UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

TESI DI LAUREA

Machine Learning Fairness: un'analisi empirica dello stato della pratica

RELATORE

Prof. **Fabio Palomba**

Università degli studi di Salerno

CANDIDATO

Carmine Ferrara

Matricola: 052250090

Anno Accademico 2021/2022

When you think of A.I., it's forward-looking.
But A.I. is based on data, and data is a reflection of our history - Joy Buolamwini

ABSTRACT

L'applicazione di soluzioni di Machine Learning è sempre più utilizzata per rispondere alla più ampia varietà di problemi in contesti d'utilizzo reali, tra i più noti aspetti qualitativi di Accuracy, Performances, Sicurezza, etc. che caratterizzano un modulo di machine learning, è sempre più rilevante valutarne le caratteristiche etiche ed intervenire in maniera adeguata qualora si manifestino problematiche discriminatorie. Negli ultimi anni all'interno della comunità scientifica, in particolare nell'ambito dell'Ingegneria del Software per l'Intelligenza Artificiale, si sta riscontrando una notevole crescita dell'aspetto fairness come tematica di ricerca, tuttavia ciò che ancora risulta ancora essere mancante, è proprio tener conto di come le pratiche lavorative di chi si approccia quotidianamente allo sviluppo di soluzioni ML intensive fair-critical, possano agevolare e migliorare le stesse attività di ricerca. Tenendo conto di questa mancanza, questo lavoro di tesi, si è posto l'obiettivo principale di proporre alla ricerca una visione empirica di quali siano gli aspetti caratterizzanti del trattamento di fairness in ambito lavorativo, cercando contestualmente di sensibilizzare gli esperti del settore all'adottare sempre più processi di sviluppo fair-oriented standardizzati. Analizzando i risultati dello studio, ottenuti a mezzo di un Survey su larga scala, si osserva come la Fairness anche allo stato della pratica sia un aspetto strettamente dipendente dal dominio applicativo e strettamente dipendenti dallo stesso restano sicuramente validi definizione e approcci alla misurazione dei livelli di fairness noti in letteratura. Tuttavia viene anche evidenziato l'utilizzo approcci tecnico pratici molto interessanti, quali l'applicazione di metodologie empiriche, tecniche di ottimizzazione dei dati, o l'utilizzo di indicatori statistici di correlazione. Per gli esperti del settore, fairness continua ad essere ad oggi un aspetto meno standardizzato rispetto ad altre specifiche non funzionali, considerando però come il livello di maturità nel trattamento di fairness stia progressivamente aumentando. Le aziende stesse suggeriscono come praticamente fairness sia vista come un aspetto che evolve e migliora durante il ciclo di vita di un modulo di machine learning, evidenziando come strategie fair-oriented specifiche che impattino sull'evoluzione dei dati raccolti, e/o la validazione del modello siano cruciali nel trattamento di Fairness in approcci di sviluppo basati su pipeline quali MLOps, tra l'altro viene evidenziata la cruciale importanza di figure professionali come Manager o Esperti di Etica per il trattamento di fairness durante lo sviluppo di un modulo di machine learning. Da quanto emerso è possibile inoltre costatare, come la stretta correlazione tra dominio applicativo e fairness di un modulo di machine learning è probabilmente anche un'ottima chiave di lettura per la didattica accademica.

Indice

Elenco delle figure

Elenco delle tabelle

CAPITOLO 1

Introduzione

1.1 Motivazioni e Obiettivi

Al giorno d'oggi la costante e continua crescita applicativa dei sistemi di intelligenza artificiale, nei più svariati ambiti professionali, ha indotto l'intera comunità scientifica nell'ambito IT a porsi nuovi quesiti ed obiettivi che permettano la realizzazione di soluzioni dagli alti standard qualitativi ed immuni a numerose tipologie di vulnerabilità. Negli ultimi anni in particolar modo, si osserva come le soluzioni AI-Intensive, ed in particolar modo i moduli di machine learning, necessitino della formalizzazione di standard di sviluppo che tengano conto delle specifiche di sviluppo e delle caratteristiche intrinseche (e.g. la forte correlazione tra dati di training e l'algoritmo utilizzato) che differenziano in termini strutturali questa particolare tipologia di soluzioni. In particolare, questa tipologia di mancanze, ha portato ad un progressivo e costante interesse ed avvicinamento dei ricercatori nell'ambito dell'ingegneria del software alle problematiche dei moduli AI-Intensive [1] ed alla nascita di una nuova branca di studio che appunto accomuna i processi ingegneristici al contesto specifico, ovvero l'ingegneria del software per l'intelligenza artificiale. Dalla progettazione dei primi processi di analisi, progettazione e sviluppo dei moduli AI-Intesive, l'ingegneria del software ha portato alla nascita di nuovi standard di sviluppo veri e propri che caratterizzano in modo specifico l'intero ciclo di vita di una soluzione AI-Intensive, tra i più innovativi e famosi, rientra sicuramente l'approccio evolutivo di standard basati su pipeline quali MLOps [2]. Un aspetto chiave di questa particolare tipologia di standard di sviluppo è senz'altro

l'analisi intrinseca e continua dei differenti aspetti qualitativi che un modulo ML intensive deve rispettare, e proprio tra gli oramai noti e standardizzati aspetti di qualità, e.g. Sicurezza, Accuracy, Performances etc. [3], il mondo della ricerca osserva come negli ultimi anni i moduli di machine learning, siano sempre più soggetti a nuove tipologie di vulnerabilità, che portano il modulo stesso all'operare in maniera eticamente scorretta, con conseguenti risultati discriminatori per particolare tipologie di utenti appartenenti a particolari gruppi di utenti, meno o mal rappresentati rispetto l'intero gruppo di addestramento, i così detti gruppi minoritari [4].

Quello che si evidenzia dai primi studi nell'ambito è che l'aspetto fairness di un modulo di machine learning è un concetto difficilmente definibile e misurabile in modo univoco data la stretta correlazione con l'ambiente applicativo delle specifica soluzione [5]. Osservando con occhi critico lo stato dell'arte, si osserva come molti studi, abbiano portato notevoli migliorie verso il comune obiettivo di progettare e realizzare soluzioni ML-Intensive eticamente corrette, da strategie specifiche per l'analisi dei requisiti e delle migliorie dei dati fino alle più specifiche strategie di testing ed analisi, d'altra parte però è facilmente riscontrabile come le specifiche strategie progettate e realizzate dal mondo della ricerca, abbiano ad oggi poco riscontro con quelle che sono le pratiche adottate o le esigenze specifiche nel trattamento di fairness di chi professionalmente realizza soluzioni ml-intensive in ambito aziendale. Tenendo conto di tale mancanza, questo lavoro di tesi, a mezzo di un Survey su larga scala, si pone quindi l'obiettivo di investigare a 360° su quali siano i punti di forza e debolezza del trattamento di fairness in ambito lavorativo all'attuale stato della pratica, in modo tale da fornire, non solo una visione strutturata dei plausibili spunti di ricerca futuri per la tematica specifica, ma anche la consapevolezza che in ambito professionale la tematica dell'etica di un modulo ML-Intensive, sia qualcosa di estremamente concreto, che necessita di standard e tecniche di sviluppo adeguate al pari di altre caratteristiche qualitative che appunto rendono un modulo ML-Intensive adatto ad essere applicato in un contesto d'uso reale.

1.2 Risultati

Tramite l'analisi empirica condotta, quello che questo lavoro di tesi ha effettivamente osservato è che allo stato della pratica, il concetto di software fairness necessiti ancora di un mirato supporto della ricerca al fine di maturare in modo opportuno al fine di essere trattato sistematicamente al pari di altre specifiche non funzionali di un modulo di machine learning,

in particolare si evidenzia come:

- Nonostante l'ampio catalogo di definizioni ed approcci noti per il trattamento alla fairness, il trattamento formale dei requisiti etici di una soluzione intelligente sia attualmente poco standardizzato, in particolare si notano visioni poco concordanti riguardo gli approcci noti in letteratura e l'utilizzo di altre strategie specifiche non totalmente coperte dagli attuali studi;
- Fairness sia ancora un requisito poco maturo per essere considerato effettivamente primario rispetto ad altre specifiche più note quali accuracy o sicurezza, viene anche osservato come questo principio sia variabile a seconda dei specifici domini applicativi;
- La progettazione di strategie specifiche o il riutilizzo di quanto già a disposizione in letteratura nel trattamento di fairness, sia effettivamente necessario e utile, soprattutto adoperando modelli di sviluppo evolutivi, e.g. MLOps;
- La suddivisione di responsabilità e il coinvolgimento di esperti possa essere un fattore estremamente rilevante nel trattamento di fairness in ciascuna delle fasi di sviluppo;
- Le aziende siano consapevoli che non è più prescindere dal trattamento e risoluzione di vulnerabilità discriminatorie nelle soluzioni prodotte, infatti è osservabile come la maggior parte degli esperti coinvolti ritengano che le proprie aziende trattino le problematiche etiche a differenti livelli di maturità. In generale si osserva come gli esperti si collochino massivamente a livelli basilari della scala proposta, ma osservando il campione nella sua interezza, è intuibile come nei prossimi anni ci possa essere una notevole tendenza al miglioramento.

Ovviamente lo studio condotto, non fornisce risultati definitivi per trattare le problematiche etiche in maniera definitivamente corretta, ma data la natura ad ampio raggio dell'analisi, si cerca appunto di lasciare punti di osservazione che possano essere utili per future analisi che tengano conto sempre di più di ciò che i professionisti del settore necessitino o ritengano utile.

