UNIVERSITÀ
DI PAVIA

DIPARTIMENTI DI GIURISPRUDENZA,
INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE,
SCIENZE ECONOMICHE E AZIENDALI,
SCIENZE POLITICHE E SOCIALI,
STUDI UMANISTICI

CORSO DI LAUREA INTERDIPARTIMENTALE IN
COMUNICAZIONE DIGITALE

L'IA GENERATIVA A SUPPORTO DELLA PRODUTTIVITÀ: IL CASO GITHUB COPILOT IN SPINDOX

Relatore:

Chiar.mo Dott./Prof. Paolo Costa

Correlatore:

Chiar.mo Dott./Prof. Massimiliano Vaira

Tesi di laurea di

Lorenzo Podda

Matricola: 544554

ANNO ACCADEMICO 2024/25

INDICE

Introduzione

L'evoluzione tecnologica ha storicamente rappresentato uno dei principali fattori di trasformazione del lavoro e delle organizzazioni. A partire dalla meccanizzazione della produzione industriale fino alla diffusione delle tecnologie digitali, ognuna di queste fasi ha modificato l'organizzazione delle attività lavorative, la distribuzione delle competenze e le forme di coordinamento all'interno delle organizzazioni. In quest' ottica, l'attuale diffusione dell'intelligenza artificiale generativa sta contribuendo alla ridefinizione del lavoro umano, alla riorganizzazione del lavoro e alla richiesta di nuove competenze.

Tra gli ambiti nella quale questa trasformazione appare particolarmente evidente vi è quello dello sviluppo software, dove strumenti di intelligenza artificiale generativa come GitHub Copilot stanno assumendo un ruolo gradualmente più centrale nel supportare attività di programmazione. Questi strumenti sono infatti in grado di assistere lo sviluppatore nella scrittura e analisi del codice, nella risoluzione di problemi e nella gestione di compiti complessi, portando forti effetti positivi sulla produttività individuale e avendo di conseguenza effetti sull'organizzazione del lavoro.

Sulla base di queste considerazioni, la presente tesi ha l'obbiettivo di analizzare l'impatto dell'intelligenza artificiale generativa nei processi di sviluppo software, con particolare attenzione al caso di GitHub Copilot, al fine di comprendere se e in che misura il suo utilizzo contribuisca a trasformare l'attività lavorativa e la sua organizzazione. In particolare, l'analisi si concentra sugli effetti dello strumento sulla produttività individuale, sulla qualità del prodotto, sulla percezione degli sviluppatori e sulle possibili implicazioni organizzative e occupazionali connesse alla sua diffusione.

La tesi combina riflessione teorica e analisi empirica articolandosi in cinque capitoli.

Il primo capitolo ricostruisce l'evoluzione storica dei processi di automazione, mettendo in evidenza come le trasformazioni tecnologiche abbiano progressivamente ridefinito il lavoro umano, le competenze richieste e le forme di coordinamento organizzativo, fino ad arrivare al tema dell'automazione cognitiva.

Il secondo capitolo inquadra l'intelligenza artificiale generativa nel contesto organizzativo, approfondendone le caratteristiche tecniche e funzionali e soffermandosi sulle modalità attraverso cui questi sistemi vengono integrati nei processi di lavoro.

Il terzo capitolo si concentra sul caso di GitHub Copilot, analizzato sia dal punto di vista strutturale e operativo sia attraverso la letteratura che ne ha studiato gli effetti sulla produttività.

Il quarto capitolo presenta la metodologia della ricerca empirica, illustrando gli obbiettivi dello studio, le domande di ricerca, il disegno metodologico, gli strumenti di raccolta dei dati e le metriche adottate per la valutazione dei risultati.

Il quinto capitolo, infine, discute i risultati emersi dall'analisi, integrando i dati raccolti tramite questionario e test sperimentale, al fine di comprendere in che misura l'utilizzo di GitHub Copilot incida sulla produttività individuale, sulla qualità del lavoro e, più in generale, sull'organizzazione delle attività di sviluppo software.

CAPITOLO 1 - DALL'AUTOMAZIONE SEMPLICE ALL'AUTOMAZIONE COGNITIVA

Questo capitolo ricostruisce l'evoluzione storica dei processi di automazione, mettendo in luce come le trasformazioni tecnologiche abbiano comportato una riorganizzazione dei processi produttivi piuttosto che una semplice riduzione dell'occupazione lavorativa. Nel corso della storia, infatti, lo sviluppo tecnologico è stato costantemente orientato all'estensione delle capacità umane: dalle prime macchine fino ai moderni sistemi informatici la tecnologia ha progressivamente ridefinito le dinamiche del lavoro umano e le modalità con cui esso viene coordinato all'interno delle organizzazioni.

Oggi l'intelligenza artificiale generativa appare come una tecnologia che, se utilizzata come strumento di lavoro, può ampliare il potenziale del lavoratore. Questa è infatti in grado di amplificare lo sforzo intellettuale umano, supportando processi decisionali e valorizzando le competenze cognitive del lavoratore. Al tempo stesso, l'introduzione di tecnologie di IA generativa, solleva importanti tensioni organizzative legate al rischio di standardizzazione del lavoro e perdita di autonomia, a seconda delle modalità con cui le organizzazioni ne orientano l'utilizzo.

L'analisi è condotta ponendo particolare attenzione al pensiero organizzativo, esaminando le modalità attraverso le quali le tecnologie vengono integrate nei sistemi produttivi, contribuendo a ridefinire il lavoro, i ruoli e le forme di coordinamento. Il capitolo si articola quindi in un'analisi storica dei processi di automazione, seguita da un approfondimento sul ruolo dell'intelligenza artificiale generativa come tecnologia di supporto al lavoro cognitivo.

1.1 Prima Rivoluzione Industriale: la nascita della macchina

Il manifestarsi della Prima Rivoluzione Industriale tra la fine del 1700 e primi decenni del 1800 rappresenta uno dei passaggi più radicali nella storia dei processi industriali e dell'organizzazione sociale del lavoro. Questa è il derivato di un precesso graduale di trasformazione sociale, economica e urbana. Arnold Toynbee evidenziò proprio come l'aumento della popolazione, l'emergere di un'economia di mercato, il miglioramento delle comunicazioni, l'espansione del commercio e la sostituzione dell'industria domestica con il sistema di fabbrica furono tra i principali elementi che portarono al processo di cambiamento[1]. La portata innovativa deriva soprattutto dall'introduzione della macchina come nuovo protagonista tecnico ed economico, capace di trasformare i metodi produttivi, l'organizzazione del lavoro e la struttura della fabbrica.

Prima della rivoluzione industriale, la produzione era basata sull'impiego di strumenti che amplificavano la capacità dell'operaio, rimanendo dipendenti alla sua abilità, al suo ritmo e soprattutto alla sua forza. È proprio quest'ultima, secondo Marx, la chiave che differenzia lo strumento dalla macchina: mentre nel primo caso la forza motrice è l'uomo, nel secondo si tratta di una forza artificiale generata dalla macchina stessa. Il macchinario è composto da tre parti: macchina motrice, meccanismo di trasmissione e macchina operatrice. La macchina motrice genera e regola il movimento, mentre il meccanismo di trasmissione lo distribuisce e lo trasmette alle macchine operatrici. Quest'ultima sfrutta il moto ricevuto per compiere le stesse operazioni prima compiute con strumenti analoghi dagli operai.[2] In sintesi, la macchina sostituisce l'operaio che utilizza un singolo strumento con

---

[1] Toynbee, Arnold J. *"La rivoluzione industriale."* Milano: Garzanti, 1961

[2] Marx, Karl. "Macchine e grande industria." *Il capitale*, vol. I, a cura di Giacomo Breda, Roberto Fineschi, Riccardo Schimmenti e Paolo Sgrò. Torino: Einaudi (2024–2025).

un meccanismo che riesce ad utilizzare contemporaneamente un elevato numero di strumenti.

Le macchine operatrici, che in quel momento iniziano a funzionare contemporaneamente formando una struttura unitaria, danno vita a quello che viene chiamato da Marx *"sistema di macchine"*[3]. Il sistema di macchine porta quindi ad una automatizzazione sequenziale delle fasi di produzione e, cioè, ad un processo in cui le macchine sono funzionalmente integrate in un'unica catena automatizzata e ogni macchina è dipendente dalle altre per la continuità del processo. Dunque, questa può essere vista come la nascita dell'automazione industriale.

Dal punto di vista organizzativo e produttivo, l'introduzione del sistema di macchine ha apportato grandi cambiamenti all'interno della fabbrica. Se prima il processo era adattato all'operaio adesso è l'operaio ad essere adattato al processo. Infatti, la diffusione del sistema di macchine rese necessaria una profonda riorganizzazione del lavoro in fabbrica. La produzione prima basata sulle capacità individuali degli artigiani viene ora suddivisa in una sequenza di operazioni semplici, standardizzate e strettamente coordinate con il ritmo imposto dalle macchine. Prima della fabbrica un artigiano autonomo realizzava i propri prodotti seguendo un metodo a lotti: completava un certo numero di pezzi per una fase della lavorazione, poi passava a quella successiva, e così via fino al termine del prodotto.[4] Questo sistema risultava poco efficiente, poiché il continuo cambio di operazione rallentava il lavoro e generava spesso accumuli di semilavorati, materiali parzialmente lavorati che occupavano spazio e risorse senza produrre valore immediato. Adesso, con la divisione del lavoro, la produzione assume quindi una forma verticale, nella quale il lavoro sul prodotto è suddiviso in fasi,

---

[3]Marx, Karl. "Macchine e grande industria." *Il capitale*, vol. I, a cura di Giacomo Breda, Roberto Fineschi, Riccardo Schimmenti e Paolo Sgrò. Torino: Einaudi (2024–2025).

[4] Smith, Adam. *An Inquiry into the Nature and Causes of the Wealth of Nations, volume 1*. Vol. 1. Oxford: Clarendon Press, 1869.

ciascuna affidata a un operaio specializzato in un'unica funzione e mansione produttiva che non ha più bisogno di competenze per ogni fase di lavorazione. [5]

La standardizzazione del prodotto, la semplificazione delle operazioni e la crescente interdipendenza delle postazioni trasformano profondamente sia il processo produttivo che la composizione della forza lavoro necessaria al suo funzionamento. Poiché la complessità tecnica viene progressivamente incorporata nel macchinario e le singole operazioni vengono ridotte a gesti elementari, il sistema di fabbrica non richiede più lavoratori dotati di competenze artigianali elevate, ma operatori capaci di adattarsi al ritmo regolare imposto dalla macchina.

In questo contesto, la nuova struttura organizzativa contribuisce a creare un'apertura del lavoro industriale anche a donne e bambini, almeno per quelle operazioni produttive che non richiedevano un particolare sforzo fisico. L'automazione, applicata a certe fasi del processo produttivo riduce la necessità dell'erogazione dello sforzo fisico, permettendo l'impiego di manodopera giovane o femminile per operazioni in cui la sostituibilità del lavoratore è elevata, e il costo del lavoro può essere drasticamente ridotto. Parallelamente, il moto continuo delle macchine modifica radicalmente la struttura temporale del lavoro. Poiché il sistema di macchine può operare in modo costante e potenzialmente ininterrotto, la giornata lavorativa tende ad allungarsi per massimizzare il rendimento del capitale investito. La continuità del processo produttivo impone che la macchina non rimanga inattiva, e ciò si traduce in turni prolungati, in un'intensificazione del ritmo e nell'obbligo per l'operaio di mantenere una presenza costante e sincronizzata con il movimento del meccanismo. Questi elementi mostrano come la trasformazione tecnica introdotta dal sistema di macchine non riguardasse soltanto la struttura del processo produttivo, ma ridefinisse profondamente la natura dell'attività lavorativa, la percezione della professionalità e il ruolo del lavoratore all'interno della fabbrica.

---

[5] Leijonhufvud, Axel. *Capitalism and the factory system*. No. 184. Diskussionsbeiträge-Serie A, 1984.

Il sistema di fabbrica oltre ad aver generato una trasformazione tecnica del lavoro, ha portato anche alla costruzione di un nuovo regime disciplinare, indispensabile per garantire il funzionamento regolare del sistema di macchine. I regolamenti di fabbrica, formalizzati e impersonali, divennero uno dei simboli più evidenti delle nuove relazioni industriali. Se molti si occupavano principalmente di norme di disciplina e comportamento, altri arrivavano a definire in modo dettagliato l'organizzazione interna dell'azienda, i compiti degli operai e il corretto uso dei macchinari. L'obiettivo era quello di costruire una nuova disciplina industriale fondata sull'obbedienza, sulla regolarità e sull'interiorizzazione dei tempi del sistema produttivo.[6] Il regolamento aveva quindi il compito di far abbandonare agli operai le abitudini di lavoro irregolari per adattarli alla regolarità immutabile del complesso automa.[7] Questa trasformazione culturale ha costituito una parte essenziale della Prima Rivoluzione Industriale contribuendo a ridefinire in modo radicale il rapporto tra lavoratori, tempo e produzione industriale.

In tale sistema, il lavoro umano viene quindi riorganizzato in base ad un sapere tecnico-scientifico che non appartiene più al singolo operaio. È in questo contesto che Marx, nel Frammento sulle macchine dei Grundrisse, elabora il concetto di "*general intellect*", individuando nella scienza, nella conoscenza sociale e nelle capacità cooperative accumulate della società la nuova forza produttiva centrale.[8] Nel nuovo sistema industriale, infatti, la macchina rappresenta la materializzazione di un sapere collettivo: competenze ingegneristiche, scienza meccanica, sperimentazioni tecniche, forme di organizzazione del lavoro. Questa conoscenza, che si trova dentro il capitale fisso -vale a dire il complesso macchine-, diventa una componente del processo produttivo, al punto che la produttività della fabbrica dipende sempre meno

---

[6] Pollard, Sidney. "Factory discipline in the industrial revolution." *Economic History Review* (1963): 254-271.

[7] Ure, Andrew. *Philosophy of manufactures*. Routledge, 2013.

[8] Marx, Karl. *"Lineamenti fondamentali della critica dell'economia politica."* Torino: Einaudi, 2005.

dall'abilità dell'operaio, e sempre più dalla complessità incorporata nelle macchine. La macchina, per Marx, trasforma quindi il lavoro umano in "lavoro morto", subordinato al funzionamento dei sistemi produttivi, rendendo il valore del prodotto estraneo alle abilità manuali dell'operaio. Studi successivi, a partire dalla seconda metà del Novecento (Touraine, Kern & Schumann), hanno però mostrato come il lavoratore possa assumere nuovi ruoli di supervisione, controllo e gestione dei processi automatizzati.

Oggi l'intelligenza artificiale generativa può essere considerata una forma moderna del "general intellect", capace di supportare e ampliare il lavoro cognitivo umano.

La creazione della macchina e il suo utilizzo come complemento del lavoratore rappresentano un passaggio storico fondamentale. Questo ha innescato un processo di riorganizzazione del lavoro, in cui le attività produttive vengono progressivamente scomposte, standardizzate e integrate in un sistema che ridefinisce ruoli, competenze e forme di coordinamento, anticipando le dinamiche nell'era dell'automazione cognitiva. In questo contesto, la macchina appare come complemento del lavoratore, anticipando una logica di collaborazione tra uomo e tecnologia che caratterizzerà le fasi successive.

1.2 Seconda Rivoluzione industriale: l'automazione del lavoro

La Seconda Rivoluzione Industriale, tra la fine del 1800 e la prima metà del 1900, è caratterizzata dall'affermazione di nuove tecnologie: elettricità, motore a combustione, nuove industrie dell'acciaio e della chimica, sistemi di trasporto e comunicazione più rapidi. Queste innovazioni rendono possibile una produzione su scala molto più ampia e continua, in cui la fabbrica viene organizzata come grande impianto integrato.

La fase descritta da Marx, nella quale la macchina aveva portato all' esigenza di ridefinire il processo produttivo, frammentando i compiti e limitando sempre di più l'autonomia del lavoratore, al modello taylorista e fordista segna un passaggio cruciale nella storia dell'automazione. Se prima il problema riguardava soltanto la sostituzione del gesto umano, adesso riguarda anche l'*"organizzazione scientifica del lavoro"* [9]. In questa direzione si muovono Taylor e Ford, estendendo l'automazione alla dimensione del processo, introducendo una forma di controllo che non agiva più soltanto sulla tecnologia, ma direttamente sul comportamento umano, sul tempo, sui movimenti, sulle sequenze.

Il taylorismo e il fordismo si affermano quindi come tentativi di dare ordine alla nuova fabbrica industriale: un contesto sempre più complesso, in cui la frammentazione del lavoro e la perdita di autonomia operaia richiedevano metodi capaci di coordinare, standardizzare e stabilizzare il processo produttivo. Questo passaggio è essenziale nell'evoluzione dell'automazione perché la razionalità produttiva oltre ad essere incorporata nelle macchine viene incorporata all'intera organizzazione del lavoro, anticipando le forme di coordinamento algoritmico contemporanee.

1.3 Dal Taylorismo a Fordismo

Con il termine Taylorismo viene indicata l'organizzazione scientifica del lavoro, ovvero il sistema organizzativo - manageriale messo a punto dall'ingegnere americano Frederick Taylor. [10] Il sistema prevede la scomposizione delle attività in operazioni semplici, standardizzate e controllate, con l'obiettivo di aumentare il

[9] Taylor, Frederick Winslow. *The principles of scientific management*. NuVision Publications, LLC, 1911.

[10] Bonazzi, Giuseppe. "Taylorismo." Enciclopedia delle Scienze Sociali, vol. VI. Treccani, 1998.

rendimento della manodopera attraverso l'organizzazione scientifica del lavoro. In questa prospettiva, l'intensificazione razionale della manodopera è considerata il presupposto per la creazione di un surplus economico, ritenuto in grado di produrre benefici sia per l'impresa sia per i lavoratori[11]

Taylor opera in un contesto industriale americano attraversato da una profonda tensione tra le potenzialità tecniche di sviluppo e l'arretratezza dell'organizzazione del lavoro. Alla fine del 1800 il progresso tecnologico all'interno del processo industriale ha permesso di avanzare lungo due direzioni: la standardizzazione dei prodotti e la specializzazione delle macchine utensili. Alain Touraine osserva come questo progresso abbia trasformato radicalmente l'organizzazione interna dell'azienda e nello specifico la divisione del lavoro e i ruoli aziendali.

Le condizioni tecniche per la produzione di massa erano quindi mature, ma si scontravano con l'arretratezza della struttura organizzativa ancora primitiva e arbitraria. L'assetto dominante fino a quel momento era infatti il "drive system" che Sanford Jacoby definì come un sistema basato sulla pressione nei confronti dei lavoratori per l'aumento dell'output[12].

Il sistema che Taylor chiama Scientific Management, orientato al *one best way* e cioè al raggiungimento dell'unico modo migliore e più efficiente per svolgere un determinato lavoro, è articolato in quattro principi fondamentali:

1. studio scientifico dei migliori metodi di lavoro in rapporto alle caratteristiche dei lavoratori e delle macchine;
2. selezione e addestramento scientifico della manodopera;
3. instaurazione di rapporti di stima e di cordiale collaborazione tra direzione e manodopera;

---

[11] Bonazzi, Giuseppe. *Storia del pensiero organizzativo*. Vol. 136. FrancoAngeli, 2008.

[12] Jacoby, Sanford M. *Employing bureaucracy: Managers, unions, and the transformation of work in the 20th century*. Psychology Press, 2004.

4. distribuzione uniforme del lavoro e delle responsabilità tra amministrazione e manodopera[13];

Secondo Taylor, l'applicazione di questo sistema avrebbe dovuto produrre quindi benefici significativi per l'azienda, per i lavoratori e, più in generale per la società, grazie all'aumento della produttività e alla riduzione degli sprechi. Il Taylorismo sancisce una nuova forma di controllo manageriale sul lavoro che ridefinisce in modo strutturale il modo in cui le attività vengono scomposte, integrate e standardizzate in un sistema che ridefinisce ruoli, competenze e modalità di coordinamento. In sintesi, il Taylorismo costituisce il primo tentativo sistematico di razionalizzare il lavoro industriale, ma resta un modello focalizzato principalmente sull'ottimizzazione del singolo gesto e sulla disciplina del lavoratore. La sua portata è profonda, ma ancora limitata all'organizzazione immediata del compito. Soltanto negli anni successivi questa logica verrà sviluppata su larga scala, fino ad investire la struttura stessa della produzione.

È in quest'ultimo passaggio che si colloca Henry Ford, il quale non si limita ad applicare i principi tayloristi, ma li reinterpreta costruendo un nuovo sistema tecnico-organizzativo capace di riorganizzare l'intero flusso produttivo e, con esso, la fabbrica moderna. Ford, infatti, interviene sull'architettura della fabbrica con un paradigma che durerà fino almeno agli anni Settanta del Novecento. Con l'introduzione della catena di montaggio, della movimentazione meccanica dei pezzi e della continua sincronizzazione tra le diverse fasi, egli costruisce un flusso produttivo continuo che integra uomini e macchine. Questo flusso, che prima era regolato dal controllo gerarchico, viene con Ford subordinato al controllo automatico della produzione tramite la catena di montaggio. Sono due i concetti fondamentali della filosofia di Ford: la standardizzazione dei componenti e la specializzazione dei compiti. La standardizzazione consente di scomporre il prodotto in parti perfettamente intercambiabili e combinabili, la specializzazione,

[13] Bonazzi, Giuseppe. *Storia del pensiero organizzativo*. Vol. 136. FrancoAngeli, 2008.

invece, prevede la cooperazione passiva degli operai, orientata alla rigorosa esecuzione di quanto stabilito in fase di pianificazione.[14]

Il sistema fordista trasformò quindi l'intero tessuto sociale delle economie industriali. La produzione di massa e la centralità del flusso continuo hanno messo le basi per un nuovo equilibrio tra impresa, lavoro e Stato, portando a nuove forme di regolazione sociale come: salari più stabili, consumi standardizzati, nuove figure professionali e un'organizzazione del tempo più rigida e scandita.

