UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Sicurezza Informatica

VALUTAZIONI DI ASSURANCE DI MODELLI DI AI

Relatore:
Marco ANISETTI
Correlatore:
Nicola BENA

Tesi di Laurea di:
Michele MASTROBERTI
Matricola: 45805A

Anno Accademico 2024/2025

Ringraziamenti

Desidero esprimere la mia sincera gratitudine a tutte le persone che hanno reso possibile la realizzazione di questa tesi. Ringrazio il mio Relatore, il Prof. Marco Anisetti ed il mio correlatore, il Dott. Nicola Bena, per il loro preziosissimo aiuto e il loro sostegno durante tutto il percorso di ricerca.

Un ringraziamento speciale va ai miei genitori e alla mia famiglia per il loro amore, il loro incoraggiamento costante e il loro supporto. Senza di loro, questo traguardo non sarebbe stato raggiunto.

Desidero anche ringraziare i miei colleghi e amici per le interessanti discussioni, le collaborazioni stimolanti e il sostegno morale nei momenti di difficoltà. Ogni conversazione e ogni confronto hanno arricchito il mio percorso di studio.

Infine, voglio ringraziare l'intera comunità accademica e tutte le persone che, direttamente o indirettamente, hanno contribuito a questo lavoro. Siete stati fonte di ispirazione e di motivazione.

Grazie a tutti coloro che hanno reso possibile questo viaggio accademico. Sono grato per tutte le esperienze e le conoscenze acquisite e guardo al futuro con entusiasmo e gratitudine.

Prefazione

Nel contesto dell'attuale espansione dell'intelligenza artificiale nei settori critici della società e dell'industria, la necessità di strumenti affidabili per la verifica e la validazione delle proprietà non funzionali dei sistemi AI è diventata centrale. Il tema della compliance, in particolare rispetto alle nuove direttive europee come l'AI Act, ha spinto la ricerca e lo sviluppo di soluzioni capaci di garantire tracciabilità, equità e trasparenza nei processi decisionali automatizzati.

La presente tesi si inserisce in tale scenario proponendo un approccio metodologico e applicativo all'AI Assurance, attraverso la progettazione e la sperimentazione di probe specializzate e la loro integrazione nella piattaforma Moon Cloud. L'obiettivo è costruire un framework modulare che permetta la verifica sistematica di proprietà cruciali come la qualità dei dati, la fairness e la spiegabilità, rendendo effettivamente praticabile la conformità normativa e il controllo continuo sui sistemi intelligenti.

Organizzazione della tesi

La tesi è articolata come segue:

- **Capitolo 1** introduce il contesto scientifico e normativo dell'AI Assurance, delineando le motivazioni e gli obiettivi dello studio, con particolare riferimento ai rischi e alle opportunità aperte dall'AI Act.
- **Capitolo 2** presenta lo stato dell'arte in materia di assurance per sistemi AI, analizzando le principali metodologie, framework e normative di riferimento che informano la ricerca e la pratica.
- **Capitolo 3** approfondisce il formalismo teorico delle funzioni di assurance e i modelli matematici utilizzati per la verifica delle proprietà non funzionali nei sistemi AI.
- **Capitolo 4** si concentra sulla proprietà di well-formedness del dataset, descrivendo le metriche e la pipeline di verifica automatizzata implementata tramite la piattaforma.
- **Capitolo 5** affronta il tema della fairness, con un focus sulle metriche quantitativo-statistiche, la loro formalizzazione e la concreta applicazione tramite probe dedicate.

- **Capitolo 6** analizza la proprietà di explainability, introducendo tecniche interpretative avanzate e presentando la verifica automatica della trasparenza nei modelli AI.
- **Capitolo 7** illustra il caso di studio e gli esperimenti condotti, integrando la dashboard di AI Assurance all'interno di Moon Cloud e discutendo la validazione delle proprietà su dati e modelli reali.
- **Capitolo 8** raccoglie le conclusioni, sintetizzando i risultati ottenuti e indicando possibili sviluppi futuri, con particolare attenzione al contributo metodologico verso una AI responsabile e conforme.

L'auspicio è che il lavoro possa rappresentare un contributo concreto per la costruzione di sistemi di AI trasparenti, affidabili e conformi alle aspettative etiche e normative contemporanee.

Indice

Capitolo 1

Introduzione

1.1 Contesto e Motivazioni

Negli ultimi anni l'intelligenza artificiale ha assunto un ruolo strategico nell'evoluzione dei sistemi informativi, abilitando applicazioni avanzate in ambiti quali sanità, giustizia, mobilità, industria e finanza. La capacità dei modelli di AI di apprendere regole dai dati e di prendere decisioni in modo autonomo ha portato a una crescita esponenziale della loro diffusione, rendendo l'AI una componente imprescindibile nelle infrastrutture di nuova generazione. Dalle reti neurali profonde ai Large Language Models, dai sistemi di raccomandazione ai servizi di automazione, sempre più processi critici vengono affidati a soluzioni intelligenti, capaci di analizzare grandi moli di dati, estrarre conoscenza e supportare decisioni operative su scala globale.

Tuttavia, questa espansione ha messo in rilievo nuove sfide, spesso trascurate nella progettazione dei sistemi tradizionali. La complessità dei modelli, la dipendenza dai dati e la variabilità dei contesti applicativi rendono difficile garantire che le soluzioni di AI siano effettivamente affidabili, robuste, eque e trasparenti. Emergono problematiche legate alla gestione del rischio, alla sicurezza, alla privacy, ma soprattutto alla validazione delle proprietà non funzionali dei modelli: la correttezza e la qualità dei dati di input, l'equità delle decisioni rispetto ad attributi sensibili (fairness), l'interpretabilità delle logiche predittive (explainability), la tracciabilità e la conformità ai requisiti regolatori.

Rispondere a queste sfide è diventato ancora più urgente con l'introduzione di regolamentazioni specifiche a livello europeo e internazionale. In particolare, il Regolamento Europeo sull'Intelligenza Artificiale (AI Act) pone requisiti stringenti per i sistemi ad alto rischio: qualità e rappresentatività dei dati, auditabilità delle decisioni algoritmiche, assenza di discriminazioni ingiustificate, trasparenza, documentazione e supervisione umana. Tali obblighi impongono ai progettisti e agli operatori di integrare nei propri processi strumenti di verifica sistematica (assurance), capaci di misurare e certificare le proprietà essenziali di affidabilità e responsabilità dei modelli AI.

L'AI assurance, dunque, si configura come disciplina emergente e trasversale,

volta a garantire che i modelli e i dati utilizzati nei sistemi intelligenti siano allineati agli standard etici, tecnici e normativi richiesti. In questo scenario, diventa fondamentale investire in metodologie, framework e piattaforme che consentano una valutazione automatizzata, rigorosa e tracciabile delle proprietà non funzionali, promuovendo un'adozione responsabile e sostenibile dell'intelligenza artificiale nelle organizzazioni e nella società.

1.2 Soluzioni e Normative

Il panorama attuale degli studi e delle applicazioni sull'AI assurance si articola su diversi fronti, riflettendo la crescente complessità delle sfide poste dall'adozione dei sistemi intelligenti in ambiti reali. La letteratura specializzata ha evidenziato come la valutazione della correttezza e della robustezza dei modelli rappresenti solo un primo passaggio verso la mitigazione del rischio, a cui si affiancano temi quali la governance, la gestione della sicurezza operativa e il controllo del ciclo di vita delle soluzioni basate su AI.

Negli ultimi anni sono emerse metodologie sempre più strutturate, capaci di affrontare la rilevazione e il trattamento dei bias nei dati e nei modelli, la certificazione della trasparenza e la spiegabilità delle decisioni, il monitoraggio continuo delle prestazioni e delle dipendenze dai dati in ambienti dinamici. La ricerca ha prodotto una varietà di framework, tool e workflow automatizzati, spesso ideati in risposta a specifiche esigenze settoriali e di compliance regolatoria.

Sul versante normativo, le recenti direttive europee e internazionali hanno fortemente influenzato l'evoluzione delle pratiche di assurance. Il Regolamento Europeo sull'Intelligenza Artificiale (AI Act) stabilisce requisiti chiari e misurabili per i sistemi ad alto rischio, imponendo obblighi stringenti in termini di documentazione tecnica, qualità dei dati, auditabilità delle decisioni e supervisione umana. In parallelo, standard come il NIST AI Risk Management Framework (USA) e le linee guida ISO/IEC 23894 stanno guidando l'adozione di processi sistematici e automatizzati, orientati alla trasparenza, al controllo del rischio e alla accountability degli algoritmi.

Questi contributi hanno portato alla proliferazione di soluzioni e piattaforme specialistiche, sia in ambito accademico che industriale, con lo sviluppo di probe modulari, suite di auditing automatico, strumenti di explainability e interfacce integrate per la gestione della compliance. Nel complesso, l'AI assurance si sta consolidando come disciplina interdisciplinare, capace di integrare criteri scientifici, valori etici e obblighi normativi, promuovendo la fiducia nell'adozione responsabile delle tecnologie intelligenti.

1.3 Approccio Proposto

Questo lavoro propone un approccio integrato e rigoroso alla verifica delle proprietà non funzionali dei sistemi di intelligenza artificiale, con l'obiettivo di coniugare le più recenti esigenze normative e scientifiche con soluzioni concrete e automatizzabili per la compliance. L'attenzione è rivolta in particolare alla qualità e struttura dei dati (well-formedness), alla parità di trattamento tra gruppi (fairness) e alla trasparenza esplicativa dei modelli (explainability), individuate come dimensioni fondamentali sia dalla letteratura sia dal quadro regolatorio.

L'impianto sperimentale e applicativo della tesi si sviluppa a partire dall'analisi critica dello stato dell'arte e delle raccomandazioni normative, per poi tradursi nella progettazione di un framework modulare di AI assurance supportato dalla piattaforma Moon Cloud. In questo contesto, vengono concepite e realizzate probe specializzate per la valutazione automatica di ciascuna proprietà, integrate in un workflow unitario e facilmente configurabile tramite dashboard dedicata. L'approccio proposto privilegia la replicabilità, la trasparenza e l'auditabilità dei risultati, così da favorire un allineamento concreto con le aspettative di compliance richieste dai regolatori e dagli stakeholder aziendali.

La struttura del lavoro riflette questa impostazione organica: dopo una discussione sul contesto e le motivazioni alla base dello studio, vengono presentati sia i riferimenti normativi sia le metodologie più consolidate, seguiti dalla formalizzazione teorica delle funzioni di assurance e dalla descrizione dettagliata delle componenti progettate. La tesi si conclude con una valutazione sperimentale sul caso studio selezionato e una riflessione sulle criticità, i risultati ottenuti e le opportunità future di evoluzione dei paradigmi di AI assurance.

Capitolo 2

Stato dell'Arte

2.1 Intelligenza Artificiale

L'Intelligenza Artificiale (AI) rappresenta un insieme di tecniche computazionali che consentono ai sistemi di apprendere, adattarsi e prendere decisioni basate su dati. Nel corso degli ultimi decenni, l'AI ha subito una rapida evoluzione, passando da modelli simbolici basati su regole esplicite a sistemi guidati dai dati, in cui l'apprendimento automatico (Machine Learning, ML) e le reti neurali artificiali giocano un ruolo centrale. Questi sviluppi hanno reso possibile la creazione di modelli in grado di riconoscere schemi complessi, effettuare previsioni e supportare la presa di decisioni in scenari sempre più sofisticati.

L'AI può essere suddivisa in diverse categorie, tra cui *AI debole (Weak AI)* e *AI forte (Strong AI)*. L'AI debole si riferisce a sistemi specializzati nel completamento di compiti specifici, come il riconoscimento di immagini, la traduzione automatica e la generazione di testo. Questi modelli operano attraverso la generalizzazione di conoscenze apprese da grandi volumi di dati, ma non possiedono una comprensione del contesto in senso umano. Al contrario, l'AI forte teoricamente sarebbe in grado di replicare processi cognitivi simili a quelli umani, dimostrando consapevolezza e capacità di ragionamento astratto, sebbene tale livello di sviluppo sia ancora ipotetico e non realizzato nella pratica.

Un aspetto fondamentale dell'AI moderna è la distinzione tra *apprendimento supervisionato, non supervisionato e rinforzo*. Nell'apprendimento supervisionato, il modello viene addestrato su dati etichettati, apprende a mappare input con output desiderati e ottimizza i propri parametri riducendo l'errore rispetto a un insieme di esempi di riferimento. L'apprendimento non supervisionato, invece, non si basa su etichette predefinite, ma estrae schemi nascosti all'interno dei dati attraverso tecniche come il clustering o la riduzione della dimensionalità. Infine, l'apprendimento per rinforzo impiega un approccio basato su agenti che interagiscono con un ambiente, apprendendo strategie ottimali attraverso un sistema di ricompense e penalità.

Negli ultimi anni, l'adozione diffusa di tecniche basate su reti neurali profonde ha ulteriormente potenziato le capacità dell'AI. Il *Deep Learning (DL)* sfrutta

architetture con molteplici livelli di trasformazione non lineare per elaborare informazioni ad alta dimensionalità, consentendo applicazioni avanzate in settori quali la visione artificiale, l'elaborazione del linguaggio naturale e la guida autonoma. Algoritmi basati su modelli come le *reti convoluzionali (CNN)* e le *reti neurali ricorrenti (RNN)* hanno dimostrato prestazioni straordinarie nel riconoscimento di immagini e nella modellazione di sequenze temporali, mentre più recentemente i *Transformers*, come BERT e GPT, hanno ridefinito il panorama delle applicazioni legate al linguaggio naturale.

Parallelamente alla crescita delle capacità computazionali e alla disponibilità di dataset su larga scala, l'AI ha trovato applicazione in numerosi ambiti, tra cui sanità, finanza, mobilità e automazione industriale. L'analisi di immagini mediche assistita dall'AI ha migliorato la diagnosi precoce di malattie, mentre nei mercati finanziari, algoritmi di apprendimento automatico vengono impiegati per la previsione delle tendenze e l'ottimizzazione del trading. Nel settore della guida autonoma, modelli basati su reti neurali analizzano in tempo reale dati provenienti da sensori per supportare la navigazione dei veicoli senza conducente. Tuttavia, nonostante i progressi, la crescente complessità dei modelli AI pone sfide significative, tra cui la scalabilità, la gestione dell'incertezza e la necessità di modelli interpretabili e affidabili.

Infine, lo sviluppo di *modelli generativi* ha aperto nuove prospettive nell'ambito dell'AI. Architetture come le *Generative Adversarial Networks (GANs)* e i *modelli di diffusione* vengono utilizzati per la sintesi di immagini, la generazione di dati artificiali e la creazione di contenuti sintetici, dimostrando un impatto crescente in settori quali l'arte digitale, il design e la simulazione di ambienti virtuali.

L'Intelligenza Artificiale continua a rappresentare un'area di ricerca attiva e in continua espansione, con implicazioni significative in molteplici discipline. La sfida attuale risiede nel bilanciamento tra lo sviluppo di modelli sempre più performanti e la necessità di garantire la loro sicurezza, robustezza e affidabilità, aspetti fondamentali per una loro integrazione efficace e sostenibile nei contesti applicativi reali.

2.2 Assurance nei Sistemi Informatici

L'*assurance* nei sistemi informatici si riferisce all'insieme di pratiche, metodologie e strumenti volti a garantire che un sistema soddisfi determinati requisiti di sicurezza, affidabilità e correttezza operativa. In un contesto tecnologico sempre più complesso e interconnesso, l'assurance rappresenta un elemento fondamentale per la validazione di software e infrastrutture critiche, assicurando che i sistemi operino in modo conforme alle specifiche e siano resilienti a errori e vulnerabilità.

L'assurance può essere suddivisa in diverse aree chiave, tra cui:

- **Security Assurance**: garantisce che il sistema sia protetto da minacce e attacchi informatici, attraverso tecniche come l'analisi delle vulnerabilità, la verifica formale della sicurezza e l'adozione di modelli di threat modeling.

- **Reliability Assurance**: si occupa della capacità di un sistema di operare senza guasti per un determinato periodo di tempo, attraverso metodi come il testing su larga scala, l'analisi della tolleranza ai guasti e le simulazioni di stress.
- **Functional Assurance**: verifica che il sistema soddisfi i requisiti funzionali specificati, mediante test di conformità, validazione automatizzata e prove di regressione.
- **Performance Assurance**: valuta la capacità del sistema di mantenere livelli di prestazione adeguati sotto carico, garantendo efficienza e scalabilità in ambienti di produzione.
- **Compliance Assurance**: assicura che il sistema sia conforme a normative e standard internazionali, come ISO/IEC 27001 per la sicurezza delle informazioni o ISO/IEC 25010 per la qualità del software.

Le tecniche utilizzate per garantire l'assurance di un sistema variano in base al contesto applicativo e al livello di criticità dell'infrastruttura. I metodi tradizionali includono test funzionali e di sicurezza, analisi formale, monitoraggio continuo e strategie di tolleranza ai guasti. Il *testing e validazione* comprende test di conformità ai requisiti specifici, mentre l'*analisi formale* si avvale di modelli matematici per dimostrare proprietà fondamentali del software. Il *monitoraggio e audit* permettono di rilevare anomalie e garantire la tracciabilità operativa, mentre tecniche di *fault tolerance* assicurano la continuità operativa anche in presenza di guasti.

L'evoluzione tecnologica e la crescente diffusione di sistemi basati su AI hanno reso necessario un adattamento delle metodologie di assurance tradizionali. A differenza dei sistemi software deterministici, i modelli di AI introducono variabilità e dipendenza dai dati, rendendo più complessa la verifica della correttezza e della robustezza. La presenza di fenomeni come bias nei dati, comportamento non interpretabile e sensibilità alle perturbazioni impone lo sviluppo di nuovi approcci per garantire affidabilità, equità e trasparenza. L'assurance per questi sistemi deve quindi integrare tecniche specifiche per valutare proprietà emergenti legate alla generalizzazione, all'adattabilità e alla stabilità dei modelli.

2.3 Assurance nei Sistemi di Intelligenza Artificiale

L'*assurance* dei sistemi di Intelligenza Artificiale riguarda la valutazione sistematica delle proprietà di sicurezza, affidabilità e correttezza dei modelli di apprendimento automatico. A differenza del software tradizionale, il cui comportamento è definito esplicitamente dal codice, i sistemi di AI apprendono dai dati, introducendo una variabilità che rende complessa la verifica del loro funzionamento. Per questo motivo, le metodologie di assurance adottate nei sistemi convenzionali devono

essere adattate o completamente ripensate per garantire che i modelli AI rispettino requisiti critici come robustezza, equità e interpretabilità.

In particolare, l'application dell'AI in settori regolati — come la sanità, la finanza e i sistemi di trasporto autonomi — impone requisiti rigorosi di assurance, poiché un errore del sistema può comportare conseguenze gravi per la sicurezza, l'equità o la continuità operativa dei servizi. In questi ambiti, è fondamentale che i modelli siano sottoposti a processi di testing approfonditi e a monitoraggio continuo, per rilevare eventuali anomalie o degradazioni. Viene inoltre frequentemente adottato un approccio *human-in-the-loop*, in cui le decisioni automatiche vengono supervisionate da operatori umani qualificati, in grado di intervenire nei casi di incertezza o criticità.

L'**AI Assurance** si sviluppa attraverso una combinazione di tecniche di verifica, testing e controllo, con l'obiettivo di garantire che i modelli soddisfino criteri di qualità predefiniti. Le metodologie adottate possono essere suddivise in diverse categorie:

- **Validazione e Testing dei Modelli**: la verifica delle prestazioni di un modello avviene attraverso metriche di accuratezza, precisione e recall, ma nei contesti di assurance è necessario integrare queste valutazioni con test di robustezza. Tecniche come *perturbation testing* e *adversarial testing* consentono di analizzare la stabilità del modello rispetto a variazioni nei dati di input, mentre approcci basati su *out-of-distribution detection* mirano a individuare scenari in cui il modello potrebbe fallire.

- **Bias Detection e Fairness Assurance**: i sistemi di AI devono essere valutati per individuare possibili distorsioni introdotte dai dati di addestramento. Metodi di fairness assurance includono l'analisi di distribuzioni demografiche e metriche come il *demographic parity* e il *equalized odds*, utilizzate per misurare l'equità dei risultati rispetto a gruppi distinti. In alcuni casi, vengono applicate tecniche di *re-weighting* o *post-processing* per mitigare eventuali bias.

- **Explainability e Interpretabilità**: data la natura spesso opaca dei modelli di deep learning, strumenti di interpretabilità come *SHAP (Shapley Additive Explanations)* e *LIME (Local Interpretable Model-Agnostic Explanations)* vengono impiegati per fornire una spiegazione dell'output del modello. L'assurance dell'AI include strategie per garantire che le decisioni del modello siano comprensibili e giustificabili agli utenti finali, specialmente in contesti normati come sanità e finanza.

- **Verifica Formale e Metodi di Certificazione**: alcuni ambiti, come i sistemi critici, richiedono l'uso di tecniche di verifica formale per garantire la correttezza del comportamento del modello. Metodi basati su *model checking* o *theorem proving* vengono utilizzati per dimostrare proprietà specifiche del sistema, come la conformità a vincoli di sicurezza o l'assenza di determinati tipi di errori.

- **Monitoraggio e Controllo in Produzione**: poiché le prestazioni di un modello possono degradare nel tempo a causa di *data drift* o *concept drift*, è fondamentale implementare strategie di monitoraggio continuo. Tecniche come la *drift detection* e il *continual learning* vengono impiegate per rilevare cambiamenti nei dati e adattare il modello in modo dinamico.

2.3.1 Normative di Riferimento

L'evoluzione dell'intelligenza artificiale ha portato alla definizione di normative e framework di governance specifici volti a regolamentare l'uso e la certificazione dei modelli di IA in contesti critici. A livello internazionale, diversi enti normativi stanno sviluppando standard per garantire la sicurezza e l'affidabilità dell'IA.

AI Act dell'Unione Europea

L'**AI Act dell'Unione Europea** rappresenta uno dei primi tentativi organici di regolamentazione dell'IA, introducendo una classificazione dei sistemi in base al livello di rischio e stabilendo requisiti stringenti per quelli ad alto rischio. Questo regolamento mira a garantire che i sistemi di IA utilizzati nell'UE siano sicuri, trasparenti, tracciabili, non discriminatori e rispettosi dell'ambiente. I sistemi di IA dovrebbero essere supervisionati da persone, anziché da automazione, per evitare conseguenze dannose.

Classificazione dei rischi

L'AI Act classifica le applicazioni di IA in quattro categorie di rischio:

- **Rischio inaccettabile**: applicazioni vietate che rappresentano una minaccia per la sicurezza, i mezzi di sussistenza e i diritti delle persone, come i sistemi di punteggio sociale utilizzati dai governi.
- **Rischio alto**: applicazioni che possono influenzare significativamente la vita delle persone, come quelle legate ai trasporti, all'istruzione e all'occupazione. Questi sistemi richiedono una valutazione di conformità obbligatoria prima dell'immissione sul mercato.
- **Rischio limitato**: applicazioni che devono rispettare specifici obblighi di trasparenza, informando gli utenti che stanno interagendo con una macchina.
- **Rischio minimo**: applicazioni che rappresentano un rischio minimo o nullo, come i filtri antispam.

Requisiti per i sistemi ad alto rischio

I sistemi di intelligenza artificiale classificati come *ad alto rischio* secondo l'AI Act dell'Unione Europea sono soggetti a una serie di requisiti normativi rigorosi, volti a garantire un elevato livello di affidabilità, sicurezza e trasparenza. Tali

requisiti riflettono l'esigenza di mitigare i potenziali impatti negativi che tali sistemi potrebbero avere sulla vita, sui diritti e sulle libertà fondamentali delle persone. Di seguito si approfondiscono i principali obblighi richiesti:

1. Valutazione e mitigazione del rischio. I fornitori devono attuare un processo sistematico di identificazione, analisi e gestione dei rischi legati al funzionamento del sistema AI. Questo include la previsione di scenari di errore, la valutazione delle conseguenze di un eventuale malfunzionamento e l'implementazione di misure preventive per limitare i danni. La gestione del rischio deve essere documentata, periodicamente aggiornata e proporzionata alla complessità e all'impatto del sistema.

2. Qualità dei dati di addestramento, validazione e test. Uno degli aspetti più critici per garantire il comportamento corretto di un sistema AI è la qualità dei dati utilizzati nelle fasi di sviluppo. I dataset devono essere pertinenti, rappresentativi, privi di errori e, soprattutto, non discriminatori. È necessario adottare strategie che prevengano l'introduzione di bias sistematici e che assicurino un'adeguata copertura delle variabili sensibili e dei diversi gruppi demografici.

3. Tracciabilità e registrazione delle attività. Per consentire audit esterni e garantire la responsabilità operativa, i sistemi ad alto rischio devono mantenere registri dettagliati delle loro operazioni. Questo implica l'adozione di meccanismi di logging che documentino le decisioni prese dal modello, gli input ricevuti, le configurazioni attive e gli eventuali eventi anomali riscontrati durante l'esecuzione.

4. Documentazione tecnica e funzionale. Il sistema deve essere accompagnato da una documentazione esaustiva che ne descriva in modo trasparente l'architettura, gli obiettivi funzionali, i limiti noti, i dati utilizzati e le metriche di performance. Questa documentazione è essenziale per consentire valutazioni indipendenti e per supportare i processi di certificazione e conformità.