1.3 Struttura della tesi

Il documento di tesi è strutturato nel seguente modo:

- **Capitolo 2 - Background**: Il capitolo di background riporta una visione generale dei macro-ambiti in cui il lavoro di tesi si colloca, fornendo una panoramica ad alto livello dei concetti basilari di ingegneria del software ed intelligenza artificiale, fino ai concetti specifici di ingegneri del software per l'intelligenza artificiale direttamente richiamati nei capitoli successivi;

- **Capitolo 3 - Stato dell'Arte**: Il capitolo stato dell'arte, fornisce dettagli di ricerca sulla tematica cardine del lavoro di tesi ovvero la Software Fairness nei modelli di intelligenza artificiale, si forniscono dettagli su definizioni ed esempi noti per il trattamento di fairness che hanno impattato la ricerca negli ultimi anni, per poi concludere con una panoramica di alcuni studi specifici (related works) inerenti la software fairness nell'ambito dell'intelligenza artificiale e dell'ingegneria del software;

- **Capitolo 4 - Design**: Il capitolo di design e progettazione, fornisce dettagli su quelli che sono gli obiettivi di ricerca del lavoro di indagine empirica, per poi passare ai dettagli di progettazione e diffusione del Survey sottomesso ai lavoratori dell'ambito;

- **Capitolo 5 - Analisi**: Il capitolo di analisi, raccoglie tutte le procedure utili all'elaborazione dei dati ottenuti a seguito della chiusura del Survey, passando dalle metodologie di data cleaning e pre-processing effettuate sul dataset originale, per poi passare alla vera e propria analisi dei dati e generalizzazione dei risultati con relativi punti di discussione e implicazioni derivanti;

- **Capitolo 6 - Conclusioni**

- **Bibliografia**

CAPITOLO 2

Background

Al giorno d'oggi è sempre più frequente adottare soluzioni che facciano uso di moduli intelligenti all'interno dei più svariati ambiti lavorativi, ed è oramai risaputo come tali soluzioni debbano rispettare tutta una serie di standard qualitativi, affinché possano essere ritenuti attendibili ai loro scopi. Tuttavia recenti studi hanno dimostrato un nuova gamma di difetti che evidenziano una serie di vulnerabilità, fin ora non riscontrate all'interno dello sviluppo software (tra cui, come verrà approfondito in seguito in questo capitolo, quelle dovute alla la Software Fairness), legati all'operare in maniera imparziale ed equa nel loro contesto di utilizzo [4]. Per capire come il mondo dello sviluppo software (ed in particolare quello delle soluzioni "intelligenti"), siano influenzati dagli aspetti di equità o un qualsiasi altro aspetto qualitativo in generale è doveroso introdurre gli aspetti essenziali che si pongono alla base dello sviluppo di tali applicativi.

2.1 Ingegneria del Software

Prima di tutto è senz'altro necessario far riferimento all'ingegneria del Software, disciplina che nasce proprio in risposta alle problematiche di sviluppo di prodotti software di qualità in un preciso tempo, con uno specifico budget [6]. L'ingegneria del Software si pone l'obiettivo di applicare tutta una serie di attività che facciano dello sviluppo software un vero e proprio processo ingegneristico [7]. Le attività cardine della materia, cosi come definite da Bernd Bruegge e Allen Dutoit, [6], sono:

- La *modellazione*: capacità standard di focalizzarsi sui dettagli rilevanti ignorando tutto il resto, tramite svariate strategie di raffinamento e astrazione;
- Il *problem solving*: l'utilizzo di modelli per la ricerca di soluzioni a specifici problemi;
- L' *acquisizione di conoscenza*: la raccolta di singoli dati e informazioni di uno specifico dominio, per poi formularne informazioni e conoscenza prima di applicare standard di sviluppo;
- La *ricerca di un razionale*: cioè la ricerca delle motivazioni e delle necessità che si pongono a priori dello sviluppo di un tool software;

Si osserva quindi come effettivamente un tool software per essere progettato in maniera congrua a molteplici aspetti qualitativi - ad esempio: sicurezza, manutenibilità e adattabilità (o qual si voglia tipologia di attributo non funzionale) - che soddisfino le aspettative del cliente che lo commissiona, sia necessario applicare un processo standard di ingegnerizzazione. Nello specifico, anche problematiche legate all'emergente tematica dell'equità e dei vincoli di imparzialità devono essere trattati sullo stesso piano di un qualsiasi altro attributo qualitativo e la ricerca si sta muovendo sempre di più in questa direzione [4].

Ovviamente, è molto complesso cercare di riassumere in poche parole ogni peculiarità dell'ingegneria del software, molteplici sono gli aspetti di disciplina che caratterizzano un prodotto software, e.g. il suo ciclo di vita, la sua manutenzione ed evoluzione, e tutti gli aspetti correlati alla sua gestione. Lo stesso testo di riferimento [6], definisce l'ingegneria del software, come un attività complessa, non come un algoritmo standard, quindi è importante osservare come essa si adatti in maniera dinamica, agli specifici problemi, e a seconda dei casi faccia uso in maniera oculata di strumenti di vario tipo, come la sperimentazione e la misurazione, il riuso di pattern o l'evoluzione incrementale dei sistemi [6] e altre variegate tecniche e metodologie che possano offrire pronta risposta alla più ampia gamma di problematiche e necessità specifiche. Volendo porre un esempio più specifico di come l'ingegneria del software vada a specializzarsi nei contesti specifici di sviluppo, è possibile far riferimento all'ampiamente utilizzato sviluppo software Object-Oriented, che specializza un *tradizionale* ciclo di sviluppo software in sei attività di sviluppo [6]:

- *Raccolta e Analisi dei requisiti*, grazie alle quali gli ingegneri del software, analizzano il problema con il cliente al fine di definire, i confini del dominio applicativo;

- *System Design*, fase in cui gli ingegneri del software analizzano il problema e lo dividono in piccoli pezzi, al fine di selezionare strategie generali di design per il problema proposto;
- l'*Object Design*, fase in cui vengono selezionate soluzioni dettagliate per ogni *sotto-problema*;
- l'*Implementazione del sistema*, che corrisponde solo ad uno degli ultimi passi del ciclo di sviluppo, durante la quale gli sviluppatori *traducono* la soluzione al problema in codice sorgente;
- Il *Testing*, fase in cui gli sviluppatori cercano differenze tra il sistema implementato e i suoi modelli sulla base di ben definiti campioni di dati di input.

Come osservato, l'ingegneria del software, offre anche tutta una serie di attività gestionali che interessano ed influenzano un canonico progetto software. Le attività di gestione si focalizzano sulla pianificazione di un progetto software, il monitoraggio del suo status di avanzamento, il tracciamento dei suoi cambiamenti e la coordinazione delle risorse, al fine di consegnare un prodotto di alta qualità in relazione a quelli che sono i vincoli a cui esso è soggetto, come ad esempio quelli di tempo e di costo [6].

Altra branca dell'ingegneria del software, che da attività finale di un canonico modello ciclo di vita del software è diventa un processo a se stante che caratterizza l'evoluzione del prodotto per tutta la sua durata di vita è la *manutenzione*, macro-processo che include tutte quelle attività che occorrono dopo la consegna del sistema al cliente. La manutenzione è un aspetto complesso, ma sempre più necessario al fine di garantire il successo dei sistemi software, sempre più orientati al cambiamento [6]. Basti pensare che già nel 1974, uno dei più noti studiosi del campo Henry Lehman, osservava l'importanza dell'evoluzione e della manutenzione, con la formulazione delle prime leggi sull'evoluzione del software. Le leggi di Lehman, sono ancora oggi dei principi cardine degli studi sull'evoluzione del software, e se si pensa anche solo alle prime due, si intuisce l'importanza del processo di manutenzione [8] :

- *Prima Legge di Lehman*: I programmi di tipo evolutivo, devono necessariamente essere adattati *al cambiamento*, altrimenti essi diventano progressivamente meno soddisfacenti;

- *Seconda Legge di Lehman*: Così come un programma (sistema software) evolve, così la sua complessità aumenta, a meno che del lavoro non venga fatto al fine di manutenerla e ridurla.

Per concludere, in maniera opportuna l'introduzione all'ingegneria del software è doveroso far riferimento alle metodologie agili e allo sviluppo incrementale, che nel corso degli ultimi decenni stanno prendendo sempre più piede nel contesto ingegneristico. Secondo le più note fonti presenti in letteratura, questa tipologia di approcci e metodologie, sono caratterizzati da alcuni principi innovativi che si discostano quasi del tutto dall'idea di ciclo di sviluppo tradizionale. Ian Sommerville, nel suo volume *Software Engineering*, riassume il core dei processi agili, facendo riferimento alle seguenti caratteristiche[9]:

- Il rilascio di nuove release dei sistemi caratterizzate da piccoli cambiamenti ogni due o tre settimane circa;
- Il coinvolgimento continuo dei clienti al fine di ottenere rapidi feedback e tracciare i continui cambiamenti;
- La riduzione della documentazione, *al minimo indispensabile*, preferendo l'uso di comunicazioni informali, piuttosto che meeting formali con documenti scritti;

2.2 Intelligenza Artificiale e Machine Learining

L'Intelligenza Artificiale ed in particolare i moduli di Machine Learning, stanno diventando sempre di più una parte fondamentale delle applicazioni commerciali e dei progetti di ricerca nell'ambito IT. In particolare si osserva come i moduli addestrati di maggior successo, siano realizzate a partire dalla generalizzazione di esempi noti (basi di conoscenza), sulle quali i moduli stessi vengono addestrati, al fine di produrre, sulla base di dati di input forniti, un output desiderato senza che l'utente umano dia informazioni aggiuntive. La branca del Machine Learning, nella quale vengono racchiusi tutti gli algoritmi che si basano su tali tuple di *input-output* viene definita come **Apprendimento Supervisionato** [10]. Dalla stessa fonte possiamo osservare come esempi di apprendimento supervisionato di interesse possano essere:

- L'identificazione di *topix* da un blog o un sito internet;
- l'identificazione di pattern di accesso anomali in un sito web;

- la suddivisione dei clienti di uno shop per preferenze similari.