Se il Taylorismo disciplina il gesto del singolo lavoratore, il Fordismo estende questa logica all'intero sistema produttivo, mostrando come uomini e macchine possano operare in coordinamento ridisegnando ruoli, gerarchie, competenze e rapporti sociali all'interno del contesto lavorativo. Taylor e Ford mostrano quindi come l'automazione non elimini il ruolo umano, ma ne riorganizzi le competenze, creando un collegamento stretto tra lavoratore e macchine, anticipando la logica dell'automazione cognitiva odierna.

1.2.2 Dal fordismo alla società automatizzata

L'aspetto che emerge dai paragrafi precedenti è come la tecnologia abbia modificato le strutture fisiche e l'organizzazione del lavoro all'interno della fabbrica. Un punto estremamente importante se si vuole riuscire a capire come oggi l'intelligenza artificiale generativa possa apportare cambiamenti nei processi lavorativi. È quindi necessaria una approfondita analisi degli effetti che le innovazioni tecnologiche hanno sul lavoro umano, partendo dai primi studi.

L'indagine di Alain Touraine, sociologo francese, si è concentrata proprio sugli effetti che le innovazioni tecnologiche hanno sul lavoro umano. La sua

[14] Della Rocca, Giuseppe, and Vincenzo Fortunato. *Lavoro e organizzazione: dalla fabbrica alla società postmoderna*. GLF editori Laterza, 2006.

ricerca, effettuata negli anni Quaranta del Novecento all'interno dello stabilimento Renault, mira a capire come l'evoluzione tecnologica nel settore dell'automobile stesse modificando l'organizzazione del lavoro e del sistema di fabbrica. La scelta della fabbrica Renault non è casuale poiché caratterizzata dalla compresenza stratificata di diverse tecnologie e forme di organizzazioni del lavoro, appartenenti a fasi storiche differenti: da sistemi pre-tayloristici a modelli tayloristici fino alle nuove soluzioni basate sulle macchine transfer.

Touraine identifica quindi tre fasi del lavoro che si sviluppano dal progressivo superamento, in età preindustriale, del modello artigianale, nella quale il lavoratore agiva direttamente sulla materia attraverso l'uso manuale degli utensili

Fase A – "*Sistema professionale di lavoro*" (pre-tayloristica)

Nasce con la creazione della macchina utensile e la incorporazione di questa nei processi di fabbrica. Queste, permettono la produzione in serie, offrendo al lavoratore un uso flessibile. Le macchine sono polivalenti e l'operaio ne conosce il funzionamento, ciò gli permette di adattarla, ripararla e intervenire per eliminare le imperfezioni. In questa fase, quindi, il processo di produzione è controllato dagli operai qualificati che formano dei gruppi naturali cooperativi.

Fase B – "*Fase capitalista*" (tayloristica)

Questa fase è caratterizzata dall'utilizzo di macchine specializzate in un processo continuo di lavoro a catena. Le macchine sono limitate alle lavorazioni per cui sono state progettate e disposte lungo la catena di montaggio che imposta il tempo di lavorazione mentre il lavoro è imposto per via gerarchica. Questa rappresenta il passaggio dal lavoro qualificato dell'operaio a quello dequalificato di alimentazione della macchina, i vecchi mestieri scompaiono e lasciano il posto a compiti semplici e limitati.

Fase C – "*Post industriale*" (macchine transfer)

Con la diffusione delle nuove macchine speciali e di quelle transfer capaci rispettivamente di eseguire sequenze complesse senza risettaggio e di integrare più stazioni di lavoro in un'unica linea automatizzata, viene eliminato il lavoro

direttamente produttivo. Le macchine in questa fase assorbono anche il lavoro semplificato e parcellizzato che prima veniva svolto dall'operaio. Questo produce due conseguenze: la prima è che diminuiscono gli operai non specializzati coinvolti direttamente nella produzione, mentre la seconda è che aumentano gli operai specializzati nel controllo, manutenzione e riparazione delle macchine. Queste tre fasi mostrano come l'evoluzione tecnologia abbia progressivamente ridefinito il ruolo del lavoratore, spostandolo dal controllo diretto della materia alla supervisione e gestione di sistemi sempre più complessi e delineando così la prima forma di complementarità tra lavoratore e macchina nel processo produttivo.

È in questa logica che Friedrich Pollock definisce l'automazione come un insieme di metodi tecnici di produzione e lavorazione automatica di beni, così come la raccolta ed elaborazione sistematica di informazioni. Si tratta quindi di un processo che ristruttura radicalmente le attività produttive e amministrative sostituendo progressivamente la forza lavoro fisica e intellettuale umana. Lo scopo dell'automazione è infatti, secondo Pollock, la sostituzione della forza lavoro umana nelle funzioni di servizio, comando e sorveglianza delle macchine, così come nel controllo dei prodotti, fino al punto in cui nessuna mano umana debba intervenire nel processo. L'automazione applicata opera in modo analogo negli uffici, rimpiazzando l'uomo nelle attività di calcolo, registrazione, trattamento statistico e controllo dell'informazione, oltre che nell'esecuzione di numerose operazioni prima svolte manualmente.[15]

Come osservato nei paragrafi precedenti, la progressiva sostituzione della macchina all'uomo nei processi produttivi apporta cambiamenti anche nella struttura organizzativa e nella ripartizione dei ruoli. Kern e Schumann, esaminano proprio come le innovazioni favoriscano la riqualificazione del lavoro. Secondo questi studiosi, le nuove tecnologie ribaltano proprio la modalità di organizzazione

[15] Pollock, Friedrich. "Automation: Materialien zur Beurteilung der oekonomischen u. sozialen Folgen." *Frankfurter Beiträge zur Soziologie* (1970).

della produzione e del lavoro tradizionale. Ma con l'introduzione di una tecnologia più complessa si rende necessario un controllo da parte di lavoratori qualificati e specializzati. Nella loro ricerca Kern e Schumann individuano una nuova figura lavorativa altamente specializzata: "*il conduttore di sistema*"[16]. Quest'ultimo è direttamente coinvolto nella produzione con funzioni di manutenzione, controllo e diagnosi delle macchine, intervenendo in autonomia per garantire la regolarità e qualità del processo.

Secondo Kern e Schuman se da una parte queste innovazioni portano all'ingresso di nuove figure specializzate, dall'altra portano anche ad una eliminazione di una quota rilevante di lavoratori non più necessari a causa delle nuove modalità di produzione. Da una parte si trovano quindi lavoratori qualificati nei nuovi metodi di produzione e dall'altro lavoratori meno qualificati il cui livello di competenza non è più richiesto. Un ulteriore prospettiva da prendere in considerazione è quella di Hanry Braverman, secondo cui l'innovazione tecnologica provoca uno spostamento del lavoro verso altre occupazioni e altri settori. In questo senso l'introduzione di nuove tecnologie elimina alcune mansioni tradizionali, creandone allo stesso tempo di nuove, e ridefinendo la composizione della forza lavoro.[17] L'automazione non andrebbe percepita come sinonimo di riduzione dei posti lavoro, ma piuttosto come processo di riorganizzazione. La conoscenza della macchina nel processo continuo viene quindi sostituita dalla conoscenza per il monitoraggio e la gestione delle nuove tecnologie.

---

[16] Kern, Horst, and Michael Schumann. *Das Ende der Arbeitsteilung?: Rationalisierung in der industriellen Produktion; Bestandsaufnahme, Trendbestimmung*. Beck, 1985.

[17] Braverman, Harry. *Labor and monopoly capital: The degradation of work in the twentieth century*. nyu Press, 1998.

1.3 Terza Rivoluzione Industriale: la nuova società dell'informazione

Nella Terza Rivoluzione Industriale, sviluppatasi a partire dalla seconda metà del 1900, l'evoluzione tecnologica conosce un passaggio ulteriore rispetto all' automazione descritta nel paragrafo precedente. Se il sistema di macchine aveva come obiettivo principale la sostituzione de lavoro umano e la razionalizzazione dei gesti manuali, l'introduzione delle tecnologie dell'informazione sposta l'attenzione sui processi informativi e decisionali che strutturano il lavoro e le organizzazioni.

Con la diffusione di calcolatori elettronici, microprocessori e sistemi informatici integrati, l'automazione influisce sulla raccolta, l'elaborazione e la circolazione delle informazioni. La tecnologia informativa entra simultaneamente in ambienti manifatturieri e d'ufficio, automatizzando operazioni e procedure amministrative. Questa possibilità di elaborare elettronicamente le informazioni rende il lavoro più chiaro, tracciabile e misurabile, creando nuove possibilità di controllo, coordinamento e apprendimento organizzativo.

In questo contesto si colloca la riflessione di Shoshana Zuboff, che interpreta la tecnologia come un dispositivo intrinsecamente ambivalente. Da una parte la nuova tecnologia può essere impiegata per automatizzare le attività, proseguendo la traiettoria della razionalizzazione industriale e della riduzione della dipendenza dalle competenze umane. Dall'altra, la stessa tecnologia produce nuove informazioni sui processi organizzativi, rendendo visibili attività prima non chiare, creando le condizioni per nuove forme di apprendimento, cooperazione e ridefinizione dei ruoli.

Queste ambivalenze si ritrovano nel dibattito attuale sull'intelligenza artificiale generativa. Infatti, i nuovi strumenti IA vengono presentati sia come mezzi per aumentare la produttività, ridurre il carico di lavoro e valorizzare le competenze sia come tecnologie utili alla standardizzazione dei processi, alla

dequalificazione di alcune mansioni e alla sostituzione all'uomo in molti ambiti lavorativi.

La distinzione proposta da Zuboff mostra come la tecnologia apra uno spazio di scelta organizzativa tra una funzione automatizzante, orientata alla standardizzazione delle operazioni e alla riduzione dei costi, e una funzione informativa e cognitiva, che amplia il contenuto conoscitivo del lavoro e redistribuisce l'accesso alle informazioni all'interno dell'organizzazione. Questa distinzione mette luce come la tecnologia informativa ponga le organizzazioni davanti ad una scelta strategica.

L'automazione informatica richiede che le attività vengano scomposte, formalizzate e trasformate in istruzioni codificabili. Questo processo porta all'esecuzione automatica delle operazioni, e alla rappresentazione informativa dettagliata del lavoro stesso. Tuttavia, molte organizzazioni si limitano a sfruttare il potenziale di automazione, trascurando il valore conoscitivo generato dalla tecnologia stessa.[18]

Manuel Castells affronta la questione concentrandosi sulla dimensione dell'organizzazione nel suo complesso. Castells parla di economia informazionale e di network society per descrivere una fase in cui la principale fonte di produzione è la capacità di analizzare le informazioni in tempo reale. La singola organizzazione entra così a far parte della rete di organizzazioni autonome e interdipendenti, collegate da flussi informativi continui e coordinate intorno a progetti e processi condivisi. La burocrazia verticale fordista viene affiancata dalle organizzazioni con burocrazia orizzontale e gerarchie appiattite.[19]

La differenza tra una tecnologia che rafforza il controllo gerarchico e una che favorisce nuove forme di autonomia e partecipazione sembra quindi non risiedere nell'artefatto tecnico in sé, ma nelle scelte organizzative che ne orientano

---

[18] Zuboff, Shoshana. *In the Age of the Smart Machine: The Future of Work and Power.* New York: Basic Books, 1988.

[19] Castells, Manuel. *The Rise of the Network Society.* Oxford: Blackwell, 1996.

l'uso. La società dell'informazione rappresenta un terreno di tensione tra continuità e trasformazione, nella quale si pongono le basi per le successive forme di automazione cognitiva.

La tecnologia informativa amplifica le capacità cognitive dell'uomo, creando nuove opportunità di collaborazione e controllo, e dimostra come la complementarità tra lavoro umano e macchine si possa estendere alla gestione delle informazioni.

1.4 La digitalizzazione

La Terza Rivoluzione Industriale, descritta nel paragrafo precedente, ha conosciuto un ulteriore evoluzione a partire dagli anni 2000. Questa fase è caratterizzata da uno sviluppo tecnologico estremamente rapido, in cui la digitalizzazione rappresenta l'elemento fondamentale della vita economica, sociale e produttiva. Come per le altre rivoluzioni industriali, anche questa è caratterizzata dalla trasformazione sistemica di modelli produttivi, organizzativi e di business.

Le nuove tecnologie consentono di facilitare e automatizzare numerose attività in una grande molteplicità di settori, in particolare quello industriale e commerciale, grazie all'adozione di sistemi online, dell'automazione avanzata e di una maggiore integrazione dei dati. Tuttavia, come ogni trasformazione tecnologica introduce anche nuove sfide, soprattutto in termini di occupazione, competenze e adattamento della forza lavoro, rendendo necessario un cambiamento profondo nel modo di lavorare e di produrre.

Un ruolo centrale è svolto dalle tecnologie di interconnessione, come l'Internet of Things, che permettono a macchine, dispositivi e sensori di comunicare tra loro e scambiare dati in modo continuo. In questo contesto si parla

anche di *"Industrial internet"* [20], inteso come l'infrastruttura digitale che connette macchine, persone e sistemi informativi, rendendo possibile una gestione integrata dei processi produttivi attraverso l'elaborazione e l'analisi dei dati. Internet diventa quindi l'infrastruttura all'interno del quale vengono progettati, eseguiti e monitorati molti processi lavorativi: applicazioni, piattaforme e servizi risiedono "in rete" e operano in tempo reale sui flussi continui di dati.

Questa crescente interconnessione ha comportato tuttavia anche un aumento della complessità dei sistemi produttivi e organizzativi. Le macchine non operano più come unità isolate, ma come nodi di rete integrate che richiedono nuove modalità di coordinamento e gestione. Diventa quindi necessario sviluppare nuovi modelli di organizzazione e progettazione del lavoro, capaci di integrare competenze umane e sistemi automatizzati.

In questa fase si assiste al passaggio dall'economia industriale, basata sull' automazione e sulla catena di montaggio ad una economia della conoscenza basata su tecnologie software digitali dell'informatica. In tale contesto emergono nuovi attori economici globali il cui valore non deriva dalla produzione materiale, quanto dalla gestione di informazioni, dati e conoscenza.

Allo stesso tempo l'automazione nell'attività manufatturiera diventa sempre più avanzata, le macchine sono in grado di prendere decisioni operative, adattarsi al contesto e ottimizzare i processi produttivi. Accanto a questa evoluzione, internet favorisce la nascita di nuove professioni legate allo sviluppo software, all'analisi dei dati, alla gestione delle piattaforme digitali e dei sistemi informativi. Figure come sviluppatori di codice, data analyst e system designer assumono un ruolo centrale all'interno delle organizzazioni.

In questo contesto il cambiamento interessa sia le organizzazioni nel loro complesso, con la nascita di nuove tipologie e forme di organizzazioni sia le strutture interne, attraverso nuove strutture organizzative, nuove strategie e una

---

[20] Industrial Internet Consortium - What is the Industrial Internet? 2015. URL: http://www.industrialinternetconsortium.org/about-industrial-internet.htm

diversa organizzazione del lavoro. I confini organizzativi, prima netti e definiti, diventano più fluidi e flessibili, favorendo la diffusione di "Network Organizzations"[21], ovvero organizzazioni che collaborano in network per sviluppare progetti o prodotti e servizi.

All'interno di queste strutture la gerarchia tradizionale tende a lasciare spazio a nuovi modelli sempre più decentrati e orizzontali, basati su gruppi di lavoro che condividono competenze e conoscenze. Il lavoro umano si sposta progressivamente da attività manuali e ripetitive verso attività cognitive, di analisi progettazione e decisione.

È proprio questa evoluzione che apre la strada al concetto di automazione cognitiva, creando le condizioni per l'introduzione e la diffusione dell'intelligenza artificiale generativa, capace di supportare e automatizzare attività tradizionalmente svolte dal lavoro cognitivo dell'uomo. La digitalizzazione mostra come l'uomo e le macchine possano collaborare, aumentando la produttività.

1.5 L'automazione cognitiva

Oggi ci troviamo davanti ad una nuova fase dell'evoluzione tecnologica, quella dell'automazione cognitiva, esito di un processo storico in cui, come visto nei paragrafi precedenti, la tecnologia ha progressivamente sviluppato la capacità di compiere attività cognitive prima svolte dall'essere umano. L'avvento dell'automazione cognitiva ha alla base la capillare diffusione dell'intelligenza artificiale, in particolare di quella generativa. Tale tecnologia, che incorpora forme

[21] Snow, Charles C., Raymond E. Miles, and Henry J. Coleman Jr. "Managing 21st century network organizations."

avanzate di sapere sociale e scientifico, incarna perfettamente il concetto marxiano di *general intellect*.

L'idea di macchine capaci di svolgere funzioni tipicamente associate all'intelligenza umana non è recente. Alan Turing, già negli anni Cinquanta del Novecento aveva posto l'attenzione sulla "*macchina pensante*" e cioè una macchina che potenzialmente potesse eguagliare la mente umana.[22]

Sebbene l'intelligenza artificiale sia quindi oggetto di studio da diversi decenni, solo negli ultimi anni è avvenuta una accelerazione significativa delle tecnologie nel campo dell'apprendimento automatico e dei modelli generativi che ha ampliato in modo sostanziale l'ambito delle attività di applicazione.

L'utilizzo dell'intelligenza artificiale generativa è adesso entrato a far parte di tutti i contesti aziendali, e non solo. Le attività per il quale può essere utilizzata sono sempre più innumerevoli, e vanno dalla pianificazione di attività e operazioni, al controllo autonomo di sistemi complessi fino alla programmazione automatica e alla analisi e scrittura di testi. In tutti questi casi l'IA amplifica il potenziale cognitivo del lavoratore, agendo come strumento di supporto.

Il dibattito sulla pericolosità e complessità che questa nuova tecnologia sta apportando e può apportare all'interno del mondo del lavoro è aperto e complesso. David Autor osserva che le tecnologie operano come una forza che scompone alcune attività e riorienta il lavoro verso altre attività lavorative complementari alla nuova tecnologia[23]. In questo senso l'utilizzo sempre più frequente dell'intelligenza artificiale generativa non implica necessariamente una scomparsa del lavoro umano nei settori nel quale viene applicato, piuttosto un aumento di potenza tecnica e una conseguente ristrutturazione organizzativa delle attività.

Questa impostazione si integra con la prospettiva di Braveman, per cui l'innovazione non elimina il lavoro in sé ma lo ridefinisce, e richiama l'analisi di

---

[22] Turing, Alan M. "Computing machinery and intelligence (1950)." *Mind* 59.236 (2021): 33-60.

[23] Autor, David H. "Why are there still so many jobs? The history and future of workplace automation." *Journal of economic perspective*

Touraine sul passaggio del lavoro diretto a quello di sorveglianza e gestione dei sistemi automatizzati. In modo analogo, nella fase attuale una quota crescente di attività lavorative si concentra sempre più su funzioni di integrazione, validazione e controllo della qualità dei risultati prodotti dai sistemi automatizzati.

Un punto fondamentale è che le tecnologie, dalla prima macchina motrice all' intelligenza artificiale, non hanno eliminato il lavoro in quanto tale, ma hanno sostituito specifiche attività e ridefinito il contenuto del lavoro umano. Quello che può essere osservato è che la differenza risiede nel metodo di utilizzo della tecnologia, se utilizzata come complemento, questa è in grado di estendere le capacità del lavoratore, migliorando efficienza, creatività e qualità del lavoro.

In questo senso, le modalità di utilizzo delle tecnologie informative dipendono dalle scelte manageriali. Sono infatti il management e la leadership organizzativa a decidere se orientare l'uso della tecnologia come metodo di standardizzazione o come complemento del lavoratore, rafforzandone così le capacità e migliorando il processo produttivo.

Questa tesi mira proprio ad analizzare quale sia l'aumento di produttività che l'intelligenza artificiale possa apportare all'interno di un'organizzazione, ponendo attenzione però alle dinamiche lavorative e sociali che questa influenza. Alla luce di questo quadro teorico, nel capitolo successivo verrà fatta un'analisi delle tecnologie di intelligenza artificiale con particolare attenzione a quella generativa e al suo utilizzo nei contesti aziendali, approfondendone le caratteristiche e le modalità di utilizzo, per poi esaminare i loro effetti concreti sui processi di lavoro e sull'organizzazione delle attività.

CAPITOLO 2 – L'IA GENERATIVA NEL CONTESTO ORGANIZZATIVO

Questo capitolo ha l'obiettivo di inquadrare l'intelligenza artificiale generativa dal punto di vista tecnico e funzionale, per delimitare con precisione la tecnologia oggetto di analisi della tesi. Dato che l'impatto delle tecnologie sul lavoro dipende in larga misura dalle loro caratteristiche operative e dalle modalità di integrazione nei contesti organizzativi, è necessario chiarire quale tipologia di intelligenza artificiale sia rilevante per l'analisi.