5. Trasparenza e informazione all'utente. Gli utenti che interagiscono con un sistema AI devono essere chiaramente informati circa la natura automatizzata del processo decisionale, i suoi limiti e le modalità con cui possono richiedere chiarimenti o contestare i risultati. Tale trasparenza è particolarmente rilevante nei settori regolati, dove la consapevolezza dell'utente gioca un ruolo cruciale per la tutela dei suoi diritti.

6. Supervisione umana. I sistemi ad alto rischio devono includere meccanismi che garantiscano un adeguato livello di controllo umano. Questo può avvenire sia a priori, durante la progettazione e la validazione, sia a posteriori, attraverso la possibilità per operatori qualificati di monitorare il comportamento del sistema, intervenire in caso di malfunzionamenti o sospendere l'operatività qualora necessario.

7. Robustezza, sicurezza e accuratezza. Infine, i sistemi devono essere progettati per funzionare in modo affidabile anche in condizioni avverse. Ciò include la resilienza a input errati o avversari, la protezione contro attacchi informatici, la gestione degli errori interni e il rispetto di soglie minime di accuratezza. La robustezza non si limita alla sola prestazione tecnica, ma coinvolge anche la continuità del servizio e la capacità del sistema di mantenere comportamenti

accettabili nel tempo.

Implementazione e governance

Per garantire l'applicazione uniforme dell'AI Act, sono stati istituiti diversi organismi:

- **Autorità nazionali competenti**: ogni Stato membro nomina autorità responsabili della supervisione dell'applicazione del regolamento.
- **Comitato europeo per l'intelligenza artificiale**: organismo che promuove la cooperazione tra le autorità nazionali e fornisce orientamenti sull'applicazione del regolamento.
- **AI Office**: ufficio dell'Unione Europea incaricato di supportare l'implementazione e l'applicazione del regolamento.

Requisiti operativi e progettuali

Dal punto di vista tecnico, i requisiti normativi si traducono spesso in specifiche misurabili e verificabili. Alcuni esempi pratici includono:

- **Robustezza**: i sistemi devono essere resilienti a input avversari o perturbazioni. Ad esempio, un sistema di riconoscimento facciale ad alto rischio non dovrebbe alterare le sue decisioni in presenza di piccoli cambiamenti (es. occhiali o variazioni di luce).
- **Tracciabilità dei dati**: l'intero ciclo di vita dei dati deve essere documentato. Questo implica versionamento dei dataset, logging delle trasformazioni e tracciamento dei modelli addestrati con quei dati.
- **Equità (fairness)**: l'output del modello non deve variare in modo ingiustificato al variare di caratteristiche sensibili come sesso, etnia o età. Ad esempio, un algoritmo di selezione del personale non dovrebbe penalizzare i candidati in base al genere.
- **Accuratezza minima garantita**: in ambiti critici (es. medicina, giustizia predittiva), i modelli devono rispettare soglie prestazionali minime su dataset di validazione certificati.
- **Interpretabilità**: i modelli devono poter essere spiegati, ad esempio tramite tecniche come LIME, SHAP o saliency maps nel caso di reti neurali, per permettere audit e verifiche indipendenti.

Questi esempi mostrano come la conformità normativa richieda una progettazione dell'IA non solo etica, ma anche ingegneristicamente rigorosa. L'AI Assurance si pone dunque come ponte tra i requisiti legali e le scelte architetturali, facilitando l'adozione responsabile dell'intelligenza artificiale.

Standard internazionali e iniziative globali sull'AI Assurance

Oltre al quadro normativo europeo delineato dall'AI Act, il tema dell'AI assurance ha ricevuto crescente attenzione anche a livello internazionale, con la definizione di specifici standard e framework per la governance e il rischio nei sistemi intelligenti. Tra le principali iniziative, si segnalano:

- **NIST AI Risk Management Framework (AI RMF)**: sviluppato dal National Institute of Standards and Technology (USA), propone un approccio strutturato alla valutazione, gestione e monitoraggio dei rischi legati all'IA, includendo principi come trasparenza, affidabilità, equità e privacy. Il framework è pensato per essere applicabile in contesti pubblici e privati, indipendentemente dall'adesione legislativa locale [1].

- **IEEE 7000 series**: una famiglia di standard pubblicati dall'Institute of Electrical and Electronics Engineers che trattano di gestione etica dei sistemi autonomi e delle implicazioni sociali dell'IA (es. IEEE 7000-2021 sul design etico). Offrono linee guida operative, spesso focalizzate su processi di sviluppo, design e auditing, in una prospettiva multidisciplinare [2].

- **ISO/IEC 23894:2023**: standard internazionale per la gestione del rischio nei sistemi di IA, che approfondisce requisiti di robustezza, sicurezza e conformità in linea con le best practice globali in materia di gestione delle tecnologie emergenti [3].

Sebbene l'AI Act europeo si concentri principalmente sul rischio per diritti fondamentali e sicurezza, gran parte degli standard sopra citati enfatizza anch'essa l'importanza di trasparenza, verifica indipendente, responsabilità e tracciabilità. Tuttavia, gli standard internazionali tendono a privilegiare un approccio flessibile, adattabile ai diversi contesti organizzativi, rispetto all'impostazione prescrittiva e basata su classi di rischio dell'AI Act. Per una strategia di AI assurance completa e realmente efficace, appare dunque fondamentale integrare gli obblighi normativi UE con le migliori pratiche e le linee guida sviluppate in ambito extra-europeo.

Conclusioni

L'adozione di normative come l'AI Act dell'Unione Europea e altri standard internazionali rappresenta un passo fondamentale verso la regolamentazione dell'uso dell'intelligenza artificiale, garantendo che lo sviluppo e l'implementazione di tali tecnologie avvengano in modo sicuro, etico e rispettoso dei diritti fondamentali. Questi framework forniscono linee guida chiare per le aziende e le organizzazioni che sviluppano e utilizzano sistemi di IA, promuovendo la fiducia del pubblico e l'adozione responsabile di queste tecnologie emergenti.

2.4 Moon Cloud

2.4.1 Introduzione

Moon Cloud è una piattaforma innovativa progettata per fornire una valutazione continua della conformità e un'analisi dell'assurance per applicazioni e infrastrutture ICT, inclusi i sistemi basati su intelligenza artificiale e machine learning.

La piattaforma offre una verifica completa dei servizi durante il loro funzionamento, concentrandosi principalmente sull'assicurazione della sicurezza e delle prestazioni.

2.4.2 Sistema

L'architettura di Moon Cloud è stata progettata come un sistema distribuito cloud-native, composto da microservizi che collaborano tra loro per fornire una piattaforma completa di assurance continua. L'obiettivo è quello di valutare automaticamente proprietà non funzionali di sistemi ICT, compresi quelli che integrano modelli di intelligenza artificiale.

Moon Cloud si distingue per una verifica della conformità basata su modelli e sulla raccolta strutturata di evidenze, mediante template adattabili alle esigenze specifiche del dominio. La piattaforma è progettata per essere completamente automatizzabile e personalizzabile, grazie a regole configurabili e controlli schedulabili nel tempo. La copertura è completa lungo tutto lo stack, dalla rete all'applicazione: ogni livello può essere analizzato tramite probe specializzate. È inoltre prevista l'integrazione con strumenti esistenti di monitoraggio e verifica esterni. L'esecuzione dei controlli avviene in ambienti isolati tramite container Docker, orchestrati su cluster Kubernetes. Infine, sono disponibili probe specifiche per la valutazione di modelli di AI/ML, al fine di verificare proprietà come fairness ed explainability.

L'architettura include i seguenti componenti principali:

- **Dashboard**: è l'interfaccia web (basata su Angular) che consente all'utente di avviare verifiche, visualizzare i risultati e gestire risorse come target e profili.
- **API backend**: componente centrale sviluppato in Django. Espone un'interfaccia REST per la gestione dell'inventory, delle valutazioni e delle statistiche. Coordina le richieste e si connette al database Postgres.
- **K8s Manager**: scritto in Go, riceve le richieste di esecuzione e le traduce in job Kubernetes, schedulati dinamicamente o periodicamente (cron jobs). È responsabile della mappatura tra livelli logici e fisici di esecuzione.
- **NATS**: sistema di messaggistica pub/sub che collega in modo asincrono componenti backend e probe.

- **Vault**: gestisce le credenziali necessarie alle probe per connettersi ai target da verificare, mantenendo elevati standard di sicurezza.
- **Evidence Writer**: sviluppato in Go, riceve le evidenze delle probe tramite NATS, le normalizza e le memorizza nel database.
- **Database**: unico punto di persistenza condiviso da API, K8s Manager ed Evidence Writer. Contiene asset, configurazioni, risultati e metadati.

L'intero sistema è orchestrato tramite Kubernetes, che si occupa della gestione automatica dei job, del bilanciamento del carico e della scalabilità. Grazie all'uso di container Docker e a un'architettura completamente modulare, Moon Cloud è facilmente estendibile, testabile e adattabile a molteplici contesti operativi.

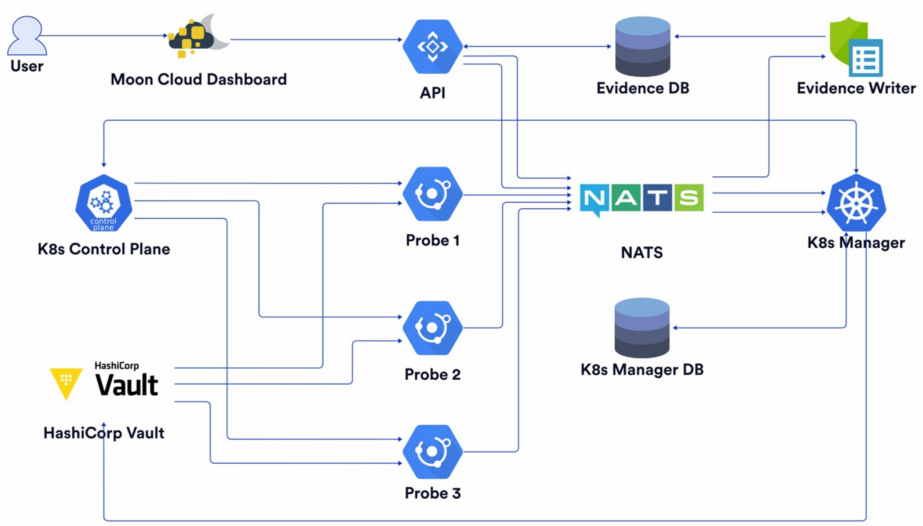

Figura 2.1: Architettura del sistema Moon Cloud

Questa architettura distribuita consente al sistema di essere robusto e affidabile anche in scenari dinamici, offrendo una piattaforma evoluta per la valutazione automatica di proprietà non funzionali come sicurezza, affidabilità e trasparenza, come descritto in [4].

2.4.3 Probe

Le *probe* sono i componenti esecutivi del sistema Moon Cloud. Ogni probe rappresenta un'unità di verifica autonoma, incaricata di controllare una specifica proprietà di un target, come ad esempio la corretta configurazione di un protocollo, la presenza di vulnerabilità, o il rispetto di requisiti legati a fairness, explainability o *well-formedness* dei dati.

Ogni probe è implementata in Python, secondo una struttura standard definita dal progetto Moon Cloud. È progettata per essere eseguita come container Docker all'interno di un cluster Kubernetes, garantendo così isolamento, portabilità e facilità di integrazione.

Input e Output

Le probe ricevono in ingresso un dizionario JSON che definisce tutti i parametri necessari per l'esecuzione. Questo può includere:

- **Indirizzo del target**: hostname o IP del sistema da analizzare.
- **Credenziali**: chiavi o password per accedere via SSH o API.
- **Parametri specifici**: come porte, soglie, path, livelli attesi, ecc.

L'output è composto da tre elementi chiave:

- **Integer result**: codice intero che indica l'esito del controllo. Ad esempio, 0 per successo, 1 per errore di input, 2 per errore di connessione, 3 per errore interno.
- **Pretty result**: messaggio sintetico, facilmente interpretabile, che riassume il risultato in forma leggibile.
- **Extra data**: contenuto strutturato con evidenze raccolte durante il controllo. Può includere risultati grezzi, valori misurati, raccomandazioni, messaggi diagnostici o output di tool esterni.

Struttura interna

Le probe sono modellate come macchine a stati finiti. Ogni stato corrisponde a una fase operativa distinta, e le transizioni determinano l'avanzamento logico del controllo. La struttura è suddivisa in due catene:

- **Forward chain**: rappresenta la sequenza standard di azioni (es. connessione, verifica, valutazione, salvataggio).
- **Rollback chain**: attivata automaticamente in caso di errore per annullare eventuali modifiche o pulire risorse temporanee.

Ogni stato è implementato come un metodo Python e deve restituire un oggetto JSON parziale che aggiorna l'output globale della probe.

Gestione degli errori e rollback

Le probe gestiscono eccezioni interne in modo controllato. Ogni errore (es. connessione fallita, input mancante, parsing errato) viene intercettato e mappato in uno stato specifico, che può attivare una sequenza di rollback.

Il rollback è pensato per garantire la non intrusività: eventuali modifiche temporanee sul target vengono annullate (es. file creati, configurazioni modificate), mantenendo il sistema in uno stato invariato.

Esecuzione

Quando l'utente avvia una valutazione tramite Dashboard o API, la richiesta viene inoltrata al *K8s Manager*, che genera un job Kubernetes con la probe corrispondente. L'input viene montato nel container e la probe viene eseguita. Al termine, il risultato viene inviato via *NATS* all'*Evidence Writer*, che struttura e salva l'output nel database.

Tutti i risultati sono poi visualizzabili tramite la Dashboard. La separazione tra orchestrazione (API/K8s) e logica (probe) consente aggiornamenti indipendenti, testabilità modulare e un'elevata affidabilità del sistema.

Conclusione

Le probe rappresentano il meccanismo fondamentale attraverso cui Moon Cloud implementa le sue funzioni di assurance. Grazie alla loro architettura modulare, alla containerizzazione e alla chiarezza dei formati input/output, esse permettono di costruire verifiche automatiche e ripetibili, adattabili a una vasta gamma di contesti e target.

Capitolo 3

Framework Teorico

3.1 Formalismo delle Funzioni di AI Assurance

Nel contesto della valutazione formale dei sistemi di intelligenza artificiale, una proprietà viene considerata verificata se il comportamento del modello, o di un suo artefatto, soddisfa un criterio misurabile. Tale verifica può essere modellata come una funzione matematica che associa al modello un insieme di valori numerici, successivamente confrontati con soglie predefinite.

Formalmente, si definisce una funzione di *AI Assurance* come una mappatura:

$$F : A \to \mathbb{R}^n \tag{3.1}$$

dove:

- A è lo spazio dei modelli di AI o degli artefatti associati (ad esempio dati di addestramento, configurazioni, output);
- $\mathbb{R}^n$ rappresenta lo spazio delle valutazioni quantitative restituite dalla funzione.

Il risultato della funzione viene quindi confrontato con un insieme di soglie $\theta \in \mathbb{R}^n$, determinando se la proprietà considerata sia soddisfatta:

$$\text{verifica}(a) = \begin{cases} \text{True} & \text{se } F(a) \models \theta, \\ \text{False} & \text{altrimenti.} \end{cases} \tag{3.2}$$

La funzione F deve soddisfare alcune proprietà fondamentali che garantiscono la coerenza e l'affidabilità delle valutazioni di assurance nei modelli di AI. Le principali proprietà richieste sono:

Stabilità

La stabilità della funzione F assicura che piccole variazioni nei modelli di intelligenza artificiale, o nei loro parametri, non portino a cambiamenti drastici

nella valutazione dell'assurance. Questa proprietà è particolarmente rilevante nei contesti in cui modifiche marginali nei dati o nell'architettura del modello non dovrebbero influire in modo significativo sulla misura della proprietà valutata.

Formalmente, la stabilità può essere espressa come continuità della funzione F tra due spazi metrici:

$$\forall \epsilon > 0, \quad \exists \delta > 0 \quad \text{tale che} \quad d_A(a_1, a_2) < \delta \Rightarrow d_B(F(a_1), F(a_2)) < \epsilon. \tag{3.3}$$

dove:

- A è lo spazio dei modelli o artefatti, dotato di una metrica d_A;
- $B \subseteq \mathbb{R}^n$ è lo spazio delle valutazioni della funzione F, dotato di una metrica d_B;
- $a_1, a_2 \in A$ sono due modelli "vicini" secondo d_A;
- ϵ, δ sono quantità positive arbitrariamente piccole.

Questa formulazione impone che, se due modelli sono simili (cioè la loro distanza $d_A(a_1, a_2)$ è piccola), anche le rispettive valutazioni $F(a_1)$ e $F(a_2)$ debbano essere simili (cioè con distanza d_B piccola). In altri termini, *la funzione non deve amplificare piccole perturbazioni.*

La stabilità è fondamentale per evitare effetti di discontinuità nella valutazione: ad esempio, un'alterazione minima nei dati di input non dovrebbe portare a una variazione radicale nell'esito dell'assurance. Ciò garantisce una maggiore affidabilità nelle decisioni che dipendono da tali valutazioni.

Monotonicità

La monotonicità garantisce che, se un modello a_1 è migliore di un altro modello a_2 rispetto a una determinata proprietà, la funzione di assurance deve riflettere questa relazione, ovvero:

$$a_1 \succ a_2 \quad \Rightarrow \quad F(a_1) \geq F(a_2). \tag{3.4}$$

Dove $\succ$ rappresenta un ordinamento parziale o totale su A rispetto a una proprietà specifica. In altre parole, se un modello presenta una qualità superiore in termini di fairness, robustezza o interpretabilità rispetto a un altro, la valutazione di assurance dovrebbe rispettare questa gerarchia.

Generalizzabilità

La generalizzabilità si riferisce alla capacità della funzione F di mantenere la coerenza delle sue valutazioni quando applicata a differenti distribuzioni di dati o

architetture di modelli. Questa proprietà è essenziale affinché i risultati dell'assurance siano validi in diversi contesti operativi, senza dipendere da fattori specifici del dataset o dell'implementazione del modello.

Matematicamente, la generalizzabilità può essere formalizzata richiedendo che, a parità di modello $a \in A$, le valutazioni ottenute su distribuzioni diverse di dati restino prossime tra loro:

$$\forall a \in A, \quad \forall D_1, D_2 \in \mathcal{D}, \quad \|F(a; D_1) - F(a; D_2)\| < \epsilon \tag{3.5}$$

dove:

- A è lo spazio dei modelli o degli artefatti;
- $\mathcal{D}$ è l'insieme delle possibili distribuzioni di dati;
- $F(a; D)$ è il valore restituito dalla funzione F applicata al modello a in relazione alla distribuzione D;
- $\epsilon > 0$ è una tolleranza arbitrariamente piccola;
- $\|\cdot\|$ è una norma su $\mathbb{R}^n$, ad esempio la norma euclidea.

Questa formulazione impone che, cambiando i dati di riferimento, la valutazione dell'assurance non subisca variazioni significative, garantendo così una certa indipendenza dal contesto. La stessa formulazione può essere estesa a variazioni.

3.2 Contesto di Applicazione delle Funzioni di Assurance

Una funzione di assurance può essere applicata in momenti differenti del ciclo di vita di un sistema di intelligenza artificiale, a seconda della proprietà che si intende verificare e dell'artefatto su cui viene condotta l'analisi.

In linea generale, le funzioni di assurance possono agire su:

- **Artefatti statici**, disponibili prima del deployment del modello. In questo caso l'oggetto della verifica può essere, ad esempio, un dataset di addestramento, un file di configurazione o una rappresentazione del modello (ad esempio, un grafo computazionale). Le proprietà comunemente valutate in questa fase includono la *well-formedness*, il rispetto di vincoli strutturali, la presenza di bias nei dati o il corretto rispetto dei contratti di input.
- **Comportamenti dinamici**, osservabili durante l'esecuzione del modello in produzione. In questo caso, la funzione di assurance opera su stream di input e output del modello, valutando proprietà quali la *fairness comportamentale*, la *robustezza a perturbazioni*, la *coerenza rispetto a vincoli semantici* o la *stabilità dell'inferenza*.

La distinzione tra questi due ambiti riflette la natura dell'informazione disponibile: nel primo caso, l'analisi si basa su elementi pienamente osservabili e immutabili; nel secondo caso, la verifica si confronta con la variabilità dei dati in tempo reale e può richiedere meccanismi di aggregazione, monitoraggio o adattamento.

In alcune situazioni, una stessa funzione di assurance può essere definita in modo tale da operare sia su artefatti statici che dinamici, purché l'input rispetti il dominio previsto dalla funzione. Tuttavia, l'efficacia e il significato della valutazione possono cambiare in modo sostanziale in funzione del contesto applicativo.

Infine, è opportuno evidenziare che l'adozione di funzioni di assurance in fase pre-deployment non esclude l'utilità di ulteriori verifiche durante l'esecuzione, in quanto alcune proprietà (come la fairness o l'accuratezza) possono degradare nel tempo in risposta a mutamenti nei dati o nel contesto operativo.

3.2.1 Esempio Applicativo

Si consideri un'applicazione di intelligenza artificiale progettata per supportare la valutazione del rischio di complicanze in pazienti sottoposti a interventi cardiaci. Il sistema elabora dati clinici, demografici e diagnostici per fornire al personale medico una stima probabilistica del rischio associato a una specifica procedura.

Secondo l'AI Act , un sistema di questo tipo rientra nella categoria dei sistemi *ad alto rischio*, poiché incide direttamente sulla salute e sulla sicurezza delle persone, in conformità con l'Allegato III del regolamento [5].

In quanto sistema ad alto rischio, è soggetto a obblighi normativi specifici. In particolare, l'Articolo 10 dell'AI Act richiede che i dati utilizzati nei processi di addestramento, validazione e test soddisfino requisiti rigorosi. Tali dati devono essere:

- **Pertinenti**: i dati utilizzati devono essere direttamente collegati alla finalità funzionale del sistema di IA. Devono quindi riflettere le variabili realmente necessarie per il compito da svolgere, evitando l'inclusione di informazioni irrilevanti che potrebbero introdurre rumore o distorsioni nel modello.

- **Rappresentativi**: il dataset deve includere esempi che rispecchino in modo fedele la popolazione su cui il sistema verrà effettivamente applicato. Questo implica una copertura adeguata di tutte le sottopopolazioni rilevanti, per ridurre il rischio di risultati distorti o non generalizzabili.

- **Accurati e privi di errori**: i dati devono essere verificati per garantirne la correttezza e l'affidabilità. La presenza di errori, ambiguità o etichette scorrette può compromettere l'apprendimento del modello e portare a decisioni errate o non sicure.

- **Completi**: le informazioni contenute devono essere sufficienti per consentire una valutazione coerente e affidabile. L'assenza di variabili chiave o la

presenza di dati mancanti in modo sistematico può limitare l'efficacia del sistema e compromettere la validità delle sue previsioni.

In aggiunta, il regolamento impone la documentazione delle modalità di raccolta, selezione e trattamento dei dati, nonché l'indicazione esplicita della loro provenienza. Viene inoltre richiesto di adottare misure tecniche e organizzative atte a prevenire l'introduzione di distorsioni sistematiche nei dataset, anche attraverso l'impiego di tecniche di mitigazione del bias e controlli di coerenza interni.

Questo esempio evidenzia come, nei sistemi ad alto rischio, il controllo sulla qualità e sull'idoneità dei dati non rappresenti solo una buona pratica progettuale, ma costituisca un requisito vincolante a livello normativo. L'inosservanza di tali disposizioni può compromettere l'affidabilità delle valutazioni fornite dal sistema e condurre a effetti discriminatori o dannosi per gli utenti finali.

Per rispettare tali requisiti, è necessario adottare strumenti tecnici in grado di analizzare le caratteristiche strutturali, semantiche e statistiche dei dati utilizzati. Tali strumenti devono essere in grado di rilevare incongruenze, incompletezze, anomalie o squilibri nella distribuzione dei campioni, restituendo indicatori interpretabili e documentabili.

Infine, l'intero processo di verifica deve essere tracciabile, ripetibile e integrabile all'interno di una pipeline di sviluppo conforme. In questo senso, le funzioni di assurance giocano un ruolo centrale nell'implementazione di un ciclo di vita responsabile e normativamente conforme per i sistemi di intelligenza artificiale.

