L'AI non è una tecnologia: è un metodo

Comprare l'intelligenza artificiale non basta, come non bastò il motore elettrico alle fabbriche del 1900. Quello che rende è il metodo che porta con sé: partire dai casi, misurare, ricavare la regola. Le fonti, i limiti e sette pratiche per usarlo.

endjneer · paithon.it · 11 novembre 2026 · versione 1
Articolo: https://paithon.it/lai-non-e-una-tecnologia-e-un-metodo/
doi.org/10.5281/zenodo.24143743
Testo con licenza CC BY 4.0.

Nei ricettari italiani c'è un'abbreviazione che nessun ingegnere accetterebbe in una specifica: *q.b.*, quanto basta. Sale q.b., farina q.b. Chi ha scritto la ricetta non era pigro. Quanta acqua voglia l'impasto del pane dipende dalla farina, dall'umidità della giornata, dal calore delle mani, e chi lo sa fare non saprebbe dirlo in grammi. Lo ha imparato guardando qualcuno, sbagliando qualche impasto, assaggiando. **Ha visto dei casi e ne ha ricavato una regola che non sa scrivere.**

Nel 1966 il filosofo Michael Polanyi diede a questa condizione una formula: «possiamo sapere più di quanto sappiamo dire». Vale per il pane, per riconoscere un volto, per capire dal tono di un'e-mail se un cliente sta per andarsene. E per decenni è stato il confine del software. Un programma tradizionale vuole la regola prima, scritta per esteso, e poi la applica ai casi: quello che non si sapeva dire non si poteva programmare. L'economista David Autor lo ha chiamato *paradosso di Polanyi*.

> **La frase di Polanyi.** Viene da *The Tacit Dimension* (1966). Autor la cita in apertura di «Polanyi's Paradox and the Shape of Employment Growth», National Bureau of Economic Research, 2014.

L'intelligenza artificiale di oggi nasce dal rovesciamento di quell'ordine. Alla macchina si danno i casi e le risposte giuste, e la regola la ricava lei. Per questo considerarla una tecnologia in più, da comprare e installare come si è fatto con il gestionale o con il cloud, porta fuori strada. L'AI è prima di tutto un metodo: partire dalle osservazioni, ricavarne una regola, controllarla su casi mai visti. La tecnologia è il veicolo con cui quel metodo è arrivato sulle scrivanie.

Quel metodo lavora su due piani. Il primo sta dentro la macchina: è il modo in cui un modello viene costruito, per induzione dai casi. Il secondo sta fuori, e riguarda anche chi una rete neurale non la addestrerà mai. Una regola che nessuno ha scritto, nessuno la può leggere: per sapere che cosa farà un modello non resta che metterlo alla prova, un caso dopo l'altro, come si fa in laboratorio. **È il primo piano a portare il secondo su ogni scrivania.**

Prima i casi, poi la regola

Lo schema più chiaro del rovesciamento lo ha disegnato François Chollet nel manuale *Deep Learning with Python*. Nella programmazione classica entrano le regole e i dati, ed escono le risposte. Nel *machine learning* entrano i dati e le risposte, ed escono le regole (figura 1).