In particolare, il capitolo si concentra sui sistemi di intelligenza artificiale generativa e sul modo in cui questi vengano utilizzati in contesti lavorativi. Questo consente di distinguere tali sistemi sia dalle precedenti forme di automazione industriale sia dalle applicazioni di intelligenza artificiale tradizionali.

Attraverso la ricostruzione delle principali componenti tecnologiche dell'IA generativa e dei contesti di utilizzo il capitolo fornisce quindi le basi necessarie per comprendere il funzionamento di sistemi come GitHub Copilot.

2.1 L'intelligenza artificiale

Attualmente, non esiste una definizione di intelligenza artificiale universalmente riconosciuta. Il fenomeno è complesso e la rapidità con cui evolve rende difficile circoscrivere in maniera definitiva cosa significhi intelligenza artificiale. In ogni modo, esistono diverse definizioni che ne descrivono la natura, cercando di coglierne gli aspetti essenziali.

La prima definizione risale alla conferenza di Dartmouth del 1956 dove l'intelligenza artificiale venne definita da John McCarthy come "la scienza e l'ingegneria di macchine intelligenti, capaci di simulare processi mentali umani, attraverso algoritmi che permettano ai computer di svolgere compiti che richiederebbero intelligenza se fatti da un essere umano".[24] Questa coglie l'aspetto fondamentale, e cioè la capacità della macchina di simulare il pensiero umano. Da allora, il concetto di intelligenza artificiale si è ampliato notevolmente, arrivando a comprendere, secondo il Parlamento dell'Unione Europea, sistemi con capacità di ragionamento, apprendimento, pianificazione e creatività[25].

È opportuno chiarire che l'intelligenza artificiale non ha una forma unica e indivisibile, può infatti essere distinta in intelligenza artificiale generale (AGI) e intelligenza artificiale ristretta (ANI). La prima esiste solo in forma ipotetica e designa un'IA in grado di apprendere autonomamente e risolvere problemi mostrando una flessibilità cognitiva paragonabile a quella umana. Al contrario, la seconda può espletare compiti specifici e definiti. Tutte le applicazioni esistenti rientrano in quest'ultima categoria e costituiscono il fondamento dell'IA moderna.

L'intelligenza artificiale non ha avuto uno sviluppo lineare. Questa ha infatti subito periodi di grande entusiasmo e investimenti, seguiti da fasi di delusione e riduzione dei finanziamenti, note come gli "inverni dell'intelligenza artificiale[26]". Il primo inverno, negli anni Settanta fu legato al fallimento della traduzione automatica insieme alle dichiarazioni dell'Automatic Language Processing Advisor Commitee (ALPAC) che confutò la fattibilità della traduzione automatica. Il secondo inverno, al termine degli anni Ottanta, riguardò invece i sistemi basati

---

[24] McCarthy, John. *Dartmouth Summer Research Project on Artificial Intelligence*, Dartmouth College, 1956

[25] Che cos'è l'intelligenza artificiale? | Tematiche | Parlamento europeo

[26] Jiang, Yuchen, et al. "Quo vadis artificial intelligence?" *Discover Artificial Intelligence* 2.1 (2022).

su regole “se-allora”, che si rivelarono inefficienti, portando allo spostamento dell’interesse verso tecnologie emergenti, come i computer generici.

Soltanto a partire dagli anni Novanta e soprattutto dagli anni Duemila, l’IA ha conosciuto una nuova fase di ascesa. Tra i motori di spinta si trova la grande disponibilità di dati utili per l’addestramento dei modelli, il significativo aumento della capacità di calcolo dei nuovi sistemi e il successo del machine learning. Quest’ultimo è stato il motore di avvio della fase che ha portato alla costituzione delle prime teorie, come l’albero decisionale proposta da J.R Quinlan e il random Forest proposto da Breiman. All’interno del campo dell’intelligenza artificiale, il machine learning è emerso proprio come il metodo più affidabile per lo sviluppo di software come l’elaborazione del linguaggio naturale e il riconoscimento di immagini.

2.2 Machine Learning

Con machine learning si intende un programma per computer che apprende dall’esperienza E rispetto a una classe di compiti T e a una misura di prestazione P. Si può affermare che un programma “apprende” se la sua prestazione nei compiti T, misurata da P, migliora con l’esperienza E[27]. In altre parole, il machine learning è una disciplina che studia come costruire sistemi computazionali capaci di migliorare le proprie prestazioni nello svolgimento di specifici compiti attraverso l’esperienza, sfruttando dati invece di regole programmate manualmente.

È possibile distinguere tre diversi tipologie di machine learning in base al tipo di addestramento dell’algoritmo: apprendimento supervisionato che utilizza esempi etichettati con l’obiettivo di apprendere funzioni in grado di effettuare previsioni sui nuovi dati; apprendimento non supervisionato, che opera invece

[27] Mitchell, Tom M. "Artificial neural networks." *Machine learning* 45.81 (1997)

senza etichette e punta ad individuare strutture o schemi nascosti nei dati; apprendimento per rinforzo, infine, che si basa sull'interazione con un ambiente, nei quali apprende attraverso un processo di tentativi ed errori, ricevendo feedback sotto forma di ricompense o penalità.[28]

A seconda del tipo di apprendimento esistono diverse categorie di algoritmi. Tra queste si trovano: modelli di regressione, alberi decisionali, metodi bayesiani e reti neurali artificiali. Oggi, le reti neurali artificiali sono tra le tecniche di machine learning più utilizzate, soprattutto in ambiti come l'elaborazione del linguaggio naturale.

Le reti neurali artificiali sono una classe di modelli che, per via della loro struttura flessibile, possono essere adattate ad un'ampia varietà di contesti. Nello specifico una rete neurale è una funzione parametrizzata, ispirata alle reti neurali biologiche, che mappa un insieme di variabili di input in un insieme di variabili output attraverso una rete formata da layer ovvero unità interconnesse chiamate neuroni, i cui parametri vengono regolati durante l'apprendimento.[29] I principali parametri di una rete neurale includono i pesi associati alle connessioni tra neuroni, i bias, le funzioni di attivazione, il learning rate, il numero di epoche di addestramento e la funzione di perdita, che misura l'errore tra output previsto e output atteso. In sintesi, si tratta quindi di un modello matematico che analizza i dati in ingresso e li elabora attraverso diversi strati (layer) di neuroni artificiali collegati tra loro, migliorando, durante la fase di addestramento, la propria capacità di fornire risposte corrette.

All'interno delle reti neurali, come all'interno di un cervello umano, si trovano delle unità di elaborazione interconnesse chiamate neuroni artificiali; ognuna di queste unità trasmette segnali alle altre. Solitamente, i neuroni sono

---

[28] Janiesch, Christian, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning." *Electronic markets* 31.3 (2021): 687.

[29] Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. No. 4. New York: springer, 2006.

organizzati in reti con diversi strati, uno strato di input che riceve dati in ingresso e uno strato di output che produce il risultato finale. Tra questi vi possono essere zero o più strati nascosti, responsabili dell'apprendimento di una mappatura tra input e output [30]. Quando una rete neurale è caratterizzata dalla presenza di una grande quantità di strati nascosti, si parla di reti neurali profonde, che costituiscono la base del cosiddetto deep learning (Fig.1).

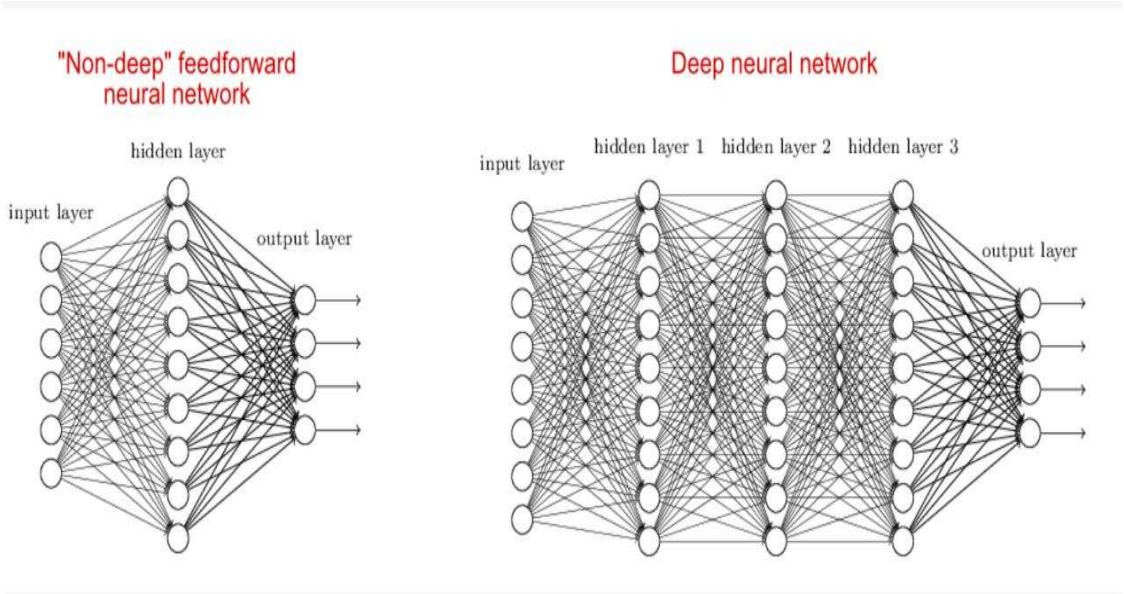

Figura 1 - Struttura delle reti neurali profonde (fonte: N. Boldrini. Deep Learning, cos'è l'apprendimento profondo, come funziona e quali sono i casi di applicazione)

Le reti neurali profonde sono formate da molteplici strati nascosti organizzati in strutture elaborate che utilizzano neuroni evoluti, capaci di svolgere operazioni complesse. Questa peculiarità consente alle reti neurali profonde di essere alimentate con dati input grezzi come immagini, testi e suoni, imparando automaticamente quali caratteristiche sono rilevanti per risolvere un determinato compito, senza che le regole vengano definite manualmente.

[30] Janiesch, Christian, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning." *Electronic markets* 31.3 (2021): 687.

Il deep learning consente a modelli computazionali, composti da più strati di elaborazione, di apprendere rappresentazioni dei dati a più livelli di astrazione.[31] In altre parole questo permette al modello di "capire" i dati costruendo progressivamente caratteristiche sempre più complesse: i primi strati individuano informazioni base, gli strati più profondi combinano queste informazioni in pattern più articolati, mentre gli strati finali consentono di prendere decisioni o fare previsioni più accurate. Il deep learning risulta quindi utile in ambiti nei quali sono presenti grandi quantità di dati complessi. Per questo motivo le reti neurali profonde superano molti algoritmi di machine learning tradizionali nella maggior parte delle applicazioni in cui devono essere elaborati dati testuali, immagini, video e audio.

È opportuno chiarire che esistono differenti tipologie di reti neurali, ciascuna con caratteristiche e ambiti di applicazione specifici. Tra le più utilizzate troviamo:

- Reti neurali feedforward: è una delle forme più semplici di reti neurali artificiali, in cui i dati o gli input si muovono in un'unica direzione. I dati passano attraverso i nodi di input e vengono restituiti dai nodi di output. Questa può avere o meno strati nascosti. In altre parole, presenta un'elaborazione in avanti senza ricorrenza.[32]
- Reti neurali convoluzionali (CNN): è un tipo di rete neurale feedforward. Sono costituite principalmente da un numero elevato di nodi computazionali in grado di estrarre dei dati mediante operazioni di convoluzione. Tipicamente è composta da tre strati: convoluzionali, di pooling e completamente connessi. I primi due servono a estrarre le

[31] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436

[32] Islam, Mohaiminul, Guorong Chen, and Shangzhu Jin. "An overview of neural network." *American Journal of Neural Networks and Applications* 5.1 (2019): 7-11.

caratteristiche dai dati di input, mentre l'ultimo trasforma le informazioni nel risultato finale, ad esempio una classificazione.[33]

- Reti ricorrenti (RNN): sono reti neurali che possiedono uno stato interno o una memoria a breve termine grazie a connessioni di retroazione ricorrenti, che le rendono adatte a trattare problemi sequenziali.[34]
- Reti generative: sono due reti neurali interconnesse, una generatrice e l'altra discriminatrice. Il generatore crea dati sintetici simili a quelli reali, mentre il discriminatore cerca di distinguere tra dati reali e falsi. Durante l'addestramento, le due reti sono coinvolte in una competizione continua, spingendo il generatore a produrre dati sempre più realistici e il discriminatore migliorare la sua capacità di riconoscimento.[35]
- Transformer: è una tipologia di rete neurale profonda che utilizza un meccanismo di self-attention per comprendere le relazioni contestuali all'interno di dati sequenziali. I transformer permettono l'elaborazione parallela degli input, migliorando l'efficienza e le prestazioni nei compiti come l'elaborazione del linguaggio naturale. [36]

La tecnologia Transformer oltre ad essere una architettura efficace per elaborare sequenze, costituisce anche la base strutturale dei moderni modelli di Large Language Model (LLM) che sfruttano questa tecnologia per comprendere e generare testo naturale su larga scala.

---

[33] Yamashita, Rikiya, et al. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9.4 (2018): 611-629.

[34] Koutnik, Jan, et al. "A clockwork rnn." *International conference on machine learning*. PMLR, 2014.

[35] Mienye, Ibomoiye Domor, and Theo G. Swart. "A comprehensive review of deep learning: Architectures, recent advances, and applications." *Information* 15.12 (2024): 755.

[36] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

2.3 Intelligenza artificiale generativa

L'intelligenza artificiale generativa (IAG) rappresenta una significativa evoluzione all'interno del campo dell'intelligenza artificiale, riconducibile ai progressi nel deep learning e nello sviluppo dei modelli generativi profondi.

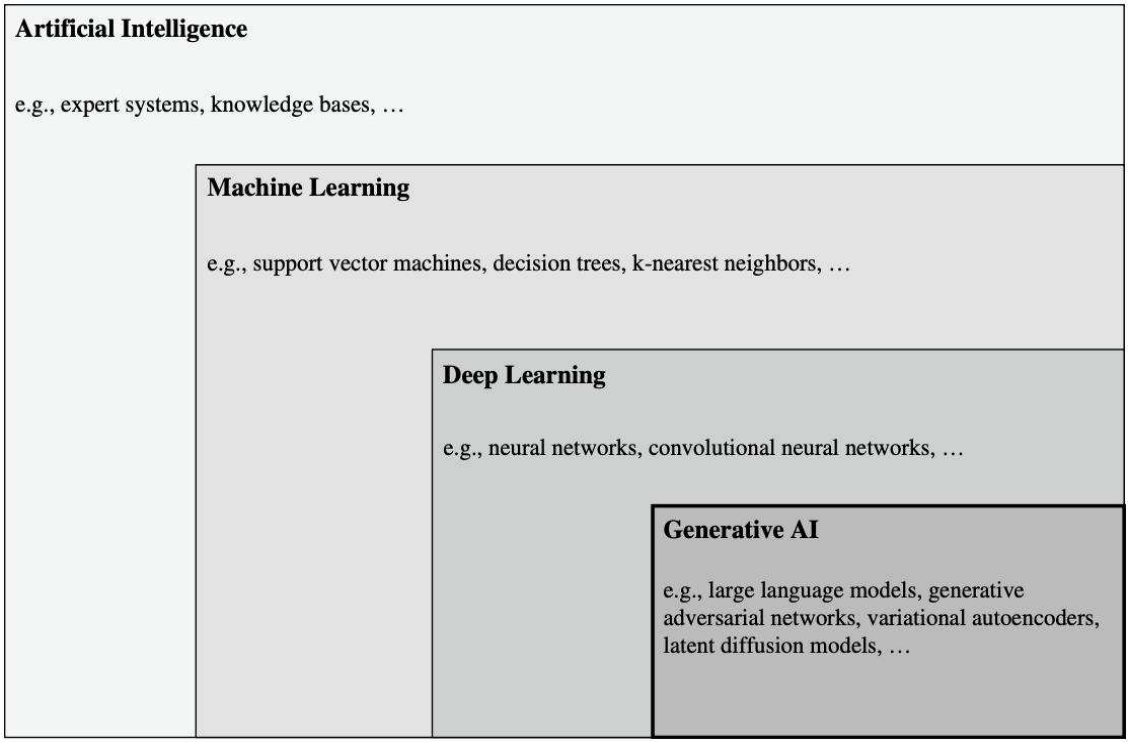

Figura 2- IA generativa e componenti (Fonte: Banh, Leonardo, and Gero Strobel. "Generative artificial intelligence.",2023)

Questa comprende modelli di apprendimento automatico che, a partire dall'analisi dei dati di addestramento, apprendono pattern e relazioni latenti al fine di produrre output nuovi e plausibili.

A differenza delle applicazioni di IA tradizionali, orientate prevalentemente a compiti discriminativi come classificazione, previsione o raccomandazione, l'intelligenza artificiale generativa si distingue per la capacità di produrre nuovi

contenuti come testi, immagini, audio o video[37], a partire da semplici input forniti dall'utente.

L'utilizzo dell'intelligenza artificiale generativa può avvenire tramite prompting, una tecnica di interazione che consente agli utenti finali di utilizzare il linguaggio naturale per interagire con le applicazioni IAG e istruirle a creare output desiderati come testo, immagini o altri tipi di contenuto. [38]

Le modalità utilizzate per l'addestramento dei modelli IAG sono differenti. I principali approcci comprendono l'addestramento supervisionato, in cui il modello apprende a partire da dati etichettati per predire risultati noti, l'addestramento non supervisionato in cui il modello identifica autonomamente pattern e strutture all'interno dei dati senza etichette predeterminate e l'addestramento per rinforzo, in cui il modello apprende attraverso un sistema di ricompense e penalità, migliorando le proprie decisioni.

Occorre ora fare una distinzione tra modelli IAG e sistemi IAG. Per modello di intelligenza artificiale generativa si intende un modello che sfrutta architetture di apprendimento automatico come il deep learning per apprendere da pattern di addestramento e generare output coerente con questi. Per sistema di intelligenza artificiale generativa si intende, invece, l'intera infrastruttura, inclusi modello, componenti di elaborazione dei dati e componenti dell'interfaccia utente. Un esempio di sistema è l'integrazione di modelli di deep learning come GPT all'interno di piattaforme interattive come GitHub Copilot, che supportano gli utenti nella programmazione.[39]

I grandi modelli di IAG, capaci di modellare l'output in modo completo e versatile all'interno di domini o tipologie di dati specifici o trasversali, sono spesso

---

[37] Generative AI | OECD

[38] Banh, Leonardo, and Gero Strobel. "Generative artificial intelligence." *Electronic Markets* 33.1 (2023): 63.

[39] Feuerriegel, Stefan, et al. "Generative ai." *Business & Information Systems Engineering* 66.1 (2024): 111-126.

definiti “foundation models”[40]. I modelli di intelligenza artificiale generativa possono essere classificati in modelli unimodali e multimodali in base alle tipologie di dati (testo, immagini, audio, video) che sono in grado di elaborare. I primi operano su una singola tipologia di dati e producono output della stessa natura, accettano input testuali e generano output testuali. I secondi, invece, integrano e combinano informazioni provenienti da più modalità. Possono ricevere input differenti, come testo e immagini simultaneamente, e generare output che possono appartenere a una o più modalità diverse.

2.4 Large Language Model (LLM)

Tra le diverse tipologie di modelli generativi, un ruolo centrale è oggi occupato dai Large Language Model (LLM), progettati per comprendere e generare testo naturale su larga scala. Questi modelli si basano sull’architettura transformer, che consente di modellare in modo efficace le relazioni contestuali all’interno del linguaggio. Gli LLM vengono addestrati su grandi quantità di dati testuali per comprendere e generare testo che riproduce in modo sempre più fedele il linguaggio umano. Tali modelli hanno apportato progressi significativi nel campo del Natural Language Processing (NLP) e trovano applicazione in numerosi settori, tra cui la programmazione.

Un LLM deve avere quattro caratteristiche fondamentali per essere considerato tale. Innanzitutto, deve essere in grado di comprendere e interpretare profondamente il linguaggio naturale, così da elaborare le informazioni e svolgere compiti come la traduzione o l’analisi di testi. In secondo luogo, deve essere in

[40] Bommasani, Rishi. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).

grado di generare testo coerente e naturale, in risposta a un prompt. In terzo luogo, deve possedere consapevolezza contestuale, tenendo conto della conoscenza specifica del dominio di riferimento, ossia della capacità di utilizzare informazioni contestuali e specialistiche. Infine, deve eccellere nella risoluzione di problemi e nel processo decisionale, sfruttando le informazioni presenti nei testi, rendendoli particolarmente adatti a compiti come il recupero di informazioni e sistemi di domanda e risposta.[41]

I Large Language Model non hanno tutti la stessa architettura. Distinguiamo modelli encoder-only e modelli decoder-only. I primi sono progettati principalmente per la comprensione del testo e risultano particolarmente efficaci in compiti di analisi e classificazione linguista. I secondi, invece, sono orientati alla generazione di linguaggio naturale, producendo testo in modo sequenziale sulla base del contesto precedente.