Capitolo 4

Proprietà: Well-Formedness del Dataset

4.1 Contesto

La qualità e la struttura dei dati rappresentano elementi imprescindibili per il corretto funzionamento dei sistemi di intelligenza artificiale (AI). I modelli di AI, e in particolare quelli basati su apprendimento automatico, dipendono in modo cruciale dalla natura e dalla qualità dei dati su cui vengono addestrati. Un dataset non adeguatamente costruito o che presenti anomalie può introdurre distorsioni (bias), compromettendo significativamente le prestazioni predittive e la capacità del modello di generalizzare correttamente su dati nuovi. Inoltre, dataset malformati possono generare effetti discriminatori non intenzionali, con conseguenze rilevanti in particolare negli ambiti di applicazione sensibili come la sanità, la finanza e la giustizia. Per queste ragioni, uno dei primi e più fondamentali ambiti di verifica nel processo di AI assurance riguarda la valutazione della *well-formedness* del dataset, ossia la sua "buona formazione" rispetto a requisiti specifici, che ne garantiscano l'idoneità per l'addestramento e l'utilizzo dei modelli. In tale contesto, la *well-formedness* si riferisce alla conformità del dataset rispetto a un insieme di criteri formali e statistici che ne certificano la coerenza interna, la rappresentatività rispetto alla distribuzione della popolazione di interesse, la correttezza del formato, la completezza informativa, nonché l'assenza di anomalie o dipendenze spurie tra le variabili. L'importanza di questi controlli non è unicamente di natura tecnica, ma si lega anche a esigenze normative e regolamentari che mirano a garantire trasparenza, equità e sicurezza delle soluzioni di AI. In particolare, l'Articolo 10 dell'AI Act dell'Unione Europea richiede che i dati impiegati per lo sviluppo di sistemi AI ad alto rischio siano pertinenti, rappresentativi, privi di errori e completi [5]. Ciò comporta l'implementazione di sistemi di controllo rigorosi e documentabili in grado di verificare sistematicamente che i dataset rispettino tali requisiti, sia in fase di progettazione che nel corso del ciclo di vita del modello. Dal punto di vista operativo, la verifica della *well-formedness* costituisce quindi un passaggio preliminare e imprescindibile all'interno di un framework

di assurance automatizzato, poiché da essa dipende la validità delle valutazioni successive sulle prestazioni, la fairness, l'affidabilità e la spiegabilità del modello AI. Un dataset che non rispetta tali requisiti può infatti indurre errori sistematici difficilmente correggibili nelle fasi successive, compromettendo l'intero processo decisionale automatizzato. Per facilitare la comprensione delle problematiche correlate ai dataset malformati, si può considerare un esempio intuitivo: un modello di classificazione del rischio creditizio addestrato con dati in cui alcune variabili risultano fortemente correlate in modo spuriamente artificiale potrebbe non riuscire a riconoscere correttamente i segnali reali, portando a decisioni errate e potenzialmente discriminatorie verso categorie di utenti specifiche. La corretta verifica della *well-formedness* aiuta a individuare e gestire tali situazioni prima che impattino negativamente sul modello. In questo capitolo verranno approfondite le caratteristiche teoriche della proprietà di *well-formedness* e illustrate le metodologie con cui essa può essere verificata mediante strumenti automatizzati di ispezione dati, ponendo le basi per una valutazione rigorosa e sistematica all'interno di un contesto di AI assurance.

4.2 Descrizione Teorica

La verifica automatica della qualità strutturale di un dataset rappresenta una delle prime attività da eseguire nel processo di assurance, poiché costituisce la base su cui si costruiscono tutte le valutazioni successive. Un dataset formalmente corretto, coerente e informativamente valido è essenziale per garantire che il modello di intelligenza artificiale sviluppato sia in grado di apprendere comportamenti generalizzabili e affidabili.

Seguendo la definizione di funzione di AI Assurance introdotta nel Capitolo 3, la verifica della proprietà di *well-formedness* del dataset può essere formalizzata come una funzione:

$$F_{\text{well-formedness}} : \mathcal{D} \to \mathbb{R}^n$$

dove $\mathcal{D}$ rappresenta lo spazio dei dataset e $F_{\text{well-formedness}}(D)$ restituisce un insieme di valutazioni quantitative sulla qualità strutturale dei dati, successivamente confrontate con soglie di accettabilità predefinite.

L'implementazione di questa funzione si basa sull'utilizzo combinato di diverse tecniche, quali::

- *Independent Component Analysis* (ICA) per trasformare il dataset in un insieme di componenti statisticamente il più possibile indipendenti;
- Analisi della *kurtosi* delle componenti ottenute, al fine di misurare la deviazione dalla gaussianità e, indirettamente, il grado di indipendenza informativa.

Questo metodo consente di esplorare in profondità la composizione statistica del dataset e di verificare se le variabili contengano informazione utile in modo

non ridondante. La combinazione di ICA e kurtosi fornisce un indicatore sintetico della struttura informativa dei dati, da utilizzare in fase pre-deployment all'interno di un framework di assurance automatizzabile.

4.2.1 Independent Component Analysis (ICA)

L'Independent Component Analysis (ICA) è una tecnica di decomposizione statistica non supervisionata utilizzata per trasformare un insieme di variabili osservabili in un insieme di componenti statisticamente indipendenti. Si tratta di un metodo particolarmente adatto a evidenziare strutture nascoste nei dati, in quanto mira a separare le fonti informative latenti che, combinate, generano le variabili osservate.

A differenza della Principal Component Analysis (PCA), che ha l'obiettivo di decorrelare le variabili e massimizzare la varianza lungo gli assi principali, l'ICA impone un vincolo più forte: l'indipendenza statistica delle nuove componenti. Mentre la decorrelazione rimuove la dipendenza lineare, l'indipendenza richiede che la distribuzione congiunta delle variabili fattorizzi nel prodotto delle distribuzioni marginali, ovvero:

$$p(s_1, s_2, \ldots, s_n) = \prod_{i=1}^{n} p(s_i)$$

dove s_i sono le componenti stimate.

Dal punto di vista operativo, dato un insieme di osservazioni $X \in \mathbb{R}^{m \times n}$, in cui ogni riga rappresenta un'osservazione e ogni colonna una variabile, l'ICA assume che X sia una combinazione lineare di sorgenti latenti indipendenti S, tali che:

$$X = AS$$

dove $A \in \mathbb{R}^{n \times n}$ è una matrice di mescolamento sconosciuta. L'obiettivo è stimare una matrice di separazione W tale che:

$$S = WX$$

in modo che le componenti di S risultino il più possibile indipendenti tra loro.

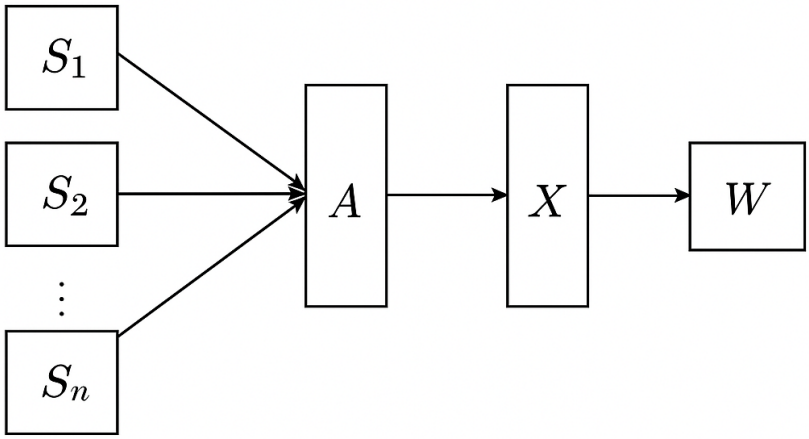

Figura 4.1: Schema concettuale del processo di separazione tramite ICA: osservazioni miscelate X, sorgenti indipendenti S, matrice di mescolamento A, e matrice di separazione W

Questa rappresentazione visiva mostra il processo alla base dell'ICA: le osservazioni miscelate X sono modellate come combinazioni lineari di componenti sorgente S tramite la matrice A, mentre il processo di separazione cerca una trasformazione W in grado di recuperare S da X. Questo schema evidenzia il flusso concettuale della tecnica e fornisce una guida utile alla sua interpretazione nei contesti applicativi.

Poiché l'indipendenza statistica non è direttamente osservabile, l'ICA si basa su criteri di ottimizzazione che sfruttano la *non-gaussianità* delle componenti.

Per il Teorema del Limite Centrale, la somma di variabili indipendenti tende alla gaussiana; pertanto, trovare proiezioni non gaussiane equivale a identificare sorgenti indipendenti. Tra le misure di non-gaussianità più utilizzate vi sono la *kurtosi* e l'*entropia negata*.

In contesto applicativo, l'ICA è utile per identificare variabili latenti in problemi di separazione di segnali (ad esempio, separazione di fonti audio), ma trova impiego anche in ambito biomedicale, neuroscienze e — in questo caso specifico — nella valutazione automatica della struttura informativa di un dataset.

4.2.2 Kurtosi come metrica di indipendenza

Per valutare il grado di indipendenza delle componenti ottenute tramite ICA, si utilizza come metrica la *kurtosi* (o curtosi), una misura della forma della distribuzione di probabilità di una variabile reale. In particolare, si considera la *kurtosi centrata e normalizzata* secondo la definizione di Fisher, che assegna valore zero a una distribuzione perfettamente normale.

La kurtosi misura la concentrazione della massa di probabilità nelle code e attorno al centro della distribuzione. Formalmente, la kurtosi di una variabile casuale X è definita come:

$$\text{kurt}(X) = \frac{\mathbb{E}[(X-\mu)^4]}{\sigma^4} - 3$$

dove μ è la media e σ la deviazione standard di X. La sottrazione di 3 rende la misura centrata sulla distribuzione normale, che ha kurtosi zero. In pratica:

- Kurtosi > 0 indica una distribuzione *leptocurtica*, con code pesanti e picco accentuato;
- Kurtosi < 0 indica una distribuzione *platicurtica*, con code leggere e picco piatto;
- Kurtosi $= 0$ corrisponde alla distribuzione normale.

Nel contesto della ICA, una componente con kurtosi significativamente diversa da zero indica una deviazione dalla gaussianità, che a sua volta è un segnale utile di potenziale indipendenza. Questo legame si fonda sull'assunto che la somma di variabili indipendenti tende alla normalità (Teorema del Limite Centrale); pertanto, per trovare sorgenti indipendenti si cercano le direzioni di proiezione con maggiore non-normalità.

Dal punto di vista applicativo, viene definita una soglia empirica (ad esempio $|\text{kurt}| > 1$) oltre la quale la componente viene considerata sufficientemente distante dalla gaussianità da poter essere trattata come informativamente indipendente. Le componenti con kurtosi prossima allo zero vengono invece interpretate come potenzialmente ridondanti o dipendenti.

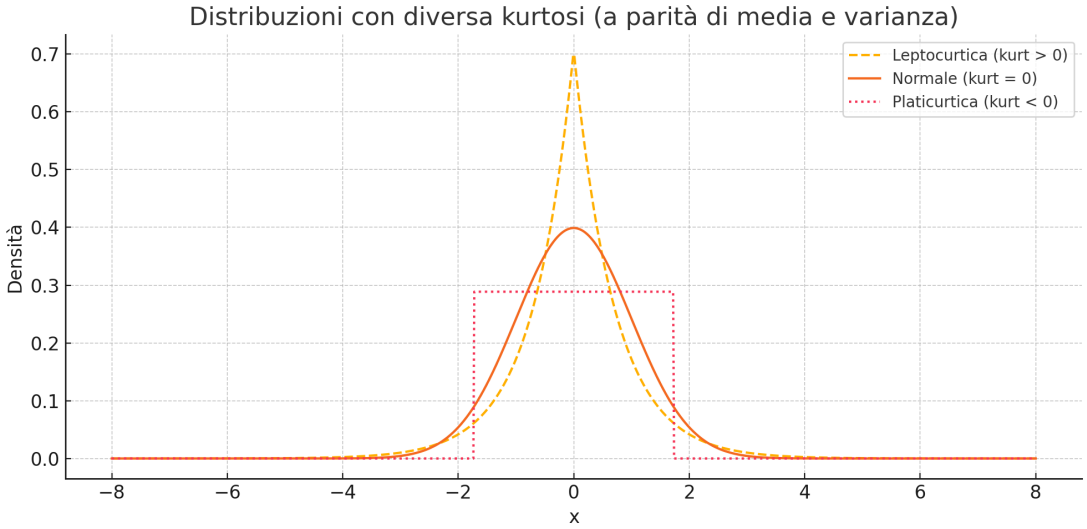

Figura 4.2: Distribuzioni platicurtica, normale e leptocurtica: effetto della kurtosi sulla forma delle code

La figura mostra graficamente la differenza tra tre distribuzioni a parità di media e varianza ma con valori di kurtosi differenti. Questa rappresentazione aiuta

a comprendere come il valore della kurtosi influisca sulla forma della distribuzione, rendendo visibile la deviazione dalla normalità.

Infine, l'esame dei valori di kurtosi su tutte le componenti indipendenti fornisce un indicatore sintetico e quantitativo della struttura informativa del dataset. L'approccio è completamente automatizzabile e può essere integrato all'interno di una pipeline di assurance per evidenziare dataset potenzialmente mal strutturati, ridondanti o non informativi.

4.2.3 Indipendenza statistica e qualità del dataset

L'indipendenza statistica delle variabili all'interno di un dataset rappresenta un elemento chiave per garantire la robustezza dell'addestramento e la validità dei modelli derivati. Dataset caratterizzati da forte correlazione tra le feature tendono a introdurre ridondanze informative, generare modelli meno generalizzabili e aumentare il rischio di overfitting, con conseguente perdita di capacità predittiva su dati reali.

In un contesto di assurance, la valutazione automatica dell'indipendenza statistica fornisce un indicatore della qualità latente dei dati, andando oltre i semplici controlli sintattici o superficiali. Tecniche come l'ICA, combinate con metriche come la kurtosi, permettono di stimare la struttura informativa reale del dataset, evidenziando anomalie, componenti ridondanti o variabili poco informative. L'uso sistematico di queste verifiche consente di prevenire la propagazione di difetti strutturali nei modelli di machine learning.

Nel ciclo di vita MLOps, l'integrazione di questo tipo di analisi all'interno delle fasi pre-deployment rappresenta un tassello fondamentale per garantire la qualità dei modelli prima dell'introduzione in ambienti produttivi. In contesti più avanzati, l'analisi della dipendenza può essere estesa anche in fase operativa, come monitoraggio di regressioni silenziose o segnali di degrado nei dati in input, soprattutto in ambienti dinamici.

Dal punto di vista normativo, l'AI Act sottolinea la necessità di utilizzare dati privi di errori, rappresentativi e completi [5]. Una struttura informativa coerente e non ridondante rappresenta una condizione necessaria per il rispetto di tali requisiti, e l'indipendenza statistica ne è una manifestazione misurabile.

L'adozione di un approccio formale, basato su trasformazioni matematiche e soglie metriche esplicitamente definite, fornisce una base oggettiva e ripetibile per supportare la dichiarazione di conformità ai requisiti di qualità dei dati. Inoltre, questi strumenti si integrano facilmente con pipeline di verifica automatica e auditabili, facilitando anche la tracciabilità documentale, elemento richiesto nei sistemi AI soggetti a regolamentazione.

In sintesi, l'impiego dell'ICA e della kurtosi nella valutazione dell'indipendenza tra le variabili costituisce una metodologia efficace per elevare la qualità dei dati da una prospettiva tanto tecnica quanto regolatoria, rafforzando l'intero processo di assurance e la fiducia nei modelli sviluppati.

4.2.4 Struttura Operativa

Nei paragrafi precedenti è stata introdotta la funzione di assurance come mappatura matematica formale, in grado di associare a un artefatto (ad esempio, un dataset) un insieme di valutazioni quantitative sulle sue proprietà strutturali. Tuttavia, affinché tale formalismo assuma significato operativo nell'ambito di un framework di verifica automatizzata, è necessario esplicitare come i diversi "blocchi costitutivi" (ad es. ICA, kurtosi, soglie empiriche) vengano effettivamente composti nella costruzione della funzione di verifica eseguibile.

A livello concettuale, la funzione di assurance per la **well-formedness** si articola nella seguente sequenza logica:

1. **Preparazione**: il dataset D viene preprocessato (esclusione colonne label, gestione di eventuali NaN, standardizzazione).

2. **Estrazione delle componenti**: viene applicata l'Independent Component Analysis (ICA), ricavando un insieme di componenti statisticamente indipendenti $S_1, \ldots, S_n$.

3. **Calcolo degli indicatori**: per ciascuna componente S_j si calcola il valore di kurtosi centrata e normalizzata $\kappa(S_j)$ secondo la definizione adottata.

4. **Aggregazione**: i valori di kurtosi vengono confrontati con la soglia prefissata e aggregati in un unico verdetto, secondo la regola empirica (tutte le componenti devono soddisfare $|\kappa(S_j)| > 1$).

Si noti che ogni passaggio della catena operativa non è solo una scelta tecnica, ma risponde anche a requisiti specifici di trasparenza, replicabilità e auditabilità richiesti dai moderni framework normativi sull'intelligenza artificiale (cfr. AI Act, art. 10). La soglia empirica adottata, ad esempio, è motivata dalla necessità di garantire che ogni direzione informativa individuata sia effettivamente ricca di contenuto statistico e non ridondante, prevenendo l'introduzione di bias o anomalie strutturali nei dati.

Per chiarire ulteriormente la natura sequenziale di questi passaggi, si può ad esempio considerare che, a valle della decomposizione ICA, ciascuna componente viene "testata" individualmente attraverso il calcolo della kurtosi, fornendo così un'analisi puntuale del contributo informativo di ogni feature latente. Solo una valutazione aggregata positiva consente di dichiarare il dataset conforme dal punto di vista della well-formedness.

In sintesi, questa struttura non solo concretizza gli aspetti formali già introdotti nel Capitolo 3, ma pone anche le premesse operative per una verifica automatizzata delle proprietà di interesse, nel pieno rispetto dei criteri di accountability richiesti dalle linee guida europee ed internazionali.

Il *core* della funzione di verifica, dunque, è rappresentato da una sequenza deterministica di trasformazioni e confronti che, a partire dal dato grezzo, conducono a una valutazione oggettiva e documentabile della proprietà di interesse.

Il processo può essere schematizzato, in termini generali, come segue:

$$F_{\text{well-formedness}}(D) = \{\kappa_1, \ldots, \kappa_n\}$$

$$\text{verifica}_{\text{ica}}(K) = \begin{cases} \text{True} & \text{se } |\kappa_j| > 1 \ \forall \kappa_j \in K \\ \text{False} & \text{altrimenti} \end{cases}$$

$$\text{dove } K = F_{\text{well-formedness}}(D)$$

La logica sottostante resta compatibile con la definizione formale del capitolo 3, ma si arricchisce di concretezza e tracciabilità: ogni step, pur derivato da principi teorici, trova una corrispondenza esplicita in un'operazione eseguibile e validabile. In particolare, la modulazione della soglia di accettazione, la scelta della metrica (kurtosi) e l'ordine delle operazioni sono completamente trasparenti e replicabili, a beneficio di audit, debugging o estensioni successive.

Questa struttura, benché declinata nello specifico per la proprietà di well-formedness, trova analogia nei capitoli dedicati alle altre proprietà (fairness, explainability), ciascuna delle quali è riconducibile all'istanza di una funzione di verifica composta da metriche elementari, regole di aggregazione e soglie operative.

La sezione successiva illustra come tali passaggi trovino implementazione sistematica all'interno della *probe* dedicata alla well-formedness, articolando in dettaglio fasi, input e gestione delle eccezioni.

4.3 Implementazione in Probe

In questa sezione viene descritta l'implementazione di una funzione di assurance finalizzata alla valutazione della qualità strutturale dei dataset, mediante l'applicazione di tecniche di decomposizione statistica e misure di forma delle distribuzioni.

4.3.1 Comportamento Generale e Assunzioni

La `probe` si occupa di valutare la struttura informativa di un dataset, con l'obiettivo di misurarne l'indipendenza statistica tra le variabili. Tale valutazione si basa sull'applicazione dell'*Independent Component Analysis* (ICA), una tecnica di decomposizione lineare che presuppone la totale assenza di valori mancanti (NaN) nei dati.

Per questo motivo, prima di applicare l'ICA è necessario un preprocessing che include:

- l'esclusione delle colonne di etichetta (`label_columns`), specificate come input di configurazione;
- la verifica e la rimozione di eventuali valori NaN residui, in quanto incompatibili con il processo di decomposizione.

Gli input richiesti dalla probe sono organizzati in un dizionario JSON e comprendono:

- `target`: URL del repository GitLab o GitHub contenente il dataset;
- `repo_type`: tipo di repository (es. `gitlab`);
- `project`: nome del progetto o repository;
- `branch`: branch da cui recuperare l'artefatto;
- `artifact_path`: percorso relativo del file CSV da analizzare;
- `job_name` o **artifact_name**: informazioni specifiche per il recupero del file;
- `label_columns`: elenco delle colonne da escludere prima dell'analisi.

Un esempio di input è il seguente:

```
{
  "config": {
    "target": "https://repository.v2.moon-cloud.eu/",
    "repo_type": "gitlab",
    "project": "nbena/sample-ml-apps",
    "branch": "master",
    "artifact_path": "sample-ml/data/train.csv",
    "job_name": "split_dataset",
    "label_columns": ["label"]
  }
}
```

Listing 4.1: Configurazione di input per la probe

La probe opera scaricando l'artefatto indicato, caricando i dati, eseguendo il preprocessing, e successivamente applicando l'ICA per ottenere componenti indipendenti. I risultati vengono valutati tramite l'analisi della *kurtosi* di ogni componente, per fornire una misura sintetica della qualità del dataset in termini di struttura informativa.

4.3.2 Struttura della Probe

La logica della probe è organizzata in una catena di operazioni (forward chain) che includono il parsing dell'input, il caricamento e la preparazione del dataset, l'applicazione dell'ICA e la valutazione basata sulla kurtosi. Ogni fase è associata a specifiche funzioni Python.

Applicazione dell'Independent Component Analysis

Una volta eseguito il preprocessing, i dati vengono normalizzati ed elaborati tramite l'algoritmo di *FastICA*. Il processo è il seguente:

```python
def apply_ica(self, X, n_components=None):
    X_scaled = StandardScaler().fit_transform(X)
    ica = FastICA(n_components=n_components, max_iter=20000)
    X_transformed = ica.fit_transform(X_scaled)
    return X_transformed
```

Listing 4.2: Applicazione dell'ICA ai dati preprocessati

Valutazione tramite Analisi della Kurtosi

Il dataset trasformato viene valutato analizzando la *kurtosi* di ciascuna componente:

```python
def evaluate_dataset(self, X_transformed):
    kurtosis_vals = kurtosis(X_transformed, fisher=True, axis=0)
    if all(k > 1 or k < -1 for k in kurtosis_vals):
        self.result.integer_result = result.INTEGER_RESULT_TRUE
        self.result.pretty_result = "The dataset is considered
    good."
    else:
        self.result.integer_result = result.INTEGER_RESULT_FALSE
        self.result.pretty_result = "The dataset might not be
    optimal."
    self.result.put_raw_extra_data("kurtosis_details", {...})
```

Listing 4.3: Valutazione tramite analisi della kurtosi

4.3.3 Conclusioni sulla Implementazione

La probe realizzata consente una valutazione oggettiva della qualità strutturale dei dataset, integrandosi nei framework di MLOps e AI Assurance, migliorando la robustezza e l'affidabilità dei modelli basati su dati. Inoltre, la verifica automatizzata della *well-formedness* rappresenta un requisito fondamentale per garantire la conformità ai criteri di qualità dei dati richiesti dall'AI Act. L'integrazione di questi controlli in pipeline MLOps consente di migliorare la robustezza, la tracciabilità e l'affidabilità dei modelli, promuovendo un'adozione responsabile e normativa delle tecnologie basate su intelligenza artificiale.

4.4 Conclusioni

In questo capitolo è stata analizzata la proprietà di *well-formedness* dei dataset, evidenziando come la qualità strutturale dei dati rappresenti un prerequisito essenziale per garantire l'affidabilità e la robustezza dei modelli di intelligenza artificiale.

È stato mostrato come la verifica di questa proprietà possa essere formalizzata tramite una funzione di AI Assurance e implementata nella pratica attraverso l'uso combinato della *Independent Component Analysis* (ICA) e dell'analisi della

kurtosi. La capacità di rilevare anomalie strutturali, ridondanze informative o dipendenze latenti consente di prevenire la propagazione di difetti nei modelli addestrati, migliorandone la generalizzazione e riducendo il rischio di overfitting.

Dal punto di vista normativo, il rispetto della *well-formedness* si configura come un elemento chiave per la conformità ai requisiti di qualità dei dati stabiliti dall'AI Act, in particolare per i sistemi ad alto rischio.

L'integrazione automatizzata di questi controlli nelle pipeline di sviluppo e monitoraggio rappresenta un passo fondamentale verso una gestione più responsabile, tracciabile e sicura dei modelli di intelligenza artificiale.

Capitolo 5

Proprietà: Fairness

5.1 Contesto

La *fairness* nei sistemi di intelligenza artificiale è una proprietà centrale nell'ambito dell'AI Assurance, in particolare per quei modelli impiegati in contesti ad alto impatto sociale come la selezione del personale, il credito, la sanità o la giustizia predittiva. L'adozione di algoritmi in questi ambiti ha evidenziato il rischio concreto che decisioni automatizzate possano perpetuare o amplificare disuguaglianze esistenti, a causa della presenza di bias nei dati o nella modellazione.

In ambito tecnico, la *fairness* si riferisce alla capacità di un modello predittivo di non discriminare sistematicamente gruppi o individui sulla base di attributi sensibili, come genere, etnia, età o altri fattori protetti. Tuttavia, definire e misurare la *fairness* non è un processo univoco: esistono infatti diverse metriche, ognuna con implicazioni operative distinte. Tra le più note si annoverano la *demographic parity*, la *equalized odds* e il *disparate impact*, che valutano rispettivamente l'equilibrio delle predizioni, l'equità condizionata all'output corretto e l'effetto proporzionale su gruppi diversi.

Dal punto di vista regolatorio, l'**AI Act** dell'Unione Europea considera la *fairness* un requisito fondamentale per i sistemi di intelligenza artificiale ad alto rischio, richiedendo che essi siano progettati per evitare distorsioni ingiustificate e garantire trattamenti non discriminatori [5]. Tale esigenza è condivisa anche da altri framework di governance internazionali, come le linee guida dell'OECD e i principi di trasparenza e responsabilità promossi a livello europeo e globale.

