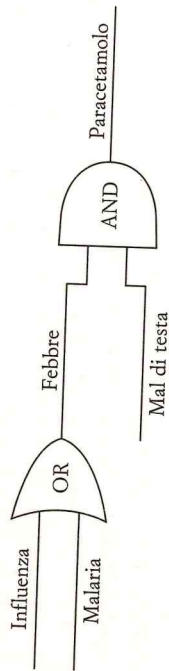
I.

La rivoluzione del machine learning

Viviamo nell'era degli algoritmi. Solo una o due generazioni fa, erano pochissimi a sapere cosa si intendesse per *algoritmo*. Oggi, con la loro onnipresenza nella vita quotidiana, gli algoritmi sono un ingrediente fondamentale della civiltà moderna. Non sono solo nel vostro cellulare o nel laptop, ma anche nelle automobili, a casa vostra, nei vostri elettrodomestici e nei giocattoli. La vostra banca è un intreccio gigantesco di algoritmi, e gli esseri umani si limitano a girare qualche manopola qua e là. Gli algoritmi decidono l'orario dei voli e governano gli aeroplani. Fanno funzionare le fabbriche, comprano e spediscono merci, ne incassano i proventi e tengono la contabilità. Se all'improvviso tutti gli algoritmi smettessero di funzionare, sarebbe la fine del mondo così come lo conosciamo.

Un algoritmo è una sequenza di istruzioni che dice a un computer cosa fare. I computer contengono miliardi di minuscoli interruttori, i transistor, che gli algoritmi accendono e spengono miliardi di volte al secondo. L'algoritmo più semplice che si possa immaginare? Far scattare un interruttore. Lo stato di un transistor corrisponde a un bit di informazione: vale 1 quando il transistor è acceso, 0 quando è spento. Nel computer della vostra banca, da qualche parte, c'è un bit che dice se il vostro conto è in rosso. Un altro bit, questa volta nei computer della Previdenza sociale, dice se siete vivi o morti. Il secondo algoritmo più semplice è il seguente: combinare due bit. Claude Shannon, più noto come il padre della teoria dell'informazione, fu il primo a capire che i transistor, con il loro accendersi e spegnersi in risposta ad altri transistor,

non fanno altro che ragionare (la tesi che scrisse per il master al MIT parlava proprio di questo: fu la più importante tesi di master mai scritta). Se il transistor A si accende solo quando entrambi i transistor B e C sono accesi, ciò che sta facendo non è altro che un piccolissimo ragionamento logico. Se A si accende quando B o C sono accesi, abbiamo un'altra minuscola operazione logica. Se A si accende quando B si spegne, e viceversa, le operazioni diventano tre. Che lo crediate o no, ogni algoritmo, per quanto complesso, può essere ricondotto a queste tre sole operazioni: AND, OR e NOT. Un algoritmo semplice può essere rappresentato con un diagramma in cui le operazioni AND, OR e NOT sono associate a simboli differenti. Ad esempio, supponiamo che uno stato febbrile possa essere causato dall'influenza o dalla malaria, e che nel caso in cui alla febbre si aggiunga il mal di testa dobbiate prendere del paracetamolo. Il ragionamento può essere espresso con il diagramma seguente:



Combinando un gran numero di operazioni di questo tipo possiamo portare a termine sequenze di ragionamento logico molto complesse. Sono in tanti a pensare che nei computer sia tutto una questione di numeri, ma non è così. Nei computer, è tutta questione di logica: i numeri e le operazioni aritmetiche sono frutto di operazioni logiche, così come ogni altra cosa. Volete sommare due numeri? Ci pensa una combinazione di transistor. Volete battere il campione di *Jeopardy!*? Anche in questo caso, c'è una combinazione di transistor (molto più grande, naturalmente) che può riuscirci.

Se dovessimo costruire un nuovo computer ogni volta che vogliamo fare qualcosa di diverso, però, i costi sarebbero proibitivi. Un computer moderno, invece, è formato da un numero enorme di transistor che può fare tante cose diverse, a seconda degli elementi attivati. Michelangelo disse che lui si limitava a intuire la statua che si celava nel blocco di marmo e a togliere il marmo in

eccesso finché non si riusciva a vederla. Analogamente, un algoritmo elimina dal computer tutti i transistor superflui finché non emerge la funzione desiderata, che può essere il pilota automatico di un aereo di linea o un nuovo film della Pixar.