Negli ultimi anni, diversi LLM si sono affermati come modelli di riferimento per capacità, diffusione e ambiti applicativi. Pur differendo per architettura, dimensione, strategia di addestramento e grado di apertura del codice, condividono l'obbiettivo comune di comprendere e generare linguaggio naturale. Tra i modelli più rilevanti si collocano:

- GPT (Generative Pre-trained Transformer), sviluppata da OpenAI, sono modelli basati su un'architettura decoder- only. Questi sono ampiamente utilizzati in applicazioni conversazionali, nel supporto alla programmazione e nella sintesi e generazione di testi complessi. In particolare, GPT 3 rilasciato nel 2022 e GPT 4 rilasciato nel 2023 hanno contribuito alla diffusione e all'adozione dei LLM in ambito organizzativo.
- Claude, sviluppato da Anthropic e rilasciato nel 2023, è progettato anche esso per compiti conversazionali e di generazione testuale.

[41] Yang J., Jin H., Tang R., Han X., Feng Q., Jiang H., Yin B., Hu X. Harnessing the power of llms in practice: A survey on chatgpt and beyond

- PaLM 3, sviluppato da Google e rilasciato nel 2023, è un LLM progettato per applicazioni conversazionali, ricerca informativa e generazione testuale avanzata. Bard era il nome commerciale originario dell'applicazione conversazionale basata su questo modello. Nel 2024 PaLM è stato sostituito dal modello Gemini e Bard è stato rinominato con lo stesso brand Gemini.
- LlaMa 3, sviluppata da Meta e rilasciato nel 2023, è un modello open-source destinato ad uso accademico ed aziendale, con capacità avanzate di comprensione e generazione di testo

I modelli LLM vengono prevalentemente utilizzati attraverso la loro integrazione in sistemi applicativi, all'interno del quale costituiscono il componente intelligente alla base delle funzionalità del sistema. L'utente interagisce con il modello tramite piattaforme software progettate per specifici ambiti applicativi. Un esempio è ChatGPT, sistema conversazionale basato sui modelli della famiglia GPT, utilizzato come assistente nello svolgimento di compiti differenti di carattere generale. Oltre ai sistemi generalisti, i modelli LLM sono integrati anche in sistemi con funzionalità specifiche come GitHub Copilot, che utilizza un modello GPT progettato per supportare lo sviluppo software e basato su un modello della famiglia GPT.

2.5 Il prompting

Gli LLM sono dei modelli di predizione, i quali prendono del testo sequenziale come input (prompt) per restituire output. Il prompt engineering è il processo di progettazione e perfezionamento delle richieste input per ottenere

risposte desiderate dai Large Language Models[42]. In altre parole, può essere anche considerata come l'abilità di porre le domande giuste all'modello, il che porta ad una sperimentazione per trovare il prompt migliore, tramite l'ottimizzazione della lunghezza insieme alla valutazione dello stile e della struttura in relazione al compito da svolgere[43]. Bisogna tenere in considerazione che richieste più precise e accurate, arricchite da informazioni e istruzioni specifiche aiutano il modello a generare l'output attesto ed evitare risposte irrilevanti. La logica su cui si muovono i modelli è infatti quella della Garbage in = Garbage out, per il quale ad un input di scarsa qualità corrisponde un output di altrettanta scarsa qualità.[44]

2.6 Applicazioni degli LLM nel contesto organizzativo

L'intelligenza artificiale generativa e, in particolare, i sistemi basati su Large Language Model, rappresentano una svolta significativa nel contesto organizzativo grazie alla loro capacità di comprendere, generare e manipolare dati complessi. Questi hanno infatti dimostrato di avere grandi capacità in diversi contesti tra i quali:

- Generazione di testo: producono riassunti astrattivi di documenti in risposta a query specifiche e testi di varia lunghezza e stile, applicabili a notizie, abstract scientifici, verbali di riunioni o documenti legali. Nella scrittura creativa possono generare storie, poesie, dialoghi e sceneggiature, affrontando sfide di coerenza, dialoghi e controllo stilistico. Supportano

[42] Marvin, Ggaliwango, et al. "Prompt engineering in large language models." *International conference on data intelligence and cognitive informatics*. Singapore: Springer Nature Singapore, 2023.

[43] Boonstra, Lee. "Prompt engineering." *Google, https://www. kaggle. com/whitepaper-prompt-engineering* (2025).

[44] Lemeš, Samir. "Prompt Engineering." *Artificial intelligence in industry* 4 (2024): 159-170.

inoltre la creazione di contenuti di marketing, pubblicità, documentazione tecnica e attività accademiche.

- Traduzione e multilinguismo: traducono tra lingue differenti, incluse quelle per cui esistono pochi dati disponibili. Sono anche in grado di affrontare traduzioni "zero-shot", cioè tra lingue per cui non hanno mai visto esempi durante l'addestramento. Inoltre, riescono anche a interpretare e generare testi in cui più lingue sono mischiate, mantenendo coerenza e significato.
- Ragionamento e problem solving: comprendono ragionamento matematico (aritmetica, algebra, calcolo, analisi libera), logico (deduttivo, induttivo, abduttivo e analogico) e del senso comune, inclusi eventi fisici, sociali, temporali e spaziali.
- Comprensione e generazione di codice: completano funzioni e programmi, sintetizzano codice da specifiche in linguaggio naturale, comprendono, documentano e rilevano errori nel codice, effettuano traduzioni e refactoring tra linguaggi e supportano test e debugging. Un esempio è GitHub Copilot.
- Capacità multimodali: lavorano con testi, immagini, audio e video, generando descrizioni, rispondendo a domande visive, creando immagini da testo, comprendendo documenti strutturati, analizzando e sintetizzando contenuti audio e video.[45]

Modelli con queste competenze possono essere usati come assistenza automatizzata, ampliando le competenze e la creatività umana. L'ingegno umano viene così combinato a strumenti che operano come suo complemento. Un esempio è ChatGPT, il quale opera come partner piuttosto che come sostituto del lavoratore[46]. In altre parole, si potrebbe dire che soluzioni come le LLM aumentano l'efficienza complessiva dell'uomo, riducendo il lavoro manuale. In un

[45] Hays, Hasi. "Encyclopedia of Large Language Models and Foundation Models."

[46] Holmström, Jonny, and Noel Carroll. "How organizations can innovate with generative AI." *Business Horizons* (2024).

contesto aziendale questi strumenti aiutano le aziende a rimanere competitive, soprattutto in contesti in rapida evoluzione.

Grazie a queste capacità, che fanno di loro modelli trasversali, i LLM possono essere infatti utilizzati in una vasta gamma di settori. In ambito sanitario, ad esempio, vengono impiegati per l'analisi, la sintesi e il supporto alla gestione della documentazione clinica. Nel settore legale, invece, facilitano la ricerca giurisprudenziale e l'analisi di contratti e testi normativi. Nella ricerca scientifica contribuiscono ad accelerare la revisione della letteratura e supportare la formulazione di nuove ipotesi. Nell'ambito della programmazione i modelli possono comprendere e generare codice, suggerire completamenti automatici, individuare errori, supportare attività di refactoring e facilitare la traduzione tra linguaggi di programmazione, contribuendo così ad aumentare la produttività degli sviluppatori e a ridurre i tempi di sviluppo.

2.7 Etica, privacy e sicurezza

L'adozione dell'intelligenza artificiale generativa nei contesti organizzativi solleva una serie di questioni etiche, di sicurezza e di tutela della privacy che risultano centrali per comprendere i suoi effetti sul lavoro e sui processi decisionali. Da un punto di vista etico è necessario considerare che l'utilizzo di questi strumenti può indurre a fenomeni di eccessivo affidamento e eccessiva fiducia.[47]

[47] Golda, Abenezer, et al. "Privacy and security concerns in generative AI: a comprehensive survey." *IEEE Access* 12 (2024): 48126-48144.

L’integrazione di questi sistemi all’interno dell’ambiente lavorativo può infatti portare al fenomeno chiamato “automation bias” [48] e cioè all’ eccessivo affidamento all’intelligenza artificiale generativa, anche quando le risposte fornite dai sistemi sono errate o dannose per la qualità del lavoro svolto. Questo è un rischio particolarmente rilevante nei contesti in cui l’IA generativa viene impiegata come strumento di supporto alle attività lavorative, come ad esempio la produzione di documentazione o la scrittura di codice software. In queste situazioni, l’output del sistema può apparire plausibile e coerente, inducendo il lavoratore a ridurre le attività di verifica e controllo. Appare quindi evidente che ciò che viene prodotto dall’intelligenza artificiale generativa non debba essere inteso come sicuro e veritiero a priori e come l’attenzione e l’esperienza dell’utilizzatore sia fondamentale per la valutazione dei risultati.

Un'ulteriore criticità riguarda l’eccessiva fiducia nei confronti di questi strumenti, che spesso spinge i lavoratori a considerarli fonti affidabili di conoscenza, utilizzandoli in modo disinvolto, anche per attività o informazioni sensibili. In ambito organizzativo, questo può portare alla condivisione non controllata di dati riservati o strategici, esponendo l’impresa a rischi legati alla protezione delle informazioni e alla conformità normativa.

Dal punto di vista organizzativo, queste considerazioni rendono evidente la necessità di adottare una strategia di IA responsabile, che includa, insieme alla dimensione tecnica, aspetti formativi, etici e procedurali. Formazione dei lavoratori, definizione di linee guida sull’utilizzo dei sistemi generativi e processi di supervisione sono elementi fondamentali per garantire che l’intelligenza artificiale agisca come strumento efficace di supporto al lavoro e non come fattore di rischio o di impoverimento delle competenze.

In questo senso si muove l’IA act, normativa adottata a livello europeo del 2 agosto 2025 che ha introdotto un quadro regolatorio basato sul rischio,

[48] Skitka, L.J., Mosier, K.L., Burdick, M.: Does automation bias decision-making? Int. J. Hum-Comput St. 51, 991–1006 (1999).

imponendo specifici obblighi in termini di trasparenza, governance e supervisione umana per i sistemi di IA utilizzati in ambito professionale. Parallelamente standard internazionali, come quelli sviluppati in ambito ISO/IEC, forniscono linee guida per la gestione responsabile dell'intelligenza artificiale, con particolare attenzione alla sicurezza, all'affidabilità e al controllo dei rischi. Questi strumenti mostrano che l'intelligenza artificiale generativa non vada utilizzata come un semplice strumento tecnologico, ma debba essere integrata all'interno dei processi lavorativi attraverso scelte organizzative consapevoli, attività di formazione, regole d'uso chiare e pratiche supervisionate adeguate.

CAPITOLO 3 – GITHUB COPILOT

3.1- Copilot

GitHub Copilot è un sistema di intelligenza artificiale generativa introdotto da GitHub nel 2021 come strumento di supporto alla scrittura di codice software. Il sistema nasce all'interno di GitHub, società controllata da Microsoft, e si fonda su modelli di linguaggio di grandi dimensioni (LLM). Esattamente come Microsoft Copilot, anche GitHub Copilot è una piattaforma multi-modello che instrada le richieste verso LLM differenti in base alle attività e alle preferenze di configurazione dell'utente. La differenza fra i due sistemi è che Microsoft Copilot supporta la produttività generale, mentre GitHub Copilot è un assistente IA specializzato per sviluppatori.

Nel corso della sua evoluzione, GitHub Copilot ha adottato diversi modelli, riflettendo i progressi nell'ambito dei large language model e l'espansione delle possibilità offerte dalle piattaforme di IA. Tra i maggiormente utilizzati si trovano i modelli della famiglia GPT-4 che sono diventati parte centrale nel sistema di GitHub Copilot, insieme a modelli come Claude di Anthropic.

Dal punto di vista funzionale, GitHub Copilot è progettato per supportare lo sviluppo in un'ampia gamma di linguaggi di programmazione comunemente utilizzati in ambito professionale. Tra questi rientrano linguaggi come Python, JavaScript, TypeScript, Java, C#, C++ e Go, oltre a linguaggi e tecnologie per lo sviluppo web come HTML e CSS[49].

Strutturalmente GitHub Copilot opera come uno strumento integrato direttamente all'interno dell'ambiente di sviluppo utilizzato dal programmatore

[49] Zhang, Beiqi, et al. "Practices and challenges of using github copilot: An empirical study." *arXiv preprint arXiv:2303.08733* (2023).

(Plugin), attraverso un'estensione installata nell'IDE. Quest'ultimo, ovvero l'ambiente di sviluppo integrato, è un'applicazione software che raccoglie in un'unica piattaforma gli strumenti necessari alla scrittura, alla gestione e all'esecuzione del codice. Ai fini dell'analisi, l'utilizzo di GitHub Copilot viene esaminato con riferimento all'ambiente di sviluppo integrato Visual Studio Code. (figura 3)

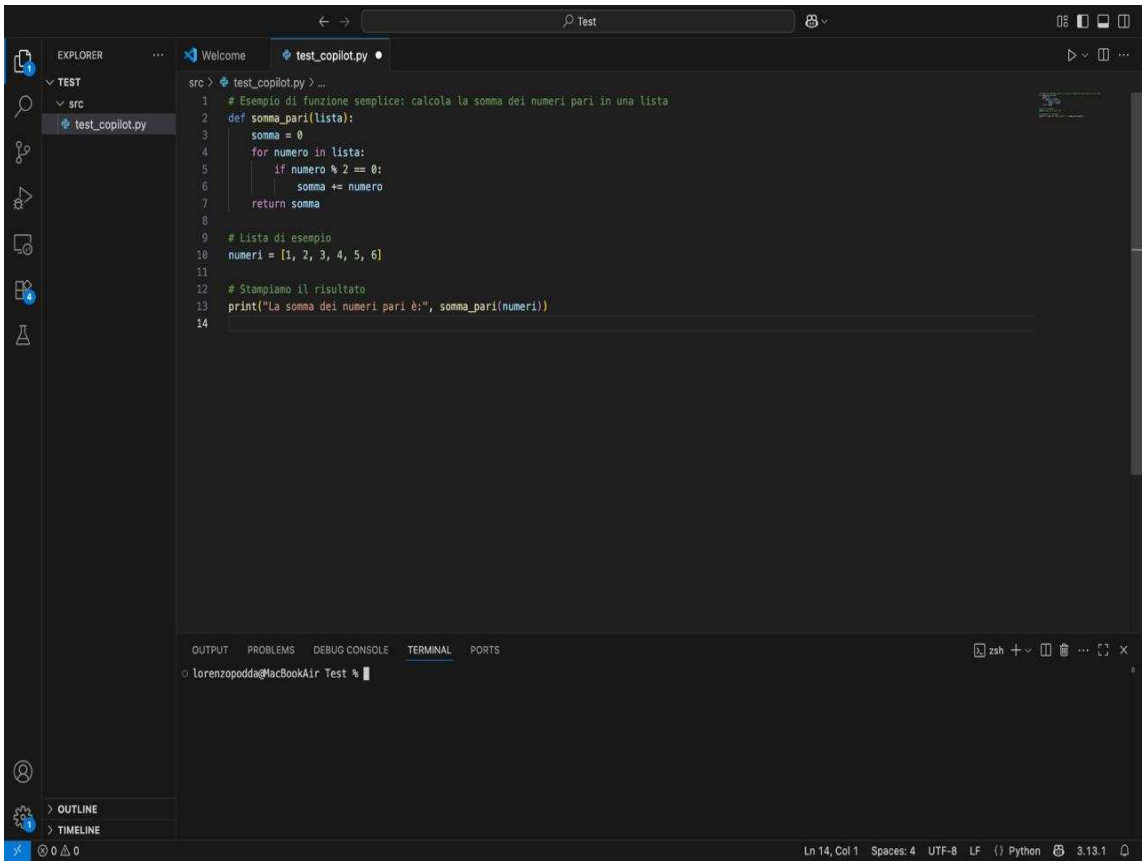

Figura 3: interfaccia base Visual Studio Code

All'interno di questo contesto operativo, GitHub Copilot mette a disposizione diverse funzionalità pensate per facilitare il lavoro degli sviluppatori. La funzionalità più immediata è quella dei suggerimenti inline, che vengono proposti direttamente nell'editor mentre lo sviluppatore scrive codice. GitHub Copilot analizza ciò che è presente nel file, considerando il codice già scritto, i

nomi delle variabili, la struttura dei file e gli eventuali commenti in linguaggio naturale, per generare completamenti automatici coerenti con l'attività in corso. Questa modalità consente di accelerare la scrittura di codice e ridurre il carico cognitivo associato a operazioni ripetitive. Ad esempio, scrivendo un commento come "calcola la somma dei numeri pari in una lista", GitHub Copilot è in grado di produrre immediatamente una possibile soluzione, lasciando allo sviluppatore la libertà di accettarla interamente o modificarla secondo le proprie esigenze (Figura 4).

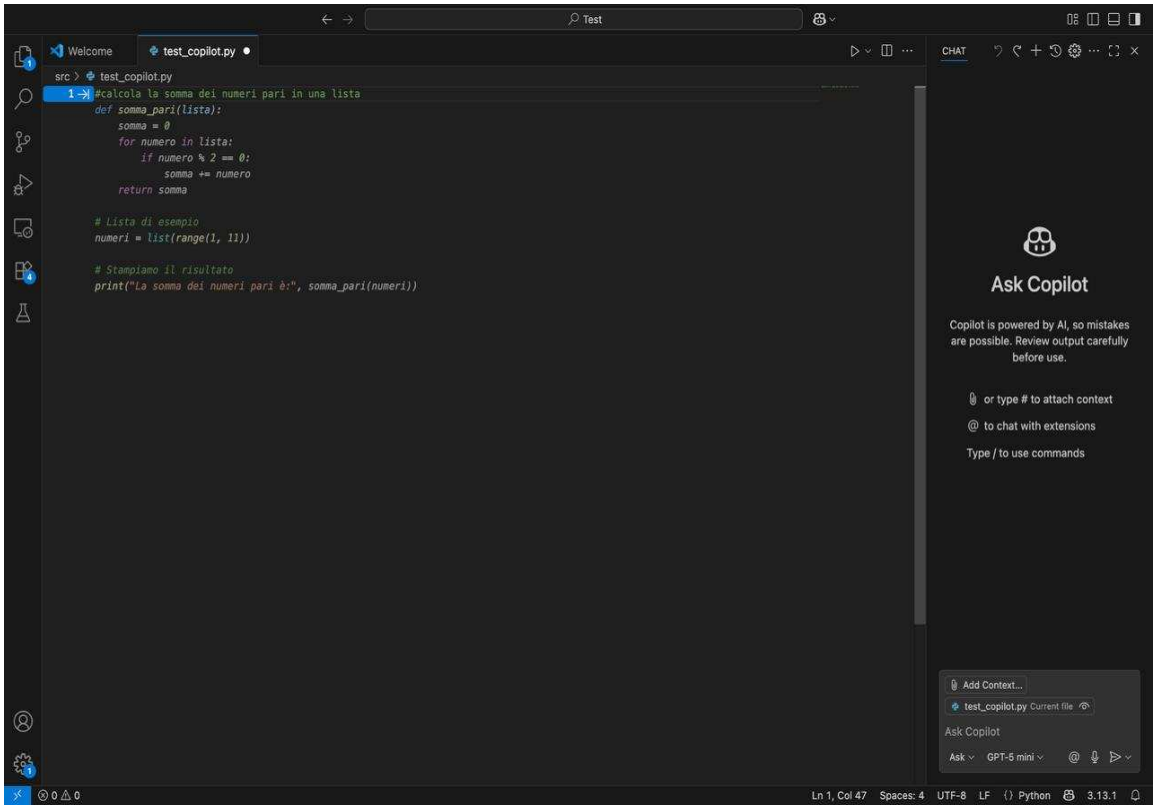

Figura 4: completamenti automatici

Un'altra funzionalità è rappresentata dalla chat integrata nell'interfaccia dell'IDE, generalmente collocata nella parte destra della schermata di Visual Studio Code. Attraverso questa interfaccia conversazionale è possibile interagire direttamente con GitHub Copilot utilizzando il linguaggio naturale. All'interno della chat, GitHub Copilot mette a disposizione due principali modalità di

interazione, comunemente indicate come modalità Ask e modalità Agent, che si differenziano per il livello di autonomia e di supporto offerto allo sviluppatore.

La modalità Ask consente allo sviluppatore di porre domande a GitHub Copilot relative al codice, chiedendo chiarimenti, suggerimenti di implementazione o supporto nella risoluzione di problemi specifici. In questa modalità, GitHub Copilot è in grado di generare porzioni di codice coerenti con il contesto del singolo file nella quale lo sviluppatore sta operando. Tuttavia, il ruolo di GitHub Copilot rimane quello di assistente reattivo: le soluzioni proposte vengono presentate come suggerimenti che lo sviluppatore può valutare, modificare e integrare manualmente.

La modalità Agent (figura 3), invece, rappresenta un livello di supporto più avanzato e proattivo. Mentre Ask si limita a rispondere a domande o a completare singole porzioni di codice, la modalità Agent è pensata per gestire attività più complesse e articolate all'interno di un intero progetto. In pratica, non guarda solo al file su cui si sta lavorando, ma analizza la struttura complessiva del progetto, valutando quali file, collegamenti e dipendenze sono rilevanti prima di proporre modifiche. Questo permette a GitHub Copilot di suggerire soluzioni coerenti e integrate, ad esempio per aggiornamenti su più file contemporaneamente o per operazioni di refactoring su larga scala.