In un framework di assurance automatizzato, la valutazione della *fairness* non può essere limitata a un'analisi statica ex-ante, ma richiede controlli continui sulle predizioni del modello in fase di addestramento, validazione e deployment. La natura multidimensionale della *fairness* impone un'analisi teorica approfondita, che tenga conto sia delle diverse formulazioni proposte in letteratura, sia delle implicazioni pratiche delle metriche adottate. La valutazione dell'equità algoritmica richiede infatti l'adozione di criteri rigorosi e coerenti, capaci di bilanciare esigenze di accuratezza, non discriminazione e affidabilità operativa.

5.2 Descrizione Teorica

5.3 Definizione Generale

La **fairness** nei sistemi di intelligenza artificiale è intesa come la capacità del modello di non produrre, durante le proprie decisioni, disparità sistematiche tra gruppi individuati da attributi sensibili quali il genere, l'etnia o altri fattori protetti. L'analisi di questa proprietà si inserisce in un quadro quantitativo, dove la conformità rispetto ai principi di equità viene valutata tramite indicatori misurabili e confrontabili con soglie operative.

Da un punto di vista generale, la funzione di verifica associata alla fairness opera confrontando le probabilità condizionate di un determinato esito tra tutti i gruppi presenti nella variabile sensibile considerata. Se indichiamo con A una variabile sensibile e con $\hat{Y}$ la predizione restituita dal modello, è possibile definire una misura di disparità come segue:

$$\Delta(A) = \max_{i,j} \left| P(\hat{Y} = 1 \mid A = i) - P(\hat{Y} = 1 \mid A = j) \right|$$

dove la quantità $\Delta(A)$ rappresenta la massima differenza di probabilità di esito positivo osservabile tra i diversi gruppi.

La funzione di verifica stabilisce poi la conformità rispetto a una soglia δ, restituendo un valore booleano di accettazione o rigetto:

$$f_{\text{fairness}}(D) = \Delta(A)$$

$$\text{verifica}_{\text{fairness}}(D) = \begin{cases} \text{True} & \text{se } f_{\text{fairness}}(D) \leq \delta \\ \text{False} & \text{altrimenti} \end{cases}$$

Questa impostazione facilita un'analisi oggettiva ed automatizzabile della fairness, consentendo di integrare la valutazione della proprietà all'interno di un framework di assurance strutturato, orientato sia alla trasparenza che alla riproducibilità delle verifiche.

5.3.1 Fairness individuale e fairness di gruppo

Nel campo della valutazione della *fairness* nei modelli di intelligenza artificiale, si distinguono due approcci principali: la *fairness individuale* e la *fairness di gruppo*. Queste due categorie concettuali riflettono modalità differenti, e talvolta complementari, di interpretare e formalizzare il concetto di non discriminazione.

La fairness individuale si basa sull'assunto che soggetti simili dovrebbero ricevere trattamenti simili. Tale principio implica la presenza di una funzione di similarità che permetta di confrontare due individui sulla base di attributi rilevanti per il compito decisionale. Se due soggetti sono considerati comparabili rispetto a tale funzione, allora le predizioni del modello nei loro confronti dovrebbero essere coerenti.

Dal punto di vista formale, per ogni coppia di individui $x_1, x_2 \in \mathcal{X}$, se la distanza $d(x_1, x_2)$ è minore di una soglia δ, allora le rispettive predizioni devono essere simili entro un margine ϵ:

$$d(x_1, x_2) \leq \delta \quad \Rightarrow \quad |f(x_1) - f(x_2)| \leq \epsilon$$

dove $d(\cdot, \cdot)$ rappresenta una metrica di similarità e f la funzione predittiva del modello.

La fairness di gruppo, invece, valuta il comportamento del modello rispetto a sottopopolazioni identificate da attributi sensibili. L'obiettivo è garantire che certe statistiche aggregate — come tassi di predizione positiva, tassi di errore o precisione — siano comparabili tra gruppi distinti.

In questo caso, la valutazione non si fonda sulla similarità tra individui, ma sulla distribuzione degli outcome a livello collettivo. Un esempio comune è la *demographic parity*, che richiede che la probabilità di assegnare un output positivo sia indipendente dall'appartenenza a un certo gruppo:

$$P(\hat{Y} = 1 \mid A = 0) = P(\hat{Y} = 1 \mid A = 1)$$

dove A è la variabile sensibile e $\hat{Y}$ l'output del modello.

Sebbene entrambi gli approcci mirino a garantire la non discriminazione, la loro applicabilità e le implicazioni che ne derivano possono essere molto diverse. La fairness individuale presuppone la disponibilità di una nozione affidabile di similarità tra soggetti, spesso difficile da definire in modo oggettivo. La fairness di gruppo, invece, è più facilmente applicabile in contesti istituzionali o regolamentati, ma può tralasciare casi di ingiustizia su scala individuale.

Questi due concetti rappresentano le fondamenta su cui si basano le metriche di fairness più diffuse. La loro distinzione è cruciale per la progettazione di funzioni di assurance che intendano valutare la proprietà in modo conforme ai vincoli applicativi, normativi o etici del dominio considerato.

5.3.2 Metriche di fairness

La valutazione della *fairness* nei modelli di intelligenza artificiale si basa su un insieme di metriche quantitative che consentono di esprimere il grado di equità osservato tra gruppi o individui. Queste metriche derivano da formulazioni statistiche e sono tipicamente confrontate con soglie predefinite per determinare se un sistema può essere considerato equo in un contesto specifico.

Di seguito sono presentate le metriche più comuni e rilevanti dal punto di vista teorico e regolatorio.

Demographic Parity

La *demographic parity*, anche nota come *statistical parity*, è una metrica che valuta la fairness a livello di gruppo. Essa richiede che la decisione automatica del modello non dipenda dalla variabile sensibile, come il genere, l'etnia o l'età. Più precisamente, il tasso di assegnazione di un certo output, ad esempio una predizione positiva, dovrebbe essere identico tra i diversi gruppi. In altre parole, tutti dovrebbero avere le stesse probabilità di ottenere un determinato risultato, indipendentemente dall'appartenenza a un gruppo protetto o meno.

Tale criterio è spesso utilizzato in ambiti in cui si vuole garantire l'uguaglianza di opportunità iniziale, come nei processi di selezione per colloqui di lavoro. Tuttavia, non tiene conto delle caratteristiche individuali che possono giustificare predizioni diverse. Ad esempio, se un algoritmo di screening CV seleziona il 60% dei candidati uomini e il 40% delle donne, soddisfa la demographic parity, ma potrebbe comunque discriminare all'interno dei gruppi se non usa criteri coerenti. Uno studio recente ha evidenziato che vincolare un modello al rispetto della demographic parity può introdurre bias induttivi indesiderati, compromettendo la generalizzazione su dati reali non bilanciati [6].

Equalized Odds

La metrica nota come *equalized odds* si concentra sul bilanciamento degli errori predittivi tra gruppi. Essa richiede che, a parità di condizione reale (ad esempio, l'effettiva presenza o assenza di una malattia), la probabilità di una predizione corretta (o errata) sia la stessa per ogni gruppo.

Questo criterio si applica soprattutto in contesti sensibili, dove le conseguenze degli errori possono essere gravi, come nella medicina o nella giustizia predittiva. Ad esempio, se un algoritmo diagnostico ha un tasso di falsi negativi molto più alto per un determinato gruppo etnico, viola la equalized odds, anche se le predizioni totali sembrano bilanciate. In letteratura è stato osservato che il rispetto della equalized odds non è sempre garantibile tramite modelli deterministici. Studi recenti evidenziano come approcci stocastici, in cui le predizioni sono espresse in termini probabilistici, possano facilitare il bilanciamento degli errori tra gruppi e aumentare la robustezza complessiva del sistema in contesti applicativi complessi [7].

Predictive Parity

La *predictive parity* si focalizza sulla qualità delle predizioni effettuate. Essa richiede che, quando il modello prevede un risultato positivo, la probabilità che questo sia effettivamente corretto sia la stessa per tutti i gruppi. In altre parole, si chiede che il valore predittivo positivo sia equamente distribuito.

Questa metrica è particolarmente rilevante quando l'output del modello viene utilizzato come base per decisioni che comportano costi o benefici reali, come la

concessione di un prestito o l'accesso a un programma sanitario. Ad esempio, la predictive parity assicura che la percentuale di soggetti realmente affidabili tra coloro che ricevono una predizione positiva (es. approvazione di credito) sia bilanciata tra gruppi sensibili come genere o etnia.

Recenti analisi teoriche hanno dimostrato che, in presenza di eterogeneità nei tassi di base tra i gruppi, l'adozione isolata della predictive parity può generare effetti collaterali indesiderati. In particolare, l'allineamento del valore predittivo positivo può implicare soglie decisionali differenti per ciascun gruppo, con il rischio di compromettere la coerenza interna del sistema decisionale [8].

Disparate Impact

Il concetto di *disparate impact* si riferisce alla presenza di un effetto disparato, ovvero a un impatto significativamente diverso delle decisioni automatizzate su gruppi distinti. A differenza delle metriche precedenti, si tratta di una misura spesso utilizzata in ambito legale e normativo per identificare discriminazioni indirette, anche in assenza di intento discriminatorio.

L'idea centrale è che, se un gruppo riceve sistematicamente un trattamento meno favorevole rispetto a un altro, anche quando il sistema non utilizza esplicitamente attributi sensibili, si configura un impatto disparato. Un esempio pratico: se un sistema di selezione universitaria ammette il 70% dei candidati di un gruppo e solo il 40% di un altro, pur applicando criteri apparentemente neutri, si può sospettare un effetto discriminatorio implicito.

In ambito legale, una delle formulazioni più diffuse è quella della *disparate impact ratio*, definita come:

$$\text{DIR} = \frac{P(\hat{Y} = 1 \mid A = a)}{P(\hat{Y} = 1 \mid A = b)}$$

dove A rappresenta la variabile sensibile e $\hat{Y} = 1$ l'output favorevole. Secondo le linee guida dell'**Equal Employment Opportunity Commission (EEOC)**, un valore di *DIR* inferiore a 0.8 (la cosiddetta *regola dell'80%*) è considerato indicativo di un potenziale effetto disparato [9].

Studi recenti hanno proposto una generalizzazione della metrica tradizionale di disparate impact, al fine di renderla applicabile anche in presenza di attributi sensibili continui. Questa estensione migliora la flessibilità e la precisione delle analisi di fairness, consentendo di modellare dipendenze più complesse tra output predittivi e caratteristiche sensibili in scenari reali [10].

Confronto sintetico tra metriche

Per facilitare la scelta operativa della metrica più adeguata in relazione al contesto applicativo, si propone nella Tabella 5.1 un confronto sintetico tra le principali metriche di fairness trattate, con riferimento alla loro formulazione matematica, punti di forza e limiti noti.

Tabella 5.1: Confronto tra le principali metriche di fairness

Metrica	Definizione	Punti di Forza	Criticità
Demographic Parity	$P(\hat{Y} = 1 \mid A = a_i) = P(\hat{Y} = 1 \mid A = a_j)$	Semplice da calcolare, applicabile anche senza ground truth, adatta a contesti ex-post	Può penalizzare l'accuratezza, ignora le differenze legittime tra gruppi
Equalized Odds	$P(\hat{Y} = y \mid Y = y, A = a_i) = P(\hat{Y} = y \mid Y = y, A = a_j)$	Bilancia gli errori tra gruppi, adatta a contesti ad alto rischio	Richiede ground truth; difficile da soddisfare insieme ad altre metriche
Predictive Parity	$P(Y = 1 \mid \hat{Y} = 1, A = a_i) = P(Y = 1 \mid \hat{Y} = 1, A = a_j)$	Utile per decisioni operative (credito, sanità), misura la precisione equa	Richiede distribuzioni di base simili; può portare a soglie diverse tra gruppi
Disparate Impact	$\text{DIR} = \frac{P(\hat{Y}=1\|A=a)}{P(\hat{Y}=1\|A=b)}$	Standard normativo; efficace per auditing e compliance	Non valuta la correttezza; soggetta a overflagging in dati squilibrati

5.3.3 Tensioni tra metriche e impossibility theorem

L'impiego di metriche di fairness nei sistemi di intelligenza artificiale è soggetto a vincoli teorici ben documentati. Diverse definizioni formali di fairness — come *demographic parity*, *equalized odds* e *predictive parity* — risultano spesso incompatibili tra loro, specialmente in presenza di differenze nei tassi di prevalenza tra gruppi.

Questo fenomeno è stato formalizzato nel cosiddetto *teorema di impossibilità*, che evidenzia l'incompatibilità strutturale tra equità predittiva, accuratezza e distribuzioni di base disomogenee. In tali condizioni, l'ottimizzazione di una metrica comporta inevitabilmente il sacrificio parziale o completo di un'altra.

Tuttavia, ricerche empiriche condotte su ampi dataset pubblici hanno mostrato che, in contesti applicativi concreti, è spesso possibile ottenere il rispetto simultaneo — entro soglie di tolleranza accettabili — di più metriche di fairness. Un'analisi sistematica ha evidenziato come sia possibile raggiungere una parità approssimata dei tassi di falsi positivi, falsi negativi e valori predittivi positivi, anche in presenza di moderate disuguaglianze nei tassi di base tra gruppi. Questi risultati suggeriscono un approccio pragmatico alla fairness, orientato all'ottimizzazione congiunta piuttosto che al rispetto assoluto dei vincoli teorici [11].

Questo orientamento implica che le limitazioni teoriche evidenziate dal teorema non escludono l'adozione pratica di soluzioni efficaci. Al contrario, richiedono strategie di ottimizzazione multi-obiettivo, definizione di soglie operative flessibili

e contestualizzazione delle metriche in base ai requisiti normativi e funzionali del sistema.

5.3.4 Strategie pratiche per il bilanciamento della fairness

L'adozione di metriche di fairness in sistemi di intelligenza artificiale introduce la necessità di definire strategie operative per il loro bilanciamento. Tali strategie non mirano necessariamente al raggiungimento di una parità perfetta, bensì all'identificazione di soglie e tecniche che consentano di contenere la disparità entro limiti accettabili e documentabili.

Una delle pratiche più diffuse consiste nel calcolo della disparità relativa tra gruppi definiti da variabili sensibili, come genere, etnia o età. In particolare, si valuta la differenza nei tassi di output positivi assegnati a ciascun gruppo. Questa forma semplificata di *statistical parity* è spesso adottata nei contesti applicativi per la sua facilità di calcolo e per la compatibilità con pipeline automatizzabili.

Per valutare la fairness in maniera sistematica, è comune definire una soglia operativa, ad esempio 0.1, oltre la quale si considera che la disparità osservata sia potenzialmente discriminatoria. L'uso di soglie consente di introdurre un margine di tolleranza che tiene conto della variabilità statistica e della complessità dei dati reali. In ambienti industriali, questa soglia può essere configurabile e adattata al contesto normativo o settoriale.

Un ulteriore vantaggio di tali strategie risiede nella possibilità di integrare il controllo della fairness in fase di valutazione pre-deployment, come parte di un framework di assurance automatizzabile. In questo contesto, la valutazione produce un risultato binario (fair/unfair) accompagnato da informazioni di supporto, come la massima disparità osservata e la descrizione dei gruppi coinvolti.

Queste pratiche riflettono un approccio pragmatico alla fairness, che bilancia l'aderenza ai principi teorici con l'esigenza di tracciabilità, spiegabilità e integrazione operativa. La misurazione della disparità osservata rispetto a gruppi sensibili, basata su output predetti, costituisce pertanto una base solida per sviluppare meccanismi di verifica ripetibili e oggettivi all'interno di pipeline MLOps e sistemi regolati da norme di responsabilità algoritmica.

5.3.5 Considerazioni conclusive sull'integrazione della fairness

La fairness nei sistemi di intelligenza artificiale non può essere considerata una proprietà assoluta, ma deve essere trattata come un requisito operativo da misurare, documentare e ottimizzare all'interno di vincoli tecnici, normativi e applicativi. Le metriche formali offrono strumenti quantitativi per l'analisi della disparità, ma pongono inevitabilmente delle sfide concettuali, specialmente nei contesti reali caratterizzati da diseguaglianze strutturali e distribuzioni squilibrate.

La presenza di tensioni tra le diverse metriche, come evidenziato dai teoremi di impossibilità, richiede un approccio flessibile e contestuale, in cui la scelta

della metrica da ottimizzare deve essere giustificata in relazione agli obiettivi del sistema e ai rischi associati. A tal fine, l'integrazione della fairness nei processi di assurance comporta l'adozione di soglie operative, l'automatizzazione delle verifiche e la tracciabilità dei risultati.

Dal punto di vista della formalizzazione, la fairness può essere ricondotta al paradigma delle funzioni di assurance, in cui un modello M, una variabile sensibile A e una funzione di output $\hat{Y}$ danno origine a un insieme di metriche $\{f_i(M, A, \hat{Y})\}$, confrontate con soglie definite a priori. Il risultato dell'assessment è dunque binario o continuo, e può essere usato per guidare decisioni di rilascio, audit o miglioramento.

Questa prospettiva consente di trattare la fairness come una proprietà formalizzabile, misurabile e verificabile, alla pari di altre dimensioni di assurance come la robustezza o l'accuratezza. Il capitolo successivo sarà dedicato all'implementazione concreta di una funzione di questo tipo, in grado di valutare automaticamente la fairness di un dataset o di un modello, integrandosi in pipeline operative e strumenti di verifica automatica.

5.3.6 Fairness nelle Regolamentazioni Europee

Nel contesto normativo europeo, la *fairness* è esplicitamente riconosciuta come un requisito fondamentale per i sistemi di intelligenza artificiale ad alto rischio. Il Regolamento AI Act, proposto dalla Commissione Europea e recentemente approvato dal Parlamento, impone una serie di vincoli che mirano a prevenire effetti discriminatori sistematici derivanti da algoritmi.

In particolare, l'**Articolo 10** del regolamento stabilisce che i dati utilizzati per addestrare modelli ad alto rischio debbano essere "*rilevanti, rappresentativi, privi di errori e completi*", specificando che devono essere adeguatamente controllati per evitare "*bias e distorsioni che possano portare a risultati discriminatori*" [12].

In aggiunta, il **Recital 44** sottolinea la necessità di adottare misure tecniche e organizzative per prevenire rischi di esclusione o trattamento iniquo nei confronti di gruppi vulnerabili, mentre il **Recital 46** introduce il concetto di *data governance* orientata alla non discriminazione e alla robustezza dei modelli.

Dal punto di vista pratico, queste disposizioni normano indirettamente l'adozione di metriche di fairness come parte integrante dei controlli di qualità. I framework di AI Assurance, inclusi quelli descritti nei capitoli precedenti, rappresentano strumenti coerenti con questa visione regolatoria, permettendo di integrare verifiche automatizzate della fairness all'interno delle pipeline di sviluppo e auditing dei sistemi AI.

L'approccio europeo si distingue così per un orientamento *ex ante*, che impone il controllo della fairness già in fase di progettazione e validazione, in contrasto con approcci più reattivi presenti in altri ordinamenti. In questo contesto, l'adozione di probe basate su metriche come la *demographic parity* risponde all'esigenza di valutare e documentare il rispetto dei requisiti normativi in modo trasparente, tracciabile e ripetibile.

5.3.7 Caso di Studio: Il Caso COMPAS

Uno dei casi più emblematici e discussi in ambito di fairness algoritmica è rappresentato dall'utilizzo del sistema COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) negli Stati Uniti. Sviluppato dalla società Northpointe (ora Equivant), il sistema era utilizzato da giudici e operatori della giustizia penale per valutare il rischio di recidiva degli imputati, al fine di supportare decisioni su cauzione, sentenze e rilascio.

Nel 2016, un'indagine condotta da *ProPublica* ha evidenziato forti segnali di disparità razziale nei risultati forniti da COMPAS [13]. In particolare, è stato osservato che il modello:

- Sovrastimava il rischio di recidiva per imputati afroamericani, classificandoli erroneamente come ad alto rischio in misura maggiore rispetto ai bianchi;
- Sottostimava il rischio di recidiva per imputati bianchi, classificandoli erroneamente come a basso rischio più frequentemente rispetto agli afroamericani.

Tali distorsioni sono state analizzate secondo diverse metriche di fairness. Il modello violava in modo sistematico:

- La *equalized odds*, in quanto i tassi di falsi positivi e falsi negativi risultavano significativamente diversi tra gruppi razziali;
- La *predictive parity*, poiché la probabilità che una previsione positiva (alto rischio) fosse corretta differiva marcatamente tra gruppi.

Dal punto di vista delle implicazioni pratiche, questo caso ha mostrato come un sistema predittivo, anche privo di attributi sensibili espliciti come la razza, possa produrre effetti discriminatori indiretti attraverso correlazioni con variabili proxy. Il caso COMPAS è diventato un riferimento centrale nel dibattito sulla necessità di valutare e garantire la fairness dei modelli di intelligenza artificiale, specialmente in ambiti ad alto impatto sociale come la giustizia.

Questo esempio evidenzia l'importanza di un approccio rigoroso alla valutazione della fairness, che includa analisi quantitative basate su metriche formalizzate, audit indipendenti, e trasparenza sui criteri decisionali automatizzati. La sua rilevanza continua a essere citata in letteratura come uno snodo cruciale per la nascita della disciplina della *algorithmic fairness*.

5.3.8 Dalla Fairness Teorica alla Verifica Operativa

Il percorso che conduce dalla definizione matematica della fairness alla sua concreta applicazione nella piattaforma di verifica automatizzata si realizza attraverso un'integrazione naturale fra teoria e pratica. All'interno della pipeline sviluppata, le metriche introdotte nei paragrafi precedenti trovano una manifestazione operativa che permette di valutare, in modo oggettivo e ripetibile, il rispetto dei principi di equità all'interno dei dati e delle predizioni.

Questa evoluzione si traduce in un insieme di passaggi metodici in cui le proprietà teoriche vengono progressivamente calate nel contesto d'uso. Ogni attributo sensibile considerato viene analizzato attraverso il calcolo empirico delle distribuzioni di output, seguendo la logica sottostante la metrica scelta. In tal modo, il criterio della fairness diventa parte integrante della sequenza procedurale, guidando la valutazione automatica e conferendo coerenza al processo di assurance.

Il risultato è una funzione di verifica che, pur radicata nel rigore del formalismo iniziale, si presenta come uno strumento operativo adattabile ai diversi scenari applicativi e pienamente tracciabile. La transizione dal principio astratto alla procedura concreta costituisce così il nucleo della probe, permettendo di coniugare esigenze normative e affidabilità tecnica.

La sezione che segue illustra in dettaglio le soluzioni implementative adottate, chiarendo come le regole quantitative enunciate finora vengano rese effettive nella probe definita.

5.4 Implementazione in Probe

In questa sezione viene descritta l'implementazione di una funzione di assurance finalizzata alla valutazione della fairness nei dataset, con particolare riferimento alla *demographic parity*.

5.4.1 Comportamento Generale e Assunzioni

La `probe` si occupa di valutare la presenza di disparità nei tassi di assegnazione dell'output predittivo tra i diversi gruppi definiti da una o più variabili sensibili. L'obiettivo è quello di determinare se l'assegnazione dell'etichetta positiva da parte del modello sia indipendente dalla variabile sensibile, come previsto dalla *demographic parity*.

A livello concettuale, il controllo si traduce nella stima delle probabilità condizionate di output positivo per ciascun gruppo e nella loro successiva comparazione. La misura della distanza massima tra tali valori costituisce l'indicatore di disparità.

La funzione di assurance assume come input:

- la colonna target, contenente le predizioni effettuate dal modello;
- l'elenco delle colonne sensibili, rispetto alle quali si intende valutare la parità;
- la soglia di tolleranza sulla differenza accettabile tra i gruppi.

Una disparità inferiore alla soglia configurata consente di dichiarare il dataset *fair*, altrimenti viene sollevata una segnalazione automatica.

Gli input richiesti sono specificati all'interno di un dizionario JSON come segue:

```
1 {
2   "config": {
```

```
    "target": "https://repository.v2.moon-cloud.eu/",
    "repo_type": "gitlab",
    "project": "sample-ml-apps",
    "branch": "master",
    "artifact_path": "sample-ml/data/predictions.csv",
    "job_name": "predict_model",
    "target_column": "prediction",
    "target_value": 1,
    "sensitive_columns": ["gender", "ethnicity"]
  }
}
```

Listing 5.1: Esempio di configurazione JSON per la probe

5.4.2 Applicazione del Principio di Demographic Parity

Nel contesto della probe di fairness, viene adottato il criterio della **demographic parity**, uno degli approcci più utilizzati per valutare la presenza di bias sistematici nelle predizioni algoritmiche rispetto a variabili sensibili come genere o etnia. Questo criterio si fonda sul confronto delle probabilità condizionate di assegnazione di un determinato esito (ad esempio, la predizione di recidiva) tra i diversi gruppi individuati dalla variabile sensibile.

Matematicamente, la *demographic parity* richiede che la proporzione di outcome positivi sia la stessa per ciascun gruppo, secondo la regola:

$$\max_{a_i, a_j} \left| P(\hat{Y} = 1 \mid A = a_i) - P(\hat{Y} = 1 \mid A = a_j) \right| \leq \delta$$

dove A rappresenta la variabile sensibile (ad esempio, `race` o `sex`), $\hat{Y}$ la predizione binaria prodotta dal modello e δ una soglia di accettabilità (tipicamente 0.1). Il termine a sinistra misura la massima differenza osservabile tra tutte le coppie di gruppi; valori elevati evidenziano l'assenza di equità statistica.