Soprattutto negli ultimi anni, si ci sta accorgendo che l'applicazione combinata di discipline come la statistica, la teoria dell'informazione e il machine learning, stanno portando alla creazione di una scienza sempre più solida, con una ferma base matematica, e a tool sempre più potenti. In particolare, se si fa riferimento al learning supervisionato tecniche e algoritmi come alberi di decisione, regressione lineare, regressione logistica, clustering, sono alcune delle tecniche più utilizzate per la progettazione di componenti AI, per innumerevoli campi applicativi, ed in particolare quello che ne risulta dall'applicazione di una generica tecnica di learning supervisionato, su un dataset di addestramento, è definito come processo di classificazione [11].

Per capire praticamente la problematica di classificazione in ambito machine learning, si può pensare di far riferimento al comune esempio di identificazione delle mail di spam, tecnica automatica di filtering basata su di grandi data set ampiamente utilizzata dai gestori di mailing. In pratica, sulla base dei dati (strutturali, storici e similari) a disposizione di mail già contrassegnate o meno come "SPAM" è possibile addestrare un modello, al fine di classificare una nuova mail che un generico utente riceve come "SPAM MAIL" oppure "NON SPAM MAIL". Volendo dare quindi una definizione semi formale: il problema di classificazione del Machine Learning consiste nell'*individuare una* ***categoria*** *(ad esempio MAIL di SPAM oppure MAIL NON DI SPAM) per una* ***nuova osservazione*** *(e.g. una nuova mail ricevuta), sulla base di precedenti* ***dati di addestramento*** *contenenti informazioni già classificati secondo* ***categorie note*** *(ad esempio archivio di mail già classificate come SPAM oppure NON SPAM a disposizione del modulo addestrato*".

Questa problematica è particolarmente attenzionata dalla ricerca, dato che è molto complesso generare classificatori che non soffrano di alcuni problemi noti in letteratura, come ad esempio, la stretta dipendenza dal dataset di partenza [11], il così detto fenomeno del *garbage-in,garbage-out*, cioè la presenza di Bias nei dataset di addestramento, che poi tenderanno a riflettersi all'interno delle predizione dei moduli di previsione stessi [12]. Volendo porre un esempio pratico, nell'ambito dell'etica del software, è facile che un classificatore addestrato con un dataset, non immune a dei bias su specifici attributi sensibili (quali razza o sesso), possa essere addestrato in modo tale da effettuare predizioni imparziali, spesso rivolte a favore degli individui del campione di addestramento dei gruppi di maggioranza (ovvero quegli individui che posseggono il valore più ricorrente dell'attributo sensibile) [12] .

2.2.1 Algoritmi di ricerca e Algoritmi Genetici

Il machine learning è sicuramente uno degli aspetti più caratterizzanti dell'intelligenza artificiale, al fine di fornirne una panoramica più ampia è interessare considerare anche gli algoritmi di ricerca. Essi per definizione sono usati per restituire informazioni memorizzate all'interno di strutture dati o ricercare soluzioni in un complesso spazio di ricerca formalmente definito sulla base del dominio del problema, il tipo di informazioni gestite dagli algoritmi di ricerca possono essere formate da valori discreti e continui. Di algoritmi di ricerca ne esistono in letteratura di vari tipi, basati su vari approcci: algoritmi basati sull'uso di una funzione di costo (e.g. depth-first, bread-first e cost-uniform first...), basati sull'utilizzo di euristiche (e.g. A* greedy best-first), algoritmi di ricerca locale (e.g. hill-climbing search) ecc., ma tra i più usati in ambito di ricerca ci sono sicuramente gli algoritmi di tipo evolutivo ed in particolare quelli di tipo genetico. Gli algoritmi genetici, sono algoritmi basati sui principi della selezione naturale e della genetica, introdotti negli anni '70 da J.Holland e ispirati alle teorie dell'evoluzione degli esseri viventi [13]. Gli algoritmi genetici astraggono lo spazio del problema come una popolazione di individui, e provano ad esplorare le caratteristiche degli individui, "producendone" di nuovi in maniera iterativa. I GA evolvono la popolazione da individui iniziali (spesso generati casualmente) in individui di alta qualità, laddove ogni individuo rappresenta una soluzione al problema di interesse codificata in stringa. La qualità di ogni individuo è misurata da una funzione matematica detta funzione di fitness che è formulata in modo tale da valutare in maniera quantitativa la bontà di un individuo, a seconda di alcune caratteristiche qualitative definite dallo sviluppatore. A seconda del problema la funzione di fitness, può essere di massimizzazione, di minimizzazione, a singolo obiettivo o multi obiettivo, e spesso rappresenta l'ostacolo più grande nella progettazione di un algoritmo genetico.

Durante ogni generazione, tre operatori di base della genetica sono applicati in sequenza con una certa probabilità: selezione, crossover e mutazione [13] . I passaggi base di un algoritmo di ricerca genetico sono:

1. Generazione randomica di una popolazione di n individui (rappresentanti di soluzioni randomiche al problema);
2. Valutazione di ogni individui tramite la funzione di fitness;
3. Selezione di due individui per generarne di nuovi (nuova generazione), le tecniche

di selezione possono essere diverse, tra le più famose ci sono: la roulette wheel e l'approccio a torneo;

4. Con una certa probabilità, viene applicata un operazione di cross over al fine di mischiare singole parti degli individui selezionati per generarne di nuovi (anche qui ne esistono vari tipi). Se il crossover non viene applicato, in sostituzione vengono copiati "i genitori";

5. Con una certa probabilità ai nuovi individui vengono applicate operazioni di mutazione (ovvero vengono cambiati uno o più dati della stringa casualmente);

6. Gli individui generati, vengono aggiunti alla popolazione, al fine di far ripartire l'algoritmo;

7. Se uno degli individui generati soddisfa le condizioni di accettazione e di arresto dell'algoritmo, allora si ritorna la soluzione migliore trovata fino a quel momento;

8. Altrimenti l'algoritmo riparte dal passo 2;

2.3 Ingegneria del Software nell'Intelligenza artificiale

Si può osservare come negli ultimi decenni, l'intelligenza artificiale e l'ingegneria del software, si siano evolute separatamente, oggi giorno, si osserva però come la ricerca attuale, stia portando alla specifica e alla costituzione di nuovi studi e approcci allo sviluppo di soluzioni AI-Intensive, che tengano proprio conto dell'intersezione che c'è tra l'ingegneria del software e l'intelligenza artificiale [1]. L'Intelligenza artificiale odierna fa riferimento a sistemi che sulla base degli input che ricevono in considerazione devono assumere rischi, fare predizioni o assumere comportamenti in risposta a specifici problemi [14], in dettaglio, nell'era dei computer dalle elevate prestazioni e dei Big Data, molte soluzioni AI-Intensive, sono sviluppate in risposta ai bisogni che la quotidianità sociale necessita, ma come noto in letteratura, molti sistemi software di grandi dimensioni, non sono privi di bug, ed in particolare i sistemi di Intelligenza Artificiale e di Machine Learning non fanno eccezione. Per questa tipologia di sistemi, bug di progettazione o addestramento, possono essere causa di crush di sistema, output errati fino all'esecuzione troppo lenta che non rende possibile l'utilizzo di tali soluzioni nell'ambiente di lavoro [15]. In casi critici gli errori dei sistemi intelligenti, possono addirittura portare alla morte di chi anche involontariamente interagisce con essi, il caso più noto è quello della ciclista Elaine Herzberg, morta investita da un'auto

con pilota automatico, per errore di valutazione del modulo di guida [14]. Quindi come è possibile produrre soluzioni AI Intensive per il mondo reale, che tengano conto di tali problematiche? L'applicazione dell'ingegneria del software cerca in qualche modo di dare risposta a questa tipologia di problematica, in dettaglio, si cerca di includere metodi di Requirement Engineering, Design Engineering, Code Engineering e Project Management, che possono essere di supporto, per lo sviluppo di Sistemi Ai-Intensive efficienti. La ricerca stessa ne fa utilizzo, ed infatti molti ambiti di studio sono nati dall'intersezione tra SE e AI, in particolare vale la pena citare l'Agent Oriented Software Engineering oppure l'Ambient Intelligence [16]. L'Agent Oriented Software Engineering, si concentra sullo sviluppo di soluzioni che siano, intelligenti, agili e pro-attive. In dettaglio il suo scopo principale è quello di sviluppare soluzioni AI-intensive tramite dei riadattamenti stessi delle tecniche di Ingegneria del Software, per lo sviluppo di Agenti di piccoli e grandi dimensioni. Nel dettaglio risultano essere di rilievo l' *Agent UML*, per la progettazione dei moduli agente, oppure metodi di specifica e valutazione formali dei sistemi, i quali hanno lo scopo intrinseco di validare gli obiettivi, i comportamenti di un singolo agente, e soprattutto le interazioni nei sistemi multi agente [1]. l'Ambient Intelligence invece si pone lo scopo di progettare ambienti di addestramento (secondo tecniche specifiche dell'ingegneria del software) che siano sensibili ed adattabili agli stimoli esterni e in conseguenza, sistemi reattivi che siano informati circa i bisogni, le abitudini e le emozioni degli utenti per supportare il loro lavoro quotidiano[1].