La modalità agente funziona quindi in modo dinamico, elaborando le richieste nei cicli iterativi. Quando gli viene assegnato un compito, GitHub Copilot procede in più fasi: individua i file e le dipendenze pertinenti, suggerisce ed esegue modifiche coerenti con la struttura complessiva del progetto, esegue comandi del terminale necessari come compilazione del codice, installazione delle dipendenze o esecuzione di test, e infine monitora l'output affinando i risultati attraverso ripetute iterazioni. Questo processo permette a GitHub Copilot di migliorare progressivamente la qualità delle soluzioni, intercettando errori, perfezionando le

modifiche mantenendo lo sviluppatore al centro del controllo. [50] La peculiarità risiede quindi nella sua capacità di visione globale e nel supporto proattivo (figura 5)

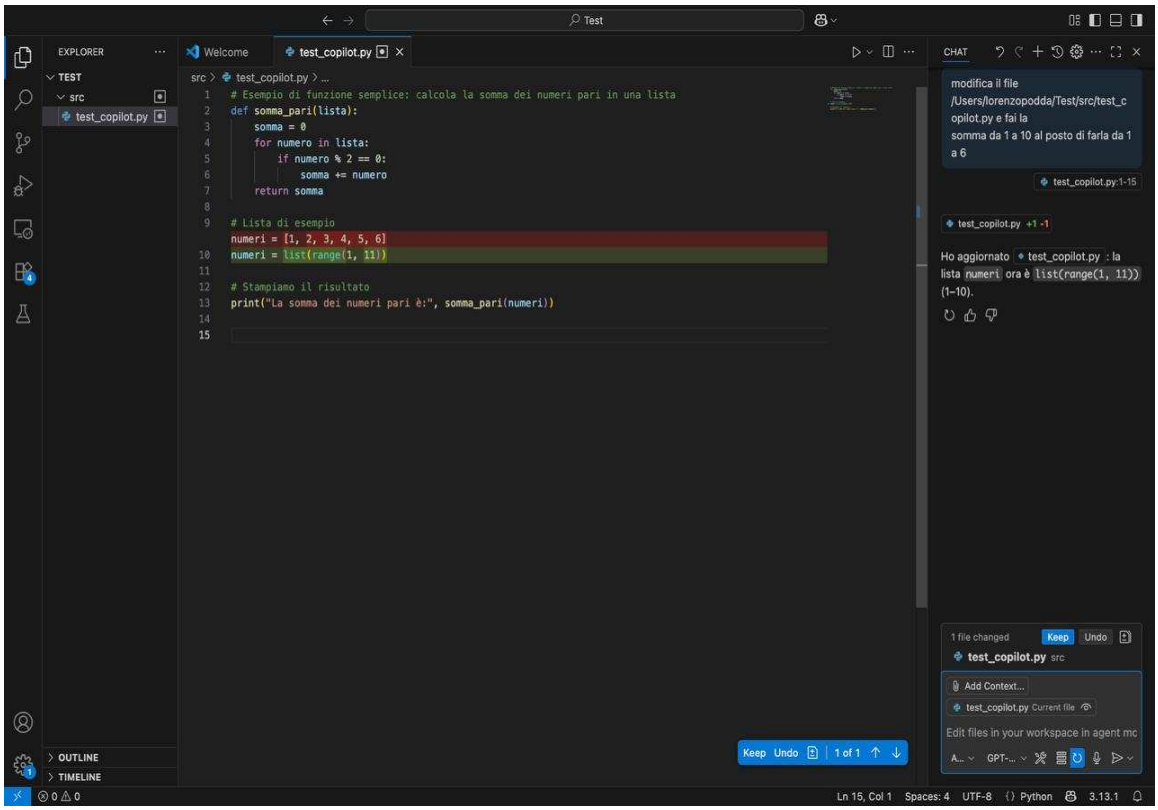

Figura 5:modalità Agent

Accanto alle funzionalità di supporto alla scrittura e all'implementazione del codice, GitHub Copilot offre anche strumenti dedicati alla revisione del codice (Figura 6). Questa funzionalità ha l'obiettivo di supportare lo sviluppatore nell'individuazione di possibili miglioramenti, errori o criticità all'interno del codice prodotto. Attraverso l'analisi automatica delle modifiche apportate, GitHub Copilot è in grado di generare suggerimenti utili alla revisione, evidenziando aspetti legati alla qualità del codice, alla leggibilità, alla coerenza con le pratiche

[50] "Che cos'è la modalità agente di GitHub Copilot?" *Microsoft Learn*. Che cos'è la modalità agente di GitHub Copilot? - Training | Microsoft Learn

di sviluppo adottate e alla presenza di potenziali problemi. Questi suggerimenti non sostituiscono il processo di revisione umana, ma rappresentano un supporto aggiuntivo che può facilitare l'individuazione di errori o aree di miglioramento, soprattutto nelle fasi preliminari della revisione.

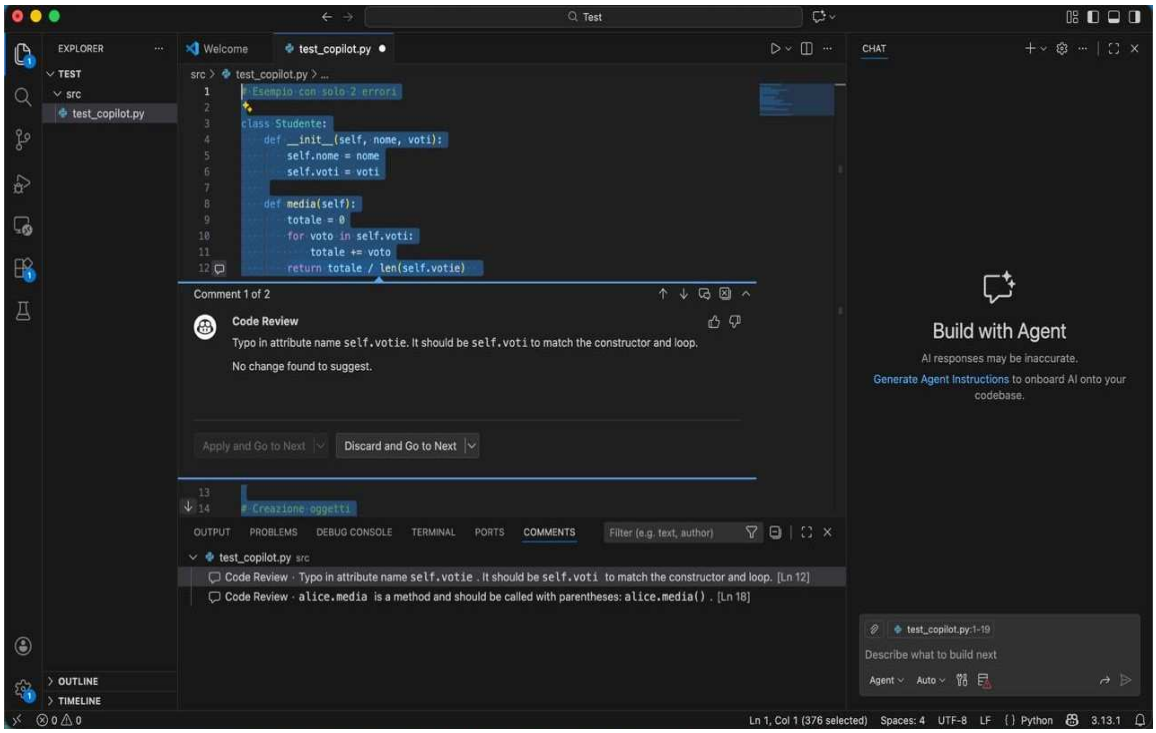

Figura 6: analisi del codice

3.2 GitHub Copilot come strumento di aumento della produttività

GitHub Copilot ha attirato fin dalla sua introduzione un forte interesse per la sua potenziale capacità di poter aumentare la produttività in contesti lavorativi in ambito di programmazione software. Diversi studi hanno analizzato l'impatto che questo strumento di intelligenza artificiale generativa può avere all'interno di

contesti lavorativi reali. Di seguito vengono presentati alcuni dei contributi più rilevanti.

Il primo studio preso in considerazione è quello condotto da Peng et al. (2022), nel quale gli autori hanno realizzato un esperimento controllato volto a misurare l'incremento di produttività derivante dall'uso di GitHub Copilot. Lo studio ha coinvolto 95 programmatori professionisti, suddivisi in due gruppi, ai quali è stato assegnato un compito di programmazione. I risultati hanno mostrato che il gruppo con accesso a GitHub Copilot ha completato il task in un tempo inferiore del 55,8% rispetto al gruppo di controllo51.

Il secondo studio rilevante è quello di Cui et al. (2024), che analizza l'impatto di GitHub Copilot sulla produttività in un contesto lavorativo reale, coinvolgendo un ampio campione di sviluppatori impiegati presso Microsoft e Accenture. A differenza dello studio precedente, la produttività non viene misurata tramite un compito artificiale, ma attraverso il confronto delle prestazioni degli sviluppatori prima e dopo l'adozione di GitHub Copilot, utilizzando dati di telemetria. Le metriche includono il numero di pull request, commit e build completate. I risultati hanno questa volta mostrato un incremento di produttività compreso tra il 12,9% e il 21,8% per gli sviluppatori di Microsoft e tra il 7,5% e l'8,7% per quelli di Accenture.52

Infine, lo studio di Weber et al. (2024) analizza l'effetto di diverse modalità di utilizzo degli assistenti basati su IA. In questo caso sono stati reclutati 24 programmatori, suddivisi in tre gruppi: un gruppo di controllo senza accesso a GitHub Copilot, un gruppo con accesso alla funzionalità di chat e un gruppo con accesso ai suggerimenti automatici di completamento del codice. I risultati indicano un incremento medio della produttività pari a circa il 65% nei gruppi che

---

[51] Peng, Sida, et al. "The impact of ai on developer productivity: Evidence from github copilot." *arXiv preprint arXiv:2302.06590* (2023).

[52] Cui, Kevin Zheyuan, et al. "The Productivity Effects of Generative AI: Evidence from a Field Experiment with GitHub Copilot." (2024).

hanno utilizzato GitHub Copilot, suggerendo come esso possa influenzare significativamente le prestazioni degli sviluppatori.[53]

Gli studi osservati forniscono evidenze rilevanti sull'impatto di GitHub Copilot in termini di aumento della produttività degli sviluppatori; tuttavia, presentano alcune limitazioni che rendono opportuno approfondire ulteriormente il tema.

Un primo elemento da considerare riguarda il fattore temporale. Tutti gli studi citati sono stati condotti tra il 2022 e il 2024, in una fase di rapida evoluzione dei modelli di intelligenza artificiale generativa. In particolare, i progressi registrati negli ultimi due anni nel campo degli LLM, come ad esempio l'introduzione della modalità Agent in grado di agire sul progetto e non solo sul singolo file, suggeriscono che i risultati ottenuti in studi precedenti potrebbero non riflettere pienamente le capacità attuali degli strumenti di programmazione assistita.

Un secondo aspetto rilevante riguarda la considerazione del livello di esperienza ai fini di un risultato attendibile. In alcuni degli studi osservati i partecipanti venivano divisi in due gruppi distinti, uno con accesso a GitHub Copilot e uno senza, assumendo che la randomizzazione fosse sufficiente a garantire l'omogeneità dei gruppi. In campioni di piccole dimensioni, questo approccio può introdurre dei bias, perché le differenze nel livello di esperienza o nelle competenze individuali tra i gruppi potrebbe influenzare i risultati in modo significativo. L'aumento di produttività potrebbe essere infatti attribuibile solo in parte all'utilizzo dello strumento, e in parte alle caratteristiche dei soggetti coinvolti. Questo elemento sottolinea quindi l'importanza di considerare disegni sperimentali che permettano di ridurre l'impatto della variabilità interindividuale, considerato che l'effetto di Copilot potrebbe essere influenzato dalle competenze individuali degli utilizzatori.

---

[53] Weber, Thomas, et al. "Significant productivity gains through programming with large language models." *Proceedings of the ACM on Human-Computer Interaction* 8.EICS (2024): 1-29.

Un ulteriore limitazione negli studi precedenti riguarda la definizione stessa di produttività. Le analisi si concentrano quasi esclusivamente sulla velocità di completamento o sulla quantità di output prodotto, dando poca attenzione alla qualità del codice generato. Un aumento della velocità di sviluppo non implica necessariamente un miglioramento complessivo del processo, soprattutto se accompagnato da problemi di manutenibilità, leggibilità o correttezza del software.

In breve, nonostante le ricerche precedenti abbiano fornito importanti indicazioni sull'impatto di Copilot sulla produttività, hanno evidenziato risultati variabili e presentando anche limiti metodologici e temporali. Questo contesto giustifica quindi la necessità di ulteriori studi che combinino misure precise di tempo, quantità e qualità, tenendo conto della variabile aggiornati agli attuali strumenti di programmazione assistita basati sull'intelligenza artificiale.

CAPITOLO 4 – METODOLOGIA DELLA RICERCA

Nel capitolo precedente è stato presentato GitHub Copilot dal punto di vista strutturale e funzionale, esponendo anche le principali analisi presenti in letteratura sull'impatto di questo strumento nella produzione di codice software. L'analisi ha evidenziato come gli strumenti di programmazione assistita basati sull' intelligenza artificiale possano incidere in modo significativo sul lavoro degli sviluppatori, in particolare in termini di produttività, sebbene i risultati delle analisi presentate precedentemente non siano uniformi e contengano alcuni limiti metodologici. Tra i limiti assumono particolare rilievo la rapida evoluzione degli strumenti, la difficoltà di misurare in modo univoco la produttività nello sviluppo software e la necessità di considerare, accanto alla velocità di esecuzione, anche la qualità del risultato prodotto e del codice generato.

Sulla base di queste considerazioni, il presente capitolo descrive il disegno metodologico adottato per analizzare l'impatto di GitHub Copilot da una prospettiva empirica, attraverso la somministrazione di un questionario e lo svolgimento di test di programmazione. Il questionario è stato distribuito ad un gruppo di 80 dipendenti dell'azienda Spindox, selezionati in quanto in possesso della licenza GitHub Copilot, strumento oggetto dell'analisi. I test di programmazione, invece, sono stati condotti su un sotto-campione di 10 dipendenti della stessa azienda, che hanno collaborato su base volontaria. Lo studio è stato costruito con l'obiettivo di integrare due livelli di osservazione complementari: da un lato la percezione generale degli sviluppatori nei confronti degli strumenti di intelligenza artificiale generativa e dello strumento GitHub Copilot, rilevata tramite questionario; dall'altro la misurazione diretta delle prestazioni di sviluppo, osservata attraverso un test sperimentale di programmazione svolto in condizioni controllate.

Le sezioni successive presentano quindi gli obiettivi della ricerca, le domande di ricerca, la struttura dell'esperimento, le modalità di raccolta dei dati, le metriche utilizzate per la valutazione dei risultati e i limiti che sono stati riscontrati.

4.1 Obiettivo e domande di ricerca

L'obiettivo principale di questa tesi è analizzare l'impiego di strumenti di intelligenza artificiale generativa, ponendo particolare attenzione a GitHub Copilot, nel contesto organizzativo e comprendere se l'utilizzo di tali strumenti stia apportando un cambiamento nell'attività lavorativa e nella sua organizzazione. L'analisi è quindi orientata a comprendere quali siano gli effetti sul lavoro individuale degli sviluppatori (in termini di produttività e qualità), quale sia la percezione dei lavoratori stessi quali possano essere i cambiamenti organizzativi connessi all'adozione di tali strumenti.

Di seguito vengono riportare le domande di ricerca:

1. **RQ1** – *Qual è l'effetto dell'uso di GitHub Copilot sulla produttività individuale nello sviluppo software, considerando sia misure oggettive sia percezioni degli sviluppatori?*
2. **RQ2** – *In che misura l'uso di Copilot influenza la qualità del risultato e del codice prodotto, e quali sono le criticità percepite?*
3. **RQ3** – *Qual è il livello di soddisfazione degli sviluppatori nei confronti di strumenti di Intelligenza Artificiale Generativa e nello specifico di GitHub Copilot?*
4. **RQ4** – *Quali sono gli effetti di questa dinamica sul mercato del lavoro?*

Per raggiungere questi obiettivi, lo studio combina:

- misure oggettive, ottenute tramite un test pratico di programmazione;
- misure soggettive, raccolte tramite un questionario sulla percezione e l'utilizzo di Copilot.

L'integrazione di queste due misure consente infatti di osservare se emergono cambiamenti nelle prestazioni e quale sia la percezione degli sviluppatori sull'utilizzo di un assistente generativo nel processo di sviluppo.

4.2 Descrizione del campione

La raccolta dei dati si è articolata in due momenti distinti.

La prima fase ha previsto l'invio di un questionario ad un gruppo di 80 sviluppatori, dipendenti dell'azienda Spindox e dotati di licenza GitHub Copilot, con differenti livelli di esperienza professionale. Di questi 45 hanno risposto e 33 hanno dichiarato di utilizzare GitHub Copilot. Questo primo insieme di dati ha consentito di raccogliere informazioni percettive sull'utilizzo dello strumento, in linea con la componente quantitativa ed esplorativa prevista dal disegno di ricerca.

Il test sperimentale è stato invece somministrato a un gruppo più ristretto di sviluppatori, selezionati su base volontaria. Il sotto campione iniziale era composto da 10 partecipanti, tutti con un livello di esperienza senior. Questa scelta è stata adottata con lo scopo di ridurre l'impatto della variabilità interindividuale dovuta all'esperienza e osservare l'effetto dello strumento limitando possibili bias dovuti a livelli di esperienza professionale differenti.

Dei 10 test raccolti, tuttavia, soltanto 7 sono risultati pienamente validi e utilizzabili ai fini dell'analisi.

Nel complesso, il gruppo selezionato presenta quindi una duplice articolazione: da un lato un insieme più ampio di rispondenti al questionario, utile per cogliere percezioni e orientamenti generali; dall'altro un gruppo ristretto e controllato, impiegato per il confronto diretto delle performance con e senza supporto di GitHub Copilot. Lo studio deve quindi essere considerato come preliminare, con l'obbiettivo di fornire evidenze empiriche sull'impatto di GitHub Copilot senza generalizzare il risultato all'intera popolazione aziendale.

4.3 Descrizione questionario

Lo scopo principale della survey è stato quello di analizzare in modo generale la percezione degli sviluppatori nei confronti degli strumenti di intelligenza artificiale generativa, con particolare riferimento a GitHub Copilot.

In particolare, il questionario può essere suddiviso in tre parti.

Nella prima parte sono state raccolte informazioni di contesto relative al background dei partecipanti, come il livello di istruzione, le modalità attraverso cui sono stati appresi i fondamenti della programmazione e la frequenza con cui vengono aggiornate le competenze tecniche. Sono state anche raccolte informazioni sui linguaggi di programmazione utilizzati più frequentemente e sulla percezione del proprio livello di produttività nello sviluppo del codice.

La seconda parte del questionario è stata dedicata all'esperienza degli sviluppatori con strumenti di assistenza alla programmazione basati su intelligenza artificiale, con particolare riferimento a GitHub Copilot. In questa parte sono state raccolte informazioni sull'utilizzo attuale degli strumenti di IA generativa, sul livello di familiarità con essi e sul grado di comfort percepito nell'integrarli nel proprio flusso di lavoro. Inoltre, il questionario ha esplorato la percezione degli sviluppatori rispetto all'impatto degli strumenti di IA generativa sul processo di

sviluppo software, chiedendo ai partecipanti di esprimere il proprio giudizio sull'eventuale incremento di produttività associato al loro utilizzo, sulla qualità del codice prodotto con il supporto dell'AI e sul livello di apertura delle organizzazioni all'adozione di queste tecnologie.

Infine, la terza parte del questionario si concentra sulla presenza o meno di aspetti di resistenza o esitazione nell'utilizzo di strumenti di IA generativa nello sviluppo software. A tal fine sono state incluse domande volte a esplorare possibili criticità percepite dagli sviluppatori, come preoccupazioni relative alla sicurezza del codice, alla perdita di autonomia nello sviluppo, alla mancanza di formazione specifica o a barriere di tipo organizzativo e culturale.

4.4 Descrizione Test programmazione

Il test ha previsto la trasformazione di un file CSV di input in un file CSV di output secondo specifiche regole di elaborazione dei dati. L'obiettivo del compito assegnato è stato quindi simulare una tipica attività di data processing, frequentemente presente nello sviluppo software, che richiede la lettura, la trasformazione e l'aggregazione di dati strutturati.

Il file di input, denominato *input.csv*, conteneva diverse informazioni relative a una serie di transazioni. In particolare, ogni riga del dataset rappresentava una singola transazione associata a un determinato produttore e a uno specifico paese in un determinato momento temporale. Il dataset includeva i seguenti campi informativi: un identificativo della transazione (*uid*), il paese associato alla transazione (*country*), il produttore (*manufacturer*), il valore della transazione (*transaction_import*) e un timestamp che indicava il momento in cui la transazione era stata registrata.

Il compito dei partecipanti consisteva nello sviluppare un programma in grado di leggere il file di input ed elaborare i dati secondo alcune regole specifiche. In particolare, a partire dal campo *timestamp* era necessario estrarre l'anno della transazione, che sarebbe stato successivamente utilizzato come dimensione di aggregazione dei dati. I partecipanti dovevano quindi organizzare le informazioni raggruppando le transazioni sulla base di tre attributi: il paese (*country*), il produttore (*manufacturer*) e l'anno estratto dal timestamp.