Operativamente, la verifica di questa proprietà viene implementata attraverso il calcolo della proporzione di esiti positivi per ciascun gruppo sensibile e la valutazione automatica della disparità osservata rispetto al vincolo di fairness. Di seguito si riporta uno snippet esemplificativo in Python, che riprende e generalizza la logica adottata nella pipeline di auditing del dataset COMPAS:

```
def statistical_parity_check(df, sensitive_column, target_column,
    target_value):
    proportions = (
        df[df[target_column] == target_value]
        .groupby(sensitive_column)
        .size() / df.groupby(sensitive_column).size()
    ).fillna(0)
    disparities = proportions.to_dict()
    max_disparity = max(disparities.values()) - min(disparities.
   values())
    threshold = 0.1
```

```
    fairness = max_disparity <= threshold
    return disparities, fairness
```

Listing 5.2: Verifica Statistical Parity su dataset

La funzione `statistical_parity_check` consente di calcolare, dato un dataset, le proporzioni di outcome positivi per ciascun valore della variabile sensibile e di valutarne la massima disparità rispetto a una soglia prefissata. L'output restituisce sia i valori calcolati sia un giudizio booleano (`True`/`False`) sulla fairness.

Questa procedura viene richiamata, all'interno dell'infrastruttura della probe, per ognuna delle colonne sensibili specificate dall'utente. La pipeline permette anche la visualizzazione immediata delle disparità tramite opportune funzioni di plotting (`matplotlib`), offrendo così trasparenza, auditabilità e facilità di comunicazione dei risultati:

```
def plot_disparities(disparities, sensitive_column):
    plt.bar(disparities.keys(), disparities.values())
    plt.title(f"Disparit  per {sensitive_column}")
    plt.xlabel("Gruppo")
    plt.ylabel("Proporzione di outcome positivi")
    plt.show()
```

Listing 5.3: Esempio di visualizzazione delle disparità

Dal punto di vista metodologico, il criterio di demographic parity rimane semplice da implementare e di immediata interpretabilità, sebbene sia noto che può trascurare elementi di eterogeneità legittima nei dati e non consideri la correttezza predittiva individuale. La scelta e la discussione di tale criterio, nonché dei suoi limiti applicativi, rappresentano parte integrante del percorso di validazione e auditing proposto dalla piattaforma.

5.4.3 Calcolo della Disparità e Valutazione della Fairness

Il nucleo computazionale della `probe` è rappresentato dal metodo `analyze_fairness`, che valuta il rispetto del criterio di *demographic parity* confrontando la frequenza relativa di output positivi tra gruppi definiti dalla variabile sensibile.

Il metodo esegue un filtraggio delle osservazioni in cui la colonna di predizione (definita come `target_column`) assume un determinato valore positivo (`target_value`, tipicamente 1), e ne calcola la proporzione rispetto al totale delle osservazioni in ciascun gruppo sensibile. Questa frazione rappresenta una stima empirica della probabilità condizionata $P(\hat{Y} = 1 \mid A = a_i)$.

```
def analyze_fairness(self):
    for sensitive_column in self.sensitive_columns:
        proportions = (
            self.dataset[self.dataset[self.target_column] == self
    .target_value]
            .groupby(sensitive_column).size() /
            self.dataset.groupby(sensitive_column).size()
```

```
        ).fillna(0)

        max_disparity = max(proportions.values) - min(proportions
    .values)
        fairness = max_disparity <= 0.1
        ...
```

Listing 5.4: Valutazione della fairness tramite demographic parity

La logica di valutazione si basa sul confronto tra i valori massimi e minimi della proporzione condizionata nei diversi gruppi. Il valore di `max_disparity` viene poi confrontato con una soglia di accettabilità fissata ($\delta = 0.1$) per stabilire un giudizio binario di fairness. Tale soglia può essere personalizzata in fase di configurazione, permettendo adattamenti alle esigenze normative o settoriali specifiche.

Il risultato dell'analisi viene organizzato in un dizionario strutturato per ciascun attributo sensibile, contenente:

- le proporzioni di predizioni positive osservate in ciascun gruppo (`disparities`);
- la disparità massima calcolata (`max_disparity`);
- il verdetto di conformità alla fairness (`fairness`: `True` o `False`).

Questo approccio presenta due vantaggi rilevanti: da un lato, consente una valutazione interpretabile e immediata; dall'altro, si presta ad essere automatizzato in contesti MLOps per il controllo sistematico della fairness nei modelli già addestrati.

5.4.4 Conclusione sulla Implementazione

La `FairnessProbe` descritta implementa un controllo automatico della fairness secondo la metrica della *demographic parity*, producendo un risultato strutturato e facilmente integrabile in pipeline di verifica. L'approccio, pur semplice, fornisce un indicatore chiaro e documentabile del rispetto del principio di equità tra gruppi, facilitando l'adozione di AI responsabili e conformi agli standard regolatori.

5.5 Conclusioni

Il presente capitolo ha fornito un'analisi completa della proprietà di fairness nei sistemi di intelligenza artificiale, illustrandone le principali metriche, le tensioni teoriche e le implicazioni normative. È stato mostrato come la fairness possa essere trattata in modo formale e misurabile, integrandosi in pipeline automatizzate di assurance. L'implementazione pratica di una `FairnessProbe`, basata sulla *demographic parity*, costituisce un primo passo verso il controllo sistematico della non discriminazione nei modelli AI. Nei capitoli successivi verranno approfondite ulteriori proprietà rilevanti per l'affidabilità e la trasparenza dell'intelligenza artificiale.

Capitolo 6

Proprietà: Explainability

6.1 Contesto

La *explainability* rappresenta una proprietà cruciale nei sistemi di intelligenza artificiale, in particolare per i modelli complessi e opachi (*black-box*) come le reti neurali profonde. In tali contesti, la capacità di comprendere e giustificare le decisioni del modello assume un ruolo fondamentale per garantire trasparenza, fiducia e conformità normativa.

Negli ultimi anni, la crescente adozione di modelli di grandi dimensioni, come i *Large Language Models* (LLM) basati su architetture Transformer, ha ulteriormente amplificato la necessità di tecniche di spiegazione affidabili. Questi modelli, pur ottenendo prestazioni impressionanti in compiti complessi come la generazione di testo, il question answering o il completamento semantico, operano attraverso miliardi di parametri distribuiti su reti profonde e non lineari, rendendo estremamente difficile per l'utente umano comprendere il razionale delle risposte fornite.

In ambito applicativo, l'assenza di spiegazioni può determinare rischi rilevanti: nei settori regolati come la sanità, la giustizia, o la finanza, le predizioni di un modello devono poter essere auditabili, specialmente se influiscono su decisioni che impattano i diritti fondamentali dell'individuo. l'assenza di trasparenza può minare la fiducia degli utenti e ostacolare l'adozione responsabile dell'IA.

Dal punto di vista normativo, il Regolamento Europeo sull'Intelligenza Artificiale (AI Act) introduce obblighi stringenti in materia di interpretabilità per i sistemi impiegati in contesti ad alto impatto. In particolare, viene richiesto che tali sistemi siano in grado di fornire spiegazioni comprensibili riguardo al proprio funzionamento, alla logica sottostante e al ruolo svolto nei processi decisionali automatizzati. Questi requisiti si applicano in modo prioritario ai modelli utilizzati in ambiti sensibili, come la sanità, la giustizia, il credito, l'istruzione, la gestione delle risorse umane e i sistemi di sorveglianza.

Inoltre, tali obblighi possono estendersi anche a modelli generativi, come i Large Language Models, qualora siano impiegati in contesti sensibili che ricadono nelle classi di rischio sopracitate. La presenza di meccanismi di spiegazione

verificabili diventa quindi essenziale per garantire la conformità, soprattutto in fase di valutazione ex ante e sorveglianza post-market.

In questo scenario, l'explainability si configura non solo come una proprietà tecnica desiderabile, ma come un requisito imprescindibile sotto il profilo etico, legale e tecnico per l'adozione su larga scala di sistemi basati su AI. Le tecniche di interpretazione automatica — tra cui SHAP, LIME, Attention Visualization, Concept Attribution e saliency maps — rappresentano strumenti fondamentali per facilitare la trasparenza nei confronti di utenti finali, sviluppatori, regolatori e auditor.

Tuttavia, la spiegabilità nei modelli di grandi dimensioni introduce nuove sfide: la complessità computazionale delle spiegazioni, il rischio di bias nelle interpretazioni, la difficoltà di valutare l'affidabilità delle spiegazioni stesse, e la necessità di mantenere un equilibrio tra precisione predittiva e comprensibilità del modello. Pertanto, è sempre più rilevante l'integrazione di probe automatizzate di explainability all'interno di pipeline MLOps e strumenti di assurance continua, in grado di fornire evidenze strutturate e verificabili anche per modelli di tipo black-box o generativo.

6.1.1 Definizione generale di Explainability

Nel contesto dei sistemi di apprendimento automatico, la *explainability* (o spiegabilità) è definita come la capacità di un modello di rendere accessibili, interpretabili e verificabili i razionali alla base delle proprie decisioni. Essa rappresenta una proprietà non funzionale cruciale per i sistemi ad alto impatto, in quanto consente di esporre la logica sottostante alle predizioni, mitigando l'opacità computazionale tipica dei modelli opachi (*black-box*).

L'explainability può essere formalmente modellata come una funzione ausiliaria $E : X \times Y \times M \to \mathcal{S}$, dove:

- X è lo spazio degli input,
- Y lo spazio degli output del modello M,
- $\mathcal{S}$ rappresenta lo spazio delle spiegazioni strutturate,
- e $E(x, M(x), M)$ produce una rappresentazione semantica coerente dell'inferenza eseguita su x da parte del modello M.

Tale funzione E può essere progettata per soddisfare differenti obiettivi informativi, a seconda del contesto applicativo, dell'utente di destinazione e delle proprietà del modello.

Nella letteratura si distinguono due principali categorie di spiegazioni:

- **Global Explainability**: riguarda l'analisi del comportamento complessivo del modello M sull'intero dominio o su sottoinsiemi significativi di X. Tali spiegazioni sono utilizzate per derivare metriche aggregabili (es. feature importance, sensibilità alle feature) e per comprendere le regole decisionali latenti nel modello.

- **Local Explainability**: si concentra sull'interpretazione puntuale della decisione associata a un singolo input $x \in X$. L'obiettivo è costruire una spiegazione specifica per la predizione $M(x)$, mantenendo coerenza con il comportamento locale del modello in un intorno definito di x.

Dal punto di vista computazionale, l'explainability può essere ottenuta tramite:

- metodi *model-specific*, validi solo per particolari classi di modelli (es. tree-path explanations per alberi decisionali);
- metodi *model-agnostic*, che trattano il modello come una funzione black-box, interrogandolo in modo controllato per derivarne spiegazioni induttive.

L'importanza dell'explainability è riconosciuta anche a livello normativo. In particolare, l'AI Act dell'Unione Europea richiede che i sistemi ad alto rischio garantiscano trasparenza decisionale del modello e tracciabilità delle inferenze. L'integrazione di tecniche spiegabili in fase di design e verifica risponde pertanto a esigenze sia tecniche sia regolatorie.

In un framework di assurance, l'explainability si configura come una proprietà valutabile tramite funzioni formali, misurabile con appositi indicatori di coerenza, robustezza e fedeltà rispetto al comportamento effettivo del modello.

Un aspetto concettuale particolarmente rilevante per la comprensione dell'explainability riguarda la distinzione tra *explanation* e *justification*. La prima si riferisce alla descrizione tecnica e verificabile del comportamento computazionale del modello, attraverso strumenti che esplicitano le dipendenze tra input e output. La seconda, invece, consiste nella formulazione di motivazioni che siano percepite come plausibili e rilevanti da un utente umano, spesso in un contesto decisionale o etico.

Questa distinzione è particolarmente rilevante in ambito di sicurezza, diritto e auditabilità, dove la mera trasparenza tecnica non è sufficiente: è necessario che le spiegazioni fornite siano anche *giustificabili*, ovvero compatibili con le aspettative, i valori e le conoscenze pregresse dell'utente. L'integrazione di explanation e justification costituisce una sfida aperta nei moderni sistemi AI, e rappresenta un criterio essenziale nella progettazione di interfacce esplicative efficaci.

6.1.2 Tecniche principali

Nel panorama attuale, le tecniche di explainability si suddividono in approcci *model-specific* e *model-agnostic*, a seconda che siano strettamente legate alla struttura interna del modello oppure applicabili in modo generalizzato. Tra gli strumenti più consolidati, tre metodologie si sono affermate per la loro efficacia, estensibilità e capacità di supportare sia la generazione di spiegazioni che la loro valutazione:

- **SHAP** (Shapley Additive Explanations): si basa su un formalismo derivato dalla teoria dei giochi cooperativi. L'obiettivo è assegnare a ciascuna feature

un valore di contribuzione alla predizione, calcolato come valore atteso marginale su tutte le possibili coalizioni di feature. Formalmente, per un modello f e un'istanza x, lo SHAP value associato alla feature i è definito come:

$$\phi_i(f, x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x) - f_S(x) \right]$$

dove F è l'insieme di tutte le feature e $f_S(x)$ rappresenta il modello limitato al sottoinsieme S. SHAP garantisce proprietà desiderabili come consistenza, accuratezza locale e additività, risultando particolarmente adatto per scenari normativi e di auditing.

Recenti contributi hanno esteso SHAP all'ambito dei modelli generativi e delle architetture Transformer, proponendo adattamenti che sfruttano i pesi di attenzione per migliorare la decomposizione additiva nelle architetture NLP di grandi dimensioni. Tra questi, è stato introdotto *TransformerSHAP*, un approccio specificamente pensato per i Large Language Models, in grado di fornire spiegazioni più strutturate e coerenti nel contesto dei modelli Transformer [14].

- **LIME** (Local Interpretable Model-agnostic Explanations): costruisce un modello interpretabile localmente (es. regressione lineare, albero decisionale) per approssimare il comportamento del modello black-box in un intorno definito di un'istanza target. L'approccio consiste nel perturbare l'input x, valutare le predizioni $f(x')$, e addestrare un modello interpretabile g pesato da una funzione di prossimità $\pi(x, x')$, ottimizzando:

$$\arg\min_{g \in G} \mathcal{L}(f, g, \pi) + \Omega(g)$$

dove G è lo spazio dei modelli interpretabili, $\mathcal{L}$ la perdita di fedeltà locale e $\Omega(g)$ la complessità del modello g. LIME è particolarmente utile per generare spiegazioni intuitive, ma la sua affidabilità è fortemente influenzata dalla scelta dei parametri locali e dalla stabilità del campionamento.

Una recente estensione del metodo ha portato allo sviluppo di **LIME-AutoML**, un framework che integra LIME all'interno di pipeline AutoML, automatizzando la selezione dei parametri di perturbazione e delle metriche di fedeltà. Questo adattamento migliora la robustezza delle spiegazioni e consente un'applicazione più stabile ed efficiente in scenari di deployment continuo [15].

- **Explainability Index (EI)**: metrica sintetica che quantifica, dato un vettore di importanza delle feature (ad esempio derivato da SHAP o LIME), quante dimensioni sono necessarie per spiegare una certa percentuale cumulativa del comportamento del modello. Formalmente, dato un vettore $\mathbf{w} \in \mathbb{R}^d$ contenente i valori assoluti medi di importanza per ciascuna feature,

ordinati in ordine decrescente, si definisce l'EI come il numero minimo di feature necessarie affinché:

$$\frac{\sum_{i=1}^{k} w_i}{\sum_{i=1}^{d} w_i} \geq \tau$$

dove $\tau \in (0, 1)$ è una soglia prefissata, tipicamente pari a 0,9. L'EI consente di confrontare modelli in termini di complessità interpretativa, indipendentemente dalla natura del task o dalla struttura del modello. Una soglia teorica per considerare un modello spiegabile può essere formulata come $EI \leq \alpha \cdot \log_2 d$, dove d è la dimensionalità del dato e α un fattore empirico di tolleranza.

Recenti proposte metodologiche hanno impiegato l'EI anche in contesti di spiegabilità equa, integrandolo con misure di parità demografica per valutare modelli in termini di accessibilità e responsabilità sociale [16]. L'indice viene spesso utilizzato in combinazione con tecniche di attribution per fornire una valutazione oggettiva e computabile della spiegabilità globale.

L'introduzione di indici come l'EI rappresenta un passo rilevante verso l'automatizzazione della verifica della spiegabilità in contesti regolamentati, poiché consente di affiancare ai metodi qualitativi anche misure oggettive e ripetibili, compatibili con pipeline di assurance continua.

Spiegazioni controfattuali

Un approccio alternativo alle tecniche additive o locali è rappresentato dalle **spiegazioni controfattuali** (counterfactual explanations), che mirano a rispondere alla domanda: *"Qual è la minima modifica necessaria all'input per ottenere un risultato differente?"*. Questo paradigma si basa sulla generazione di un input ipotetico x' tale che il modello produca un output desiderato $y' \neq y$, pur mantenendo x' il più simile possibile all'input originale x.

Formalmente, il problema può essere espresso come:

$$\min_{x'} \; D(x, x') \quad \text{tale che} \quad f(x') = y',$$

dove D è una misura di distanza (tipicamente normata) e f è il modello predittivo. L'ottimizzazione può includere vincoli di plausibilità o coerenza semantica per garantire che la controfattuale generata sia realistica nel dominio applicativo.

Le spiegazioni controfattuali sono particolarmente rilevanti in contesti decisionali sensibili, come il credito, la selezione del personale o l'assistenza sanitaria, poiché forniscono indicazioni pratiche su come modificare una condizione per ottenere un esito favorevole. A differenza di SHAP o LIME, che spiegano l'output corrente, l'approccio controfattuale enfatizza la trasformazione dell'output rispetto a modifiche minime dell'input.

Recenti studi, come quello di **Van Looveren e Klaise (2021)**, hanno proposto approcci generativi per produrre controfattuali realistici utilizzando autoencoder

e reti neurali invertibili [17]. Tali metodi integrano vincoli strutturali nei modelli generativi per preservare coerenza e plausibilità semantica, rendendo il framework controfattuale sempre più adatto per ambienti regolati e ad alta responsabilità.

6.1.3 Valutazione della spiegabilità

La spiegabilità non è una proprietà binaria o assoluta, ma una caratteristica complessa che può essere valutata lungo più dimensioni complementari. In ambito scientifico e industriale, diversi framework hanno identificato criteri ricorrenti per analizzare l'efficacia delle spiegazioni prodotte dai modelli di intelligenza artificiale. Tra questi, emergono con particolare rilevanza tre dimensioni fondamentali:

- **Fedeltà** (*fidelity*): indica quanto la spiegazione riflette accuratamente il comportamento del modello. Una spiegazione fedele è in grado di rappresentare in modo coerente le decisioni effettive del sistema, evitando distorsioni o semplificazioni eccessive. Strumenti come SHAP garantiscono un alto livello di fedeltà teorica, mentre tecniche locali come LIME ne ottimizzano la coerenza in prossimità dell'input analizzato. Ad esempio, in un sistema di valutazione creditizia, una spiegazione fedele dovrebbe identificare il reddito e la storia debitoria come fattori determinanti, escludendo elementi secondari non rilevanti.

- **Stabilità** (*stability*): si riferisce alla coerenza delle spiegazioni al variare leggero dei dati in ingresso. Se piccole modifiche all'input causano spiegazioni drasticamente diverse, la tecnica non può essere considerata affidabile. La stabilità è particolarmente importante nei contesti ad alto rischio, dove ogni decisione deve essere tracciabile e verificabile. Ad esempio, in ambito diagnostico, se due immagini mediche quasi identiche generano spiegazioni opposte, il metodo interpretativo perde credibilità agli occhi del clinico.

- **Semplicità** (*simplicity*): riguarda la facilità con cui la spiegazione può essere compresa da un essere umano. Una spiegazione semplice privilegia la chiarezza e l'intelligibilità, anche a scapito del dettaglio tecnico. Misure come l'Explainability Index cercano di quantificare questo aspetto stimando quante informazioni sono realmente necessarie per spiegare una decisione. Ad esempio, un albero decisionale poco profondo o una lista di tre feature significative può risultare molto più interpretabile di una lunga formula con decine di variabili.

Queste dimensioni, pur essendo concettualmente distinte, possono entrare in conflitto tra loro. Una spiegazione molto dettagliata e fedele potrebbe risultare difficile da interpretare, mentre una spiegazione semplice potrebbe omettere elementi rilevanti. Trovare il giusto equilibrio tra questi aspetti è cruciale, soprattutto quando il modello viene impiegato in contesti regolamentati o sensibili.

Un ulteriore aspetto critico riguarda la differenza tra spiegabilità percepita ed effettiva. Alcune tecniche producono spiegazioni che appaiono plausibili all'occhio umano ma che, in realtà, non riflettono il comportamento autentico del modello. Questo fenomeno, noto come *delusional interpretability*, può generare una falsa percezione di controllo, compromettendo la validità di audit e revisioni indipendenti.

Integrare criteri di valutazione della spiegabilità all'interno di un processo di assurance permette non solo di misurare la trasparenza del modello, ma anche di migliorare la fiducia degli utenti, supportare le attività di auditing e garantire la conformità a normative come l'AI Act.

6.1.4 Limitazioni e rischi

Nonostante la crescente adozione delle tecniche di explainability nei sistemi di intelligenza artificiale, è essenziale riconoscerne i limiti strutturali e i rischi operativi, specialmente in scenari ad alto rischio e soggetti a normative stringenti. L'esistenza di una spiegazione non garantisce automaticamente trasparenza, affidabilità o conformità.

- **Spiegazioni fuorvianti**: le tecniche basate su approssimazioni locali o euristiche possono produrre spiegazioni che appaiono coerenti ma non riflettono accuratamente la logica decisionale reale. Questo fenomeno, noto come *delusional interpretability*, può indurre una falsa percezione di comprensione, compromettendo i processi di validazione e audit. Nei sistemi di scoring bancario, ad esempio, spiegazioni ingannevoli possono alterare la percezione di equità agli occhi dell'utente.

- **Vulnerabilità ad attacchi adversarial**: approcci post-hoc come SHAP e LIME possono essere manipolati in modo mirato per generare spiegazioni arbitrarie pur mantenendo invariata la predizione del modello. È stato dimostrato che modifiche minime all'input o ai pesi possono alterare significativamente la spiegazione prodotta [18]. Questo espone i sistemi a nuovi vettori di attacco, con impatti critici in settori come il sanitario o il forense, dove la spiegazione rappresenta una traccia ufficiale.

- **Ambiguità e incoerenza tra metodi**: applicando metodi diversi allo stesso input e modello si possono ottenere spiegazioni discordanti, dovute a differenze nelle ipotesi teoriche o nelle metriche ottimizzate. Questa eterogeneità rende difficile validare le spiegazioni in modo oggettivo. L'assenza di standard condivisi complica ulteriormente la comparazione e limita l'affidabilità inter-metodo.

Strategie di mitigazione. Per affrontare questi rischi, la letteratura propone diverse contromisure, tecniche e organizzative:

- *Confronto sistematico* tra più tecniche di explainability per identificare incoerenze;
- *Validazione umana* attraverso audit di leggibilità e valutazioni esperte;
- *Metriche quantitative* come l'Explainability Index, la varianza intermetodo o l'accuratezza locale;
- *Robustezza computazionale* mediante test adversarial mirati sulle spiegazioni.

L'adozione di questi strumenti permette di rafforzare la fiducia nel modello e dimostrare conformità ai requisiti di trasparenza e accountability richiesti dal Regolamento Europeo sull'Intelligenza Artificiale (AI Act, Art. 13).

Verso l'implementazione. In risposta a queste criticità, diventa essenziale progettare componenti software che integrino valutazioni di spiegabilità direttamente nei flussi di monitoraggio dei modelli. La sezione seguente introduce una *Probe explainability-aware*, concepita per misurare automaticamente la qualità delle spiegazioni prodotte, in modo coerente con i vincoli imposti dalla normativa e dalle pratiche di assurance continua.

6.2 Implementazione in Probe

La formalizzazione concettuale dell'Explainability Index e l'utilizzo dei valori SHAP, presentati nella sezione precedente, costituiscono il fondamento per l'implementazione automatica della proprietà di spiegabilità all'interno dell'infrastruttura Moon Cloud.

In questa sezione viene descritto il funzionamento top-down della *ExplainabilityProbe*, ovvero il modulo software responsabile dell'analisi della trasparenza dei modelli. L'approccio adottato combina tecniche di interpretabilità locale con metriche sintetiche globali e include, inoltre, una valutazione della stabilità delle spiegazioni per garantire coerenza semantica.

6.2.1 Comportamento Generale e Assunzioni

La *probe* di explainability è progettata per operare all'interno di pipeline di AI supervisionato, con l'obiettivo di valutare automaticamente la trasparenza di modelli già addestrati. Il suo funzionamento si basa su un approccio top-down, in cui il modello viene trattato come una black-box e analizzato mediante tecniche post-hoc come SHAP o LIME.

Il componente assume che il modello sia stato precedentemente salvato in un formato serializzabile e compatibile con i principali framework di explainability. Inoltre, richiede la disponibilità di un dataset di test rappresentativo, le cui feature siano coerenti con quelle utilizzate in fase di training.

Le principali precondizioni per il corretto funzionamento della probe includono:

- un **modello serializzato** in formato `.h5`, `.pkl` o equivalente, compatibile con TensorFlow, Keras o Scikit-learn;
- un **dataset di test** strutturato secondo lo stesso schema del training set, preferibilmente già normalizzato o preprocessato;
- un **formato leggibile** (es. `.txt`, `.csv`, `.npy`) che consenta il parsing efficiente delle istanze da analizzare;
- una compatibilità strutturale con le librerie SHAP o LIME, che richiedono modelli predittivi e accesso diretto alle predizioni del modello.