Qualsiasi branca o studio di ricerca a cui si voglia far riferimento (come gli esempi riportati) definita nella così detta "intersezione tra intelligenza artificiale e ingegneria del software", non può prescindere dall'analizzare quelli che sono i requisiti non funzionali di qualità che una soluzione AI-Intensive deve rispettare. In particolare in letteratura [3], si può osservare come, ad esempio, il Machine Learning sia soggetto a specifici vincoli di qualità quali:

- *Accuracy and Performances*, come l'output di un agente risulta "corretto" se paragonato alla realtà;
- *Fairness*, Requisito che si pone l'obiettivo di rendere gli algoritmi di ML più **imparziali** e indipendenti da bias di dati;
- *Transparency*, ovvero la capacità di dimostrare come i risultati elaborati da un modulo intelligente siano affidabili e trasparenti, ricostruendo le fonti di partenza;

- *Security and Privacy*, *Testability* e *Reliability* del modulo addestrato.

Per progettare, realizzare e controllare questi aspetti qualitativi essenziali per un modulo di intelligenza artificiale, la ricerca è enormemente incentrata nello studio di questi aspetti. In particolare, si nota come questi requisiti non funzionali, siano particolarmente attenzionati dall'ingegneria dei requisiti, branca dell'ingegneria del software che si incentra nella specifica, l'analisi, la verifica e la validazione dei requisiti di un sistema software [17], la cui ricerca ricerca sta incentrando buona parte dell'effort nello studio di aspetti quali fairness, privacy, sostenibilità e modificabilità anche per tecniche di Machine Learning e per lo sviluppo di soluzioni AI-Intensive in generale [3]. In particolare, la letteratura afferma come l'industria dell'intelligenza artificiale è particolarmente incentrata nello sviluppo di soluzioni di Machine Learning e basate sugli approcci Data Driven, ed una delle principali aree di interesse per lo sviluppo di questo tipo di soluzione è il settore dell'Healthcare, particolarmente caratterizzato dalla mancanza di standard di progettazione e dalla continua evoluzione dei dati a disposizione[17], per far fronte a questa tipologia di problematiche, l'ingegneria del software ed in particolare l'ingegneria dei requisiti, pongono l'accento su nuove metodologie e tecniche che toccano vari processi di un sistema AI-Intensive: la specifica e l'analisi dei suoi requisiti, la validazione del modello (e quindi delle sue specifiche), la documentazione e il management dell'intero processo di sviluppo dei requisiti formulati. Le sfide principali che la ricerca ha davanti in quella che è stata definita *intersezione* tra intelligenza artificiale e ingegneria del software sono:

- *Lo Skill Gap*: la necessità di creare lo giusto spirito di collaborazione aziendale tra Data Scientist e Ingegneri del Software;
- *Il Data Gap*: Ovvero la necessità di rendere disponibili (big) dataset necessari alla realizzazione di soluzioni AI complesse;
- *L'Engeeniring Gap*, ovvero la necessità di creare prototipi generalizzabili dei sistemi AI, con il giusto supporto all'intero ciclo di vita della Soluzione AI-Intensive;

2.3.1 Machine Learning Life Cicle e ML Pipeline

Uno dei contributi generali che l'ingegneria del software cerca di dare all'intelligenza artificiale è proprio quello di sistematizzare la progettazione e lo sviluppo dei suoi modelli. In tal senso lo sviluppo di una soluzione AI-Intensive, così come un generico progetto di Machine Learning, può essere sistematizzato definendo opportuni processi ingegneristici e

di conseguenza un vero e proprio ciclo di vita della soluzione che si sta progettando. Ogni progetto AI-Intensive, quindi può avere uno specifico ciclo di vita, e concentrandosi nel del machine learning, Burkov e Andriy affermano che un progetto ML-Intensive è caratterizzato inizialmente dalla comprensione e dall'analisi dello scope applicativo e degli obiettivi di business il modello [2], successivamente il primo passo che porta alla nascita di un modello di machine learning è capire se il modello stesso ha dei precisi obiettivi. Gli obiettivi di un modello ML-Intensive sono ovviamente diversi da quelli di business e dettano le basi di progettazione del modello stesso, un generale obiettivo di un modello ML, deve essere formalizzato in modo tale da tener conto [2]:

1. Di cosa il modello riceve come input;
2. Di cosa genera come output;
3. Dei criteri di accettabilità (o inaccettabilità) del comportamento del modello;

Il ciclo di vita di un modello di machine learning può essere sistematizzato tramite i seguenti passaggi [2]:

1. Definizione degli obiettivi;
2. Collezione e preparazione dei dati;
3. Ingegnerizzazione delle feature;
4. Training del modello;
5. Evoluzione del Modello;
6. Deployment del Modello;
7. Fase operativa del modello e monitoraggio;
8. Manutenzione del modello.

Approcci più innovativi allo sviluppo ML-Intensive, sottolineano come aspetti di monitoraggio, architecturing e monitoring continuo, siano essenziali per far evolvere un modello di pari passo con il suo ambiente di utilizzo, uno degli approcci innovativi che cerca di rispondere a queste esigenze è la specializzazione ML di DevOps, che appunto prende il nome di MLOps.

I passaggi del ciclo di vita di vita di un modello ML-Intensive, possono essere anche organizzati e automatizzati in modo tale da configurare una vera e propria **pipeline di machine learning**, che viene definita come una sequenza di operazioni su un dataset che vanno dal suo stato iniziale, fino al rilascio del modello e la sua evoluzione. Una pipeline può includere tra i passaggi di preparazione dei dati, imputation dei dati mancanti, estrazione delle feature, data augmentation e model training[2]. In pratica, una delle innovazioni più forti portati dalla formalizzazione di una pipeline di machine learning, è che quando un modello di machine learning è deployato in produzione, in realtà è deployata l'intera pipeline[2] (rispondendo in maniera *automatizzata* alle necessità evolutive del modello. Da notare che una pipeline di machine learning è generalmente ottimizzata quando i suoi hyperparameters di configurazione sono ottimizzati (Hyperparameters tuning) [2].

2.3.2 Piattaforme per Machine Learning Pipelines e MLOps

Dare la definizione di Machine Learning Pipeline, fa capire come lo sviluppo e il deploying di applicazioni ML-Intensive, è un processo che va ben oltre i dati collezionati e i modelli addestrati e fare predizioni. Come osservato da Zhou et al. connettere queste parti ignorando la fase di manutenzione del modello, costituisce un enorme debito tecnico [18]. Costruire un workflow dal data pre-processing fino al porre il modello in run nel contesto di utilizzo, può facilmente essere un processo dispendioso in termini di tempo e soprattutto non privo di errori, tra l'altro una produzione efficiente e affidabile è fondamentalmente critico per molti contesti reali, si pensi a casi d'uso quali la guida automatica oppure l'health care[18].

In risposta a queste esigenze critiche, sono nate piattaforme di machine learning, che in maniera embedded forniscono un ciclo di vita manageriale end-to-end per applicazioni ML-Intensive, core primario di queste applicazioni cercano di accorpare sistemi stand-alone che in maniera specificano si assumono responsabilità di task quali: data-preprocessing, model training, model evolution e messa in servizio. Goal primario di queste piattaforme è quello fornire una soluzione generica per molteplici casi d'uso di sviluppo, collegando e configurando componenti indipendenti per specifiche problematiche. Con questa tipologia di configurazione, un generico workflow ML può essere orchestrato e convertito in una vera e propria Pipeline di Machine Learning, la cui esecuzione è supportata da queste piattaforme [18]. Oltretutto il livello di affidabilità, scalabilità, abilità di continuous training, fornito da questi strumenti, offre la possibilità di adattare e ed evolvere i dati oppure incrementare i

cambiamenti frequentemente [18].

Con la creazione di Tools che permettono di ingegnerizzare una pipeline di machine learning, hanno portato nell'ambito dell'ingegneria del software alla nascita di vere e proprie culture e pratiche di sviluppo specifiche. la più nota nel mondo dell'intelligenza artificiale è sicuramente *MLOps*, che specializza nell'ambito AI, la più generica cultura di sviluppo software *DevOps*.

Per comprendere bene un processo di sviluppo ML-intensive, basato sulla filosofia ML-Ops, è necessario comprendere innanzitutto, quali sono i processi generali che entrano in gioco durante il ciclo di vita di una soluzione ML-Intensive. Innanzitutto è necessario distinguere quelli che sono i processi di sviluppo, che racchiudono [19]:

- Raccolta e manipolazione dei dati grezzi;
- Analisi e processing dei dati per il training;
- Costruzione, training e testing del modello;
- Overfitting e tuning del modello;
- Validazione del modello.

da quelli che vengono invece definiti processi derivanti dall'uso del modello nell'ambiente di utilizzo, chiamati appunto processi operazionali eseguiti da figure professionali dedicate, in particolare possono essere definiti processi operazionali per una soluzione ML-Intensive [19]:

- Deployment del modello;
- Monitoring del modello;
- Formulazione di statistiche di reporting;
- Feedback retrival al team di manutenzione;

DevOps, o development operations, è una cultura più generale di sviluppo ingegneristica, che si riferisce a set di pratiche che combinano i generici processi di sviluppo software, con quelli operazionali, in modo tale da creare un set comune di pratiche che assicurino un'accelerazione dei tempi di sviluppo, oltre che una rapida risposta alle necessità di cambiamento data dal monitoraggio continuo del tool rilasciato, incorporando quindi in maniera naturale

quello che è il concetto di continuous delivery[19]. Con l'utilizzo di un tipico workflow DevOps, si osserva come i costi totali di manutenzione si riducono in maniera considerevole dato che la manutenzione stessa è integrata in maniera nativa all'interno dell'efficiente meta-processo [19]. Similarmente MLOps adotta i principi di DevOps e li specializza per i modelli di machine learning piuttosto che per un generico prodotto software. In particolare l'approccio ML-Ops, crea un meta-modello di sviluppo iterativo, che integra il ciclo di sviluppo, di responsabilità dei data scientists e dei Machine Learning engineers, con i processi operazionali del team addetto, al fine di assicurare continuous delivery di modelli di machine learning dalle alte prestazioni[19].