Un algoritmo non è solo un insieme qualunque di istruzioni: deve essere abbastanza preciso e univoco da poter essere eseguito da un computer. Una ricetta di cucina, ad esempio, non è un algoritmo perché non specifica esattamente in che ordine fare le cose o in che cosa consiste, nei minimi dettagli, ogni singolo passo. A quanto zucchero corrisponde esattamente un cucchiaino? Come ben sanno tutti quelli che si sono cimentati con una nuova ricetta, seguirla può portare a un piatto squisito o a un disastro totale. Un algoritmo, invece, dà sempre lo stesso risultato. Anche se una ricetta specifica esattamente 15 grammi di zucchero, i problemi non sono finiti, perché un computer non sa cosa sia lo zucchero, o un grammo. Se dovessimo programmare un robot da cucina per fare una torta, dovremmo dirgli come riconoscere visivamente lo zucchero, come impugnare un cucchiaino, e così via (e su questo c'è ancora un po' di lavoro da fare). Il computer deve sapere come eseguire l'algoritmo nei minimi dettagli, fino all'accensione e allo spegnimento del singolo transistor. Ecco perché una ricetta è molto diversa da un algoritmo.

Questo, invece, è un algoritmo per giocare a tris:

Se il vostro avversario ha occupato due caselle di fila, occupate la casella rimanente.

Altrimenti, se c'è una mossa che crea due coppie di caselle adiacenti in un colpo solo, scegliete quella.

Altrimenti, se la casella centrale è libera, occupatela.

Altrimenti, se il vostro avversario ha occupato una casella d'angolo, occupate la casella opposta.

Altrimenti, se c'è un angolo vuoto, occupatelo.

Altrimenti, occupate una casella vuota qualsiasi.

Questo algoritmo ha una proprietà bellissima: non perde mai! Certo, mancano ancora molti dettagli, ad esempio la rappresentazione, nella memoria del computer, della griglia di gioco e dei suoi cambiamenti a ogni mossa. Ad esempio, potremmo associare ad

ogni casella una coppia di bit che varrà 00 se la casella è vuota, 01 se contiene un cerchio e 10 se contiene una croce. In ogni caso, l'algoritmo è già abbastanza preciso e univoco da consentire a qualsiasi programmatore competente di completarlo nel modo corretto. Non dimentichiamo, poi, che non siamo realmente obbligati a definire i dettagli di un algoritmo fino al singolo transistor, ma che possiamo utilizzare come componenti gli algoritmi preesistenti, e che ce ne sono tantissimi tra cui scegliere.

Lavorare con gli algoritmi richiede rigore. Si dice spesso che non si è realmente capito qualcosa se non lo si può esprimere con un algoritmo (come disse Richard Feynman: «Se non lo so creare, non lo capisco»). Le equazioni, pane quotidiano di fisici e ingegneri, non sono altro che una classe speciale di algoritmi. Ad esempio, la seconda legge di Newton, senza dubbio l'equazione più importante di tutti i tempi, ci dice che per calcolare la risultante delle forze su un corpo bisogna moltiplicarne la massa per l'accelerazione. Ci dice anche, implicitamente, che l'accelerazione si ottiene dividendo la forza per la massa, ma il fatto di esprimerlo in forma esplicita è a sua volta il passaggio di un algoritmo. In ogni settore della scienza, se una teoria non può essere espressa sotto forma di algoritmo non è del tutto rigorosa (senza tenere conto del fatto che non si può usare un computer per risolverla, il che limita seriamente la sua utilità). Gli scienziati creano teorie, e gli ingegneri creano dispositivi. Gli informatici creano algoritmi, e che sono al tempo stesso teorie e dispositivi.