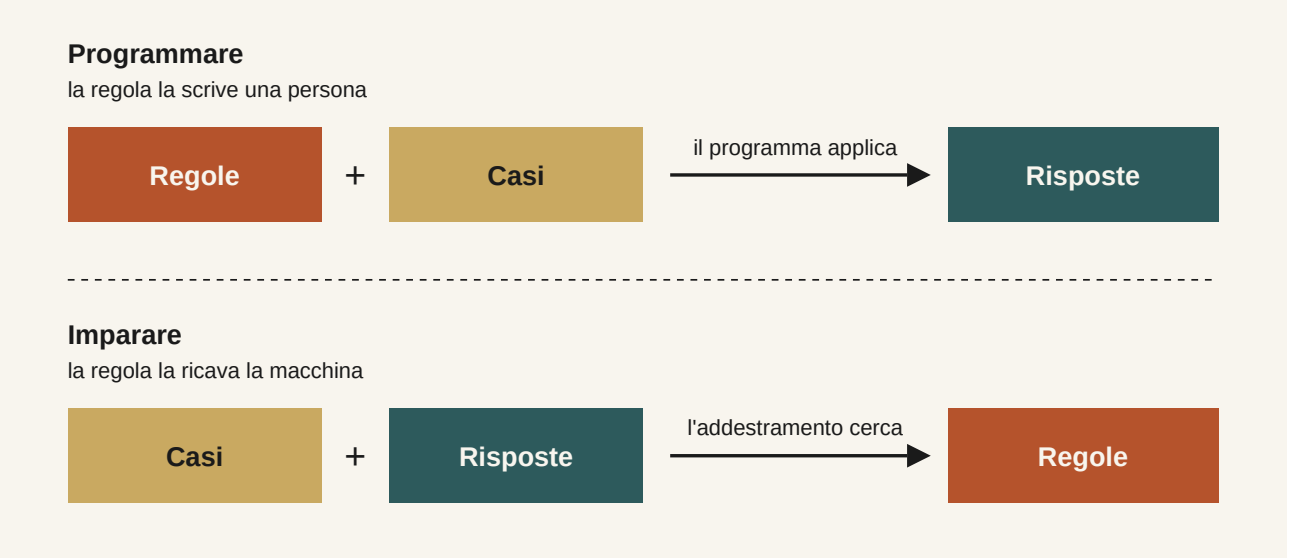


Figura 1. Programmare e imparare: gli stessi tre elementi, in due ordini.

Le stesse tre cose, in due ordini diversi. Sopra, la regola è il punto di partenza e la scrive una persona. Sotto, è il punto di arrivo e la ricava la macchina dai casi già risolti. Schema ridisegnato da quello di François Chollet.

La definizione che si studia all'università dice la stessa cosa in forma di contratto. È di Tom Mitchell, 1997: un programma impara dall'esperienza E, rispetto a una classe di compiti T e a una misura di prestazione P, se la sua prestazione nei compiti T, misurata da P, migliora con l'esperienza E. **In questa frase non c'è nessuna macchina.** Non si parla di reti neurali, di chip, di quantità di dati. Ci sono tre cose da dichiarare prima di cominciare: che cosa si vuole ottenere, come si misura se lo si è ottenuto, da quali casi si impara. È la descrizione di una procedura, e la si può seguire anche con carta e penna.

Che fosse una questione di metodo e non di attrezzi lo scrisse nel 2001 uno statistico, Leo Breiman, in un articolo intitolato *Statistical Modeling: The Two Cultures*. Le due culture sono due modi di arrivare a una conclusione partendo dai dati. La prima suppone di sapere come i dati nascono: sceglie un modello, per esempio una retta con pochi parametri, e lo adatta. La seconda tratta il meccanismo come ignoto e giudica una regola da una cosa sola, quanto bene prevede i casi che non ha visto. Breiman stimava che alla prima appartenesse il **98% degli statistici** e alla seconda il 2%, e sosteneva che per questo la disciplina stesse lasciando ad altri i problemi più interessanti.

Da allora la stessa idea è stata riscritta da persone molto diverse, ogni volta con parole più nette (tabella 1).

Tabella 1. La stessa idea, cinque volte
dai testi originali; le citazioni sono tradotte da noi

Anno	Chi	Che cosa dice
1997	Tom Mitchell, nel manuale *Machine Learning*	Imparare vuol dire migliorare in un compito, secondo una misura, grazie all'esperienza
2001	Leo Breiman, su *Statistical Science*	Due culture: supporre come nascono i dati, oppure giudicare la regola da quanto prevede
2009	Halevy, Norvig e Pereira, ricercatori di Google	«Modelli semplici e molti dati battono modelli più elaborati basati su meno dati»
2017	Andrej Karpathy, nel saggio «Software 2.0»	Il programma non si scrive per intero: si fissa un obiettivo sul suo comportamento, per esempio una raccolta di ingressi e uscite, si abbozza l'architettura di una rete e si lascia al calcolo la ricerca
2019	Richard Sutton, in «The Bitter Lesson»	In settant'anni di ricerca i metodi generali, ricerca e apprendimento, hanno battuto la conoscenza umana scritta a mano nei programmi

I cinque testi. Mitchell, *Machine Learning*, McGraw-Hill, capitolo 1. Breiman, *Statistical Science* 16(3). Halevy, Norvig e Pereira, «The Unreasonable Effectiveness of Data», *IEEE Intelligent Systems*. Karpathy, «Software 2.0», 11 novembre 2017. Sutton, «The Bitter Lesson», 13 marzo 2019.

Sutton, verso la fine del suo testo, scrive una frase che vale un programma di lavoro: «Vogliamo agenti che sappiano scoprire come sappiamo fare noi, non che contengano quello che abbiamo scoperto». In tutti e cinque i testi il soggetto è un modo di procedere. I grandi modelli linguistici e i centri di calcolo che li addestrano sono venuti dopo, e sono ciò che ha reso quel modo di procedere abbastanza economico da uscire dai laboratori. Sutton, con David Silver, ha poi spinto l'idea un passo oltre: i modelli addestrati sui dati umani starebbero toccando un tetto, e il prossimo salto verrebbe da agenti che imparano dalla propria esperienza. È il tema di «L'era dell'esperienza: Silver e Sutton oltre i dati umani».