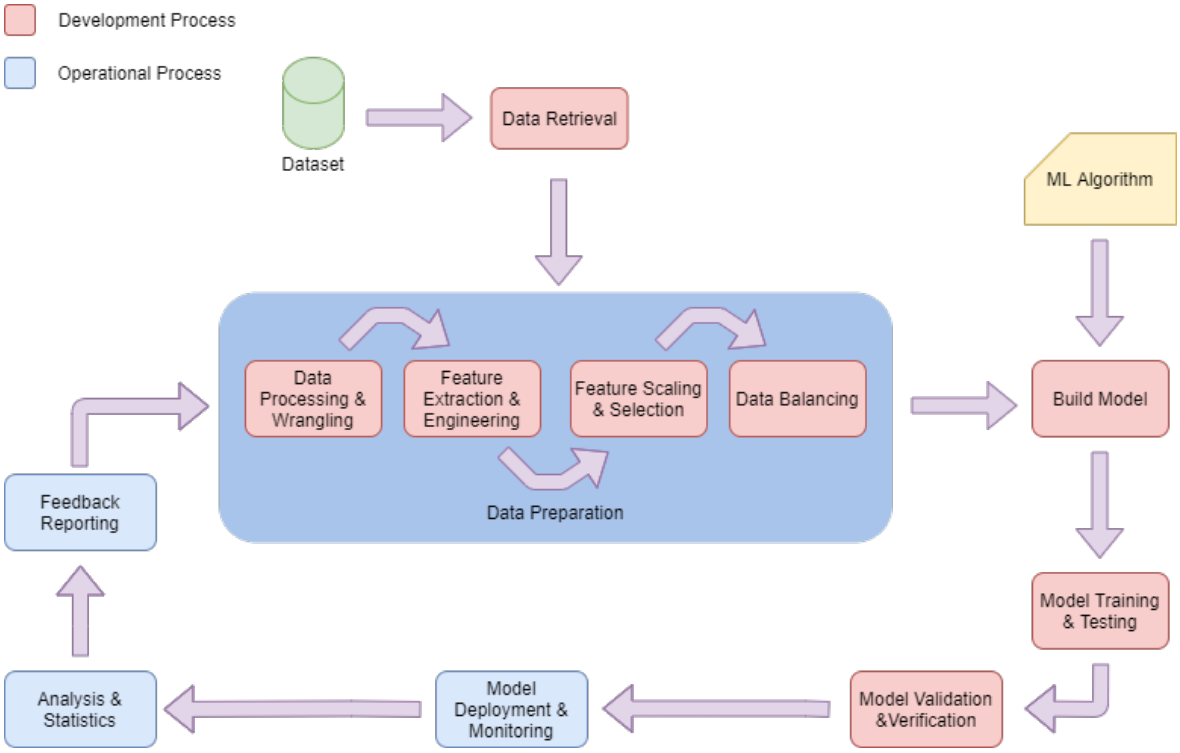

Figura 2.1: Esempio generico di pipeline di machine learning basato su MLOps

CAPITOLO 3

Stato dell'arte

3.1 Definizioni e Metriche di Software Fairness

Dopo aver introdotto il background in cui il lavoro di tesi si si colloca, è senz'altro necessario introdurre quello che è l'aspetto principale che caratterizzerà il lavoro illustrato nei successivi capitoli, ovvero la Software Fairness e il codice eticamente corretto. Horkoff introduce alla fairness come un requisito non funzionale dei moduli di machine learning, per il quale le attività di ricerca attuale sta investendo molto. In particolare si cerca di rendere gli algoritmi di machine learning più imparziali, non solo cercando di rimuovere features sensibili (come razza o sesso), ma cercando soprattutto di definire e implementare soluzioni AI-Intensive che tengano conto del livello di fairness richiesto dal dominio di applicazione[3].

Implementare un algoritmo eticamente corretto, significa definire formalmente cosa si intende per fairness, e cercare in conseguenza di misurare come ciò viene implementato[3]. Dato che concetto di Fairness è relativamente nuovo per la comunità scientifica, e soprattutto requisiti sociali e principi legali spesso ne evidenziano differenti caratteristiche, sono emerse differenti definizioni del concetto e un numero speculare di metriche per misurarne il livello nei sistemi software[12]. Prima di cercare di definire formalmente il concetto però è necessario cercare di capire in che maniera un sistema software mostra dei comportamenti scorretti e nello specifico , e come la comunità scientifica si pone rispetto ad esso.

Recenti studi, inerenti gli aspetti di qualità e il successo del processo ingegneristico del software, hanno dimostrato che i sistemi software presentano un nuovo tipo di vulnerabilità, correlate appunti alla loro abilità di operare in maniera imparziale (fair) e priva di pregiudizi. Da dove nasce, quindi il problema della software fairness? [4] come espresso in letteratura, il livello di fairness è strettamente correlato al concetto di Bias (pregiudizi algoritmici) che un sistema software ha al suo interno, questi comportamenti erronei possono emergere da vari aspetti, soprattutto appresi dai dati di addestramento (per quanto concerne le soluzioni AI-Intensive), ma anche per specifiche di requisiti incomplete, design povero, bug di implementazioni o interazioni errate tra componenti. Esistono differenti esempi interessanti di come fair bias nei sistemi software abbiano portato a spiacevoli inconvenienti, tra i più famosi vale la pena citare:

- *Software di Recruiting*: Gli specialisti di Machine Learning di Amazon, hanno reso noto che nel 2015, il loro nuovo tool di recruiting per lo sviluppo o altre posizioni tecniche, non offriva opportunità lavorative in maniera imparziale rispetto al sesso dei candidati, questo perché tale sistema era addestrato con dei dati di un periodi di 10 anni antecedente al 2015, per il quale la prevalenza degli impiegati tecnici dell'azienda è stata maschile, il gender e altri fattori (quali razza, oppure lo stesso linguaggio che gli impiegati tecnici maschili hanno adottato negli anni), sono stati classificati come feature sensibili che hanno addirittura portato Amazon, a chiudere il progetto, secondo quanto stabilito dalla loro versione ufficiale [20];

- *Healthcare*: Gli ospedali statunitensi per anni hanno utilizzato un software di predizione, per le cure mediche, che nel tempo è arrivato a "preferire" i pazienti di razza bianca, rispetto a quelli di colore. Tale comportamento "unfair" è da attribuire, purtroppo ad un bias di dati che considerava più "conveniente" la popolazione bianca, dato che quest'ultima era soggetta a spese mediche maggiori rispetto a quelle di colore, quindi "secondo il tool" preferendo tali individui su un campione di più di 200 milioni di abitanti, ci sarebbero stati maggiori guadagni per gli ospedali pubblici americani. Tale assunzione però si è convertita nella pratica nel considerare le persone di colore, maggiormente in salute rispetto quelle di carnagione chiara [21];

- *Crimine*: Nel 2014, si è osservato come un tool (denominato Correctional Offender Management Profiling for Alternative Sanctions, meglio conosciuto con l'acronimo di COMPAS), di predizione di responsabili di futuri crimini tra i più usati in Florida tra il 203 e il 2014, il cui funzionamento si basa sul confronto di analisi facciali partendo

da un dataset di immagini di fotografie di criminali condannati, avesse la tendenza a giudicare le persone di colore come più più predisposte a crimini violenti quali, la causa di questa tematica è ancora oggi molto discussa, ma si ritiene che la causa principale, sia nuovamente nei dati con cui il modello veniva addestrato, nel quale erano presenti feature sensibili, quali razza o sesso [22];

- *Traduzioni Automatiche*: Si è riscontrato come Google Traduttore, il più popolare motore di traduzione al mondo, mostrava un bias legato al sesso. in particolare, se si traduce dall'inglese al turco la frase "She is an engineer, He is a nurse", ne risulta un inversinone di soggetti: "He is an engineer, She is a nurse", quasi ad indicare come le professioni tecniche, come un generico Ingegnere, in Turchia sia automaticamente associato al sesso [22];

- *Generazione automatica di sottotitoli*:Su Youtube, se si seleziona la traduzione automatica, si osserva come effettivamente la traduzione del video risulti essere maggiormente accurata con voce maschile, rispetto a voce femminile, inoltre si osserva come che per lingue come l'inglese o l'arabo, ci siano delle differenze qualitative circa il risultato della traduzione, per alcuni specifici dialetti [23].

Dagli esempi citati si osserva, quindi, come ricercatori e ingegneri del software producano sempre di più software di qualità, in risposta alla necessità di prendere delle decisioni eticamente corrette, che riguardano sempre di più la vita umana[24]. Di conseguenza, si nota come un tool software o una soluzione AI-Intensive utilizzata su larga scala, non possa più presentare vulnerabilità che vadano ad inficiare il suo livello di Fairness. In risposta a questa nuova issiue ingegneristica, ne deriva in maniera naturale che è necessario come applicare definire formalmente il concetto di Software Fairness e conseguenti processi standard di misurazione. Negli ultimi anni questa problematica, ha attirato l'attenzione di ricercatori nell'ambito dell'intelligenza artificiale, l'ingegneria del software e la comunità legislativa, con più di venti differenti notazioni di Software Fairness proposte e ovviamente quello che ne deriva è che non è che non è possibile darne un'unica definizione specifica, ma è necessario specializzare il concetto in riferimento ad una specifica problematica analizzata [5]. Ponendo l'attenzione al problema di Classificazione del Machine Learning, già definito nel precedente paragrafo di introduzione all'intelligenza artificiale, è possibile dare numerose definizioni di Fairness per un generico classificatore ML, sulla base di alcuni strumenti ampiamente utilizzati nell'ambito dell'Intelligenza Artificiale, ovvero: Le misure statistiche di validazione, le misure di similarità e distanza e il Casual Reasoning [5].