Progettare un algoritmo non è semplice. Le trappole non mancano, e non si può dare nulla per scontato. Alcune delle vostre intuizioni si riveleranno sbagliate, e dovrete trovare un'altra strada. Dopo aver progettato l'algoritmo, dovrete scriverlo in un linguaggio comprensibile da un computer, come Java o Python, e a quel punto avete un programma. La correttezza del programma andrà verificata: bisognerà trovare tutti gli errori e correggerli finché il computer non riuscirà a eseguire l'intero programma senza andare in crisi. Ma quando finalmente avrete un programma che fa ciò che volete, sarete a cavallo. I computer eseguiranno i vostri ordini milioni di volte ad altissima velocità e senza lamentarsi. Chiunque potrà usare la vostra creatura: gratis, se così vorrete, o pagando abbastanza da trasformarvi in un miliardario se il problema è sufficientemente

importante. Un programmatore – la persona che crea gli algoritmi e li trasforma in un programma – è un dio minore, creatore di universi a piacimento. Potremmo spingerci ad affermare che persino il Dio della Genesi è un programmatore: lo strumento della sua creazione non è il lavoro manuale ma il linguaggio. Le parole diventano mondi. Oggi anche voi, seduti sul divano con il vostro laptop, potete essere un dio. Immaginare un universo e farlo diventare realtà. Le leggi della fisica sono un optional.

Con il passare del tempo, gli informatici sono partiti dagli algoritmi esistenti per crearne di nuovi. Gli algoritmi si combinano tra di loro per utilizzare i risultati di altri algoritmi, e i loro risultati finiscono in pasto a ulteriori algoritmi. Ogni secondo, miliardi di transistor in miliardi di computer cambiano stato miliardi di volte. Gli algoritmi formano un nuovo tipo di ecosistema, un'entità che cresce senza sosta, paragonabile per complessità solo alla vita stessa.

Come in tutti i paradisi terrestri, però anche qui c'è un serpente. Si chiama mostro della complessità e ha molte teste, come l'Idra. Una di queste è la complessità spaziale, cioè il numero di bit di informazione occupati dall'algoritmo nella memoria del computer. Se l'algoritmo ha bisogno di una quantità di memoria superiore a quella che può fornire il computer non serve a niente e va sostituito. Poi c'è la sorella malvagia, la complessità temporale, cioè il tempo necessario per eseguire l'algoritmo, ovvero quante volte dovrà utilizzare i transistor del computer prima di produrre il risultato desiderato. Se il tempo richiesto è più di quanto possiamo aspettare, ci ritroviamo di nuovo con un algoritmo inutile. Il volto più spaventoso del mostro della complessità, tuttavia, è la complessità umana. Quando gli algoritmi si fanno troppo complicati perché il nostro povero cervello di esseri umani possa capirli, quando le interazioni tra le varie componenti dell'algoritmo diventano troppe e troppo involute, al loro interno si insinuano errori che non riusciamo a trovare e a correggere, e l'algoritmo non fa ciò che vorremmo. E anche se, in un modo o nell'altro, riusciamo a farlo funzionare, finisce per essere troppo complicato per chi deve usarlo, non si integra con gli altri algoritmi e rischia di essere fonte di guai in futuro.

Gli informatici combattono con il mostro ogni giorno. Quando perdono, la complessità si insinua nelle nostre vite. Probabilmente

te vi siete già accorti di quante battaglie siano state perse. Ciò nonostante, continuiamo a costruire la nostra torre di algoritmi, con difficoltà sempre più grandi. Ogni nuova generazione di algoritmi viene costruita sulle precedenti e deve vedersela con le complessità passate, oltre che con le proprie. La torre diventa così alta da controllare il mondo intero, ma si fa anche sempre più fragile, come un castello di carte in procinto di crollare. Basta un errore minuscolo in un algoritmo per far esplodere un razzo da un miliardo di dollari o per lasciare senza elettricità milioni di persone. Un'interazione inaspettata tra algoritmi, e il mercato azionario crolla.

Se i programmatori sono divinità minori, il mostro della complessità è il diavolo in persona. E poco per volta, sta vincendo la guerra.

Ci deve essere un'altra soluzione.

Entra in scena il learner

Ogni algoritmo ha un input e un output: i dati vengono immessi nel computer, l'algoritmo fa quello che deve fare, e alla fine si ottiene il risultato. Il machine learning capovolge tutto quanto: si parte dai dati e dal risultato desiderato e si arriva all'algoritmo che vi fa passare dagli uni all'altro. Gli algoritmi di apprendimento, o learner, sono algoritmi che creano altri algoritmi. Con il machine learning, non dobbiamo più scrivere programmi: ci pensano i computer.