Gli economisti lo chiamano «metodo per inventare»

Nel 1957 l'economista Zvi Griliches studiò la diffusione del mais ibrido negli Stati Uniti e si accorse che la novità vera stava nel procedimento. L'ibridazione permetteva di ottenere, zona per zona, la varietà adatta a quel clima e a quel terreno. Era, scrisse, «l'invenzione di un metodo per inventare».

Sessant'anni dopo tre economisti, Iain Cockburn, Rebecca Henderson e Scott Stern, hanno ripreso quella formula per l'intelligenza artificiale. Nel loro lavoro distinguono due cose che di solito si confondono. Una *general purpose technology* è una tecnologia che entra in tutti i settori, come il motore elettrico o il microprocessore. Un metodo di invenzione è uno strumento che cambia il modo in cui si cercano le cose nuove, come il mais ibrido. Un robot industriale, per loro, non è nessuna delle due. Il *deep learning* ha le carte per essere **tutte e due insieme**, ed è dal secondo ruolo che gli autori si aspettano l'effetto più grande.

Il metodo per inventare. Cockburn, Henderson e Stern, «The Impact of Artificial Intelligence on Innovation», NBER Working Paper 24449, marzo 2018. La frase di Griliches, che citano, è nello studio sul mais ibrido uscito su *Econometrica* nel 1957.

Qui la tesi di questo articolo va corretta. Presa alla lettera, «l'AI non è una tecnologia» è falsa: è anche una tecnologia, e chi la studia per mestiere lo scrive. La versione che regge è più stretta: è una tecnologia il cui valore sta nel metodo che obbliga ad adottare.

L'esempio più pulito è la forma delle proteine. Prevedere come si ripiega una catena di amminoacidi era un problema aperto da cinquant'anni, e una delle strade battute era scrivere le forze fisiche in gioco e simularle. AlphaFold 2 ha preso l'altra: ha imparato dalle strutture che i laboratori avevano già misurato e depositato in un archivio pubblico, la Protein Data Bank, aiutandosi con le sequenze di proteine imparentate. Alla gara CASP14 del 2020, su proteine mai viste, le sue previsioni avevano un errore mediano di **0,96 ångström**, paragonabile nella maggior parte dei casi a quello degli esperimenti. Nel 2024 metà del Nobel per la chimica è andata a Demis Hassabis e John Jumper per quel lavoro.

La rete di AlphaFold, però, non parte da zero: i suoi autori scrivono di aver messo nel disegno dell'algoritmo quello che la fisica e la biologia sanno delle proteine. Il metodo dei casi funziona meglio quando chi lo usa conosce già il problema.

AlphaFold. Jumper e altri, «Highly accurate protein structure prediction with AlphaFold», *Nature* 596, 2021. Il premio: comunicato dell'Accademia svedese, 9 ottobre 2024; l'altra metà è andata a David Baker.

Il motore nuovo sugli alberi di prima

Che una tecnologia potente possa restare a lungo senza effetti lo ha mostrato lo storico dell'economia Paul David con il caso dell'elettricità. Le prime centrali di Edison sono dell'inizio degli anni Ottanta dell'Ottocento. Nel 1899 i motori elettrici fornivano **meno del 5%** della forza motrice delle fabbriche americane, e per arrivare alla metà ci vollero altri vent'anni. Fino ai primi anni Venti, quattro decenni dopo le centrali, nelle statistiche della produttività l'elettricità non si vede.

Il motivo sta nel modo in cui le fabbriche adottarono la novità. Una fabbrica a vapore aveva una grande macchina che faceva girare alberi lunghi quanto il capannone; da lì, con cinghie e pulegge, il moto arrivava ai telai e ai torni. Per decenni gli industriali fecero la cosa più ragionevole, perché gli impianti esistenti funzionavano ancora e sostituirli non conveniva: misero dei motori elettrici a far girare gli stessi alberi, uno per ogni gruppo di macchine, e lasciarono cinghie e pulegge dov'erano. È il sistema che David chiama *group drive*. Si risparmiava un po' di energia. La fabbrica restava quella di prima.

Il salto arrivò quando ogni macchina ebbe il suo motore, lo *unit drive*. Senza alberi da sostenere i capannoni potevano essere leggeri e a un solo piano, e le macchine si potevano disporre seguendo il percorso del pezzo in lavorazione invece della linea dell'albero. **L'elettricità era la stessa di vent'anni prima.** Era cambiato il progetto della fabbrica (figura 2).