3.1.1 Definizioni di Fairness basate su Metriche Statistiche

Sahil Verma e Julia Rubin, introducono a questa tipologia di definizioni basate su metriche statistiche fornendo alcuni concetti di base [5]:

- Attributi sensibili o protetti: attributo di un dato dataset, sul quale si sta effettuando classificazione, per il quale il generico modulo AI di classificazione potrebbe produrre discriminazioni (come ad esempio gender o razza);
- Diretta conseguenza degli attributi sensibili possono essere i Gruppi Protetti, gruppi di individui che assumono un valore dell'attributo sensibile che potrebbe essere soggetto a discriminazione;
- Il valore reale di classificazione: ovvero il valore (categoria di appartenenza) assegnato un individuo del dataset assume sulla base base delle sue feature di riferimento;
- La probabilità di predizione [0-1]: ovvero la probabilità condizionata che un individuo appartenga ad una data categoria, sulla base dei dati che lo caratterizzano, i classificatori, per stimare queste probabilità fanno riferimento in generale a tutte le feature del dataset, e in casi particolari possono anche riferirsi ad attributi sensibili, per particolari scelte di implementazione;
- La decisione predetta [0-1]: ovvero la categoria di appartenenza dell'individuo determinata dal classificatore, sulla base del dataset.

Le metodologie statistiche di validazione di un classificatore, ed in particolare i classificatori binari (con due singole categorie di classificazione, per convenzione una positiva e una negativa), fanno utilizzo di questi concetti, un determinato individuo può risultare:

- *Reale Positivo - True Positive* (TP), se il valore reale di predizione e la decisione predetta dal corrispondono entrambi alla categoria "positiva";
- *Reale Negativo - True Negative* (TN), se il valore reale di predizione e la decisione predetta corrispondono entrambi alla categoria "negativa";
- *Falso Positivo - False Positive* (FP), se la decisione predetta corrisponde alla categoria "positiva", mentre il valore reale di predizione è la categoria "negativa";
- *Falso Negativo - False Negative* (FN, se la decisione predetta corrisponde alla categoria "negativa", mentre il valore reale di predizione è la categoria "positiva".

,

Determinando questi valori per ciascuno degli individui del dataset, è possibile estrarre numerose metriche utili alla validazione di un generico classificatore binario, le più famose sono note in letteratura sono:

...la *precision*:

$$\frac{\sum TP}{\sum TP + FP}$$

...e la *recall*:

$$\frac{\sum TP}{\sum TP + FN}$$

Tali formule, con altre metriche statistiche ampiamente riconosciute, sono spiegate nel dettaglio, nel paper di riferimento[5], vale la pena osservare che tali metriche possono essere generalizzate anche nel caso si stia validando un classificatore con N categorie.

Questa piccola introduzione ai concetti basilari di classificazione, permette quindi di introdurre a qualche definizione di Fairness tra le più utilizzate nell'ambito della classificazione, tra le più importanti e utilizzate, vale la pena citare (N.B. Le definizioni riportate prendono come dominio di applicazione i classificatori binari, così come gli attributi sensibili con due possibili valori)[5]:

1. *Group Fairness - Statistical Parity*: Un classificatore, soddisfa questa definizione di fairness, se individui di un gruppo protetto (G1), definito per un dato attributo sensibile (g), possano essere assegnati dal classificatore alla categoria positiva (che quindi il valore predetto d sia pari ad 1) , con la stessa probabilità degli individui del gruppo non protetto (G2), per il quale l'attributo sensibile assume valore non discriminante:

   $$P(d = 1|g = X1) = P(d = 1|g = X2)$$

   dove X1 e X2 sono rispettivamente il valore discriminate e non discriminate dell'attributo sensibile g;

2. *Predictive Parity - Outcome Test*: Un classificatore soddisfa questa definizione, se entrambi i gruppi, protetto e non protetto, hanno stessa probabilità di assumere valore reale di classificazione (Y), pari ad 1 (categoria positiva), data decisione predetta pari ad 1;

   $$P(Y = 1|d = 1\&g = X1) = P(Y = 1|d = 1\&g = X2)$$

3. *False Positive Error Rate Balance*: Un classificatore soddisfa questa definizione, se, la probabilità di essere classificati come appartenenti alla categoria positiva, (decisione predetta d = 1), pur appartenendo in realtà alla categoria negativa (valore reale di classificazione Y = 1), sia equivalente sia per individui appartenenti al gruppo protetto che per quelli appartenenti al gruppo non protetto.

$$P(d = 1|Y = 0\&g = X1) = P(d = 1|Y = 0\&g = X2)$$

Queste ed altre definizioni statistiche del concetto di Software Fairness, vengono illustrate più in dettaglio nel paper di riferimento [12], anche in termini di metriche di validazione statistica quali Precision e Recall. Quello che però è importante sottolineare è l'enorme dinamicità di queste definizioni statistiche, come si nota come gli esempi riportati (come altre definizioni statistiche di Fairness presenti in letteratura), modellano concetti di probabilità differenti, ognuno avente un significato semantico ben preciso, e per avere una buona idea del livello statistico di Fairness di un classificatore sotto analisi, è necessario identificare bene quali metriche tenere in considerazione. Come è buona pratica di questi studi statistici può essere anche opportuno considerare e incrociare i risultati più metriche per formalizzare considerazioni più attendibili.

3.1.2 Definizioni di Fairness basate su Metriche di Similarità

Continuando questo excursus, nell'analisi delle definizioni di Software Fairness, nell'ambito dei classificatori ML, è possibile è possibile notare, come le definizioni statistiche, ignorino largamente tutti gli attributi di classificazione di un soggetto (sia X l'insieme degli attributi non sensibili di un generico individuo per cui si vuole effettuare classificazione), a meno di quelli sensibili G. Ad esempio, la Statistical Parity può determinare un classificatore come *Fair*, senza tener conto delle evidenze dei valori sensibili (G), tralasciando il valore di tutte le altre features (X) del dataset. Quello che ne evince è che tali misure, per quanto di largo utilizzo, possono definire unfair un classificatore, senza però tenere conto di altri vincoli di implementazione[5]. Per venire incontro a questa problematica, il paper di riferimento, propone altre definizioni che non fanno utilizzo del concetto di probabilità, ma osservano appunto la similarità statistica tra i singoli individui, tenendo anche conto degli attributi sensibili[5].

1. *Casual Discrimination*: Un classificatore soddisfa questa definizione, se produce lo stesso risultato di classificazione (d - decisione predetta), per ogni coppia di individui con i medesimi attributi X (non sensibili): dati due individui m ed f, tale definizione è rispettata se rispetta la seguente implicazione;

$$X_m = X_f \;\&\&\; G_f! = G_M \rightarrow d_m = d_f$$

2. *Fairness Through Unawareness*: un classificatore soddisfa questa definizione, se per ogni coppia di individui, con lo stesso insieme di attributi non sensibili, si ottiene la stessa decisione predetta, in altre parole, nessun attributo sensibile è coinvolto nel processo di classificazione:dati due individui I1 e I2, il classificatore rispetta questa definizione se vale la seguente implicazione;

$$X_1 = X_2 \rightarrow d_1 = d_2$$

3.1.3 Definizioni basate sul concetto di Casual Reasoning

Oltre le definizioni basate sulla statistica matematica, il paper di riferimento, riporta anche qualche definizione, basata sul concetto di Casual Reasoning: Processo di individuazione di relazioni causale (dipendenze cause/effetto tra attributi), molto utilizzato per costruire Classificatori e altri algoritmi ML [5]. L'output di un processo di Casual Reasoning per un classificatore ML, è solitamente un Casual Graph, un grafo aciclico, orientato, che connette le singole feature del dataset in relazione causa/effetto. Caratteristica interessante di un casual graph per un classificatore ML, è la presenza di un nodo terminale con solo archi entranti che rappresenta la decisione predetta dal classificatore (d). Facendo utilizzo di un Casual Graph è possibile identificare facilmente i percorsi di dipendenza (causa/effetto) che caratterizzano le scelte del classificatore corrispondente.

Il paper definisce come componenti di un Casual Graph [5], anche:

- I *Proxy Attribute*: attributi di un casual graph utilizzati per derivarne altri, che spesso sono utilizzati per ottemperare a necessarie trasformazioni matematiche all'interno di del processo di classificazione;

- I *Resolving Attribute*: attributi di un casual graph influenzati da attributi di tipo sensibile (quindi interconnessi con i medesimi), in maniera non discriminatoria;

Anche per il Casual Reasoning, il paper riporta qualche definizione di Fairness, in questo caso direttamente individuabili dalla composizione strutturale di un Casual Graph, relativo al classificatore sotto analisi [5].