Accidenti.

Computer che scrivono i propri programmi. È un'idea formidabile, non c'è che dire, e forse è anche un po' inquietante. Se i computer cominciano a programarsi da soli, come li controlleremo? In realtà, come vedremo, siamo in grado di farlo senza problemi. Un'obiezione più immediata, forse, è che un'idea del genere sembra troppo bella per essere vera. Scrivere algoritmi, ovviamente, richiede intelligenza, creatività, capacità di risolvere i problemi: da quando in qua un computer possiede tutto ciò? Come distinguere il machine learning dalla magia? In effetti oggi sono in molti a scrivere programmi che i computer non possono imparare. Più sorprendente, forse, è il fatto che i computer sono in grado di im-

parare programmi che nessuno è in grado di scrivere. Sappiamo guidare un'automobile o decifrare una frase scritta a mano, ma si tratta di capacità subcoscienti, e non riusciamo a spiegare a un computer come metterle in pratica. Con un numero sufficiente di esempi di entrambe le azioni, invece, un learner non avrà problemi a capire da solo come fare; dopo di che, potremo lasciare che se la cavi da solo. È così che negli uffici postali si leggono i codici di avviamento, ed è per questa strada che stiamo per arrivare alle automobili che si guidano da sole.

Per spiegare la potenza del machine learning, la cosa migliore, forse, è ricorrere a un'analogia a basso contenuto tecnologico: l'agricoltura. In una società industriale, i beni sono prodotti nelle fabbriche: vuol dire che gli ingegneri devono capire esattamente come assemblarli a partire dalle loro componenti, come fabbricare le componenti stesse, e così via, fino alle materie prime. Un sacco di lavoro. I computer sono i beni più complessi che siano mai stati inventati: progettarli, progettare le fabbriche che li costruiscono e i programmi che li utilizzano è un lavoro enorme. Per soddisfare una parte dei nostri bisogni, però, c'è un altro modo, molto più antico: lasciar fare alla natura. L'agricoltore pianta i semi, si assicura che ci siano acqua e cibo a sufficienza e aspetta la stagione del raccolto. Non possiamo immaginare una tecnologia che funzioni in maniera analoga? Certamente, ed è proprio quello che promette il machine learning. Gli algoritmi di apprendimento sono i semi, i dati sono il terreno e i programmi appresi sono le piante adulte. L'esperto di machine learning è come l'agricoltore: semina, irriga e concima il terreno, tiene d'occhio lo stato di salute del raccolto, ma per il resto non interferisce.

Non appena cominciamo ad analizzare il machine learning in quest'ottica balzano agli occhi due cose. La prima è che più dati abbiamo, più possiamo imparare. Niente dati? Nulla da imparare. Un sacco di dati? Molto da imparare. La crescita esponenziale di montagne di dati disponibili è la ragione per cui il machine learning sta comparso ovunque. Se lo si potesse acquistare al supermercato, sulla scatola ci sarebbe scritto «basta aggiungere i dati».

Il secondo elemento che balza agli occhi è che il machine learning è la spada con cui uccidere il mostro della complessità. Con una quantità di dati sufficiente, un programma di apprendimento

lungo poche centinaia di righe può generare senza problemi milioni di righe di codice, e può farlo a ripetizione, ogni volta per un problema diverso. Per il programmatore, la complessità si riduce in maniera fenomenale. Naturalmente, al mostro della complessità spuntano nuove teste non appena tagliamo le vecchie, proprio come all'Idra, ma all'inizio sono piccole, e il tempo che impiegano a crescere gioca a nostro favore.

Possiamo pensare al machine learning come all'inverso della programmazione, proprio come la radice quadrata di un numero è l'inverso del suo quadrato, o l'integrazione è l'operazione inversa della differenziazione. Così come possiamo chiedere «qual è il numero che elevato al quadrato dà 16?» o «qual è la funzione la cui derivata è $x + 1$?», possiamo porci la domanda «qual è l'algoritmo che produce questo output?». Tra poco vedremo come trasformare questa intuizione in algoritmi di apprendimento concreti.