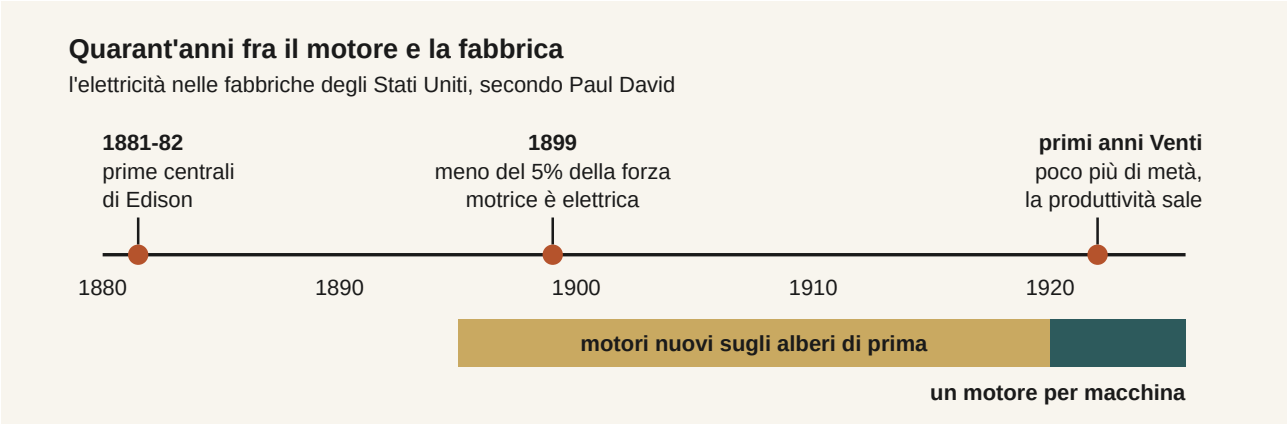


Figura 2. L'elettricità nelle fabbriche degli Stati Uniti: quarant'anni fra le prime centrali e l'effetto sulla produttività (da Paul David, 1990).

> **La dinamo.** Paul A. David, «The Dynamo and the Computer: An Historical Perspective on the Modern Productivity Paradox», *American Economic Review* 80(2), 1990. David mette in guardia dal prendere l'analogia alla lettera: serve a ricordare i tempi lunghi, non a prevedere date.

Per l'AI gli economisti si aspettano un andamento simile. Erik Brynjolfsson, Daniel Rock e Chad Syverson lo chiamano *curva a J* della produttività: una tecnologia di uso generale, scrivono, permette e insieme richiede investimenti complementari, cioè processi nuovi, modelli d'impresa nuovi, persone formate. All'inizio la produttività misurata rallenta, perché quegli investimenti costano e nei conti non compaiono. Cresce dopo, quando danno frutto.

La conferma più interessante viene da chi sostiene, in apparenza, il contrario di questo articolo. Nel 2025 Arvind Narayanan e Sayash Kapoor, informatici di Princeton, hanno pubblicato un saggio dal titolo *AI as Normal Technology*: l'AI è una tecnologia normale, come l'elettricità e internet, e va trattata come tale. Uno dei loro argomenti portanti, però, è che la diffusione ha un limite di velocità, ed è la velocità con cui persone, organizzazioni e istituzioni cambiano il modo di lavorare. Citano anche loro la dinamo di David. Chi difende con più rigore la parola «tecnologia» mette il collo di bottiglia nel metodo.

> **Curva a J e tecnologia normale.** Brynjolfsson, Rock e Syverson, «The Productivity J-Curve», NBER Working Paper 25148, poi su *American Economic Journal: Macroeconomics*, 2021. Narayanan e Kapoor, «AI as Normal Technology», Knight First Amendment Institute, 15 aprile 2025.

I numeri italiani dicono la stessa cosa con meno teoria. Secondo l'Istat le imprese con almeno dieci addetti che usano l'AI sono passate dall'8,2% del 2024 al **16,4% del 2025**: raddoppiate in un anno. Fra quelle che non la usano ma ci hanno pensato, l'ostacolo più citato è la mancanza di competenze, indicata dal 58,6%. I costi elevati, al 43%, sono quinti nell'elenco.

Un sondaggio di McKinsey del 2025 va nella stessa direzione: fra 25 caratteristiche organizzative messe a confronto, quella più legata a un effetto dell'AI generativa sull'utile operativo è aver ridisegnato i flussi di lavoro. È il sondaggio di una società di consulenza, che di riorganizzazioni vive, e va pesato per quello che è.

> **Imprese e sondaggi.** Istat, «Imprese e ICT, anno 2025», 15 dicembre 2025. McKinsey, «The state of AI: How organizations are rewiring to capture value», marzo 2025.

Stesso strumento, esiti opposti

Gli esperimenti condotti con chi fa quel lavoro di mestiere mostrano che uno stesso strumento può migliorare un risultato o peggiorarlo, secondo dove e come lo si usa (tabella 2).