1. *Conterfactual Fairness*: un classificatore rispetta questa definizione, se nessun risultato predetto d non discende in alcun modo (tramite percorsi di casualità) da un attributo identificato come sensibile;

2. *No Unresolved Discrimination*: Un classificatore rispetta questa definizione, se nel Casual Graph di riferimento, non ci sono path da attributi sensibili ai risultati predetti, a meno che nel path non siano identificabili *Resolving Attribute*;

3. *No Proxy Discrimination*: Un classificatore rispetta questa definizione, se nel Casual Graph di riferimento, non ci sono path da attributi sensibili ai risultati predetti, "bloccati" da *Proxy Attribute*, in altri termini, gli attributi sensibili non devono incidere nelle decisioni del classificatore per effetto di processi di trasformazione applicanti la strategia dei Proxy Attribute;

3.2 Software Fairness come tematica di ricerca

Autori	*Obiettivi*	*Soluzione*
* ***Paper***: Bias in Machine Learing Software: Why? How? What to Do? [24]		
Chakraborty et al.	Identificare principali cause di fair bias in un datasets.	Produzione dell'algoritmo Fair-Smoot per il Data Balancing di feature sensibili
* ***Paper***: Ignorance and Prejudice in Software Fairness [25]		
M.Zhang e Mark Harmon	Bilanciare dati e features di un dataset, al fine di elaborare un modello efficiente di Machine Learning Fair Oriented	Analisi dei dati statistica, sulla base di differenti datasets, circa il bilanciamento ottimale di feature e campioni di dati al fine di ottenere livelli di fairness ottimali secondo opportune metriche
* ***Paper***: Diversity Data Selection under Fairness Constraint [26]		
Moumoulidou at al.	Analizzare e valutare la complessità del problema della data-diversity per soluzioni ML-Intensive	Dimostrazione formale dell'NP-Completezza dell'approccio fair Max-Min Diversification e creazione di alcune approssimazioni dello stesso
** ***Paper***: Software Fairness [4]		
Yuriy Brun e Alexandra Meliou	Valutare il concetto di Software Fairness come un requisito non funzionale prioritario al pari di altri aspetti di qualità	Definizione di approcci e analisi dello status di avanzamento circa l'adozione di tecniche e metodologie specifiche per ogni fase di un canonico ciclo di vita di un prodotto software

Autori	*Obiettivi*	*Soluzione*
** ***Paper***: "Fairness Analysis" in Requirement Assignments [27]		
Finkelstein at al.	Identificare un processo standard di identificazione di specifica dei requisiti fair-oriented, in relazione con definizioni di fairness spesso contrastanti	Analisi dei requisiti correlati da fair trade-off modellata come problema di ricerca, con vincoli specifici dettati da differenti approcci matematici di Software Fairness
** ***Paper***: Fairness Testing: Testing Software for Discrimination [28]		
Yuriy Brun at al.	Definire nuovi approcci di testing fair oriented al fine di misurare se e in che modo i programmi effettuano discriminazioni	Definizione dell'approccio Themis, al fine di generare ed eseguire test suite per valutare, secondo alcune specifiche definizioni di fairness, l'impatto di feature plausibilmente discriminanti in un tool
*** Intelligenza Artificiale **Ingegneria Del Software**		

Tabella 3.1: Software Fairness Related Work

Si osserva come nell'ambito dell'intelligenza artificiale e del machine learning i fair-bias possono influenzare la creazione di dataset di addestramento, l'addestramento dei moduli di apprendimento (che di conseguenza soffriranno a loro volta di bias), e i risultati stessi di un processo di addestramento possono essere potenzialmente discriminanti per i gruppi minoritari [29]. Per tutti questi motivi, la comunità di ricerca, nell'ambito dell'intelligenza artificiale è sempre più propensa a studiare e eliminare questi bias di tipo algoritmico, al fine di produrre soluzioni AI-Intensive, e Sistemi di Machine Learning che risentano sempre di meno delle problematiche connesse alla Fairness.

Recentemente la fairness nei software di machine learning ha destato notevole interesse anche nella comunità dell'ingegneria del software [25], per esempio grandi studiosi dell'ambito, quali Yuriy Brun and Alexandra Meliou dell'università del Massachusetts, affermano che

numerose iniziative in diverse aree dell'ingegneria del software (quali: specifica dei requisiti, design, testing e verifica) necessitano di essere prese al fine di risolvere il problema. Altri studiosi invece descrivono la fairness come una vera e propria proprietà non funzionale per i sistemi di machine learning e che sostanziale effort di testing sia necessario al fine di scovare le vulnerabilità e violazioni di fairness all'interno dei moduli di machine learning.

Una panoramica completa di tutti i related work di Software Fairness presentati nel documento è riportata in tabella 3.1

3.2.1 Fairness come oggetto di studio nell'ambito AI

Bias di dati come causa di discriminazioni di un modulo AI

Rendere un modulo di machine learning, fair, significa renderlo innanzitutto indipendente dalle dipendenze correlate ai bias immessi, quindi una delle primissime attività su cui la ricerca si basa, è proprio l'individuazione di bias all'interno dei set di dati,i quali devono essere mitigati tramite intensive attività di pre-processing.

Come affermato da Chakraborty et al.[24] la principali cause dei bias, sono le decisioni prese a priori circa i dati che vengono selezionati per un modulo di Machine Learining e l'assegnazione di "label" che possono portare alla creazione di disparità di gruppo tra gli individui del dataset, esempi di "etichette" possono derivare da attributi quali sesso, razza, età, status sociale, etc (volendo porre un esempio, gli individui del dataset possono essere "etichettati" in maniera discriminante come "ricco" o "povero" in base al guadagno netto mensile). Il loro algoritmo Fair-Smoot che dati in input dataset e attributo sensibile, partiziona l'intero dataset in 4 sottogruppi (individui favoriti e privilegiati, favoriti e non privilegiati, individui non favoriti e privilegiati ed infine individui non favoriti e non privilegiati). Sinteticamente, da questa divisione iniziale, l'algoritmo va a generare nuovi *data points*, unità discrete di informazione, quindi nuove entry del dataset, per ciascuno dei sottogruppi, ad eccezione di quello che risulta averne il numero maggiore. Come risultato, tutti e 4 i sottogruppi del dataset diventeranno della stessa taglia rispetto l'attributo sensibile (la stessa del gruppo più numeroso). Lo scopo di questo algoritmo, come altri due presenti nello stato dell'arte (Fairway di Chakraborty e Oprimized pre-processing for discrimination prevention di I.Guyon) è quello di fornire bias mitigation e quindi bilanciare il dataset di partenza (con attività di pre-processing), con il compromesso di condurre a priori un analisi sui dati di partenza che non sia complessa in termini di performances (misurata in termini di Recall e F-Measure).

Statisticamente il loro algoritmo Fair-SMOTE è tra le soluzioni più promettenti effettuando data-balancing in media 200 volte più velocemente di altre soluzioni ed a seguito di studi empirici su più dataset è tra i più consigliati al fine di mitigare i bias all'interno di un dataset di addestramento.

Fairness come conseguenza delle Feature e del Training Set

Altri studi come quelli condotti da Zhang e Harmon, [25], affermano che la Fairness è naturalmente un problema specifico del dominio di utilizzo, ma che è comunque possibile generalizzare il concetto analizzando il numero di feature e l'ammontare dei dati di training. In particolare nel suddetto paper di ricerca, vengono riportati i risultati di uno studio empirico sull'impatto delle fattezze del feature set e del dataset di training quando si cerca di sviluppare *fair machine learning software*, ed in particolare viene valutate le implicazioni che questi aspetti hanno nel costruire *Fair ML Models*. Per lo studio vengono utilizzati diversi datasets presenti e noti in letteratura (quali il COMPAS score, per la valutazione di recidività nel crimine) e differenti definizioni e metriche di fairness (riconducibili a quanto illustrato nel capitolo precedente). Applicando differenti considerazioni matematiche, si arriva ad affermare che avere modello ML con un grande numero di feature, aiuta a migliorare (secondo varie definizioni matematiche) i livelli di fairness d del 38% rispetto la media e che un grande numero di dati possa provocare l'effetto contrario, ovvero una diminuzione sostanziale dei livelli di fairness.

Diversità nella selezione dei dati con vincoli di Fairness

Considerando che i dati sono generati e collezionati da tutti gli aspetti dell'attività umana, in domini come commercio, medicina e trasporti così come la misurazione scientifica, le simulazioni e il monitoring ambientale, è facile incorrere nella pratica di raggruppare i dati, in modo tale da garantire il principio di diversità il quale resta uno degli elementi molti campi applicativi dell'Intelligenza artificiale come la Summarization, la Facility Location e i sistemi di raccomandazione. Ad oggi però non sono molti gli studi che mettono a confronto la Diversificazione con il concetto di Fairness, i quali sono strettamente correlati, ma modellano concetti differenti, la prima cerca di massimizzare la dissimilarità di items in un insieme di dati, mentre la seconda cerca di raggiungere specifici livelli di rappresentazione considerando diverse categorie e gruppi.
Moumoulidou at al. [5] osservano proprio come sia importante e non banale selezionare

sotto-insiemi di di addestramento più differenti possibili (massimizzando la dissimilarità degli individui in ogni insieme), soprattutto qualora sia necessario raggiungere specifici livelli di rappresentazione di differenti categorie e gruppi, in altre parole il problema che il paper analizza è proprio quello di creare diversificazione tra gli individui di un dataset secondo specifici vincoli di Fairness (tipicamente definiti su attributi sensibili). Viene menzionato il più studiato, analizzato e frequentemente usato modello di diversificazione usato dalla comunità del Data Management, ovvero il Max-Min diversification model. Dopo aver introdotto il modello base, lo studio si concentra su una specializzazione del problema con l'introduzione di un numero k di vincoli di fairness prespecificati (per specificare i vincoli è possibile utilizzare per esempio la definizione di Statistical Parity). Quello che, infine, si osserva è che la *fair - Max-Min Diversification* sia un esempio di algoritmo NP-Completo (conclusione ricavata dimostrando prima l'NP-Completezza del problema generale), in ragion di ciò vengono mostrate delle forti approssimazioni dell'algoritmo base che garantiscono la diversità in caso di gruppi non sovrapposti (ad intersezione vuota) [26].