Una volta effettuato il raggruppamento dei dati, il programma doveva calcolare la media del valore delle transazioni (*transaction_import*) per ciascuna combinazione di paese, produttore e anno. Il risultato di questa elaborazione doveva poi essere esportato in un nuovo file denominato *output.csv*.

Il file di output doveva contenere un insieme di record aggregati, ognuno dei quali rappresentava il risultato dell'elaborazione per una specifica combinazione di paese, produttore e anno. Per ciascun record dovevano essere riportati un identificativo progressivo (*uid*), il paese (*country*), il produttore (*manufacturer*), l'anno della transazione (*year*) e il valore medio delle transazioni calcolato per quel gruppo (*transaction_average*). L'identificativo *uid* doveva essere generato dal programma come valore incrementale per ciascun record prodotto nel file di output.

Nel complesso, il compito richiedeva l'implementazione di un processo completo di elaborazione dei dati che comprendeva diverse fasi tipiche dello sviluppo software: la lettura di dati da una sorgente esterna, la trasformazione delle informazioni disponibili, l'aggregazione dei dati secondo specifiche regole e la generazione di un nuovo dataset contenente i risultati dell'elaborazione.

4.5 Raccolta dei dati

La raccolta dei dati è stata effettuata attraverso le due principali fonti: il questionario e il test sperimentale di programmazione svolto da un gruppo più ristretto di partecipanti. Questa combinazione consente di analizzare il fenomeno da due prospettive complementari: da un lato comprendere la percezione generale degli sviluppatori nei confronti degli strumenti di Intelligenza Artificiale Generativa, dall'altro osservare in modo diretto e misurabile l'effetto dell'utilizzo di tali strumenti sulle attività di sviluppo software.

Tramite il questionario sono state raccolte informazioni relative alla percezione dell'utilizzo degli strumenti di IA generativa nello sviluppo software, con particolare riferimento a GitHub Copilot. In particolare, il questionario ha indagato il grado di familiarità con questi strumenti, la percezione del loro impatto sulla produttività e sulla qualità del codice, nonché eventuali fattori di resistenza o esitazione nel loro utilizzo.

Per quanto riguarda il test sperimentale, sono stati registrati diversi tipi di informazioni relative alle prestazioni dei partecipanti. In primo luogo, è stato misurato il tempo di sviluppo, espresso in minuti, necessario per completare il compito assegnato nelle due condizioni sperimentali (con e senza supporto di IA). Questa misura consente di valutare direttamente l'efficienza del processo di implementazione e di osservare eventuali differenze nei tempi di realizzazione della soluzione tra le due modalità di sviluppo.

Oltre al tempo di sviluppo è stata analizzata la qualità funzionale del risultato prodotto. Questo aspetto è stato valutato attraverso un punteggio di correttezza ottenuto confrontando l'output generato dal programmatore in entrambe le condizioni (con e senza IA) con un risultato di riferimento (Gold). Questa misura consente di verificare se le soluzioni implementate dai partecipanti producono risultati corretti e coerenti con le specifiche del compito.

Un ulteriore elemento analizzato riguarda la complessità del codice prodotto. A tal fine sono state utilizzate metriche di analisi statica del codice che permettono di stimare il livello di complessità logica e cognitiva delle soluzioni implementate. L'analisi di queste metriche consente di osservare se l'utilizzo di strumenti di assistenza basati su AI influisca sulla struttura del codice e sulla sua potenziale manutenibilità.

Nel complesso, la combinazione di dati oggettivi derivanti dal test sperimentale e informazioni percettive raccolte tramite questionario consente di ottenere una visione più completa del fenomeno, permettendo di analizzare sia gli effetti concreti dell'utilizzo di strumenti di IA nello sviluppo software sia il modo in cui tali strumenti vengono percepiti dagli sviluppatori.

4.6 Valutazione della qualità del risultato

La qualità funzionale delle soluzioni prodotte dai partecipanti è stata valutata utilizzando DeepEval, uno strumento di valutazione automatica progettato per analizzare e confrontare un output generato dalle soluzioni implementate rispetto a un risultato di riferimento atteso detto anche "Gold standard".

DeepEval è una libreria utilizzata per la valutazione automatizzata dei risultati prodotti da sistemi software o modelli generativi. Lo strumento consente di definire criteri di confronto tra un output generato e una soluzione corretta di riferimento, permettendo di quantificare il livello di correttezza della soluzione attraverso un punteggio numerico normalizzato. Questo approccio consente di effettuare valutazioni oggettive e riproducibili, riducendo la necessità di un'analisi manuale dei risultati.[54]

[54]*DeepEval documentatiom https://deepeval.com/docs/getting-started?utm_source=chatgpt.com*

Nel contesto di questo studio, DeepEval è stato utilizzato per confrontare il file output prodotto dal programma sviluppato da ciascun partecipante con un Gold standard, ovvero un file di output corretto generato tramite un'implementazione di riferimento del compito. L'implementazione di riferimento è stata realizzata in modo da seguire rigorosamente le specifiche del problema, producendo quindi il risultato corretto atteso detto appunto Gold standard.

Il confronto tra l'output generato dai partecipanti e il Gold standard è stato effettuato tramite una metrica di valutazione personalizzata, progettata per considerare diversi aspetti della correttezza della soluzione.

Il punteggio finale viene infatti calcolato tramite la combinazione di metriche differenti. La metrica principale è la "value accuracy" che con un peso del 60% sullo score finale, misura la correttezza dei valori calcolati nel file di output. A questa si affiancano due metriche che valutano la correttezza delle soluzioni presenti nel risultato: la "recall", che misura la capacità della soluzione di produrre tutte le righe attese ed ha un impatto del 20% sul punteggio, e la "precision", che penalizza la presenza di righe non previste nel risultato ed ha un peso del 10%.

Sono inoltre stati considerati due aspetti strutturali del file di output: la correttezza dello schema del file, valutata tramite uno schema score con peso del 5%, e la corretta generazione dell'identificativo progressivo uid, anch'essa con un peso del 5%. Quest'ultima misurazione consente di verificare che la soluzione produca un identificatore progressivo coerente con le specifiche del compito.

Sulla base di queste metriche, DeepEval restituisce uno punteggio normalizzato compreso tra 0 e 1, che rappresenta il grado di corrispondenza tra il risultato prodotto e il risultato corretto.

Uno punteggio pari a 0 indica che la soluzione prodotta è completamente errata o non corrisponde ai risultati attesi, mentre uno score pari a 1 indica una soluzione completamente corretta, ovvero perfettamente coincidente con il risultato di riferimento. Valori intermedi rappresentano livelli parziali di

correttezza, indicando che alcuni elementi della soluzione sono corretti mentre altri presentano errori o discrepanze rispetto al risultato atteso.

L'utilizzo di uno strumento di valutazione automatica come DeepEval ci consente quindi di ottenere una misura quantitativa e oggettiva della qualità delle soluzioni prodotte dai partecipanti, rendendo il processo di valutazione più consistente e replicabile.

4.6.1 Analisi della complessità del codice

Oltre alla correttezza funzionale delle soluzioni prodotte, è stata analizzata anche la qualità strutturale del codice prodotto dai partecipanti. In particolare, l'analisi si è concentrata sulla complessità del codice generato in entrambe le condizioni (con e senza l'utilizzo di GitHub Copilot), valutando aspetti quali leggibilità, manutenibilità e facilità di comprensione del software.

Per questa analisi sono state utilizzate due metriche di analisi statica ampiamente adottate nella pratica dello sviluppo software e nella letteratura di ingegneria del software: Cyclomatic Complexity e Cognitive Complexity.

L'utilizzo combinato di queste due metriche consente di analizzare la complessità del codice da due prospettive differenti: la prima legata alla struttura del flusso di controllo del programma, la seconda legata allo sforzo cognitivo necessario per comprendere il codice.

4.6.2 Cyclomatic Complexity

La Cyclomatic Complexity, introdotta da McCabe nel 1976, è una metrica utilizzata per misurare la complessità logica di un programma, analizzando il numero di percorsi indipendenti presenti nel suo flusso di controllo.

Formalmente, la Cyclomatic Complexity $V(G)$è definita come:

$$V(G) = E - N + 2P$$ [55]

dove:

- E rappresenta il numero di archi (edges) nel grafo di controllo del programma
- N rappresenta il numero di nodi (nodes) nel grafo di controllo
- P rappresenta il numero di componenti connesse del grafo (generalmente pari a 1 per una singola funzione o modulo)

Questa formula deriva dalla teoria dei grafi e descrive la complessità del flusso di controllo in termini strutturali.

Nella pratica dell'analisi del codice, tuttavia, la complessità ciclomatica viene generalmente calcolata in modo più semplice. Quando si analizza una singola funzione o metodo, il grafo di controllo è normalmente costituito da una sola componente connessa ($P = 1$). In questo caso, la metrica può essere stimata contando i punti decisionali presenti nel codice, secondo la formula:

$$V(G) = D + 1$$

[55] McCabe, Thomas J. "A complexity measure." *IEEE Transactions on software Engineering* 4 (1976): 308-320.

dove *D* rappresenta il numero di decision points presenti nel codice. I decision points corrispondono tipicamente a strutture di controllo come:

- istruzioni if
- cicli for e while
- operatori logici condizionali
- costrutti switch o case

Ogni nuova struttura di controllo introduce un ulteriore percorso di esecuzione nel programma, incrementando quindi la complessità ciclomatica.

Valori più elevati di Cyclomatic Complexity indicano una maggiore complessità del flusso di controllo del codice e, di conseguenza, una maggiore difficoltà nella comprensione, manutenzione e testing del software.[56]

4.6.3 Cognitive Complexity

Accanto alla Cyclomatic Complexity è stata analizzata anche la Cognitive Complexity, una metrica introdotta da SonarSource con l'obiettivo di stimare la difficoltà cognitiva necessaria per comprendere un frammento di codice.

La Cognitive Complexity tiene conto di come le strutture logiche influenzano la comprensione del codice da parte dello sviluppatore. La metrica viene calcolata incrementando il punteggio di complessità ogni volta che nel codice compaiono strutture che richiedono uno sforzo cognitivo maggiore per essere interpretate. In particolare, la Cognitive Complexity aumenta nei seguenti casi:

---

[56] PMD. *Source Code Analyze* https://pmd.github.io/pmd/pmd_rules_java_design.html

- presenza di strutture di controllo (ad esempio if, while, for)
- nidificazioni di strutture di controllo
- combinazioni di condizioni logiche
- strutture di controllo annidate che aumentano la profondità del flusso logico

Una caratteristica fondamentale della Cognitive Complexity è che penalizza esplicitamente la nidificazione.

In questo modo, strutture di controllo annidate profondamente contribuiscono in misura maggiore alla complessità complessiva rispetto a strutture lineari.[57]

4.7 Calcolo della produttività

Per valutare congiuntamente la qualità delle soluzioni prodotte e il tempo necessario per svilupparle è stato definito un indice di produttività. L'obiettivo è combinare in un'unica misura due dimensioni dell'attività sperimentale: la correttezza del risultato ottenuto e l'efficienza temporale dello sviluppo.

Nel contesto di questo studio, la produttività è stata definita come il rapporto tra la qualità funzionale della soluzione e il tempo impiegato per realizzarla. Formalmente, l'indice di produttività Pè definito come:

$$P = \frac{punteggio}{tempo}$$

[57] PMD. *Source Code Analyze* https://pmd.github.io/pmd/pmd_rules_java_design.html

Dove:

- “punteggio” rappresenta la qualità funzionale della soluzione, misurata tramite lo score restituito dallo strumento *DeepEval*;
- “tempo” rappresenta il tempo necessario per completare il task, espresso in minuti.

Per confrontare le due condizioni sperimentali (con e senza supporto di AI), la produttività è stata stimata a livello di campione tramite una misura aggregata. In particolare, la produttività complessiva è stata calcolata come il rapporto tra lo score totale ottenuto dai partecipanti e il tempo totale impiegato dal gruppo:

$$P_{tot} = \frac{\sum_{i=1}^{n} punteggi\ o_i}{\sum_{i=1}^{n} temp\ o_i}$$

dove $punteggio_i$ e $tempo_i$ rappresentano rispettivamente lo score e il tempo associati al partecipante i.

Questa formulazione consente di stimare la quantità complessiva di risultato ottenuta per unità di tempo nelle due condizioni sperimentali. L’utilizzo del rapporto tra somme, anziché della media dei rapporti individuali, permette inoltre di ottenere una misura più stabile della produttività complessiva, riducendo l’influenza di valori estremi o di differenze marcate nei tempi individuali.

È importante sottolineare che questa metrica non rappresenta una definizione standardizzata di produttività nello sviluppo software. La produttività è infatti un concetto multidimensionale che può includere ulteriori fattori, come la manutenibilità del codice, lo sforzo cognitivo richiesto allo sviluppatore o la qualità del software nel lungo periodo. L’indice utilizzato in questo studio deve quindi essere interpretato come una misura sintetica definita specificamente per l’esperimento, utile a confrontare le due condizioni di sviluppo analizzate.

La formula adottata deve quindi essere interpretata come un indice sintetico costruito specificamente per questo esperimento, utile a combinare in modo

semplice e trasparente la qualità del risultato e l'efficienza temporale nel contesto del compito assegnato. In questo senso, essa non intende fornire una misura generale della produttività del programmatore, ma piuttosto una misura comparativa interna allo studio.

Poiché il punteggio di correttezza assume valori compresi tra 0 e 1, la metrica di produttività risulta principalmente influenzata dal tempo di sviluppo, pur mantenendo la capacità di penalizzare soluzioni parzialmente corrette o errate. Per questo motivo, l'indice di produttività viene utilizzato principalmente come misura descrittiva a supporto dell'analisi dei tempi di sviluppo e della qualità delle soluzioni, e non come unica metrica per valutare l'impatto dell'utilizzo dell'AI nel processo di sviluppo software.

4.8 Limiti

La presente analisi deve essere interpretata come uno studio preliminare volto a esplorare l'impatto degli strumenti di intelligenza artificiale generativa, nello specifico GitHub Copilot, nel processo di sviluppo software. Di conseguenza, alcuni limiti metodologici devono essere considerati nell'interpretazione dei risultati.

Il primo limite riguarda la dimensione del gruppo relativamente ridotta, avendo effettuato i test su un sottogruppo di 10 sviluppatori. L'analisi è stata condotta su un numero limitato di partecipanti che hanno completato entrambe le condizioni sperimentali. Campioni di piccola dimensione possono ridurre la potenza statistica delle analisi e limitare la generalizzabilità dei risultati.

Un secondo limite riguarda la natura del compito sperimentale. Questo è stato progettato per essere relativamente contenuto e controllabile, così da consentire una valutazione oggettiva delle soluzioni prodotte. Tuttavia, attività di

sviluppo software più complesse o progetti di dimensioni maggiori potrebbero produrre dinamiche differenti nell'utilizzo degli strumenti di AI.

Un ulteriore elemento da considerare è il rapido sviluppo delle tecnologie di intelligenza artificiale generativa. Durante il periodo in cui lo studio è stato condotto, gli strumenti di supporto alla programmazione hanno subito un'evoluzione significativa. È quindi possibile che strumenti più recenti o versioni successive degli stessi sistemi possano produrre risultati differenti rispetto a quelli osservati nel presente studio.

Infine, la produttività è stata misurata tramite un indice sintetico basato sul rapporto tra qualità della soluzione e tempo impiegato per completare il compito. Sebbene questa metrica permetta di integrare due dimensioni rilevanti del processo di sviluppo, essa rappresenta una semplificazione del concetto di produttività nello sviluppo software.

Nel complesso, queste limitazioni suggeriscono che i risultati ottenuti debbano essere interpretati come indicazioni preliminari, utili a comprendere alcune dinamiche dell'utilizzo dell'AI nella programmazione, ma che richiedono ulteriori studi con campioni più ampi e compiti più complessi per essere confermati.

CAPITOLO 5 – DISCUSSIONE DEI RISULTATI DELLA RICERCA

Il presente capitolo ha l'obiettivo di rispondere alle domande di ricerca formulate nel capitolo metodologico. In coerenza con il disegno mixed-method adottato, l'analisi integra dati quantitativi e qualitativi derivanti dal questionario e dal test sperimentale con considerazioni interpretative utili a comprendere l'impatto di GitHub Copilot sulla produttività e sulla qualità del risultato prodotto.

Dopo una breve descrizione del gruppo di programmatori coinvolto, vengono presentati i risultati relativi alla produttività, misurata attraverso i tempi di completamento del compito e la qualità del output prodotto, successivamente si passa all'analisi dei risultati relativi alla qualità del codice scritto e per concludere vengono presentati i risultati del questionario volto a misurare la percezione dei lavoratori sull'IA generativa e su GitHub Copilot.

5.1 Risultati analisi

La prima analisi, volta a misurare il tempo medio di completamento del compito e la correttezza del risultato, con e senza l'utilizzo di GitHub Copilot, evidenzia che senza il supporto dello strumento il tempo medio è pari a 53,7 minuti, mentre con il supporto di GitHub Copilot si riduce a 17,3 minuti (vedi grafico 1).

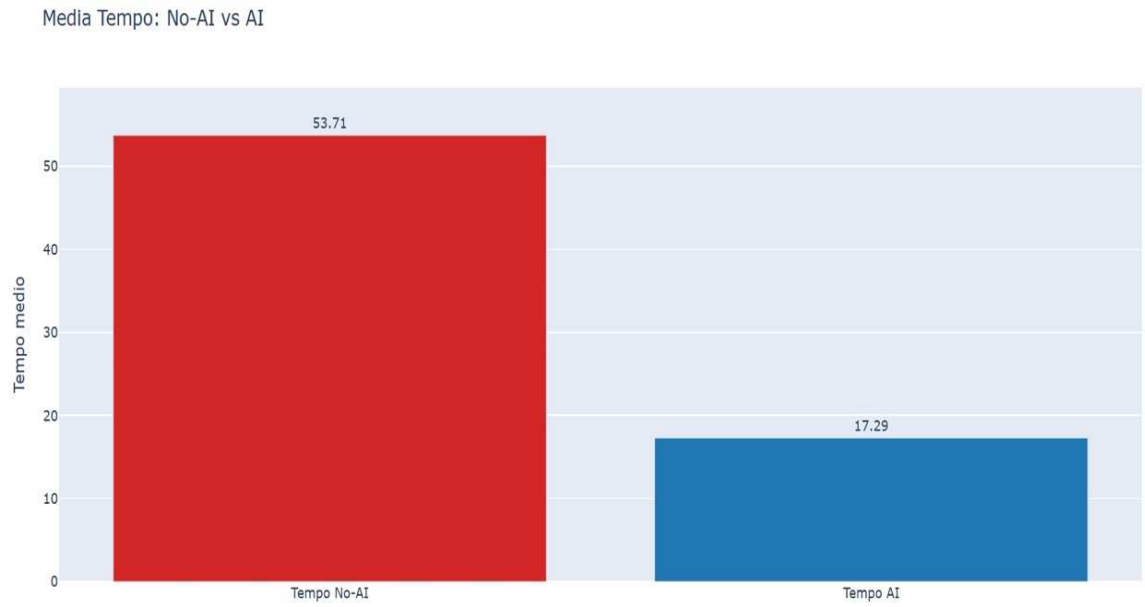

Figura 7 Media tempo di esecuzione del compito

Considerando il totale dei tempi osservati, si passa da 376 minuti complessivi nel setting senza IA a 121 minuti con IA, con una riduzione pari a circa il 67,8% del tempo impiegato nel completare il compito. Anche osservando i singoli casi, tutti i partecipanti hanno completato il compito in un tempo inferiore quando supportati da GitHub Copilot. Le riduzioni individuali variano da circa 46,2% a 83,3%, mostrando un miglioramento diffuso e non limitato a pochi soggetti isolati.

Questi risultati indicano che, nel contesto osservato, l'utilizzo di GitHub Copilot è associato a un incremento significativo della rapidità di esecuzione. L'evidenza empirica raccolta appare quindi coerente con quanto emerso nella letteratura discussa nel terzo capitolo, in cui GitHub Copilot veniva descritto come uno strumento potenzialmente in grado di accelerare il lavoro di sviluppo software. Al tempo stesso, il disegno adottato in questa ricerca, basato su un confronto diretto tra prestazioni con e senza supporto dello strumento, consente di ridurre almeno in

parte il rischio che l'aumento di produttività sia attribuibile esclusivamente a differenze pregresse tra i partecipanti.

Occorre precisare che una valutazione della produttività individuale basata esclusivamente sulla variabile tempo di completamento restituirebbe una lettura soltanto parziale del fenomeno. Per questo motivo, in questa ricerca, la produttività è stata considerata oltre che per tempo di completamento anche come la capacità di raggiungere un risultato corretto e coerente con l'obiettivo del compito. L'analisi dei tempi è stata quindi affiancata alla valutazione della qualità del risultato prodotto nei due scenari sperimentali, così da verificare se il vantaggio osservato sul piano dell'efficienza operativa si accompagnasse anche a un effettivo mantenimento, peggioramento o miglioramento, dell'accuratezza dell'output finale.

Considerando la media complessiva, il punteggio passa da 0,8086 nello scenario senza AI a 0,9345 nello scenario con GitHub Copilot (vedi grafico 2).