Tabella 2. Tre esperimenti, ciascuno con il suo strumento
confronto con chi lavora senza AI

Che cosa si misura	Risultato
Dell'Acqua e altri, 2023: 758 consulenti con GPT-4	
Compiti alla portata del modello	12,2% di compiti finiti in più, 25,1% di tempo in meno
Un compito scelto fuori portata	19 punti percentuali in meno di risposte corrette
Brynjolfsson, Li e Raymond, 2025: 5.172 addetti all'assistenza clienti	
Problemi risolti all'ora, media di tutti	15% in più
Gli addetti più esperti	piccoli guadagni di velocità, piccoli cali di qualità
METR, 2025: 16 sviluppatori esperti, 246 compiti sui propri progetti	
Tempo che credevano di aver risparmiato	20%
Tempo misurato	19% in più

> **I tre studi.** Dell'Acqua e altri, «Navigating the Jagged Technological Frontier», Harvard Business School Working Paper 24-013, 2023. Brynjolfsson, Li e Raymond, «Generative AI at Work», *Quarterly Journal of Economics*, 2025. METR, lo studio del luglio 2025 e l'aggiornamento del febbraio 2026. Sono tre contesti precisi: non autorizzano una cifra valida per tutti i lavori.

Il primo studio ha dato un nome al fenomeno: frontiera frastagliata. Dentro la frontiera il modello aiuta, appena fuori fa danni, e il confine non è segnato da nessuna parte: lo si trova provando. Lo stesso esperimento dà anche il dato più citato su chi guadagna di più: i consulenti che partivano sotto la media migliorano del 43%, quelli sopra del 17%. In quel confronto, però, lavora di nascosto la regressione verso la media: quanto ne resti lo discute «L'oblio della media».

Il secondo suggerisce da dove viene il guadagno, quando c'è. L'assistente dato agli addetti era stato messo a punto su conversazioni di assistenza riuscite e fallite, e gli autori portano indizi che funzioni perché cattura e diffonde i comportamenti degli addetti più bravi. Uno di questi, notano, è fare qualche domanda in più prima di decidere qual è il guasto. È il «quanto basta» dei migliori, ricavato dai casi e consegnato a chi è appena arrivato. Si spiegherebbe così perché i nuovi assunti guadagnano molto e i veterani quasi niente: la regola, loro, la sapevano già.

Il terzo è il più istruttivo, e per una ragione diversa da quella che lo ha reso famoso. Sedici sviluppatori esperti, al lavoro su progetti open source che conoscevano da anni, con gli strumenti di inizio 2025 hanno impiegato il 19% di tempo in più. Prima di cominciare si aspettavano di risparmiarne il 24%. **Alla fine erano convinti di averne risparmiato il 20%.** Nel febbraio 2026 METR ha riferito di aver provato a ripetere la misura: con gli strumenti più recenti gli sviluppatori sono probabilmente più veloci, ma l'esperimento non riesce più a dirlo: troppi partecipanti rifiutano di lavorare senza AI, e ci sono altri problemi di selezione e di misura. Il rallentamento, quindi, è una fotografia del 2025 e forse non vale più. Lo scarto fra impressione e misura resta: senza un conteggio, chi usa lo strumento non sa se lo sta aiutando.

Uno strumento che obbliga a osservare

C'è una ragione tecnica per cui chi lavora con questi sistemi finisce, che lo voglia o no, col ragionare per casi, ed è il punto in cui i due piani del metodo si toccano. Di un programma tradizionale si può leggere il codice e sapere che cosa farà: si ragiona per deduzione, dalla regola al comportamento. Un modello le sue regole le ha ricavate dai casi, e stanno sparse in miliardi di numeri: nessuno, nemmeno chi lo ha addestrato, sa dire leggendoli come risponderà a una domanda qualsiasi. **L'unico modo per saperlo è provare**, e una prova sola non basta, perché la stessa domanda può ricevere risposte diverse. Il metodo induttivo dentro la macchina obbliga al metodo sperimentale fuori.

Anche il modo di dargli istruzioni segue la stessa logica. Dal 2020, con GPT-3, è noto che a un modello linguistico bastano pochi esempi dentro la richiesta per cambiare comportamento: gli si mostrano tre recensioni già classificate e classifica la quarta. È la tecnica che si chiama *few-shot*. Chi scrive un *prompt* con degli esempi fa in piccolo quello che fa chi addestra un modello: sceglie i casi da cui la regola verrà ricavata.

Con un modello, però, si fa anche altro. Buona parte delle istruzioni sono regole scritte a mano, in italiano invece che in codice: «se il cliente nomina una scadenza, segnala l'email». La differenza con il software di una volta è che l'effetto di quella regola non si conosce leggendola. La si scrive, la si prova sui casi, si guarda che cosa è successo. **È un'ipotesi da collaudare**, e chi la collauda sta facendo un esperimento su una scatola che risponde in modo probabilistico.