In presenza di input non validi o incompatibili (es. numero errato di feature, formato non leggibile, assenza di metadati), la probe attiva un sistema interno di gestione degli errori, notificando l'anomalia e bloccando l'esecuzione in sicurezza. La configurabilità del sistema consente di adattare soglie, numero di campioni analizzati, metodi di fallback e livello di tolleranza ai requisiti specifici dell'ambiente in cui è integrato.

6.2.2 Struttura della Probe e Snippet Rilevanti

La probe segue una sequenza strutturata di fasi funzionali, ognuno dei quali svolge un compito fondamentale nell'analisi delle spiegazioni prodotte dal modello. La struttura della probe è progettata per essere modulare e facilmente configurabile, consentendo l'adattamento a diversi tipi di modelli e dataset.

I passaggi principali della probe sono i seguenti:

1. **Caricamento del modello:** Il primo passo consiste nel caricare il modello serializzato, che può essere un modello addestrato tramite TensorFlow, Keras, Scikit-learn o altri framework. Il modello viene caricato in memoria, pronto per essere analizzato. Il formato più comune per la serializzazione dei modelli è `.h5` o `.pkl`.

```
1     from tensorflow.keras.models import load_model
2
3     # Caricamento del modello Keras
4     model = load_model('model/optimized_model.h5')
5
```

2. **Caricamento dei dati e preprocessing:** Una volta caricato il modello, la probe carica il dataset di test, che deve essere coerente con le caratteristiche del training set. I dati vengono sottoposti a un processo di preprocessing (es. normalizzazione, gestione dei valori mancanti) per garantire che siano pronti per l'analisi. Il preprocessing deve essere consistente con quello utilizzato durante l'addestramento del modello.

```
    import numpy as np

    # Caricamento dei dati di test
    X_test = np.loadtxt('data/X_test.txt')
    y_test = np.loadtxt('data/y_test.txt')

    # Preprocessing (normalizzazione)
    X_test = (X_test - np.mean(X_test, axis=0)) / np.std(
X_test, axis=0)

```

3. **Selezione di uno o più esempi:** Successivamente, la probe seleziona uno o più esempi dal dataset di test per analizzare le spiegazioni locali. La selezione degli esempi può essere fatta in modo casuale o sulla base di determinati criteri (es. esempio con maggiore errore di predizione).

```
    # Selezione di esempi casuali dal dataset
    import random
    selected_example = random.choice(X_test)

```

4. **Calcolo delle spiegazioni locali:** Il passaggio finale consiste nel calcolare le spiegazioni per ogni esempio selezionato, utilizzando tecniche come SHAP o LIME. Le spiegazioni locali forniscono il contributo di ciascuna feature alla predizione del modello, consentendo di comprendere come il modello abbia preso una decisione su un singolo esempio.

```
    import shap

    # Inizializzazione dell'Explainer SHAP
    explainer = shap.KernelExplainer(model.predict, X_test
[:100])  # Usa un sottoinsieme dei dati come background
    shap_values = explainer.shap_values(selected_example)

    # Visualizzazione delle spiegazioni locali
    shap.initjs()
    shap.force_plot(explainer.expected_value[0], shap_values
[0], selected_example)

```

Questa sequenza di operazioni consente alla probe di generare spiegazioni affidabili e comprensibili per ogni esempio analizzato, fornendo strumenti utili per il monitoraggio e la validazione dei modelli in contesti regolamentati.

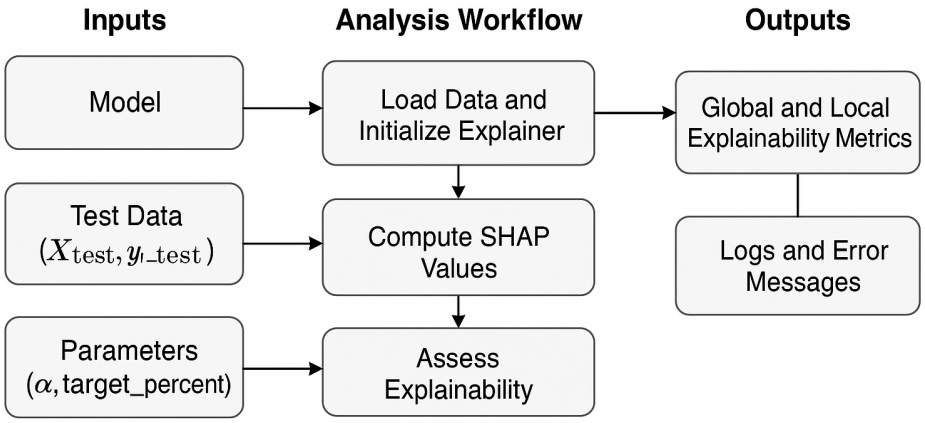

Figura 6.1: Flusso di elaborazione della Explainability Probe: a partire da modello e dati, viene eseguita una sequenza di analisi che porta al calcolo dei valori SHAP, alla valutazione dell'esplicabilità e alla generazione delle metriche, dei log e dell'output strutturato.

6.2.3 Calcolo delle Spiegazioni Locali (SHAP)

Il processo di calcolo delle spiegazioni locali si fonda su due tecniche principali: **SHAP** e **Explainability Index (EI)**. La combinazione di tali tecniche consente di ottenere spiegazioni dettagliate relativamente alle predizioni del modello. Di seguito, si analizzano i meccanismi implementativi di tali metodi, evidenziando come SHAP fornisca una misura marginale del contributo delle feature e come l'EI consenta di quantificare la spiegabilità complessiva.

Inizializzazione dell'Explainer SHAP

Il metodo SHAP si basa sulla teoria dei giochi cooperativi, attribuendo a ciascuna variabile indipendente un contributo marginale alla predizione del modello. L'oggetto `shap.KernelExplainer` viene utilizzato per l'analisi di modelli non interpretabili in modo diretto, come le reti neurali profonde.

```
1 import shap
2
3 # Caricamento del modello e selezione del background
4 explainer = shap.KernelExplainer(model.predict, X_test[:100])
```

Nel frammento di codice, l'oggetto `explainer` si occupa del calcolo dei valori SHAP, sfruttando come background un sottoinsieme del dataset di test. Tale

campione viene utilizzato per confrontare ogni istanza con un contesto medio, stimando l'impatto marginale delle singole feature.

Calcolo dei Valori SHAP

Una volta inizializzato l'explainer, è possibile determinare i valori SHAP per una specifica istanza, al fine di valutare il contributo di ciascuna variabile alla predizione.

```
# Selezione di un'istanza dal dataset
selected_example = X_test[0]

# Calcolo dei valori SHAP
shap_values = explainer.shap_values(selected_example)
```

Il risultato della funzione `shap_values` rappresenta l'importanza attribuita dal modello alle diverse feature, relativamente alla predizione per l'istanza selezionata.

Visualizzazione delle Spiegazioni

I valori SHAP possono essere rappresentati graficamente tramite la funzione `force_plot`, che consente una visualizzazione intuitiva dei contributi positivi e negativi delle feature rispetto alla predizione complessiva.

```
shap.initjs()
shap.force_plot(explainer.expected_value[0], shap_values[0],
    selected_example)
```

Tale rappresentazione evidenzia in maniera interattiva l'influenza di ciascuna variabile indipendente, distinguendo tra le componenti che incrementano o riducono la probabilità predetta.

Calcolo dell'Explainability Index (EI)

L'Explainability Index (EI) fornisce una misura sintetica della spiegabilità, quantificando il numero minimo di feature necessario per coprire una determinata soglia di contributo cumulativo alla predizione. Il calcolo avviene secondo il seguente algoritmo:

```
def calculate_explainability_index(shap_values, target_percent
    =0.9):
    """
    Calcolo dell'Explainability Index (EI) dato un array di
    valori SHAP.
    """
    mean_abs_vals = np.mean(np.abs(shap_values), axis=0)
    sorted_vals = np.sort(mean_abs_vals)[::-1]
    cumulative_contrib = np.cumsum(sorted_vals)
    total_contrib = cumulative_contrib[-1]
```

```
    ei = np.argmax(cumulative_contrib >= target_percent *
    total_contrib) + 1
    return ei
```

Il calcolo si basa sull'ordinamento dei valori SHAP in termini di importanza assoluta media e sulla determinazione del punto in cui il contributo cumulativo supera la soglia prefissata (ad esempio il 90%).

Valutazione della Spiegabilità

A partire dal valore di EI calcolato, è possibile stabilire se il modello risulti sufficientemente interpretabile. Una soglia teoricamente fondata prevede che il valore di EI non superi $\alpha \cdot \log_2 d$, dove d rappresenta la dimensionalità del dataset e $\alpha \in \mathbb{R}^+$ è un coefficiente scelto in base al contesto applicativo.

Conclusioni

L'impiego combinato dei valori SHAP e dell'Explainability Index consente un'analisi sia locale che globale del comportamento del modello. Mentre SHAP fornisce una spiegazione dettagliata a livello di singola istanza, l'EI offre una metrica compatta in grado di sintetizzare il grado complessivo di spiegabilità. Tali strumenti risultano particolarmente rilevanti in ambiti in cui la trasparenza e la validabilità delle decisioni algoritmiche assumono un ruolo critico.

Nota sui modelli multiclass. Nel caso di modelli di classificazione multiclasse, l'explainer restituisce un insieme di vettori SHAP distinti, uno per ciascuna classe predetta. In questa implementazione, si considera il vettore relativo alla prima classe restituita (`shap_values[0]`), ipotizzando che tale classe corrisponda alla classe prevalente o predetta dal modello.

Questa scelta, pur semplificata, è coerente con l'obiettivo di stimare l'Explainability Index per la decisione principale del modello, che è quella normalmente rilevante dal punto di vista applicativo. In implementazioni future, è possibile estendere il calcolo del valore EI a più classi, o selezionare dinamicamente la classe corrispondente alla predizione effettiva.

6.2.5 Valutazione della Coerenza e Stabilità

Oltre alla spiegabilità globale e locale ottenuta mediante valori SHAP e Explainability Index, è necessario valutare la *stabilità* delle spiegazioni generate. Tale proprietà si riferisce alla coerenza delle spiegazioni al variare di input simili, e risulta fondamentale per garantire l'affidabilità semantica delle interpretazioni nel tempo e in presenza di perturbazioni lievi dei dati.

Formalmente, data una funzione di spiegazione $E(x)$ che restituisce l'importanza attribuita a ciascuna feature per un input x, la stabilità locale può essere misurata come lo scostamento medio tra i vettori di importanza di x e di un insieme di istanze simili x' appartenenti a un intorno $\mathcal{N}(x)$:

$$\text{Stability}(x) = \frac{1}{|\mathcal{N}(x)|} \sum_{x' \in \mathcal{N}(x)} \|E(x) - E(x')\|_2$$

Nel contesto implementativo adottato, l'insieme $\mathcal{N}(x)$ viene approssimato considerando le k istanze più simili a x in termini di distanza coseno. A ciascuna di esse viene associato un vettore SHAP, e la distanza euclidea tra tali vettori viene impiegata come metrica di variazione. Tale procedura consente di stimare la coerenza locale delle spiegazioni prodotte per il modello in esame.

```
from sklearn.metrics.pairwise import cosine_similarity

def evaluate_stability(shap_values, X_test, k=10):
    similarities = cosine_similarity(X_test)
    stabilities = []
    for i in range(len(X_test)):
        neighbors_idx = np.argsort(similarities[i])[-(k+1):-1]
        diffs = [np.linalg.norm(shap_values[i] - shap_values[j])
    for j in neighbors_idx]
        stabilities.append(np.mean(diffs))
    return np.mean(stabilities)
```

L'indice risultante viene confrontato con una soglia δ configurabile (ad esempio $\delta = 0.2$), oltre la quale le spiegazioni sono considerate instabili. Tale soglia può essere definita a partire da analisi empiriche, considerazioni normative o metodi di validazione cross-modello.

L'adozione di un controllo esplicito della stabilità risponde a requisiti di *robustezza semantica*, riducendo il rischio di interpretazioni incoerenti dovute a rumore o anomalie locali nei dati. In scenari regolamentati, tale aspetto assume rilievo particolare per garantire auditabilità e affidabilità delle spiegazioni fornite.

Infine, il valore numerico della stabilità può essere incluso nell'output della probe come metadato supplementare, a supporto di una diagnosi più approfondita del comportamento esplicativo del modello. Il logging include, per ciascun esperimento, la misura aggregata di stabilità, la soglia utilizzata e l'esito binario della valutazione (stabile/non stabile), in coerenza con lo schema di Moon Cloud.

Nota metodologica. La scelta della distanza euclidea tra vettori SHAP come proxy della stabilità locale è motivata dalla sua semplicità computazionale, dalla simmetria e dall'interpretabilità diretta come variazione assoluta tra contributi feature-wise. In contesti ad alta dimensionalità, la distanza euclidea si comporta in modo più prevedibile rispetto ad altre metriche, risultando meno influenzata da rumore sparso e distribuzioni sbilanciate.

In alternativa, potrebbero essere adottate metriche più sofisticate come la distanza di Jensen–Shannon tra distribuzioni normalizzate di importanza o la similarità coseno. Tuttavia, tali approcci comportano una complessità computazionale maggiore e una dipendenza più forte dalla scala e dalla normalizzazione dei valori SHAP.

6.2.6 Output e Logging

L'output generato dalla *ExplainabilityProbe* è strutturato in modo da fornire, oltre all'esito binario della valutazione, una serie di evidenze strutturate utili per l'audit, la tracciabilità e l'integrazione nei sistemi di assurance continua. Il formato di output segue lo schema JSON previsto dal backend di Moon Cloud, comprendente le seguenti sezioni:

- `integer_result`: valore intero che indica l'esito della valutazione (0 = successo, 1 = modello non explainable, 2 = errore, ecc.).
- `pretty_result`: stringa sintetica che descrive in modo leggibile il risultato.
- `extra_data`: dizionario strutturato contenente le evidenze quantitative e qualitative raccolte durante l'esecuzione.

Un esempio di output generato dalla probe è il seguente:

```
{
  "integer_result": 0,
  "pretty_result": "Il modello    spiegabile e stabile",
  "extra_data": {
    "ei_global": 5,
    "ei_threshold": 6.32,
    "ei_local": 4,
    "globally_explainable": true,
    "stability_score": 0.12,
    "stability_threshold": 0.2,
    "explainer_type": "GradientExplainer",
    "samples_analyzed": 100,
    "top_global_features": [
      [123, 0.0851],
      [45, 0.0784],
      [78, 0.0725]
    ],
    "top_local_features": [
      [45, 0.1023],
      [123, 0.0942],
      [91, 0.0875]
    ]
  }
}
```

In particolare, il campo `extra_data` fornisce:

- il valore calcolato dell'Explainability Index globale (`ei_global`);
- la soglia massima ammessa in base al numero di feature (`ei_threshold`);
- l'Explainability Index locale relativo all'istanza selezionata (`ei_local`);
- l'indicatore binario di spiegabilità (`globally_explainable`);

- la misura della stabilità delle spiegazioni (`stability_score`) e la soglia di riferimento;
- il tipo di explainer utilizzato (SHAP Gradient, Deep o Kernel);
- l'elenco delle feature più rilevanti a livello globale e locale, accompagnate dal peso medio assoluto.

Tali informazioni vengono serializzate e inviate tramite il canale asincrono NATS all'Evidence Writer, che si occupa della loro normalizzazione e persistenza all'interno del database relazionale. Questo meccanismo consente la successiva consultazione tramite la dashboard Moon Cloud, nonché l'integrazione con moduli di auditing, rendicontazione e generazione di alert.

La disponibilità di metadati espliciti e coerenti con le soglie teoriche definite nel modello di assurance consente infine di integrare la spiegabilità tra le proprietà formalmente verificabili, supportando meccanismi decisionali automatizzati e conformi ai requisiti normativi dell'AI Act.

6.2.7 Gestione degli Errori

La probe di explainability implementa una gestione strutturata degli errori, ispirata ai principi di robustezza e non intrusività previsti dall'architettura Moon Cloud. Gli errori vengono intercettati in ciascuna fase della pipeline e restituiti con codici espliciti all'interno del campo `integer_result`, secondo la seguente codifica:

- `0`: esecuzione completata con successo, modello spiegabile.
- `1`: modello non spiegabile, Explainability Index sopra soglia.
- `2`: instabilità nelle spiegazioni locali.
- `3`: errore tecnico durante l'elaborazione (es. caricamento fallito, SHAP error).

Gli errori più comuni gestiti includono:

- **Incompatibilità tra modello e dataset**: viene rilevata quando la dimensione delle feature nei dati di test non corrisponde all'input atteso dal modello serializzato.
- **Mancanza o ambiguità nei nomi delle feature**: può compromettere la leggibilità delle spiegazioni e l'ordinamento delle importanze. In tal caso, la probe notifica l'assenza e assegna un indice numerico incrementale.
- **Fallimento nella computazione dei valori SHAP**: gli errori durante l'inizializzazione dell'explainer (es. `GradientExplainer`, `DeepExplainer`, `KernelExplainer`) sono catturati con fallback progressivo tra le tecniche disponibili. Se tutti i tentativi falliscono, l'esecuzione viene interrotta in sicurezza e il messaggio d'errore viene riportato nei log.

- **Eccezioni inattese**: eventuali errori non previsti vengono comunque catturati e registrati, per garantire che la probe non comprometta la stabilità complessiva della piattaforma.

La gestione degli errori è progettata per essere compatibile con il sistema di rollback automatico e con l'architettura a stati della probe. In caso di fallimento, le risorse temporanee vengono eliminate, e il messaggio diagnostico è incluso nel campo `extra_data` per facilitare attività di debugging o audit successivi.

Tale approccio consente l'integrazione affidabile della probe anche in contesti produttivi regolamentati, riducendo il rischio di comportamenti inattesi e migliorando la resilienza dell'intero sistema di assurance.

Conclusioni sull'Implementazione

L'implementazione della probe di explainability consente di valutare in modo sistematico la trasparenza dei modelli di intelligenza artificiale, combinando tecniche avanzate di interpretabilità locale (SHAP) con una metrica sintetica e formale (Explainability Index). La struttura modulare della probe, unita alla compatibilità con modelli serializzati e dataset reali, ne permette l'integrazione in pipeline MLOps e sistemi di assurance automatica.

L'adozione di soglie teoriche per l'Explainability Index, basate sulla complessità logaritmica del numero di feature, garantisce la ripetibilità e l'oggettività del giudizio di spiegabilità. La valutazione della stabilità delle spiegazioni rafforza ulteriormente l'affidabilità delle evidenze prodotte, mitigando il rischio di interpretazioni arbitrarie o incoerenti.

Grazie all'output strutturato e alla gestione robusta degli errori, la probe si integra pienamente nella piattaforma Moon Cloud, contribuendo alla tracciabilità, auditabilità e conformità dei modelli analizzati. Tale componente rappresenta un passo concreto verso l'implementazione pratica dei requisiti imposti dall'AI Act in materia di trasparenza e accountability algoritmica, favorendo un uso più responsabile e verificabile dei sistemi basati su intelligenza artificiale.

6.2.4 Conclusioni

Il presente capitolo ha introdotto e formalizzato la proprietà di *explainability*, ovvero la capacità di un modello predittivo di fornire spiegazioni interpretabili rispetto alle decisioni prodotte. Tale proprietà è diventata centrale nella regolamentazione europea (AI Act) e nei sistemi di auditing algoritmico, in quanto strettamente connessa ai principi di trasparenza, accountability e comprensibilità delle decisioni automatizzate.

A livello teorico, si è adottato un approccio quantitativo basato sull'Explainability Index (EI), che misura il numero minimo di feature necessarie per spiegare una determinata soglia di variabilità del modello. Tale indice, derivato da metriche di interpretabilità locale (valori SHAP), consente di sintetizzare in modo formale e ripetibile la spiegabilità del comportamento del modello, sia a livello globale che

locale. È stata inoltre introdotta una metrica di stabilità semantica, al fine di valutare la coerenza delle spiegazioni rispetto a piccole perturbazioni dei dati di input.

L'implementazione in probe, integrata nella piattaforma Moon Cloud, ha dimostrato la possibilità di automatizzare l'intero processo valutativo, producendo un output strutturato, interpretabile e conforme ai requisiti normativi. Tale implementazione include la gestione degli errori, strategie di fallback in caso di fallimento degli explainer, e meccanismi di logging e tracciabilità compatibili con ambienti MLOps e pipeline di AI assurance.

In prospettiva comparata, l'approccio adottato per la spiegabilità risulta concettualmente coerente con quanto discusso nei capitoli precedenti in merito alla *well-formedness* e alla *fairness*. In tutti e tre i casi, infatti, viene adottata una logica top-down: partendo da una proprietà desiderabile (completezza, equità, spiegabilità), si formalizza una metrica verificabile, si definiscono soglie teoriche di accettabilità e si sviluppa un modulo software in grado di eseguire la valutazione in modo autonomo e scalabile. L'Explainability Index, in particolare, si distingue per la sua immediata interpretabilità anche da parte di decisori non tecnici, rappresentando un utile strumento di comunicazione trasparente dei risultati.

L'integrazione sinergica di queste proprietà nella suite Moon Cloud costituisce un passo rilevante verso la costruzione di sistemi di intelligenza artificiale affidabili, verificabili e conformi alle normative emergenti.

Capitolo 7

Caso di Studio ed Esperimenti

7.1 Obiettivi

Con la crescente diffusione dell'intelligenza artificiale in settori critici, è diventato sempre più necessario disporre di strumenti che consentano una verifica sistematica, trasparente e ripetibile delle proprietà non funzionali dei sistemi AI, tra cui la qualità dei dati, l'equità delle decisioni algoritmiche e la spiegabilità dei modelli. In risposta a tali esigenze, la **dashboard di AI Assurance** descritta in questo capitolo è stata progettata come estensione e modulo di supporto operativo alla piattaforma *Moon Cloud*. La dashboard è servita come dimostratore aggiuntivo di rispetto alle probe già implementate in Moon Cloud.

Questa dashboard svolge un ruolo fondamentale nell'ecosistema Moon Cloud, fornendo un'interfaccia dedicata alla configurazione, all'attivazione e al monitoraggio dei controlli di assurance, orientati alla compliance con i requisiti normativi dell'AI Act. Le principali funzionalità della piattaforma includono:

- un workflow completamente automatizzato, configurabile e basato su probe modulari per la valutazione di well-formedness dei dataset, fairness nei modelli e explainability dei processi decisionali;
- un'interfaccia web user-friendly che guida l'utente nell'impostazione dei controlli, nell'esecuzione delle analisi e nella fruizione immediata dei risultati numerici e grafici;
- un sistema integrato di audit, con logging puntuale e versioning di tutte le analisi effettuate, esportabile in formati standard per ispezioni e conformità normativa.

Attraverso tali strumenti, la dashboard consente l'adozione pratica di processi di AI Assurance conformi alle più avanzate linee guida di regolamentazione, e si propone come punto di raccordo tra la componente infrastrutturale di Moon Cloud e le esigenze operative di data scientist, compliance officer e auditor chiamati a dimostrare responsabilità e trasparenza nell'uso dell'intelligenza artificiale.

7.2 Caso di studio: Disparità nel Recruiting Automatizzato

Negli ultimi anni, l'introduzione di sistemi di **AI generativa** (basati su Large Language Models, GenAI) per lo screening automatizzato dei curricula ha impattato in modo significativo il mondo del lavoro europeo. Questi algoritmi, inizialmente adottati per incrementare efficienza e trasparenza nei processi decisionali, hanno posto nuovi interrogativi etici, giuridici e sociali sull'uso dell'AI nella selezione del personale. Rapporti ed audit pubblicati da enti europei come **EDPS** (European Data Protection Supervisor)[19] e **AlgorithmWatch**[20] hanno certificato che, in assenza di controlli e auditing strutturati, tali sistemi hanno generato **disparità di genere** fino al 18% nelle selezioni femminili rispetto a quelle maschili[19], e **bias discriminatori** concreti, con un 22% di penalizzazione delle donne nei processi di screening[20]: a parità di esperienza e competenze, le candidate donne risultavano selezionate in proporzione significativamente inferiore rispetto agli uomini. Questo fenomeno è attribuito sia alla sottorappresentazione dei dati femminili nei dataset di addestramento, sia agli stereotipi linguistici appresi ed amplificati dalle AI generative[21, 22].

Aspetti etici e sociali

Questi scenari sollevano una serie di problematiche:

- Bias algoritmico: decisioni influenzate da pregiudizi storici presenti nei dati, difficilmente percepibili dai progettisti o dai responsabili HR senza strumenti di auditing appropriati.
- Opacità e mancanza di accountability: la natura "black box" dei modelli GenAI rende difficile risalire alle cause delle decisioni e argomentare le ragioni di una selezione o esclusione. Mancano motivazioni trasparenti ed interpretabili, negando ai candidati il diritto di conoscere e contestare l'esito decisionale.
- Riproduzione e amplificazione di stereotipi sociali: la selezione algoritmica rischia di perpetuare squilibri di genere, età e minoranza, ostacolando le politiche di equità e inclusione.
- Responsabilità etica delle organizzazioni: le aziende e le PA che si affidano a GenAI per decisioni ad alto impatto sociale devono garantire auditing, revisione, documentazione e meccanismi di intervento umano (human-in-the-loop), tutelando la dignità, i diritti e l'uguaglianza di tutte le persone coinvolte [22, 19].