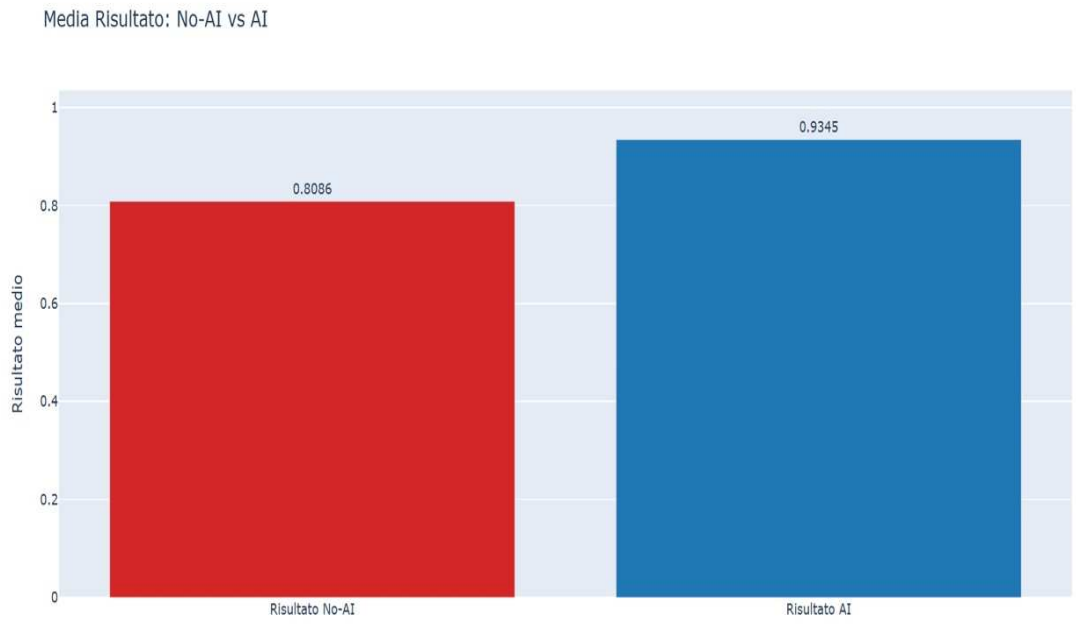

Figura 8: Media qualità del risultato prodotto

A una prima lettura, questo dato sembrerebbe indicare un miglioramento della qualità del risultato in presenza di GitHub Copilot. Tuttavia, l'interpretazione richiede cautela, poiché la media è fortemente influenzata da un caso particolarmente estremo, nel quale un partecipante passa da un punteggio pari a 0,02 senza IA a 0,9642 con IA.

Una lettura più analitica della distribuzione dei punteggi restituisce però un quadro più sfumato. In 4 casi su 7, il punteggio ottenuto con IA risulta leggermente inferiore rispetto a quello registrato senza IA, mentre in 3 casi su 7 si osserva un miglioramento. Anche il confronto tra i valori centrali conferma questa ambivalenza: la mediana passa da 0,94 senza AI a 0,9238 con AI, suggerendo una lieve riduzione della qualità tipica del risultato. In altri termini, se la media complessiva sembra premiare l'utilizzo di Copilot, la distribuzione dei dati mostra che tale vantaggio non si manifesta in maniera uniforme tra tutti i partecipanti.

Il risultato emerso suggerisce quindi che l'utilizzo di GitHub Copilot non determina in modo automatico un miglioramento sistematico della qualità del risultato finale. Piuttosto, lo strumento appare capace di produrre effetti eterogenei: in alcuni casi consente di avvicinarsi maggiormente al gold standard, mentre in altri genera output leggermente meno accurati rispetto a quelli ottenuti senza supporto dell'intelligenza artificiale. Questo sembra evidenziare che la qualità dell'output dipenda dall'utilizzo che lo sviluppatore fa dello strumento, piuttosto che dallo strumento utilizzato.

Analizzato il tempo impiegato e la qualità del risultato raggiunto è possibile calcolare una misura sintetica della produttività individuale (P) definita come rapporto tra la qualità del risultato ottenuto (Q) e il tempo impiegato per completare il compito (T), secondo la formula $P = Q/T$. Questa formula consente di analizzare la produttività come capacità di produrre un risultato corretto in un determinato intervallo di tempo.

Applicando la formula ai valori medi osservati nei due scenari sperimentali, la produttività passa da 0,0151 nello scenario senza AI a 0,0541 nello scenario con IA, con un incremento pari a circa il 259,1%. Questo suggerisce che nel campione

analizzato, l'utilizzo di GitHub Copilot non si associa soltanto a una riduzione del tempo necessario per completare il compito, ma anche a un aumento complessivo della quantità di risultato corretto prodotto per unità di tempo.

È tuttavia opportuno precisare che questo indicatore risente della presenza di un caso particolarmente estremo nel gruppo senza IA. In uno dei 7 casi validi, infatti, il punteggio di qualità registrato nello scenario privo di supporto dell'intelligenza artificiale è pari a 0,02 valore sensibilmente inferiore rispetto agli altri osservati e tale da influenzare in misura rilevante la media complessiva del gruppo. Poiché il dato risulta corretto e metodologicamente valido, esso è stato mantenuto nell'analisi principale. Per rendere più robusta l'interpretazione del risultato, è stata però considerata anche una lettura di sensibilità, escludendo tale osservazione estrema. In questo secondo scenario, la produttività passa da 0,0182 senza AI a 0,0627 con l'utilizzo di GitHub Copilot, portando un incremento pari a circa il 244,4%.

Nel complesso, l'analisi conferma che il caso estremo incide sull'ampiezza dell'effetto osservato, ma non modifica la direzione generale del risultato. Anche adottando una lettura più prudente, GitHub Copilot risulta associato a un incremento marcato della produttività individuale. In relazione alla RQ1, l'analisi consente quindi di affermare che l'utilizzo di GitHub Copilot ha un effetto positivo sulla produttività nel campione osservato, soprattutto in termini di riduzione di tempo. Inoltre, in riferimento alla RQ2 l'analisi mostra che la qualità del risultato non si distribuisce in modo uniforme tra i partecipanti. Se, da una parte la media dei punteggi risulta più elevata nella condizione con IA, dall'altro la distribuzione dei dati evidenzia un andamento più eterogeneo, dimostrando come la qualità del risultato dipenda in larga parte dall'utilizzo dello strumento da parte del programmatore.

La seconda analisi, condotta con l'obbiettivo di rispondere alla domanda di ricerca RQ2, si è basata su una valutazione automatizzata della complessità del codice generato nei due scenari sperimentali, distinguendo tra condizione con supporto di GitHub Copilot e condizione senza supporto dell'intelligenza

artificiale. Questa analisi consente di osservare la qualità del codice in modo distinto rispetto alla qualità del risultato finale, già esaminata precedentemente, concentrandosi invece su aspetti strutturali del codice. Il confronto tra le due condizioni viene riportato attraverso le medie aggregate delle metriche di complessità.

Considerando lo scope ALL, che include l'intero insieme dei simboli presenti nei branch analizzati, la media della Cyclomatic Complexity risulta pari a 1,545 nella condizione IA e a 1,374 nella condizione NO-IA. Analogamente, la media della Cognitive Complexity risulta pari a 0,797 nella condizione IA e a 0,505 nella condizione NO-IA. In entrambi i casi, dunque, i valori medi osservati risultano leggermente più elevati nella condizione supportata da GitHub Copilot (Vedi figura 3).

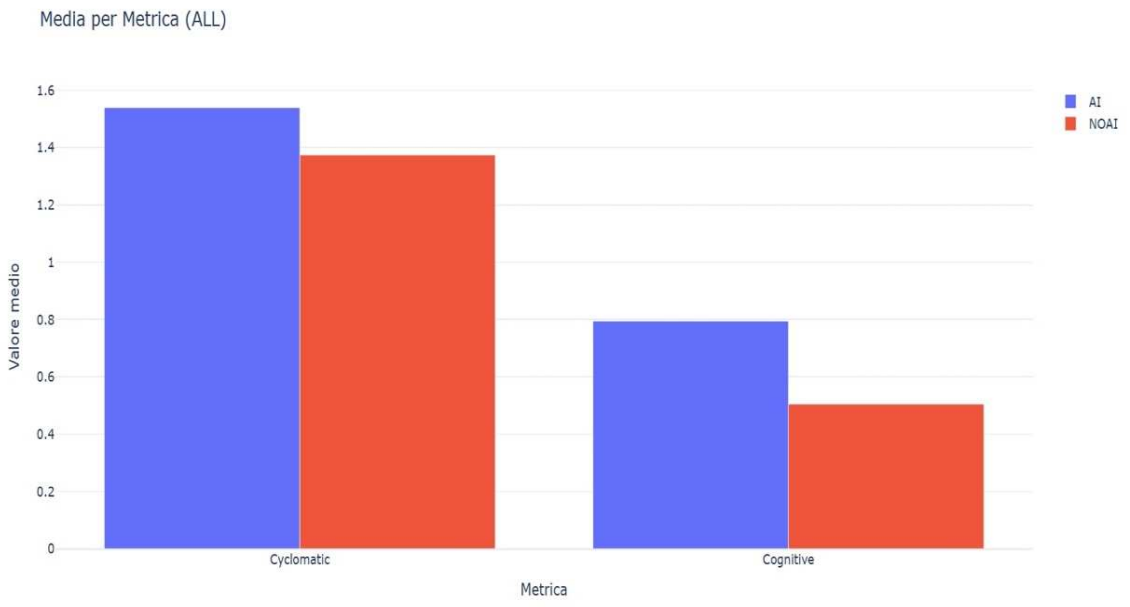

Figura 9: Media qualità codice prodotto

Tuttavia, tali differenze devono essere interpretate con cautela. Considerate alla luce delle soglie comunemente adottate dagli strumenti di analisi statica,

entrambe le condizioni si collocano infatti su livelli di complessità molto bassi: per la Cyclomatic Complexity, PMD considera generalmente bassi i valori compresi tra 1 e 4 e segnala di default i metodi con complessità pari o superiore a 10; analogamente, anche per la Cognitive Complexity le soglie di attenzione risultano sensibilmente più elevate rispetto ai valori medi osservati nel presente studio. Ne consegue che lo scarto tra le due condizioni, pur esistente, resta contenuto sul piano sostanziale e non può essere letto come indicatore di una marcata differenza nella qualità strutturale del codice prodotto.

È inoltre opportuno evidenziare che i bassi valori registrati in entrambe le condizioni sono coerenti con la natura del test sperimentale, che richiedeva la scrittura di una quantità relativamente limitata di codice. Di conseguenza, i livelli medi di complessità osservati riflettono le caratteristiche del codice generato e la ridotta estensione e articolazione del compito assegnato. In questo senso, il confronto tra IA e NO-IA assume maggiore rilevanza sul piano relativo che su quello assoluto e può essere interpretato come una prima analisi esplorativa o preliminare dell'impatto di GitHub Copilot sulla qualità strutturale del codice, più che come una valutazione definitiva e generalizzabile.

Con riferimento alla RQ2, l'evidenza empirica consente quindi di affermare che GitHub Copilot non appare associato a una riduzione sistematica della complessità del codice prodotto, pur senza evidenziare un peggioramento strutturale marcato. Questo risultato mostra che GitHub Copilot è quindi in grado di accrescere l'efficienza operativa senza mostrare un marcato miglioramento o peggioramento della qualità strutturale del codice.

La terza analisi, finalizzata a rispondere alla RQ3 e a capire quale sia il livello di soddisfazione degli sviluppatori nei confronti degli strumenti di intelligenza artificiale generativa, è stata condotta su un sottogruppo di 33 rispondenti, selezionati tra i 45 rispondenti totali che hanno dichiarato di utilizzare GitHub Copilot, in quanto tale gruppo consente una valutazione più diretta e pertinente del livello di soddisfazione nei confronti dello strumento. La soddisfazione è stata osservata attraverso tre dimensioni principali: il comfort

nell'utilizzo, il guadagno di produttività percepito e il giudizio sulla qualità del codice prodotto con il supporto di GitHub Copilot.

Per quanto riguarda il comfort d'uso, i risultati mostrano un livello complessivamente elevato di familiarità operativa con Copilot: 17 rispondenti su 33 (51,52%) dichiarano di sentirsi molto a proprio agio nell'utilizzo di strumenti di assistenza alla codifica basati su IA, mentre 11 (33,33%) si definiscono completamente a proprio agio; soltanto 5 rispondenti (15,15%) riportano invece un livello di comfort più contenuto, mentre nessuno riferisce di sentirsi per niente a proprio agio. La media dell'item è pari a 3,18 su 4, mentre la mediana è pari a 3, indicando che sia il giudizio medio sia il valore centrale della distribuzione si collocano su una valutazione positiva (Vedi figura 5)

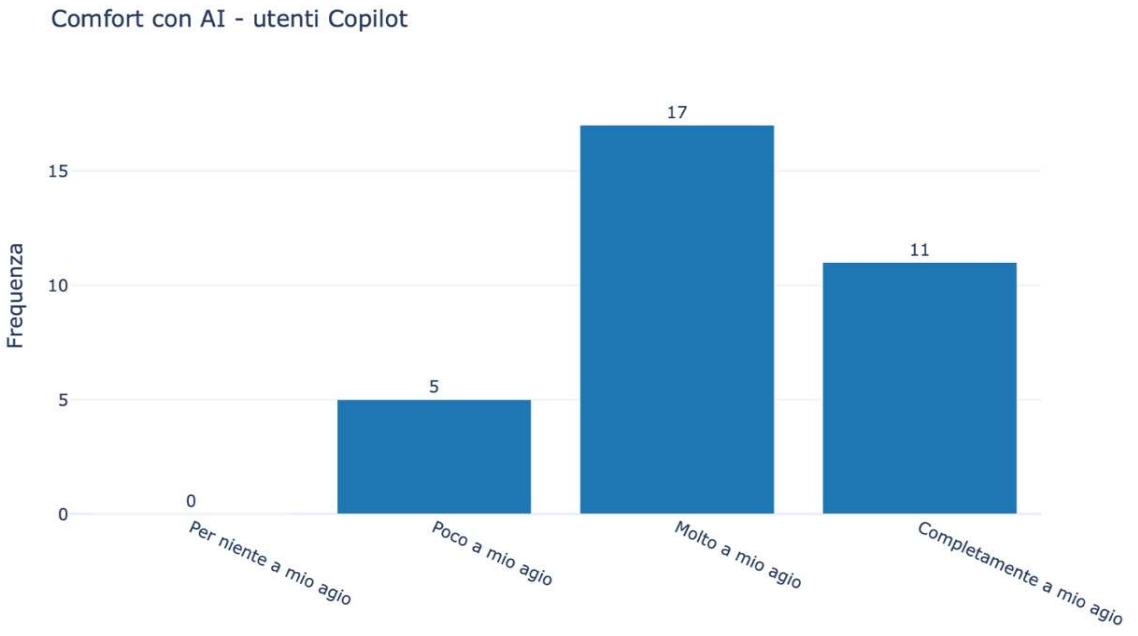

Figura 10: Risultati Comfort con IA

Questo dato suggerisce che, tra gli utenti effettivi di Copilot, l'interazione con lo strumento non venga generalmente percepita come complessa o problematica sul piano operativo, ma piuttosto come accessibile e gestibile nel lavoro quotidiano.

Una seconda dimensione rilevante riguarda il guadagno di produttività percepito, che rappresenta l'aspetto più marcatamente positivo. In questo caso, 12 rispondenti su 33 (36,36%) valutano l'incremento di produttività come buono e 10 (30,30%) lo giudicano eccellente; le valutazioni più moderate restano comunque presenti, con 8 rispondenti (24,24%) che esprimono un giudizio discreto e 3 (9,09%) che lo considerano scarso. La media dell'item è pari a 3,88 su 5, con mediana pari a 4, valore che segnala una percezione complessivamente favorevole e orientata verso la fascia alta della scala.

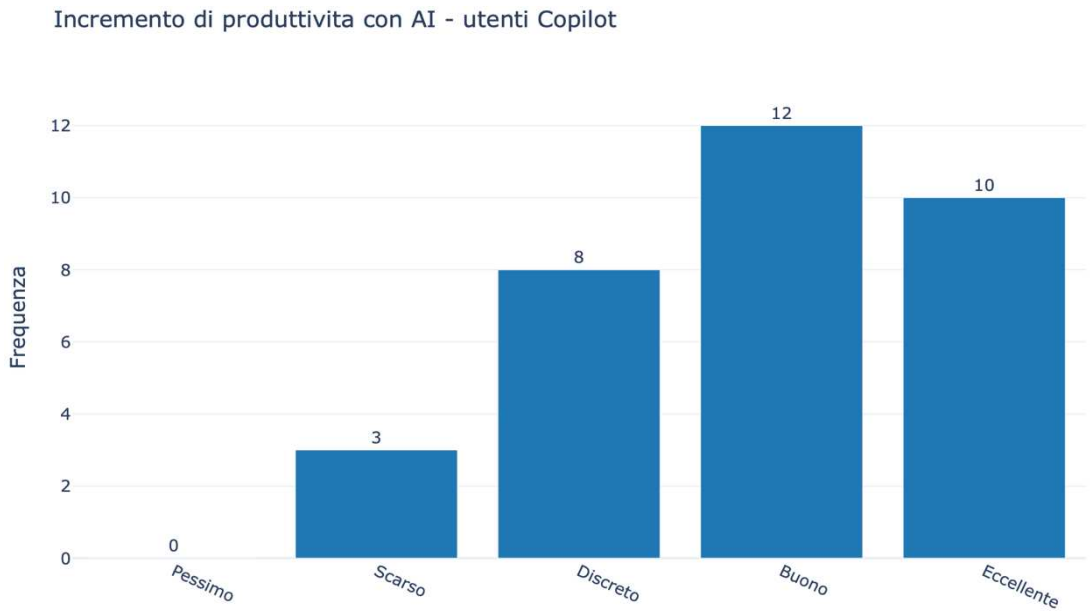

Figura 11: incremento produttività

Tali risultati evidenziano che uno dei principali fattori alla base della soddisfazione verso Copilot risiede nella percezione di un beneficio concreto in termini di velocizzazione del lavoro e supporto nello svolgimento delle attività di sviluppo software. In altri termini, gli utenti attribuiscono allo strumento soprattutto la capacità di aumentare l'efficienza operativa e di agevolare l'esecuzione di compiti ricorrenti.

Più prudente risulta invece il giudizio relativo alla qualità del codice prodotto con il supporto dell'IA. In questo caso, 13 rispondenti su 33 (39,39%)

valutano la qualità del codice come buona e altri 13 (39,39%) come discreta; le valutazioni eccellenti riguardano 4 rispondenti (12,12%), mentre 3 (9,09%) esprimono come giudizio scarso. La media dell'item è pari a 3,55 su 5, con mediana pari a 4.

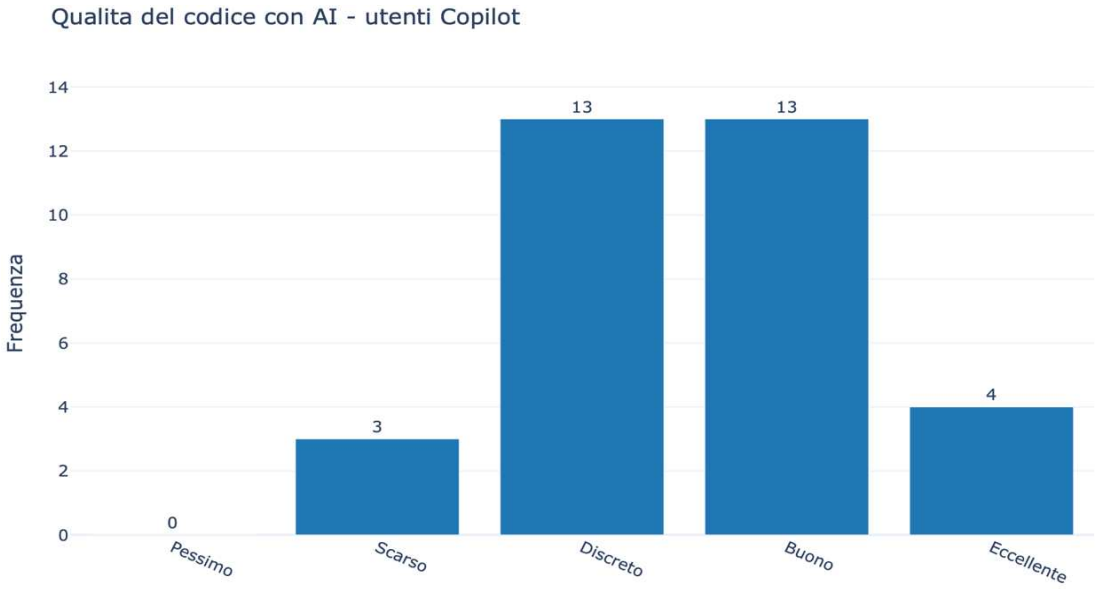

Figura 12: Qualità del codice

Anche in questo caso il giudizio si mantiene generalmente positivo, ma con un grado di cautela maggiore rispetto a quanto rilevato per la produttività. La distribuzione delle risposte suggerisce infatti che gli utenti riconoscano a Copilot un contributo utile sul piano qualitativo, senza tuttavia considerarlo pienamente autonomo o sempre affidabile nella generazione di codice di qualità elevata.