I manuali dei produttori lo dicono senza giri di parole. La guida di Anthropic alla scrittura dei prompt comincia da ciò che bisogna avere prima di scriverne uno: una definizione chiara dei criteri di successo, un modo per verificarli empiricamente, e una prima bozza da migliorare. Chi non li ha, dice la guida, spenda prima il tempo per costruirli.

Nemmeno i criteri, poi, si fissano del tutto a tavolino. Shreya Shankar e colleghi hanno osservato come le persone valutano le risposte dei modelli e hanno dato un nome a quello che succede, *criteria drift*: agli utenti servono criteri per valutare le risposte, ma è valutando le risposte che arrivano a definire i criteri. Anche quando la regola la scriviamo noi, la troviamo guardando i casi. È il paradosso di Polanyi che ritorna, un piano più su: i modelli lo hanno aggirato nel fare, e lo ritroviamo nel giudicare. Che cosa renda buona una risposta, spesso non sappiamo dirlo finché non ne vediamo una sbagliata.

> **Esempi, criteri, valutazioni.** Brown e altri, «Language Models are Few-Shot Learners», 2020. Anthropic, «Prompt engineering overview». Shankar e altri, «Who Validates the Validators?», 2024.

Come si usa: sette pratiche

Se l'AI è un metodo, usarla bene vuol dire seguire la procedura. È la stessa che il *machine learning* applica alle macchine, portata sulle decisioni di chi lavora. Prendiamo un caso piccolo: uno studio di commercialisti che vuole far leggere a un modello le email dei clienti, per segnalare quelle urgenti (figura 3).

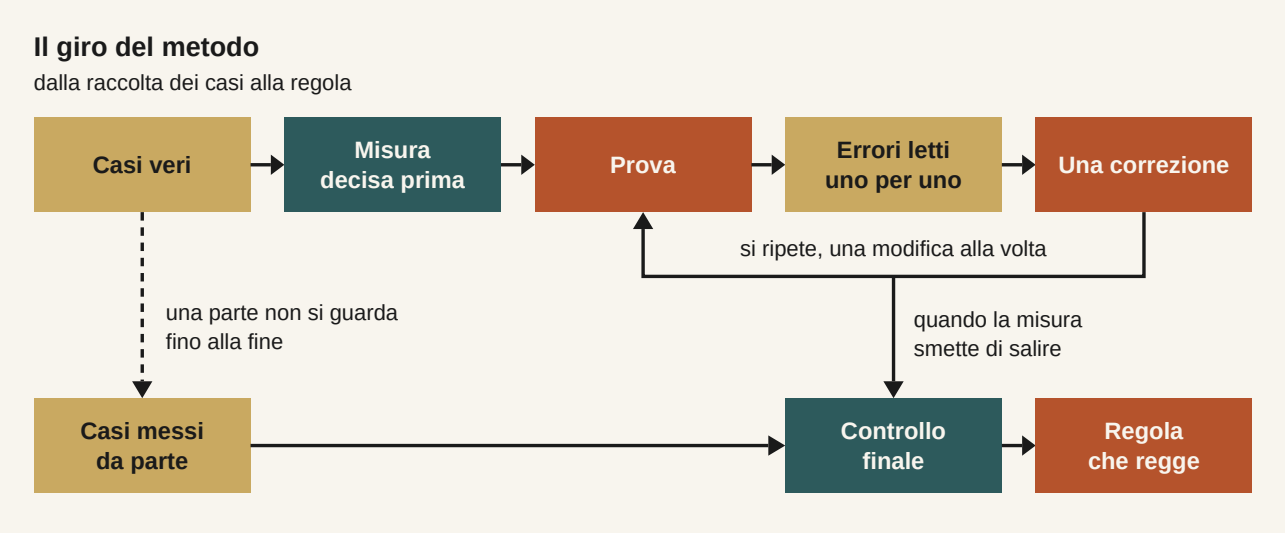


Figura 3. Il giro del metodo: dai casi veri alla regola, con una parte dei casi tenuta da parte per il controllo finale.

Parti dai casi veri. Prima di scrivere una riga di istruzioni lo studio prende qualche decina di email arrivate davvero e segna, per ciascuna, che cosa sarebbe dovuto succedere. La tentazione è cominciare dalla regola («urgente è ciò che ha una scadenza»). Una regola scritta a tavolino descrive il caso che si ha in mente, e le email vere sono piene di casi che in mente non c'erano.

Decidi prima come misuri. È la P della definizione di Mitchell. Che cosa pesa di più, un'urgenza persa o un falso allarme? Quante urgenze perse sono accettabili su cento? Deciderlo dopo aver visto le risposte è troppo tardi: a chi ha già scelto lo strumento qualunque risultato sembrerà buono.