La specialità di alcuni learner è la conoscenza; altri, invece, acquisiscono abilità pratiche. L'affermazione «Tutti gli uomini sono mortali» fa parte della conoscenza. Andare in bicicletta è un'abilità pratica. Nel machine learning la conoscenza assume spesso la forma di modelli statistici, perché gran parte della conoscenza è di natura statistica: tutti gli umani sono mortali, ma solo il 4% di questi è americano. Le abilità pratiche, dal canto loro, assumono spesso la forma di procedure: se la strada curva a sinistra, gira la ruota a sinistra; se un cervo ti balza di fronte, schiaccia i freni (purtroppo, mentre scrivo queste righe le automobili che si guidano da sole di Google prendono ancora per cervi i sacchetti di plastica trascinati dal vento). Spesso le procedure sono semplici, ed è la conoscenza su cui si fondano che è complessa. Se siete in grado di identificare una e-mail come spam, sapete quali vanno cancellate. Se sapete giudicare la bontà di una situazione sulla scacchiera, sapete anche che mossa fare (quella che porta alla posizione migliore).

Il machine learning assume forme diverse, chiamate con nomi diversi: pattern recognition, modellizzazione statistica, data mining, scoperta della conoscenza, analisi predittiva, scienza dei dati, sistemi adattivi, sistemi auto-organizzati e così via. Ognuno di questi nomi è utilizzato da comunità differenti con significati differenti. Alcuni di essi hanno una vita media lunga, altri meno. Nel

corso del libro utilizzerò l'espressione «machine learning» per riferirmi al loro insieme.

Talvolta si fa confusione tra il machine learning e l'intelligenza artificiale (AI). Da un punto di vista tecnico, il machine learning è un settore dell'intelligenza artificiale, ma è cresciuto così tanto e così bene che ormai è sul punto di eclissare la sua orgogliosa genitrice. L'intelligenza artificiale si propone di insegnare ai computer come fare ciò che attualmente gli esseri umani sanno fare meglio, e non c'è dubbio che l'apprendimento sia l'attività più importante: un computer che non sa imparare non riuscirà a competere a lungo con un essere umano; con l'apprendimento, il resto viene da sé.

Nell'ecosistema dell'elaborazione dell'informazione, i learner sono i superpredatori. I database, i crawler, gli indicizzatori e gli altri programmi simili sono gli erbivori, che ruminano pazientemente in distese sterminate di dati. Gli algoritmi statistici, i software di elaborazione analitica online e così via sono i predatori. Gli erbivori sono necessari - senza di loro, gli altri non potrebbero esistere - ma i superpredatori hanno una vita molto più eccitante. Un crawler è come una mucca; il web è il suo pascolo globale, ogni pagina è un filo d'erba. Quando il crawler è sazio, i suoi hard disk contengono una copia del web. A quel punto un indicizzatore crea un elenco delle pagine in cui compare una data parola, come nell'indice alla fine di un libro. I database sono come gli elefanti: grossi, pesanti, non dimenticano mai. Tra queste bestie pazienti sfrecciano algoritmi statistici e analitici intenti a compattare e filtrare i dati per trasformarli in informazione da dare in pasto ai learner che la digeriscono e la trasformano in conoscenza.

Gli esperti di machine learning (noti anche come machine learner) sono considerati come una casta sacerdotale elitaria anche dagli altri informatici. Molti informatici, in modo particolare quelli delle vecchie generazioni, non capiscono il machine learning come vorrebbero. Il problema è che l'informatica si è sempre basata sul pensiero deterministico, mentre il machine learning obbliga a pensare statisticamente. Ad esempio, se la regola per etichettare le e-mail come spam è accurata al 99%, non vuol dire che è sbagliata: può darsi che sia il meglio che riusciamo a fare e che sia abbastanza efficiente da risultare comunque utile. Questa differenza di atteggiamento spiega in larga misura come mai Microsoft ha

avuto molte più difficoltà a stare al passo con Google che con Netscape. Un browser, dopo tutto, non è che un software ordinario, ma un motore di ricerca presuppone una *forma mentis* differente.