Workflow di compliance e verifica

La dashboard AI compliance è stata progettata per affrontare in modo coerente le principali sfide poste dalla normazione su AI e GDPR. Nei casi studiati, ha consentito di:

- analizzare la distribuzione di genere nei risultati dei processi selettivi;
- valutare la fairness tramite metriche dedicate (*demographic parity*, *equalized odds*);
- garantire tracciabilità e trasparenza decisionale;
- attivare simulazioni di interventi correttivi, inclusa la supervisione umana.

Queste funzionalità si sono dimostrate efficaci nell'evidenziare eventuali criticità di conformità rispetto alle normative vigenti, spingendo le organizzazioni verso maggiori pratiche di audit e revisione dei dati.

Riflessione critica sulla governance etica

Questo caso studio dimostra come l'adozione indiscriminata di AI generativa in processi decisionali ad alto impatto sociale possa produrre effetti contrari agli ideali di merito, inclusione e uguaglianza. La governance responsabile dell'AI richiede:

- una **valutazione costante dei rischi**, con auditing proattivo su dati e modelli;
- l'introduzione di controlli di fairness e meccanismi di contestazione delle decisioni;
- la documentazione e la trasparenza ("right to explanation") per tutti i soggetti coinvolti[19, 22];
- una sorveglianza attiva degli organismi di controllo e la creazione di registri pubblici degli algoritmi decisionali[23].

In conclusione, l'integrazione fra strumenti di verifica automatizzata e governance etica rappresenta oggi la sfida fondamentale per garantire che l'AI sia realmente uno strumento al servizio della società, della dignità umana e dei principi democratici.

7.3 Architettura della Dashboard

La piattaforma sviluppata per questo progetto è accessibile via browser e consente di effettuare verifiche modulari sulla conformità di dataset e modelli di intelligenza artificiale. Il sistema è stato progettato per massimizzare la flessibilità, la trasparenza e la replicabilità delle analisi, con particolare attenzione alla user experience e alla modularizzazione del codice.

7.3.1 Tecnologie Utilizzate

L'interfaccia utente è stata sviluppata in **React.js**, una delle librerie JavaScript più diffuse per la creazione di Single Page Applications (SPA). La scelta di React consente di:

- mantenere un'interfaccia fluida e reattiva;
- gestire efficacemente lo stato delle configurazioni tramite componenti dinamici;
- facilitare l'integrazione con sistemi backend modulari.

La comunicazione tra frontend e backend è orchestrata tramite chiamate `API RESTful`, che permettono il caricamento dei dati, l'attivazione delle *probe* e la ricezione dei risultati.

Il backend, realizzato in Python, sfrutta un'architettura a microservizi che consente di eseguire ogni *probe* in modo isolato, favorendo la scalabilità e la manutenzione. Le principali librerie impiegate includono:

- **Pandas**, NumPy e SciPy per la gestione ed elaborazione dei dati;
- **SHAP** per l'analisi dell'explainability;
- **Scikit-learn** per l'interfacciamento ai modelli AI;
- **Matplotlib** e **Plotly** per la generazione dei grafici.

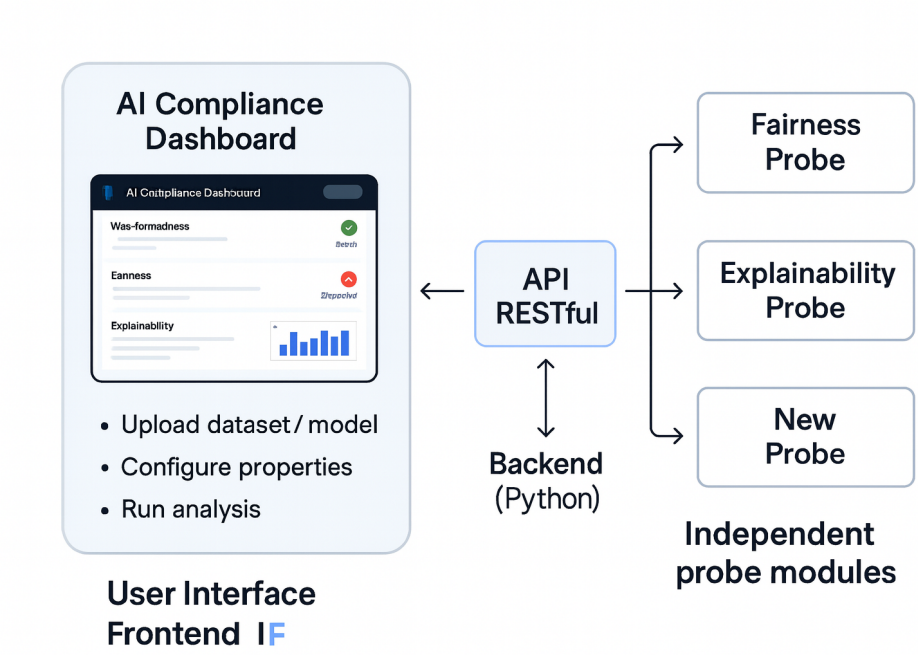

Figura 7.1: Architettura generale della piattaforma: interfaccia React, backend Python e moduli probe.

7.3.2 Struttura Modulare delle Probe

Il cuore della piattaforma risiede nella sua architettura modulare, che ruota attorno al concetto di **probe**. Ogni *probe* rappresenta un'unità funzionale autonoma incaricata di verificare una specifica proprietà del sistema AI, come la *well-formedness* del dataset, la *fairness* del modello o la sua *explainability*.

Ogni probe è progettata come un modulo indipendente, dotato delle seguenti componenti:

- **Interfaccia di configurazione**: definisce gli input richiesti dall'utente o dalla dashboard (es. variabile sensibile, soglia di valutazione, numero di campioni);
- **Motore di esecuzione**: implementa la logica dell'algoritmo di controllo, utilizzando metriche e procedure descritte nei capitoli teorici precedenti;
- **Modulo di logging e output**: restituisce un risultato strutturato, corredato da grafici, valori sintetici e commenti interpretativi, compatibile con l'interfaccia di visualizzazione.

Tale architettura consente un'elevata **estensibilità**: per aggiungere una nuova verifica è sufficiente creare un nuovo modulo probe, senza alterare la logica delle altre. Ogni probe comunica con il backend tramite API RESTful, rendendo il sistema facilmente integrabile e scalabile.

Inoltre, ogni esecuzione di probe è tracciata tramite un identificativo univoco e può essere rieseguita con gli stessi parametri, garantendo la *replicabilità* e l'*auditability* delle verifiche.

L'adozione di un pattern modulare di questo tipo ha permesso di trasformare concetti teorici, come le proprietà formali di assurance, in componenti software riutilizzabili, facilmente configurabili anche da utenti non esperti. Tale astrazione si è dimostrata particolarmente utile nel caso di studio proposto, dove sono state attivate tre probe distinte — per qualità, equità e spiegabilità — senza necessità di modificare alcun codice sorgente.

7.3.3 Esperienza Utente e Tracciabilità

La dashboard è progettata per offrire un'interazione fluida e intuitiva, garantendo all'utente un controllo completo sul processo di verifica della compliance. Attraverso un'interfaccia chiara e responsive, anche profili non tecnici possono accedere alle funzionalità principali della piattaforma, che includono:

- **Caricamento guidato** di dataset e modelli predittivi, con validazione automatica del formato e dei contenuti;
- **Selezione delle proprietà** da analizzare (es. well-formedness, fairness, explainability) tramite configurazione interattiva dei parametri di ciascuna probe;

- **Esecuzione delle verifiche** in pochi clic, con gestione asincrona delle richieste e notifica al completamento;
- **Visualizzazione dei risultati** in formato numerico, tabellare e grafico, comprensivi di spiegazioni testuali e suggerimenti per la mitigazione;
- **Esportazione dei report** in formato strutturato (es. JSON, PDF) per esigenze di documentazione, auditing o condivisione.

Ogni interazione con la piattaforma è associata a un identificativo univoco e registrata tramite un sistema di **logging strutturato**, che consente di:

- ricostruire in ogni momento le condizioni di esecuzione delle probe;
- mantenere lo storico delle analisi per finalità di audit o revisione;
- replicare esattamente un'analisi precedente per verificarne la coerenza.

La combinazione tra semplicità d'uso, tracciabilità completa e modularità rende la dashboard uno strumento adatto non solo a sviluppatori e data scientist, ma anche a figure legali, auditori e responsabili della conformità. L'interfaccia si ispira ai principi di **usabilità, trasparenza e accountability**, promuovendo una cultura operativa di AI responsabile, in linea con i requisiti dell'AI Act e delle principali linee guida etiche europee.

7.3.4 Gestione Automatizzata degli Errori con Modelli LLM

Per garantire robustezza e continuità operativa, la piattaforma integra un meccanismo di **gestione automatizzata degli errori** basato sull'utilizzo di modelli linguistici di grandi dimensioni (LLM), forniti da OpenAI. Questo componente è attivato ogniqualvolta l'esecuzione di una *probe* restituisca:

- un errore tecnico (es. eccezioni durante l'elaborazione, input mancanti, incompatibilità di formato);
- un risultato non interpretabile (es. valori `NaN`, spiegazioni nulle, anomalie logiche nei dati);
- un output semanticamente incoerente o sospetto.

Il backend attiva automaticamente un modulo di **fallback semantico**, che esegue una chiamata API verso un modello LLM. Il flusso è strutturato come segue:

1. Viene generata una richiesta contenente:
    - il nome e il tipo della probe coinvolta;
    - la configurazione dei parametri utilizzati;

- l'errore ricevuto o l'output anomalo;
- un contesto sintetico dell'obiettivo della verifica.

2. La richiesta è inviata tramite `HTTP POST` a un endpoint OpenAI, utilizzando le API ufficiali e specificando il modello desiderato (es. `gpt-4o` o `gpt-3.5-turbo`) con un `timeout` predefinito.
3. La chiave API è gestita in modo sicuro tramite variabili d'ambiente (`.env`) caricate al momento dell'esecuzione del backend e non esposte nel codice sorgente.
4. Il modello LLM restituisce una risposta in linguaggio naturale contenente:
   - una possibile spiegazione tecnica dell'errore;
   - un suggerimento per la risoluzione (es. preprocessare i dati, modificare una metrica, variare i parametri);
   - un avvertimento se l'errore è critico e impedisce ulteriori verifiche.
5. La risposta viene automaticamente salvata nel log strutturato dell'analisi, visibile all'utente nella sezione dei risultati e tracciabile per auditing.

Questo sistema permette alla piattaforma di fornire assistenza proattiva nella risoluzione dei problemi, riducendo il carico operativo e favorendo un'esperienza d'uso più fluida, specialmente in ambienti regolamentati dove la disponibilità continua è un requisito.

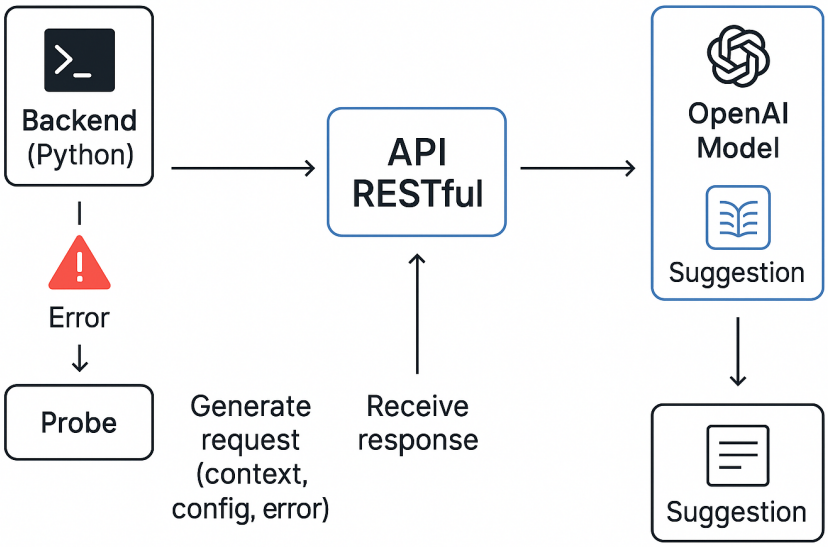

Figura 7.2: Flusso di fallback e gestione automatica degli errori tramite modello OpenAI.

7.3.5 Collegamento con il Caso di Studio

L'architettura della dashboard Moon Cloud è stata applicata con successo all'analisi di un dataset reale di candidature, simulando un contesto operativo e regolamentato legato alla selezione automatizzata di personale tramite sistemi di AI generativa. In particolare, il caso di studio ha riguardato l'utilizzo di modelli LLM (*Large Language Models*) che, impiegati per lo screening iniziale dei curricula e per la valutazione automatica dei profili, hanno manifestato problematiche di disparità di genere e bias discriminatori certificate da audit ufficiali [19, 20, 22].

La dashboard ha consentito di configurare e attivare le *probe* relative a **well-formedness**, **fairness**, ed **explainability** in modo completamente automatico, senza la necessità di modificare il codice sorgente. Sono state raccolte tutte le metriche rilevanti e, tramite moduli specifici, si è potuto valutare:

- la distribuzione di genere tra i candidati effettivamente selezionati dalla GenAI;
- la disparità e il bias nelle scelte algoritmiche rispetto a variabili sensibili quali età e minoranza;
- la spiegabilità delle decisioni prese dal sistema generativo, con restituzione delle motivazioni (quando disponibili) e delle feature determinanti nei processi di esclusione/selezione.

Questa sperimentazione ha permesso di validare l'efficacia del framework proposto in scenari concreti di **lavoro, selezione e governance etica**, mostrando come la piattaforma sia in grado di adattarsi anche a casi d'uso emergenti collegati all'impiego di AI generativa ad alto impatto sociale. I risultati sono stati prodotti in formato chiaro, tracciabile e completamente riproducibile, evidenziando la necessità di strumenti di auditing operativi, di trasparenza delle logiche decisionali e di introduzione di meccanismi di correzione nel ciclo di sviluppo e impiego dell'AI.

7.4 Valutazione delle Proprietà

In questa sezione vengono applicate al caso di studio le tre logiche formalizzate nei capitoli precedenti, con l'obiettivo di valutare in maniera sistematica alcune delle principali proprietà non-funzionali del sistema: well-formedness, fairness ed explainability. L'analisi è condotta secondo metodologie rigorose e criteri oggettivi, al fine di garantire una valutazione critica delle soluzioni proposte. I risultati empirici, ottenuti mediante l'implementazione e la sperimentazione sulla piattaforma di riferimento, verranno presentati e discussi nelle sotto-sezioni seguenti, con particolare attenzione all'identificazione di punti di forza, limiti e possibili aree di miglioramento delle strategie adottate.

7.4.1 Well-formedness del Dataset

Per garantire che un dataset sia "ben formato" (*well-formed*) e adatto all'addestramento di modelli di machine learning, si applica una verifica basata sulla decomposizione statistica delle sue feature. Sia

$$X \in \mathbb{R}^{m \times n}$$

la matrice dei dati, con m osservazioni e n variabili. Seguiamo questi passi:

1. **Standardizzazione:** Ogni colonna di X viene trasformata in media zero e varianza unitaria tramite Z-score normalization, per rendere omogenee le scale delle variabili.
2. **Independent Component Analysis (ICA):** Si calcola

$$S = W\,X,$$

dove $W \in \mathbb{R}^{n \times n}$ è la matrice di separazione studiata per massimizzare la non-gaussianità delle componenti. Il risultato $S = [S_1, \ldots, S_n]^\intercal$ contiene le *componenti indipendenti*.

3. **Kurtosi di Fisher:** Per ciascuna componente S_j, si valuta la kurtosi normalizzata

$$\kappa(S_j) \;=\; \mathbb{E}\Big[\tfrac{(S_j-\mu_j)^4}{\sigma_j^4}\Big] \;-\; 3,$$

con μ_j, σ_j media e deviazione standard di S_j. Le interpretazioni chiave sono:

- $\kappa(S_j) = 0$ indica *perfetta gaussianità*;
- $\kappa(S_j) > 0$ segnala *distribution leptocurtica* (code pesanti);
- $\kappa(S_j) < 0$ segnala *distribution platicurtica* (code leggere).

4. **Criterio di validità:** Si adopera la soglia empirica

$$|\kappa(S_j)| \;>\; 1 \quad \forall\, j = 1, \ldots, n.$$

Se *tutte* le componenti soddisfano $|\kappa(S_j)| > 1$, significa che nessuna direzione proiettata in S è prossima alla normalità. Questo garantisce che:

1. *Ogni* componente contiene informazione non ridondante,
2. il dataset non presenta variabili spurie o totalmente ridondanti,
3. il modello potrà apprendere pattern significativi senza overfitting su dipendenze lineari dominanti.

Figura 7.3: Valori di curtosi per ogni componente ICA. La soglia di accettazione $|\kappa| = 1$ è evidenziata in rosso.

Risultati: Visualizzazione della Curtosi per Componente ICA **Grafico dei valori di curtosi per ciascuna componente:**

Il grafico mostra, per le 12 componenti ottenute applicando FastICA ai dati standardizzati, i valori di curtosi centrata (Fisher) associati a ciascuna componente IC_j:

$$\kappa(IC_j) \;=\; \mathbb{E}\Big[\tfrac{(IC_j-\mu_j)^4}{\sigma_j^4}\Big] \;-\; 3.$$

1. Tutte le componenti superano la soglia $|\kappa| > 1$, indicando deviazioni significative da una distribuzione gaussiana.

2. I valori variano da $\kappa(IC_2) \approx 7.4$ fino a $\kappa(IC_{10}) \approx 55.1$, riflettendo differenze nella non-gaussianità delle sorgenti.

3. La componente IC_{12}, pur avendo la curtosi più bassa (≈ 4.3), resta comunque sopra la soglia minima, confermando l'indipendenza informativa di tutte le direzioni.

Secondo il **criterio di validità** adottato, il dataset è considerato *well-formed* se *tutte* le componenti ICA soddisfano $|\kappa| > 1$. Pertanto, il risultato conferma che il dataset è idoneo per l'addestramento: ogni componente estrae informazione non ridondante e statisticamente indipendente, riducendo il rischio di overfitting e migliorando la robustezza del modello.

Procedura di Test e Interpretazione (Versione Tecnica) La valutazione di well-formedness sfrutta un workflow strutturato in quattro fasi:

1. **Preprocessing e Standardizzazione:** rimozione delle colonne label, eliminazione dei NaN e scaling delle feature tramite `StandardScaler`, garantendo distribuzioni con media nulla e varianza unitaria.

2. **Decomposizione ICA:** applicazione di `FastICA` con `n_components` pari al numero di feature e `random_state=42` per risultati deterministici; si ottengono componenti indipendenti $S_1, \ldots, S_d$.

3. **Calcolo della Curtosi:** per ciascuna componente S_j si stima la curtosi centrata e normalizzata

$$\kappa(S_j) \;=\; \frac{\mathbb{E}\big[(S_j - \mu_j)^4\big]}{\sigma_j^4} \;-\; 3,$$

dove μ_j e σ_j sono media e deviazione standard di S_j.

4. **Verifica dei Threshold:** si dichiara il dataset *well-formed* se e solo se $\forall j \,:\, |\kappa(S_j)| > 1$. Il mancato superamento di questa condizione segnala la presenza di componenti prossime alla gaussianità, indicative di possibili dipendenze lineari o ridondanze informative.

Interpretazione dei Risultati Componenti con $|\kappa| \gg 1$ evidenziano distribuzioni leptocurtiche o platicurtiche, segnalando direzioni statistiche ad elevato contenuto informativo non-gaussiano, utili per l'inferenza in modelli predittivi robusti.“‘

Discussione Il controllo della curtosi tramite ICA, basato su soglie formalizzate ($|\kappa(S_j)| > 1$) e interamente tracciabile, fornisce una misura quantitativa e oggettiva della struttura informativa latente del dataset. Questo metodo, conforme alla definizione formale di funzione di AI Assurance esposta nel Capitolo 4, è facilmente integrabile in pipeline di assurance automatizzate e soddisfa i requisiti normativi sull'affidabilità e sulla qualità dei dati AI.

Nel caso analizzato, tutte le dodici componenti indipendenti hanno superato la soglia di validità, senza evidenza di valori di curtosi prossimi a zero. Ciò indica l'assenza di anomalie statistiche rilevanti e conferma, nella fase ICA, un elevato grado di indipendenza informativa tra le variabili del dataset.“‘

7.4.2 Fairness

La valutazione della *fairness* rappresenta uno dei pilastri del framework di AI assurance, perseguendo l'obiettivo di rilevare potenziali disparità nelle predizioni del modello tra diversi gruppi demografici, come richiesto dai principi normativi e di responsabilità etica discussi nei precedenti capitoli. Il quadro metodologico di riferimento è quello della *Demographic Parity*, definita come la differenza tra le probabilità di esito positivo ($P(\hat{Y} = 1)$) condizionata ai valori di variabili sensibili. Tale criterio fornisce una misura quantitativa del rischio di bias e della possibilità che il sistema di AI produca esiti discriminatori.

Nel caso di studio presentato, la verifica di fairness è stata interamente automatizzata ed implementata all'interno della dashboard AI Compliance, la quale guida l'utente lungo un workflow strutturato in fasi distinte, rendendo il processo trasparente, tracciabile e facilmente replicabile.

1. Configurazione iniziale — Selezione degli attributi sensibili e della variabile target La prima fase consiste nella specifica dei parametri di input necessari all'analisi (Figura 7.4). Tramite l'interfaccia, l'utente individua:

- le **colonne sensibili** sulle quali intendere valutare la fairness (ad es. `race`, `sex`);
- la **colonna target**, ovvero la variabile predetta dal modello (ad es. `label`);
- il **valore della classe target** che identifica l'esito di interesse (ad es. `1` per caso positivo).

Questa fase è fondamentale per garantire che la valutazione sia pertinente rispetto agli obiettivi di compliance richiesti e che nessun gruppo demografico rilevante venga trascurato.

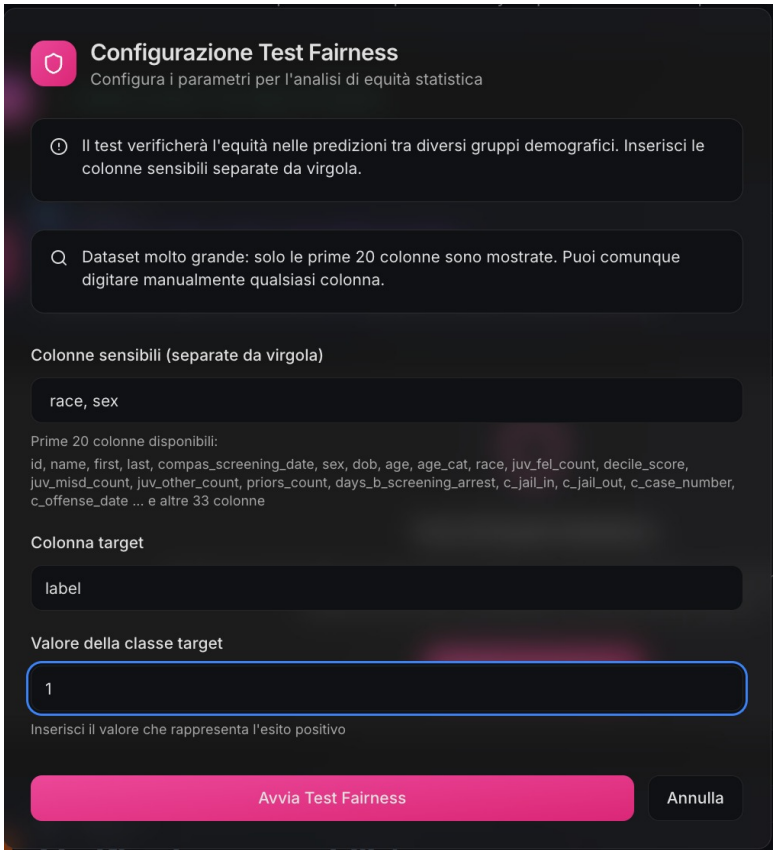

Figura 7.4: Configurazione degli attributi sensibili, della colonna target e del valore di etichetta nella dashboard di test Fairness.

2. Analisi automatica delle disparità — Calcolo delle metriche per ciascun attributo Successivamente, la dashboard esegue in modo autonomo il calcolo delle metriche di fairness definite. In particolare, per ogni attributo sensibile selezionato, viene determinata la *massima disparità* fra i tassi di outcome positivo osservati nei diversi gruppi:

$$\Delta(A) = \max_{i,j} \left| P(\hat{Y} = 1 \mid A = i) - P(\hat{Y} = 1 \mid A = j) \right|$$

dove A denota l'attributo sensibile e i, j iterano sui possibili valori/gruppi. Il risultato viene confrontato con una **soglia predefinita** (ad es. 10%), che rappresenta il limite massimo tollerabile per la disparità (Figura 7.5).

3. Visualizzazione e interpretazione dei risultati — Esempi per `race` e `sex` Ogni attributo sensibile analizzato dà luogo a una reportistica dettagliata:

- **Attributo `race`**: come mostrato in Figura 7.5, l'analisi include:
  - numero di gruppi individuati (ad es. 6: Other, African-American, Caucasian, Hispanic, Native American, Asian);

– proporzioni di esito positivo per ciascun gruppo;
– verifica del rispetto della soglia e conteggio dei gruppi conformi;
– rappresentazione grafica delle distribuzioni.