Metti da parte dei casi e non guardarli. Una parte delle email resta chiusa in un cassetto fino alla fine. Le istruzioni si mettono a punto sulle altre; il giudizio si dà su quelle mai viste. È una delle regole più antiche del machine learning, e una delle più violate: Kapoor e Narayanan hanno raccolto 22 studi che, in 17 discipline, avevano trovato lo stesso errore in **almeno 294 articoli scientifici**: l'informazione dei casi di prova era filtrata, per una strada o per l'altra, nella costruzione del modello, e in alcuni casi le conclusioni ne uscivano molto gonfiate.

> Perché una regola che va benissimo sui casi già visti può fallire sui nuovi, e come si divide un insieme di dati per accorgersene, è spiegato nel Paithon Book, il libro di paithon che si legge online.
> Leggi «Overfitting e validazione» nel libro →

Leggi gli errori uno per uno. Una percentuale dice quanto si sbaglia e tace su come. Dieci email classificate male, lette di fila, mostrano di solito due o tre motivi che ritornano (il cliente che scrive «con calma» e intende il contrario, la scadenza nascosta nell'allegato), e ogni motivo è una correzione possibile: un'istruzione in più, un esempio aggiunto, un criterio riscritto. Sono regole scritte a mano anche queste. Rispetto alla regola a tavolino cambiano due cose: nascono da un errore che si è visto, e restano solo se sui casi la misura migliora. Quello che lo studio costruisce, alla fine, è un ibrido di istruzioni affinate sui casi, esempi dentro la richiesta e criteri di controllo.

Cambia una cosa alla volta. Se in un colpo solo si riscrivono le istruzioni, si aggiungono esempi e si cambia modello, e il risultato migliora, non si sa a che cosa lo si deve. Una modifica, una nuova misura sugli stessi casi, e se ne tiene nota.

Diffida del numero che sale. «Quando una misura diventa un obiettivo, smette di essere una buona misura»: è la legge di Goodhart, nella formulazione resa celebre dall'antropologa Marilyn Strathern. I modelli la confermano a modo loro. Robert Geirhos e colleghi hanno chiamato *shortcut*, scorciatoie, le regole che vanno benissimo sui casi di prova e non reggono nel mondo, perché si appoggiano a un indizio che con il compito non c'entra. Se le email urgenti dello studio venissero quasi tutte dallo stesso cliente ansioso, il modello imparerebbe a riconoscere il cliente. È il difetto dell'agente che, addestrato su un videogioco di motoscafi, smise di correre e si mise a girare in tondo in una laguna, colpendo sempre gli stessi bersagli per fare punti, senza bisogno di finire la gara: lo racconta «Reward hacking: quando il modello vince barando».

Ricontrolla quando il mondo cambia. Una regola ricavata dal passato vale finché il presente gli somiglia. Google Flu Trends stimava i casi di influenza dalle ricerche fatte sul motore, e per un po' ci riuscì. Poi, fra l'agosto 2011 e il settembre 2013, **sbagliò per eccesso in 100 settimane su 108**: con ogni probabilità erano cambiati il motore e le abitudini di chi cercava. Lo raccontarono su *Science* quattro ricercatori, fra cui il fisico italiano Alessandro Vespignani, notando che una proiezione dei dati ufficiali vecchi di tre settimane avrebbe fatto meglio. Bertrand Russell aveva detto la stessa cosa con un pollo: l'uomo che gli ha portato da mangiare ogni giorno della sua vita, alla fine, gli tira il collo.

> **Le fonti delle pratiche.** Kapoor e Narayanan, «Leakage and the reproducibility crisis in machine-learning-based science», *Patterns*, 2023. Geirhos e altri, «Shortcut Learning in Deep Neural Networks», 2020. Lazer, Kennedy, King e Vespignani, «The Parable of Google Flu», *Science* 343, 2014. Il pollo è nel capitolo sull'induzione di *The Problems of Philosophy*, 1912.

Dove la tesi si ferma

Il metodo non è nuovo. Nel 1615 Galileo scriveva a Cristina di Lorena che nelle dispute sui problemi naturali non si dovrebbe cominciare dall'autorità dei passi della Scrittura, «ma dalle sensate esperienze e dalle dimostrazioni necessarie». Cinque anni dopo Francesco Bacone ne fece un programma nel *Novum Organum*. Partire dalle osservazioni è il metodo sperimentale, e ha quattro secoli. Di nuovo ci sono due cose: una macchina che lo applica dove la regola non si sapeva scrivere, nel territorio del «quanto basta», e il fatto che per usarlo non serva più un laboratorio.