L'altra ragione per cui i machine learner sono i super geek è che il loro numero globale è di gran lunga inferiore a quello di cui avremmo bisogno, persino per gli standard già rigidi dell'informatica. Secondo il tecnoguru Tim Reilly, la qualifica più richiesta nella Silicon Valley è «data scientist», scienziato dei dati. Il McKinsey Global Institute ha stimato che nel 2018 gli esperti di machine learning negli Stati Uniti saranno inferiori di 140.000-190.000 unità rispetto alle necessità reali, e che mancherà un milione e mezzo di manager consapevoli dell'importanza dei dati. Il boom del machine learning è stato troppo improvviso perché la scuola riuscisse a tenere il passo, e l'argomento è noto per essere ostico. I suoi libri di testo rischiano di provarvi un'indigestione di matematica. La difficoltà, però, è più apparente che reale. Tutti i concetti fondamentali del machine learning possono essere espressi senza ricorrere alle formule. Leggendo questo libro, potreste addirittura riuscire a inventare i vostri algoritmi di apprendimento senza nemmeno l'ombra di un'equazione.

La Rivoluzione industriale ha automatizzato il lavoro manuale, e la Rivoluzione dell'informazione ha fatto lo stesso con quello intellettuale. Il machine learning, invece, automatizza l'automazione stessa: se non ci fosse, i programmatori diventerebbero i colli di bottiglia che frenano il progresso. Grazie al machine learning, il progresso può accelerare. Se siete informatici pigri e non troppo brillanti il machine learning è il campo che fa per voi, perché gli algoritmi di apprendimento fanno tutto il lavoro ma vi lasciano prendere il merito. È anche vero, però, che gli stessi algoritmi di apprendimento potrebbero farci licenziare, rendendoci pan per focaccia.

Portando l'automazione a livelli mai visti, la rivoluzione del machine learning porterà grandi cambiamenti economici e sociali, proprio come accadde con Internet, il personal computer, l'automobile e la macchina a vapore nelle rispettive epoche. Esiste un'area, però, in cui i mutamenti sono già davanti agli occhi di tutti: il mondo degli affari e delle aziende.

Perché le aziende sposano il machine learning

Come mai Google vale così tanto più di Yahoo? Entrambi guadagnano attraverso gli annunci pubblicitari che mostrano sul web, ed entrambi sono tra i siti più frequentati. Entrambi mettono all'asta gli spazi pubblicitari e si servono del machine learning per prevedere la probabilità che un utente clicchi su un annuncio (all'aumentare della probabilità aumenta il valore dell'annuncio). Gli algoritmi di apprendimento di Google, però, sono molto più efficaci di quelli di Yahoo. Naturalmente non è questo l'unico motivo della differenza di quotazione dei due titoli, ma è comunque uno dei più importanti. Ogni clic previsto che non si concretizza è un'opportunità persa per il cliente e un introito perso per il sito web. I ricavi annuali di Google ammontano a 50 miliardi di dollari: migliorare le previsioni sui clic dell'1% potrebbe portare nelle casse dell'azienda un altro mezzo miliardo di dollari ogni anno. Non c'è da stupirsi se Google è un grande fan del machine learning e se Yahoo e gli altri stanno facendo di tutto per non rimanere indietro.

La pubblicità online è solo un aspetto di un fenomeno molto più vasto. In tutti i mercati, perché possa aver luogo una transazione bisogna far entrare in contatto i produttori e i consumatori. Prima dell'avvento di Internet gli ostacoli principali erano di natura fisica. Si potevano comprare libri solo in libreria, e gli scaffali del libro avevano una lunghezza limitata. Da quando si può scaricare qualsiasi libro a qualsiasi ora sul proprio lettore di e-book, invece, il problema è la quantità sconvolgente di scelte possibili. Come ci si comporta in una libreria che vende milioni di titoli? Lo stesso vale per altri prodotti legati all'informazione - video, musica, notizie, tweet, blog, e persino le care, vecchie pagine web - e per tutti i prodotti e i servizi che possono essere acquistati a distanza: scarpe, fiori, gadget, camere d'albergo, corsi, investimenti. Il discorso si applica persino alle persone che cercano un lavoro o un appuntamento. Come si fa a trovare quello che si cerca? Il problema fondamentale dell'Era dell'informazione è proprio questo, e il machine learning è un elemento importante della soluzione.

Quando un'azienda cresce, attraverso tre fasi. Nella prima, tutto viene gestito manualmente: i proprietari di un negozio a condu-