Nel caso di specie, solo 3 gruppi risultano conformi e la disparità massima (14.7%) eccede la soglia stabilita, portando ad un esito negativo del test sull'attributo `race`.

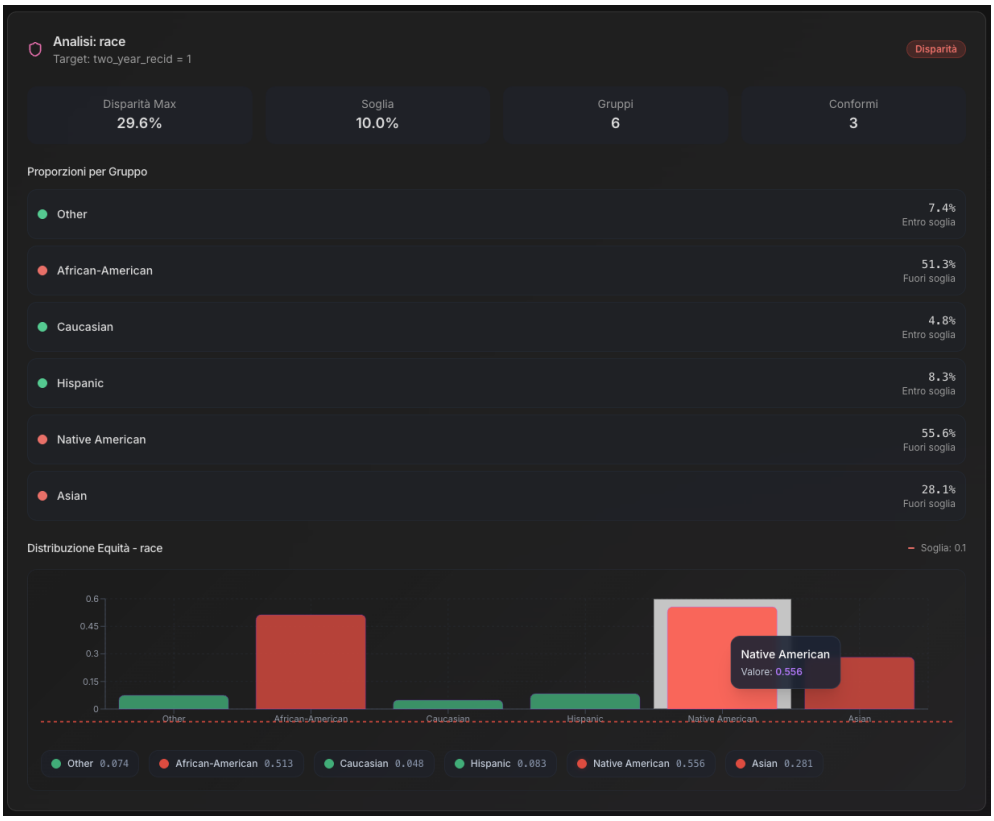

Figura 7.5: Analisi di fairness rispetto all'attributo `race`: la dashboard evidenzia i gruppi fuori soglia e la disparità massima osservata.

Questa disparità implica che:

– **Accesso diseguale alle opportunità**: candidati con qualifiche equivalenti hanno probabilità di selezione drasticamente diverse in base alla loro appartenenza etnica, violando il principio di equal opportunity;
– **Perpetuazione di bias storici**: il modello amplifica probabilmente distorsioni presenti nei dati di addestramento, replicando e istituzionalizzando pregiudizi sistemici;
– **Rischio di non conformità normativa**: la soglia del 10% superata configura una potenziale violazione dei requisiti di non discriminazione dell'AI Act europeo e delle normative antidiscriminatorie nazionali;

  - **Effetti cumulativi**: la discriminazione algoritmica può generare cicli di esclusione che, nel tempo, riducono ulteriormente la rappresentanza dei gruppi penalizzati, creando un *feedback loop* discriminatorio.

- **Attributo `sex`**: l'output della dashboard, illustrato in Figura 7.6, mostra che la disparità massima osservata tra maschi e femmine supera la soglia (10%): il gruppo `female` risulta evidenziato in rosso, indicando una situazione di non conformità. Di conseguenza, il test per `sex` è considerato non superato e si rende necessaria una revisione dei dati o del modello per eliminare il bias rilevato.

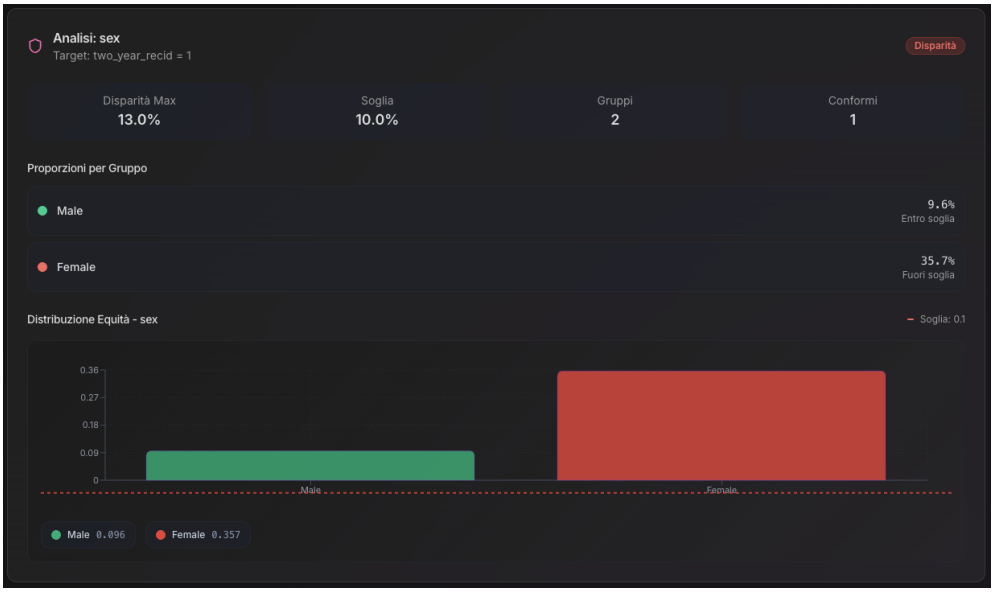

Figura 7.6: Analisi di fairness per l'attributo `sex`: entrambi i gruppi demografici sono entro la soglia di disparità ammissibile.

Questa disparità evidenziata per il gruppo `female` implica che:

- **Penalizzazione sistematica delle donne**: Le candidate presentano una probabilità di selezione inferiore rispetto ai colleghi maschi, nonostante qualifiche equivalenti, compromettendo il principio di pari opportunità nei processi decisionali automatizzati.

- **Riproduzione di bias di genere**: Il modello manifesta una tendenza a replicare e accentuare squilibri già presenti nei dati storici, favorendo involontariamente discriminazioni sistemiche contro il gruppo `female`.

- **Potenziale violazione normativa**: Il superamento della soglia di disparità del 10% configura una situazione di non conformità rispetto ai requisiti di non discriminazione imposti dall'AI Act europeo e dalle normative nazionali contro le discriminazioni di genere.

- **Effetti a lungo termine nella rappresentanza**: Il bias algoritmico può generare cicli di esclusione che riducono progressivamente la presenza femminile tra i selezionati, aggravando la disparità di rappresentanza e consolidando un meccanismo di feedback negativo nel tempo.

4. Criterio di superamento e logica aggregata Conformemente a quanto descritto nelle sezioni metodologiche, la logica di test adottata prevede che la **fairness dell'intero dataset sia garantita solo se tutti gli attributi sensibili analizzati risultano conformi** al vincolo di disparità stabilito. Anche una singola violazione comporta il fallimento della verifica e segnala la necessità di ulteriori interventi di debiasing o revisione dei dati/modelli. Ciò assicura un presidio rigoroso e trasparente, fondamentale per la gestione del rischio di bias sistemici in ambito AI compliance.

In sintesi, questa procedura consente di:

- rilevare automaticamente ogni forma di disparità rilevante a livello statistico;
- documentare in maniera riproducibile le condizioni di conformità o non conformità del dataset;
- fornire al contempo una rappresentazione visiva, quantitativa e totalmente integrabile nel ciclo di vita del modello, favorendo accountability e trasparenza verso stakeholder regolatori e aziendali.

7.4.3 Explainability

La valutazione della explainability costituisce un caposaldo del framework di AI assurance, con l'obiettivo di verificare in modo oggettivo quanto le decisioni assunte dal modello risultino trasparenti, interpretabili e giustificabili agli occhi di utenti, auditor e stakeholder regolatori. In linea con i principi dell'AI Act europeo e delle principali linee guida etiche, la piattaforma proposta integra meccanismi di analisi automatica della spiegabilità, traducendo requisiti normativi in metriche computazionali e visualizzazioni interattive.

Nel caso di studio presentato, la valutazione dell'explainability è stata condotta tramite un workflow strutturato e ripetibile all'interno della dashboard AI Compliance: l'utente può configurare le soglie di spiegabilità, selezionare il modello da analizzare e visualizzare sintesi numeriche (come l'Explainability Index) e grafici di feature importance locale e globale. Questo approccio rende il processo di verifica trasparente, documentabile e accessibile anche a profili non tecnici, consentendo una valutazione pratica e coerente con i vincoli imposti dalla normativa. L'analisi della spiegabilità così strutturata permette di evidenziare eventuali aree critiche e di supportare audit e processi decisionali basati su modelli AI, assicurando accountability e tracciabilità delle scelte algoritmiche.

Workflow di verifica: configurazione e parametri Analogamente a quanto previsto per le altre proprietà non funzionali analizzate, la valutazione della explainability è realizzata tramite un workflow sistematico. All'avvio del processo, la dashboard consente di caricare il modello predittivo di interesse, insieme al dataset di test che ne rappresenta il dominio operativo. In questa fase, l'utente individua:

- il **modello predittivo**, serializzato in un formato compatibile (ad es. `.h5`, `.pkl`), che ingloba la struttura e i pesi appresi durante la fase di training;
- la **matrice delle feature** di input (`X`), che rappresenta l'insieme delle variabili esplicative (ad es. caratteristiche demografiche, attributi comportamentali o altri predittori rilevanti);
- il **vettore degli output attesi** (`Y`), ovvero le etichette o i valori target associati agli esempi di test (ad es. classe prevista, score di rischio o probabilità stimata).

Questa fase di configurazione garantisce la tracciabilità e la riproducibilità delle analisi, assicurando che il modello sia sottoposto a verifica in condizioni rigorose e controllate.

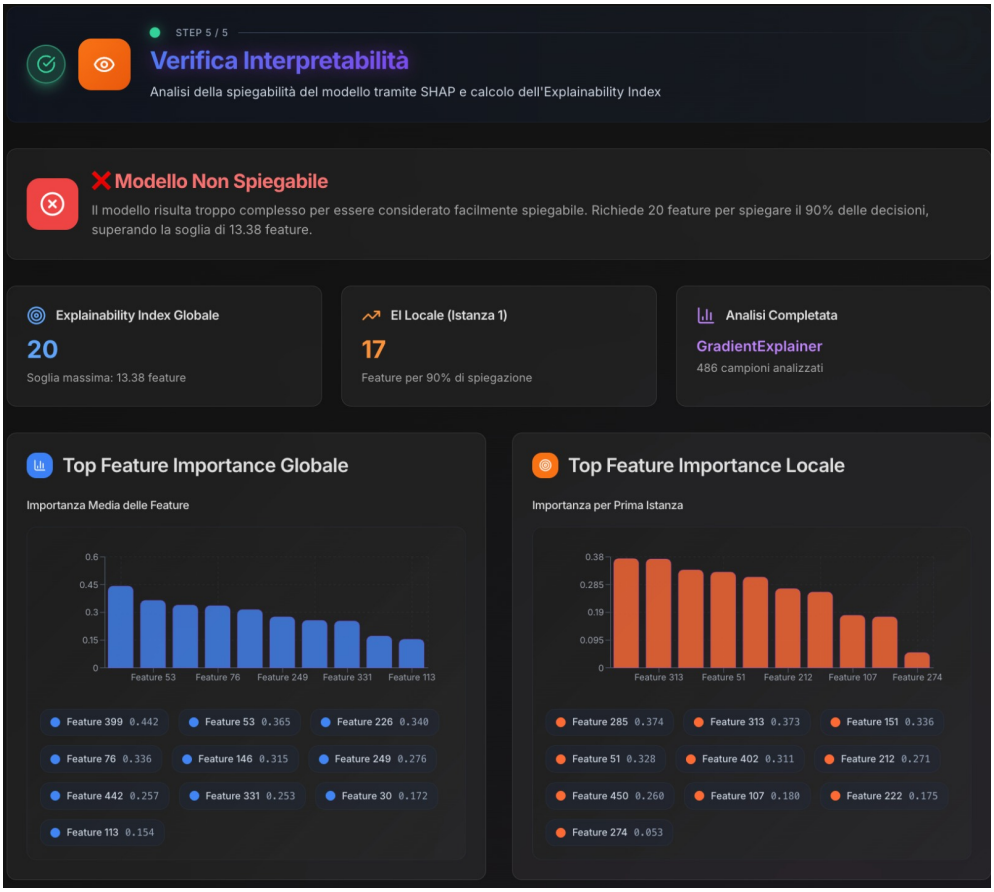

Figura 7.7: Dashboard di explainability: il sistema riassume la complessità esplicativa del modello attraverso indicatori sintetici (Explainability Index), visualizzazioni delle feature importance globali e locali, e output di analisi dettagliata sulle instance selezionate.

Calcolo automatico degli indici e visualizzazione dei risultati Successivamente alla configurazione dei parametri, la dashboard esegue in modo autonomo il calcolo delle principali metriche di explainability. In linea con le più recenti best practice, vengono computati i seguenti indicatori:

- **Explainability Index Globale**: misura il numero minimo di feature necessarie a spiegare una soglia prefissata (tipicamente il 90%) della variabilità decisionale del modello. Nel caso esaminato (Figura 7.7), ad esempio, sono richieste 20 feature per raggiungere il 90% di spiegabilità, superando la soglia massima accettabile (`13.38` feature).
- **Explainability Index Locale**: per ogni istanza campione selezionata, viene calcolato il numero di feature che contribuisce al 90% della spiegazione locale, evidenziando la complessità interpretativa nelle singole decisioni. Valori elevati indicano una spiegazione potenzialmente articolata o poco trasparente.

- **Feature Importance**: la dashboard propone grafici interattivi che visualizzano l'importanza media globale delle feature (global feature importance) e, separatamente, le importanze locali calcolate per specifiche istanze predittive di interesse (local feature importance). Questi strumenti consentono all'utente di identificare rapidamente le variabili a maggiore impatto sul comportamento del modello.

Analisi dei risultati: criteri di spiegabilità e soglie di accettazione Il giudizio di spiegabilità si basa sul confronto tra i valori calcolati dell'Explainability Index e una soglia teoricamente fondata, dipendente dalla dimensionalità dei dati in input (d) e tipicamente fissata come $\alpha \cdot \log_2 d$, dove α è un coefficiente empirico (ad esempio, $\alpha = 1.5$).

Nel caso analizzato, la dashboard segnala esplicitamente la condizione di **modello non spiegabile**: viene richiesto un numero eccessivo di feature per raggiungere una copertura esplicativa pari al 90% delle decisioni, con una soglia superata di circa il 50%. Di conseguenza, il modello viene classificato come "troppo complesso" per una spiegazione immediata e conforme ai parametri di auditabilità.

Discussione Questo risultato sottolinea il ruolo cruciale delle metriche oggettive nella valutazione di explainability. Tale approccio, oltre a fornire strumenti ripetibili e trasparenti per l'auditing, favorisce una maggiore accountability nella progettazione e nel rilascio dei modelli AI. In particolare, il mancato rispetto delle soglie di explainability può indicare la necessità di una revisione architetturale, di una selezione feature più selettiva, o dell'integrazione di ulteriori moduli di interpretabilità a beneficio degli stakeholder coinvolti.

Infine, la rappresentazione simultanea di indici sintetici, grafici di feature importance e output locali consente una diagnosi multilivello della trasparenza, coerente con i requisiti normativi e con le migliori pratiche di AI Assurance.

7.5 Discussione e validazione sperimentale della dashboard di AI Assurance

Il presente capitolo ha validato, tramite uno studio di caso reale, l'efficacia del framework di AI Assurance proposto: l'integrazione automatizzata dei controlli di well-formedness, fairness ed explainability ha permesso di rilevare problematiche concrete di bias e complessità nei modelli, superando le logiche di verifica tradizionali. In particolare, i risultati sperimentali evidenziano la capacità del sistema di:

- rilevare e segnalare carenze nella qualità e indipendenza dei dati (*well-formedness*);
- identificare disparità operative tra gruppi demografici (*fairness*) con metodi oggettivi e automatizzati;

- valutare la trasparenza e la spiegabilità dei modelli, indicando aree di miglioramento architetturale (*explainability*).

La gestione automatizzata degli errori e la tracciabilità delle analisi, unitamente a output esportabili, pongono la piattaforma in linea con i requisiti di auditabilità e accountability richiesti dalle normative europee.

Questa sperimentazione conferma come il framework di AI Assurance sia in grado di trasformare criteri formali e teorici in strumenti operativi solidi, a beneficio sia delle organizzazioni sia della governance etica dell'intelligenza artificiale, dimostrando la reale applicabilità e utilità della metodologia proposta.

Capitolo 8

Conclusioni

8.1 Riepilogo e Considerazioni Finali

Il lavoro presentato in questa tesi ha avuto come obiettivo la definizione e la sperimentazione di un approccio sistematico per la verifica delle proprietà non funzionali nei sistemi di intelligenza artificiale, nell'ottica di una compliance efficace rispetto alle più recenti esigenze normative e operative. Muovendo dal contesto attuale di espansione dell'AI e dai vincoli imposti dall'AI Act, sono stati individuati e formalizzati i criteri di well-formedness dei dati, fairness delle decisioni algoritmiche e explainability dei modelli, illustrando metodologie rigorose per la loro valutazione e certificazione.

Una parte centrale della tesi ha riguardato la progettazione e la realizzazione di un framework modulare di AI assurance, realizzato in stretta integrazione con la piattaforma Moon Cloud. Quest'ultima, approfondita nei capitoli tecnici, ha fornito l'architettura di riferimento e le logiche infrastrutturali necessarie alla configurazione, esecuzione e monitoraggio dei controlli di assurance. In particolare, sono state sviluppate probe specializzate, accessibili tramite una dashboard web, che permettono la verifica automatica delle proprietà studiando il comportamento di sistemi AI e datasets reali. Tutti i risultati, sia numerici sia visuali, sono stati tracciati e resi auditabili, contribuendo a costruire un workflow conforme, trasparente e replicabile.

L'analisi empirica condotta su casi studio concreti – come il dataset COMPAS per la fairness e il caso di recruiting automatizzato – ha permesso di validare l'efficacia delle soluzioni proposte, evidenziando punti di forza e limiti operativi sia delle metriche adottate sia delle pipeline di verifica. Nel complesso, la sinergia tra Moon Cloud e le componenti applicative sviluppate ha dimostrato come sia possibile integrare principi formali, strumenti automatici e dashboard intuitive al fine di elevare il livello di compliance e responsabilità nell'adozione di sistemi AI.

8.2 Lavori Futuri

Le attività svolte costituiscono una base solida per molteplici sviluppi futuri nell'ambito dell'AI Assurance. Un primo percorso riguarda l'estensione dei meccanismi di verifica verso la definizione di soglie "fuzzy", che superano il vincolo di parametri fissi permettendo l'adozione di criteri adattivi e più vicini alle situazioni reali. In quest'ottica, si apre la strada a pipeline capaci di gestire intervalli di accettabilità o decisioni basate su regole probabilistiche, ad esempio accettando risultati "abbastanza buoni" piuttosto che perfettamente conformi a una soglia rigida.

Un'altra direzione rilevante è la formalizzazione di proprietà e test specifici per modelli LLM: si tratta di definire nuove metriche e procedure per garantire robustezza, affidabilità e correttezza proprio nei sistemi basati su foundation models e AI generativa. In quest'ambito, l'uso stesso di LLM per l'automazione delle verifiche apre scenari inediti: ad esempio, sfruttando le capacità di ragionamento degli LLM per generare test case dinamici, individuare casi-limite o validare automaticamente la coerenza delle risposte rispetto a vincoli di fairness o trasparenza.

Infine, un'evoluzione chiave sarà l'integrazione dei framework di assurance con metriche e controlli dinamici, in grado di adattarsi a modelli agentici evolutivi e ai nuovi contesti organizzativi. Lo sviluppo di strumenti collaborativi tra data scientist e compliance officer, unito a interfacce di controllo intelligenti e adattive, favorirà soluzioni sempre più robuste, inclusive ed eticamente trasparenti.

Bibliografia

[1] National Institute of Standards e Technology. *AI Risk Management Framework (AI RMF 1.0)*. 2023. URL: https://www.nist.gov/ai-risk-management-framework.

[2] Institute of Electrical e Electronics Engineers. *IEEE Standard for Addressing Ethical Concerns during System Design (IEEE 7000-2021)*. 2021. URL: https://ieeexplore.ieee.org/document/9443135.

[3] ISO/IEC. *ISO/IEC 23894:2023 Artificial Intelligence – Guidance on risk management*. 2023. URL: https://www.iso.org/standard/79244.html.

[4] Alex Della Bruna. "Design e sviluppo di un sistema distribuito avanzato per verifiche di security assurance". In: *Master Thesis* (2024).

[5] European Commission. *Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. COM(2021) 206 final, Accessed: 2025-03-29. 2021. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206.

[6] Haoyu Lei, Amin Gohari e Farzan Farnia. "On the Inductive Biases of Demographic Parity-based Fair Learning Algorithms". In: *arXiv preprint arXiv:2402.18129* (2024). URL: https://arxiv.org/abs/2402.18129.

[7] Zeyu Tang e Kun Zhang. "Attainability and Optimality: The Equalized Odds Fairness Revisited". In: *arXiv preprint arXiv:2202.11853* (2022). URL: https://arxiv.org/abs/2202.11853.

[8] Xianli Zeng, Edgar Dobriban e Guang Cheng. "Fair Bayes-Optimal Classifiers Under Predictive Parity". In: *arXiv preprint arXiv:2205.07182* (2022). URL: https://arxiv.org/abs/2205.07182.

[9] Equal Employment Opportunity Commission. *Uniform Guidelines on Employee Selection Procedures*. https://www.ecfr.gov/current/title-29/subtitle-B/chapter-XIV/part-1607. Accessed: 2025-07-03. 1979.

[10] Luca Giuliani, Eleonora Misino e Michele Lombardi. "Generalized Disparate Impact for Configurable Fairness Solutions in Machine Learning". In: *arXiv preprint arXiv:2305.18504* (2023). URL: https://arxiv.org/abs/2305.18504.

[11] Sam Bell et al. “The Possibility of Fairness: Revisiting the Impossibility Theorem in Practice”. In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. ACM, 2023, pp. 1535–1545. DOI: 10.1145/3593013.3594046. URL: https://par.nsf.gov/servlets/purl/10437287.

[12] European Commission. *Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. COM(2021) 206 final, 2021/0106(COD). 2024. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

[13] Julia Angwin et al. “Machine Bias: There’s software used across the country to predict future criminals. And it’s biased against blacks”. In: *ProPublica* (2016). URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[14] Zihan Wang et al. “TransformerSHAP: Efficient SHAP Explanations for Transformer Models via Attention Sampling”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2023).

[15] Yixuan Tan, Qasim Ahmed e Rohit Patel. “LIME-AutoML: Robust Local Interpretability Integrated into Automated Machine Learning Pipelines”. In: *Journal of Artificial Intelligence Research* 79 (2024), pp. 45–67.

[16] Luca Giuliani, Eleonora Misino e Michele Lombardi. “FairSHAP: Combining Explainability and Group Fairness via Shapley-based Attribution Metrics”. In: *arXiv preprint arXiv:2305.18504* (2023). URL: https://arxiv.org/abs/2305.18504.

[17] Arnaud Van Looveren e Janis Klaise. “Multi-Objective Counterfactual Explanations”. In: *Proceedings of the 2021 AAAI Conference on Artificial Intelligence* 35.10 (2021), pp. 9324–9332. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17108.

[18] Dylan Slack et al. “Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods”. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 180–186.

[19] European Data Protection Supervisor. *Generative AI and the EUDPR. First EDPS Orientations for ensuring compliant and responsible use by EUIs*. 2024. URL: https://www.edps.europa.eu/system/files/2024-06/24-06-03_genai_orientations_en.pdf.

[20] AlgorithmWatch. *Bias nei sistemi GenAI nel recruitment e nella pubblica amministrazione*. 2024. URL: https://www.sos-svizzera.ch/2024-08-19-sos-ch-presa-di-posizione-ia-2-2/.

[21] Oliver Wyman Forum. *Divario di genere e bias nell'uso di GenAI nel lavoro*. 2024. URL: https://zalaconsulting.com/intelligenza-artificiale-parita-genere-tecnologia-alleata-nuova-frontiera-della-discriminazione.

[22] UNESCO. *L'UGUAGLIANZA DI GENERE e i rischi di amplificazione bias nei sistemi AI*. 2024. URL: https://www.unesco.it/wp-content/uploads/2024/11/italiano.pdf.

[23] Commissione Europea. *Segnalazione ufficiale. Database incidenti AI ad alto rischio*. 2024. URL: https://documenti.camera.it/leg19/dossier/pdf/AT026.pdf.