I casi da soli non bastano. Nel 2008 il direttore di *Wired*, Chris Anderson, annunciò la fine della teoria: con abbastanza dati, scrisse, «la correlazione basta» e dei modelli si può fare a meno. Google Flu Trends nacque quello stesso anno, e fu a lungo portato a esempio di che cosa i grandi dati sapessero fare. Pedro Domingos, in un articolo molto letto del 2012, scrisse il contrario: i dati da soli non bastano, per quanti se ne abbiano. La formula di Galileo ha due metà, e la seconda sono le dimostrazioni. L'AI ha reso economica la prima metà del metodo, le esperienze; la seconda, capire perché una regola regge, resta lavoro nostro.

Il giudizio non si ricava dai casi. Gli economisti Ajay Agrawal, Joshua Gans e Avi Goldfarb descrivono questi sistemi come macchine che abbassano il costo di una cosa sola, la previsione. Decidere richiede anche altro: sapere quanto pesa un errore rispetto a un altro, che cosa si è disposti a rischiare, per chi si lavora. Lo chiamano giudizio, ed è ciò che entra in gioco quando l'obiettivo non si riesce a scrivere. Decidere come si misura è il punto in cui il giudizio entra nel metodo, e lì la scelta resta di chi risponde della decisione.

Qualcuno deve continuare a produrre il «quanto basta». Se l'assistente consegna ai nuovi arrivati la regola dei veterani, viene da chiedersi come si diventa veterani. Le prove vanno in due direzioni. Nello studio sull'assistenza clienti gli addetti imparavano davvero: quando il sistema si fermava per un guasto, continuavano a rendere più di quanto rendevano prima che arrivasse. In un esperimento del 2026 su programmatori alle prese con una libreria nuova, invece, chi usava l'AI capiva meno i concetti e leggeva e correggeva peggio il codice, senza un guadagno di tempo significativo in media; non succedeva a chi la usava restando impegnato a capire. La psicologa Lisanne Bainbridge lo aveva scritto nel 1983 a proposito degli impianti industriali: più un sistema automatico è avanzato, più può diventare decisivo il contributo dell'operatore umano; le abilità che non si esercitano si deteriorano, e un certo tipo di conoscenza si forma soltanto usandola. Il giorno in cui il mondo cambia e i vecchi casi non bastano più, la regola nuova dovrà ricavarla qualcuno che il mestiere lo ha fatto.

> **Imparare o delegare.** Shen e Tamkin, «How AI Impacts Skill Formation», 2026. Bainbridge, «Ironies of Automation», *Automatica* 19(6), 1983. Il dato sui guasti è in «Generative AI at Work», versione del 2025.

> **Esperienze e dimostrazioni.** La lettera a Cristina di Lorena è su Wikisource. Anderson, «The End of Theory», *Wired*, 23 giugno 2008. Domingos, «A Few Useful Things to Know About Machine Learning», *Communications of the ACM*. Agrawal, Gans e Goldfarb, «Exploring the Impact of Artificial Intelligence: Prediction versus Judgment», NBER, 2018.

Quanto basta

La differenza fra le due parole è pratica. Una tecnologia si compra, e una volta comprata ce l'hanno tutti: l'abbonamento a un modello costa lo stesso allo studio che lo usa bene e a quello che lo usa male. Un metodo si impara, e si impara nel modo in cui si impara il «quanto basta»: lavorando sui casi, sbagliandone alcuni, tenendo il conto. Le fabbriche del 1900 il motore elettrico lo avevano. Mancava la pianta nuova.

In Italia l'uso è raddoppiato in un anno e l'ostacolo dichiarato sono le competenze. Conviene intendersi su quali. Scrivere una richiesta a un modello si impara in fretta. Raccogliere i casi, decidere una misura, tenerne da parte alcuni, leggere gli errori: è questo che vuol dire saper usare l'AI, e un'azienda può cominciare domani mattina, con un foglio di calcolo e trenta email.

La difficoltà sta nella disciplina più che nella tecnica. Decidere la misura prima di vedere le risposte, chiudere dei casi in un cassetto, cambiare una cosa alla volta: sono abitudini da laboratorio, e a occhio si fa prima. A occhio, però, ci si convince in fretta, come gli sviluppatori che erano sicuri di aver risparmiato tempo. Richard Feynman lo disse ai laureati del Caltech nel 1974: «Il primo principio è che non devi ingannare te stesso, e tu sei la persona più facile da ingannare».

> **Non ingannarsi.** Richard Feynman, «Cargo Cult Science», 1974.

Il ricettario, alla voce sale, continuerà a dire q.b. Chi lavora con una macchina che ha imparato dai casi deve fare come chi impara a cucinare: assaggiare, e segnarsi com'è andata.

Come citare. endjneer (2026). *L'AI non è una tecnologia: è un metodo.* paithon.it, versione 1, 11 novembre 2026. https://doi.org/10.5281/zenodo.24143743