Nel complesso, i risultati delineano quindi un profilo di soddisfazione complessivamente positivo nei confronti di GitHub Copilot. Il comfort d'uso risulta elevato, il beneficio produttivo percepito rappresenta la dimensione più marcatamente favorevole e la qualità del codice viene valutata in termini positivi, sebbene con maggiore moderazione. Con riferimento alla RQ3, i dati raccolti consentono pertanto di affermare che gli sviluppatori che utilizzano GitHub Copilot manifestano un livello di soddisfazione generalmente medio-alto,

fondato soprattutto sulla sua utilità pratica e sul contributo offerto all'efficienza operativa. Al tempo stesso, la valutazione più prudente relativa alla qualità del codice suggerisce che Copilot venga apprezzato prevalentemente come strumento di supporto, più che come sostituto autonomo del giudizio tecnico e della supervisione umana.

Un altro aspetto emerso dal questionario è il tema della sicurezza e della privacy. Il 21% dei rispondenti ha infatti affermato di essere cauto nell'utilizzo dell'IA per questi motivi. Il dato mostra quindi come la sicurezza e la privacy rimangano centrali nel lavoro dello sviluppatore, portando a maggiore prudenza nell'utilizzo di questi strumenti. Questo conferma la teoria secondo il quale lo sviluppatore deve avere il controllo dello strumento e dell'output da questo generato, e debba prestare attenzione alle informazioni condivise con tale strumento.

5.2 Implicazioni organizzative e riflessi sul mercato del lavoro

Nel primo capitolo di questa tesi è stata proposta un'analisi delle rivoluzioni industriali e dei conseguenti cambiamenti nei processi lavorativi e organizzativi. In particolare, è stato analizzato come queste rivoluzioni tecnologiche abbiano comportato l'assorbimento, la semplificazione o la ricomposizione di attività precedentemente svolte dall'uomo, modificando il ruolo dei lavoratori all'interno del processo produttivo. In questa prospettiva, Alain Touraine evidenzia come queste trasformazioni abbiano portato da una parte a una diminuzione dei lavoratori non specializzati coinvolti direttamente nella produzione e dall'altra ad un aumento dei lavoratori specializzati nel controllo e gestione delle nuove

macchine.[58] In modo analogo, Kern e Schuman sottolineano come le nuove tecnologie abbiano contribuito a ribaltare le tradizionali modalità di organizzazione della produzione e del lavoro stesso, rendendo necessaria la figura di lavoratori qualificati chiamati a svolgere funzioni di supervisione, regolazione e controllo del sistema tecnico.[59] Anche Braverman, pur in una prospettiva diversa, interpreta il cambiamento tecnologico come una ricomposizione del lavoro e della sua organizzazione, caratterizzata dalla scomparsa di alcune mansioni e dalla nascita di nuove attività.[60]

In quest'ottica, l'analisi empirica svolta non consente di rispondere direttamente alla RQ4 relativa alle implicazioni dell'IA generativa sul mercato del lavoro, anche se i risultati ottenuti permettono di formulare alcune considerazioni interpretative alla luce della letteratura esistente. L'analisi fatta ha mostrato come l'utilizzo dell'IA generativa e in particolare GitHub Copilot sia associato a un forte incremento di produttività, suggerendo che questa possa contribuire a trasformare i processi lavorativi, incidendo sia sull'efficienza individuale che sule modalità di organizzazione del lavoro. L'ipotesi che l'impiego dell'IA sia associato a un forte incremento di produttività, del resto, costituisce l'assunzione alla base degli enormi investimenti pianificati dalle imprese in tutto il mondo. Solo in Italia, per esempio, in base ai dati elaborati dall'Osservatorio Digitale del Politecnico di Milano, nel 2025 il mercato dell'Intelligenza Artificiale ha raggiunto 1,8 miliardi di euro, con una crescita del 50% rispetto all'anno precedente; con l'84% delle grandi imprese che ha acquistato licenze di IA generativa, il 47% dei lavoratori che dichiara di utilizzare strumenti di IA in azienda e con gli annunci di lavoro che richiedono competenze di IA aumentati del 93% in un anno. [61]

---

[58] Buccarelli, Filippo. "L'evoluzione del lavoro alla Renault." *Cambio: rivista sulle trasformazioni sociali: 11, 1, 2016* (2016): 245-249.

[59] Kern, Horst, and Michael Schumann. *Das Ende der Arbeitsteilung?: Rationalisierung in der industriellen Produktion; Bestandsaufnahme, Trendbestimmung*. Beck, 1985.

[60] Braverman, Harry. *Labor and monopoly capital: The degradation of work in the twentieth century*. nyu Press, 1998.

[61] https://www.osservatori.net/comunicato/artificial-intelligence/intelligenza-artificiale-italia/

Il cambiamento appare particolarmente evidente nel campo della programmazione. Per lungo tempo il lavoro dello sviluppatore si è fondato su un'impostazione sostanzialmente deterministica: analizzare un problema, tradurlo in istruzioni formali e scrivere il codice che la macchina esegue. In questo schema, il valore professionale del programmatore è stato storicamente associato, in larga misura, alla capacità di tradurre una logica in codice. L'avanzamento recente dell'intelligenza artificiale generativa, e ancor più l'emergere dell'IA generativa agentica, sembra però iniziare a modificare questo assetto. L'IA generativa si sta progressivamente trasformando da semplice strumento di supporto nelle mani del programmatore a sistema di lavoro collaborativo capace di intervenire in più fasi del processo, contribuendo alla letteratura della documentazione, alla produzione di codice, alla verifica del risultato e all'analisi di criticità o alternative.

Questo passaggio tende a suggerisce uno spostamento del baricentro del lavoro del programmatore. Più che concentrarsi esclusivamente sulla scrittura puntuale del codice, lo sviluppatore si concentra sempre più sul definire il problema, fornire contesto, esplicitare vincoli, valutare il risultato e gestire il flusso complessivo di lavoro. In altri termini, il contributo umano si sposta progressivamente dalla produzione diretta di ogni singola istruzione alla definizione degli obiettivi, alla scomposizione delle attività, alla supervisione del processo e alla validazione degli output.

Da ciò derivano implicazioni organizzative rilevanti. Se una quota crescente del lavoro tecnico può essere accelerata da, delegata a o integrata con sistemi di IA generativa, il valore professionale tenderà a concentrarsi sempre meno sulla sola esecuzione manuale dei compiti e sempre più sulla capacità di impostare il problema, governare il processo e validarne criticamente l'esito. Le conseguenze di questo cambiamento riguardano l'intera struttura organizzativa. Se il lavoro tecnico viene progressivamente ricomposto attorno all'interazione tra esseri umani e sistemi intelligenti, anche le organizzazioni tenderanno a valorizzare sempre di

più figure capaci di coordinare strumenti, flussi di lavoro e processi di validazione, piuttosto che semplicemente aumentare il numero di risorse dedicate all'esecuzione operativa.

CONCLUSIONI

Il presente lavoro di tesi ha avuto l'obiettivo di analizzare l'impatto degli strumenti di intelligenza artificiale generativa nei processi di sviluppo software, con particolare attenzione al caso di GitHub Copilot e ai suoi effetti sulla produttività individuale, sulla qualità del codice e sulle implicazioni nei processi lavorativi e organizzativi. L'interesse per questo tema nasce dalla crescente diffusione di tali strumenti nei contesti organizzativi e dal ruolo sempre più centrale che essi stanno assumendo nel ridefinire le modalità di lavoro cognitivo e digitale.

I risultati emersi consentono innanzitutto di affermare che l'utilizzo di GitHub Copilot è associato a un incremento significativo della produttività individuale, soprattutto in termini di riduzione del tempo necessario per completare il compito assegnato. Anche considerando la produttività in forma più articolata, cioè come rapporto tra correttezza del risultato e tempo impiegato, l'effetto dello strumento si mantiene chiaramente positivo. In questo senso, la ricerca conferma che Copilot può rappresentare un supporto concreto all'efficienza operativa nello sviluppo software, in linea con quanto suggerito dalla letteratura più recente sul tema.

Più sfumato appare invece il quadro relativo alla qualità. Per quanto riguarda la qualità del risultato prodotto, l'analisi non mostra un miglioramento uniforme in tutti i casi osservati. Sebbene la media complessiva dei punteggi risulti più elevata nella condizione con Copilot, la distribuzione dei dati evidenzia esiti

eterogenei, che non consentono di sostenere l'esistenza di un miglioramento sistematico e generalizzato. Analogamente, anche l'analisi della qualità del codice, osservata attraverso metriche di complessità strutturale, non evidenzia una riduzione netta della complessità nella condizione supportata da Copilot. Le differenze tra le due condizioni risultano infatti contenute e si collocano in un quadro generale di bassa complessità, coerente con la natura circoscritta del compito sperimentale. Nel complesso, la ricerca suggerisce quindi che l'intelligenza artificiale generativa possa aumentare l'efficienza del lavoro senza tradursi automaticamente in un miglioramento qualitativo del risultato o del codice prodotto.

Sul piano percettivo, il questionario mostra che gli sviluppatori che utilizzano GitHub Copilot esprimono un livello di soddisfazione complessivamente medio-alto. In particolare, il comfort d'uso risulta elevato e il beneficio più chiaramente riconosciuto riguarda la produttività percepita, mentre pi prudente appare la valutazione relativa alla qualità del codice generato. Questi elementi sono particolarmente significativi, poiché confermano che GitHub Copilot viene apprezzato come strumento di supporto operativo, utile per velocizzare il lavoro e agevolare lo svolgimento di alcune attività. Occorre sottolineare che la soddisfazione espressa dagli sviluppatori non coincide con un affidamento incondizionato allo strumento, ma si accompagna alla consapevolezza della necessità di controllo, revisione e validazione.

Per quanto riguarda le implicazioni organizzative e i riflessi sul mercato del lavoro, la ricerca non consente di formulare conclusioni definitive, ma suggerisce alcune tendenze plausibili. L'aumento di produttività osservato e la crescente diffusione di strumenti di IA generativa nelle imprese indicano infatti la possibilità di una progressiva ridefinizione delle modalità di lavoro e della struttura delle competenze richieste. Più che una scomparsa immediata di determinate figure professionali, sembra emergere una crescente centralità di competenze legate all'orchestrazione dei processi, alla validazione degli output, alla definizione degli obiettivi e alla supervisione di sistemi intelligenti. In questo senso, l'impatto

dell'IA generativa appare destinato a manifestarsi soprattutto come trasformazione qualitativa del lavoro, più che come semplice riduzione quantitativa dell'occupazione.

Durante lo sviluppo di questa tesi sperimentale l'intelligenza artificiale generativa ha continuato la sua evoluzione in modo rapido e costante, portando alla nascita dell'intelligenza artificiale generativa agentica. Questa rappresenta un forte cambiamento nel campo dell'ingegneria del software. I nuovi sistemi mostrano infatti capacità di interpretazione avanzata, pianificazione e decomposizione di compiti, utilizzo di strumenti e risorse, esecuzione e iterazione di codice, ragionamento e risoluzione di problemi, mantenimento del contesto a lungo termine, autoriflessione e autocorrezione.[62] Questa svolta amplia ancora più approfonditamente la capacità dell'IA generativa di intervenire nei processi lavorativi, configurandosi come una potenziale infrastruttura di lavoro capace di prendere in carico intere attività, lasciando all'essere umano il ruolo di definizione degli obbiettivi, supervisione del processo e validazione finale. In quest'ottica, ricerche future potrebbero indagare in che misura il passaggio da una logica di semplice assistenza a una logica di orchestrazione di agenti trasformi i processi lavorativi.

Un altro tema di rilevante importanza è il ruolo dei token come nuova variabile economica e organizzativa del lavoro svolto da sistemi di IA generativa. Se nel paradigma tradizionale il vincolo della produttività è rappresentato soprattutto dal tempo del lavoratore, in un contesto caratterizzato da agenti specializzati e attività eseguite simultaneamente, potrebbe assumere crescente rilevanza la capacità di convertire il consumo di token in valore economico. In contesti ad alta intensità digitale, il problema organizzativo principale potrebbe non essere piu soltanto come distribuire il tempo delle persone, ma come allocare

---

[62] Sapkota, Ranjan, Konstantinos I. Roumeliotis, and Manoj Karkee. "Vibe coding vs. agentic coding: Fundamentals and practical implications of agentic ai." *arXiv preprint arXiv:2505.19443* (2025)

il lavoro in modo efficiente ai vari modelli, limitare gli sprechi e valutare se la spesa in token generi effettivamente un ritorno in termini di produttività, qualità e valore per l'organizzazione.

ABSTRACT

Questa tesi analizza l'impatto degli strumenti di intelligenza artificiale generativa nei processi di sviluppo software, con particolare attenzione al caso di GitHub Copilot, al fine di comprendere in che modo l'utilizzo stia trasformando l'attività lavorativa, le competenze richieste agli sviluppatori e, più in generale, le modalità di organizzazione del lavoro.

Dal punto di vista empirico, la ricerca adotta un disegno mixed-method, combinando la somministrazione di un questionario a dipendenti dell'azienda Spindox dotati di licenza GitHub Copilot e su un test sperimentale di programmazione condotto su un sottogruppo di sviluppatori senior appartenenti alla stessa azienda. Le domande di ricerca riguardano l'effetto di GitHub Copilot sulla produttività individuale, la sua influenza sulla qualità del risultato e del codice prodotto, il livello di soddisfazione dei programmatori e le possibili implicazioni sul mercato del lavoro.

I risultati mostrano che l'utilizzo di GitHub Copilot è associato a un incremento significativo della produttività individuale. Nel test sperimentale, infatti, la produttività, misurata come rapporto tra qualità funzionale della soluzione e tempo impiegato per completare il compito mostra un incremento del 259,1 % con l'utilizzo di GitHub Copilot. La qualità del codice prodotto, misurata attraverso la Cyclomatic Complexity e la Cognitive Complexity non mostra invece particolari differenze tra il codice prodotto senza e con l'utilizzo di IA generativa. Sul piano percettivo, i dati del questionario mostrano inoltre un livello di soddisfazione complessivamente medio-alto tra gli sviluppatori che utilizzano GitHub Copilot: il comfort d'uso risulta elevato, il beneficio più chiaramente riconosciuto riguarda la produttività percepita, mentre la valutazione della qualità del codice generato appare positiva ma più prudente.

Nel complesso l'analisi suggerisce che nell'ambito dello sviluppo software il valore del contributo umano tenda a concentrarsi sempre più su attività di definizione del problema, supervisione del processo, validazione degli output e controllo della qualità, ridefinendo funzioni, competenze e assetti organizzativi

Bibliografia

Autor, David H. "Why are there still so many jobs? The history and future of workplace automation." *Journal of economic perspective*

Banh, Leonardo, and Gero Strobel. "Generative artificial intelligence." *Electronic Markets* 33.1 (2023): 63.

Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. No. 4. New York: Springer, 2006.

Bommasani, Rishi. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).

Bonazzi, Giuseppe. *Storia del pensiero organizzativo*. Vol. 136. FrancoAngeli, 2008.

Bonazzi, Giuseppe. "Taylorismo." *Enciclopedia delle Scienze Sociali*, vol. VI. Treccani, 1998.

Boonstra, Lee. "Prompt engineering." *Google, https://www. kaggle. com/whitepaper-prompt-engineering* (2025).

Braverman, Harry. *Labor and monopoly capital: The degradation of work in the twentieth century*. NYU Press, 1998.

Buccarelli, Filippo. "L'evoluzione del lavoro alla Renault." *Cambio: rivista sulle trasformazioni sociali* 11, 1 (2016): 245-249.

Cui, Kevin Zheyuan, et al. "The Productivity Effects of Generative AI: Evidence from a Field Experiment with GitHub Copilot." (2024).

Della Rocca, Giuseppe, and Vincenzo Fortunato. *Lavoro e organizzazione: dalla fabbrica alla società postmoderna*. GLF editori Laterza, 2006.

Feuerriegel, Stefan, et al. "Generative AI." *Business & Information Systems Engineering* 66.1 (2024): 111-126.

Golda, Abenezer, et al. "Privacy and security concerns in generative AI: a comprehensive survey." *IEEE Access* 12 (2024): 48126-48144.

Hays, Hasi. *Encyclopedia of Large Language Models and Foundation Models.*
Holmström, Jonny, and Noel Carroll. "How organizations can innovate with generative AI." *Business Horizons* (2024).

Hudson, Pat, Giovanni Arganese, and Carlo Bardini. *La rivoluzione industriale*. Il Mulino, 1995.

Islam, Mohaiminul, Guorong Chen, and Shangzhu Jin. "An overview of neural network." *American Journal of Neural Networks and Applications* 5.1 (2019): 7-11.

Jacoby, Sanford M. *Employing bureaucracy: Managers, unions, and the transformation of work in the 20th century*. Psychology Press, 2004.

Janiesch, Christian, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning." *Electronic Markets* 31.3 (2021): 687.

Jiang, Yuchen, et al. "Quo vadis artificial intelligence?" *Discover Artificial Intelligence* 2.1 (2022).

Kern, Horst, and Michael Schumann. *Das Ende der Arbeitsteilung?: Rationalisierung in der industriellen Produktion; Bestandsaufnahme, Trendbestimmung*. Beck, 1985.

Koutnik, Jan, et al. "A clockwork RNN." In *International Conference on Machine Learning*. PMLR, 2014.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *Nature* 521.7553 (2015): 436.

Leijonhufvud, Axel. *Capitalism and the factory system*. No. 184. Diskussionsbeiträge-Serie A, 1984.

Lemeš, Samir. "Prompt Engineering." *Artificial intelligence in industry* 4 (2024): 159-170.

Marvin, Ggaliwango, et al. "Prompt engineering in large language models." In *International conference on data intelligence and cognitive informatics*. Singapore: Springer Nature Singapore, 2023.

Marx, Karl. "Capital: Critique of Political Economy, Volume 1, cap. 13." In *Capital*. Princeton University Press, 2024.

McCarthy, John. *Dartmouth Summer Research Project on Artificial Intelligence*. Dartmouth College, 1956.

McCabe, Thomas J. "A complexity measure." *IEEE Transactions on software Engineering* 4 (1976): 308-320.

Mienye, Ibomoiye Domor, and Theo G. Swart. "A comprehensive review of deep learning: Architectures, recent advances, and applications." *Information* 15.12 (2024): 755.

Mitchell, Tom M. "Artificial neural networks." *Machine Learning* 45.81 (1997).
Peng, Sida, et al. "The impact of AI on developer productivity: Evidence from GitHub Copilot." *arXiv preprint arXiv:2302.06590* (2023).

Pollard, Sidney. "Factory discipline in the industrial revolution." *Economic History Review* (1963): 254-271.

Pollock, Friedrich. "Automation: Materialien zur Beurteilung der oekonomischen u. sozialen Folgen." *Frankfurter Beiträge zur Soziologie* (1970).

Skitka, L. J., K. L. Mosier, and M. Burdick. "Does automation bias decision-making?" *International Journal of Human-Computer Studies* 51 (1999): 991-1006.

Smith, Adam. *An Inquiry into the Nature and Causes of the Wealth of Nations*, volume 1. Vol. 1. Oxford: Clarendon Press, 1869.

Snow, Charles C., Raymond E. Miles, and Henry J. Coleman Jr. "Managing 21st century network organizations."

Taylor, Frederick Winslow. *The principles of scientific management*. NuVision Publications, LLC, 1911.

Turing, Alan M. "Computing machinery and intelligence (1950)." *Mind* 59.236 (2021): 33-60.

Ure, Andrew. *Philosophy of manufactures*. Routledge, 2013.
Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems* 30 (2017).

Weber, Thomas, et al. "Significant productivity gains through programming with large language models." *Proceedings of the ACM on Human-Computer Interaction* 8.EICS (2024): 1-29.

Yamashita, Rikiya, et al. "Convolutional neural networks: an overview and application in radiology." *Insights into Imaging* 9.4 (2018): 611-629.

Yang, J., H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu. *Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond.*

Zhang, Beiqi, et al. "Practices and challenges of using GitHub Copilot: An empirical study." *arXiv preprint arXiv:2303.08733* (2023).

Zuboff, Shoshana. *In the age of the smart machine: The future of work and power*. Basic Books, Inc., 1988.

Sitografia

Industrial Internet Consortium - What is the Industrial Internet? 2015 http://www.industrialinternetconsortium.org/about-industrial-internet.htm

Che cos'è l'intelligenza artificiale? | Tematiche | Parlamento europeo https://www.europarl.europa.eu/topics/it/article/20200827STO85804/che-cos-e-l-intelligenza-artificiale-e-come-viene-usata

Generative AI | OECD https://www.europarl.europa.eu/topics/it/article/20200827STO85804/che-cos-e-l-intelligenza-artificiale-e-come-viene-usata

"Che cos'è la modalità agente di GitHub Copilot?" Microsoft Learn. https://learn.microsoft.com/it-it/training/modules/github-copilot-agent-mode/2-what-is-agent-mode

PMD. Source Code Analyze https://pmd.github.io/pmd/pmd_rules_java_design.html

DeepEval documentation https://deepeval.com/docs/gettingstarted?utm_source=chatgpt.com

Ringraziamenti

Grazie alla mia famiglia per avermi sempre sostenuto e per avermi permesso di portare a termine questo percorso.

Grazie al mio relatore Paolo Costa e al mio correlatore Massimiliano Vaira per l'attenzione, la disponibilità e i preziosi consigli che hanno accompagnato la realizzazione di questo progetto.

Infine, grazie a Domenico Cambrea, Alberto Masala e tutti i dipendenti di Spindox che hanno contribuito a portare a termine la ricerca.