3.2.2 Fairness come oggetto di studio nell'ambito SE

Software Fairness come requisito non funzionale prioritario

Come già osservato più volte all'interno del capitolo Stato dell'Arte, progettare e produrre fair software, sta diventando un ambito che interessa sempre di più il dominio dell'ingegneria del software, Yuriy Brun e Alexandra Meliou, [4] osservano come la Software Fairness, debba essere un entità di "prima classe" in un tipico processo di ingegnerizzazione del software, al pari di altri aspetti non funzionali come qualità e sicurezza. In particolare, al fine di rendere immune un tool immune da discriminazioni, quindi ridurre quelli che sono i difetti intrinseci che diminuiscono i livelli di fairness, è senz'altro importante adottare buone pratiche di design e e algoritmi mirati, ma è necessario porre sullo stesso piano anche la necessità di supportare attività di "fairness testing". Al fine di misurare le discriminazioni software, è necessario identificare e riportare quelli che vengono definiti come "discrimination bugs", in modo tale da cambiare il codice o i dati che introducono tali discriminazioni. Cercando appunto di rispondere a tali problematiche, l'articolo evidenzia tutta una serie di challenge aperte in ciascuno degli aspetti chiave del ciclo di sviluppo del software [4]:

1. *Requirement and Specification*: ponendo l'accento sulla moltitudine di definizioni emerse per il concetto di fairness algoritmica, e sulla correlata difficoltà di definire un software "fair" in maniera univoca, il paper osserva come la consistenza dei requisiti e le analisi

correlate siano una una delle challenge aperte nell'ambito della Requirement Engineering quando si parla di Software Fairness, infatti, quando si considerano combinazioni di "fairness requirements" si può aver a che fare con più definizioni del concetto che possono essere contro intuitive e mutualmente esclusive, e allo stesso tempo analisi automatizzate possono identificare requisiti insoddisfabili o inconsistenti. Il risultato ultimo, infatti, è senz'altro la produzione di software soggetto a risultati inattesi e comportamenti inaspettati, per esempio si osserva, come un tool addestrato con la tecnica degli alberi di decisione derivato da un dataset con feature sensibili, fosse finito a discriminare in maniera molto forte sulla razza dei singoli individui. Al fine di evitare tali problematiche si sottolinea come l'analisi possa aiutare a comprendere come i requisiti di fairness influenzino gli altri requisiti (fairness trade-off) al fine ultimo di realizzare una specifica dei requisiti corretta;

2. *Architecture and design*: è noto come le inconsistenze tra le proprietà di design desiderate per i sistemi software siano comuni. Infatti, in generale una challenge di design aperta è proprio quella di creare tools che aiutino a modellare le architetture dei sistemi, identificando i conflitti. In particolare, nell'ambito del la software fairness, si osserva come una delle ricerche aperte, sia proprio quella di sviluppare stili di sviluppo e pattern di design per le proprietà di fairness, con l'obiettivo di trattare i trade-off dei fair design goals in maniera semi-automatica, come già viene fatto per altre specifiche non funzionali, ad esempio tramite l'ottimizzazione multi-obiettivo. Il paper inoltre osserva come per i sistemi ML-Intensive, il design di algoritmi fairness-aware possa produrre dei fair models soprattutto laddove lavorare con dati di training affetti da bias è critico. La ricerca da questo punto di vista è molto attiva, e molti algoritmi sono in sviluppo e molti framework di progettazione sono in sviluppo;

3. *Testing and Debugging*: È ormai noto come il primo metodo per assicurare la qualità del software sia il testing. Questa è la principale ragione per credere che ciò sia vero anche per la software fairness. In particolare il paper afferma come i Fairness bug siano comuni per sistemi con complessi input e output (si pensi alle forti dipendenze dalla lingua dei sistemi di Speach to Text, all'accuracy dei sistemi di riconoscimento facciale, strettamente dipendenti dalle informazioni demografiche come sesso e razza) e come questi sistemi siano la challenge più grande per la generazione di casi di test per questa tipologia di tool. Il paper evidenzia come il fairness testing richieda che i moduli vengano posti sotto test, svariate volte, e ricordando che il testing esaustivo è

inapplicabile per sistemi complessi, sottolinea l'importanza di eseguire test con input similari. Si evidenzia infatti come l'ottimizzazione di esecuzione incrementale, sulla base dei test già eseguiti con input similari, possa, potenzialmente, ridurre i tempi di esecuzione dei test e aumentare l'applicabilità del fairness testing per i grandi sistemi. Similarmente, la prioritizzazione e selezione dei casi di test possono migliorare l'efficienza dei sistemi di fairness testing. Contestualmente si specifica anche la necessità per gli sviluppatori di identificare e rimuovere le "root causes" dei bias, ciò comporta come la ricerca si stia attivando al fine di fornire strumenti di debugging appositi da mettere a disposizione degli sviluppatori;

4. *Verification*: al pari della correttezza del software, il paper evidenzia come anche la verificabilità sia un gola altamente desiderabile per la software fairness. L'esecuzione multipla dello stesso codice che può portare ad output diversi (*non determinismo*), la stretta dipendenza del concetto di fairness con l'esecutore (*la multi utenza*) e la *natura probabilistica* delle proprietà di fairness, sono tutti aspetti che riducono lo spettro di tecniche di verifica e validazione esistenti ed applicabili direttamente al problema. Quando si parla di fairness, è essenziale verificare il comportamento dei tool già durante lo sviluppo, perciò, si evidenzia come, creare ambienti di runtime e tool di debugging mirati all'identificazione dei bug o warning di fairness sia essenziale, al fine di verificare formalmente il comportamento dei tool. Il problema principale però è nuovamente, identificare modi per codificare le definizioni di fairness come proprietà verificabili di un programma, ciò è strettamente connesso alla natura intrinseca delle metriche, alcune sono di tipo probabilistico e plausibilmente verificabili, altre però sono di natura diversa (e.g. casual reasoning e metriche strutturali). Tutto ciò rende la verifica una sfida di ricerca ancora molto aperta e avvincente.

Il "problema fairness" nella specifica dei requisiti

Finkelstein at Al. [27], nel 2008 hanno introdotto il concetto di fairness nell'ambito dell'analisi e ottimizzazione dei requisiti. Il lavoro è particolarmente interessante perché introduce modelli valutativi, basati su funzione di valutazione multi obiettivo, al fine di bilanciare i trade-off derivanti da differenti clienti. I modelli proposti adottano scenari semplificati al fine di bilanciare il problema della fairness tra le sue differenti definizioni, infatti il primo step che il paper cita mostra come sia possibile utilizzare le tecniche di "search based optimization" al fine di giungere ad un compromesso tra le varie definizioni di fairness in specifici contesti.

L'esperimento poi dimostra come le tecniche di ricerca possano essere anche applicate a dataset reali e illustra come tali tecniche possano essere anche utilizzate per identificare i unfair bias intrinseci in tali dataset. Anche se un po' datato, questo paper evidenzia un problema che è tutt'ora attuale: aspetti intrinseci dello sviluppo software e delle tematiche AI-Intensive al giorno d'oggi mettono in luce ancora innumerevoli sfide nel campo dell'ingegnerizzazione dei requisiti e dei dati. Lo sviluppo di soluzioni AI-Intensive è strettamente influenzato da trade-off qualitativi tra cui quelli legati al mondo della fairness, e soluzioni di intelligenza artificiale, come tecniche di ricerca (e.g. algoritmi genetici), oppure modelli di ottimizzazione multi obiettivo, potrebbero sicuramente dare un'ottima risposta a queste esigenze.

Il Testing in ambito Software Fairness

Yuriy Brun et al. oltre ad essere famosi per aver posto l'accento all'emergente problema della Software Fairness, hanno condotto anche attività di ricerca specifiche come studi specifici circa il fairness testing. Uno dei loro studi più famosi del 2017 [28], propone un nuovo approccio di fair-testing, chiamato dai ricercatori Themis, al fine di misurare se e in che modo i programmi effettuano discriminazioni, focalizzandosi sulle casualità dei comportamenti discriminatori. L'approccio Themis genera test suite al fine di computare score inerenti la casual discrimination per particolari caratteristiche, e.g. secondo specifiche definizioni di fairness, il tool è in grado di generare uno score che determina quanto un sistema software discrimina contro razza ed età. Individuato un problema di discriminazione, Themis genera una test suite al fine di computare tutti i sets di caratteristiche che potrebbero essere alla base del problema. Fornendo, infine, in input al sistema di testing, una test suite manuale o auto generata, esso è in grado di verificare, su specifici input rappresentativi della popolazione, se effettivamente sono presenti feature del dataset discriminanti. L'obiettivo principale di Themis è quello di rispondere al problema di esecuzione del fairness testing per sistemi reali, (citato anche nel paper generale [4]), infatti, le tre tecniche di ottimizzazione che il sistema di testing adotta, riducono il numero di test cases necessario a computare informazioni circa i gruppi sensibili più significativi di un dataset, con l'obiettivo di individuare le cause di discriminazione che influenzano il comportamento del tool sotto test.

3.2.3 Riflessioni sullo stato dell'arte e sull'evoluzione della Software Fairness

Visione comune del mondo della ricerca, è che il concetto di Software Fairness è ancora in evoluzione, negli ultimi anni molti studi emergenti, hanno rivalutato l'importanza del

concetto cercando in vari modi di sistematizzarlo e formalizzarlo il più possibile. Come osservato in questo capitolo, sforzo principale della ricerca, è stato infatti, analizzare i vari aspetti che il concetto di Fairness racchiude, molte sono le metriche emerse che cercano di descrivere in maniera qualitativa o quantitativa i livelli di Fairness di uno specifico tool, come un generico modello di Machine Learning o una soluzione AI-Intensive.

Dagli studi analizzati, si osserva come prerogativa principale dei ricercatori nel campo dell'intelligenza artificiale, siano orientati attivamente per rendere dataset e tecniche AI-based sempre più conformi alle definizioni semantiche dell'una o l'altra metrica. A problematiche specifiche dell'intelligenza artificiale, quali il rilevamento di fair-bias nei dataset oppure applicare operazioni di fair-diversification in fase di pre- processing, l'ingegneria del software sta cercando di sistematizzare il concetto in modo tale da rendere ai Data Scientist più agevole il trattamento della Fairness. Come osservato, negli ultimi anni sono nati molti quesiti che pongono il problema del software eticamente corretto sotto una nuova luce, ovvero quella di vero e proprio requisito non funzionale prioritario per una soluzione AI-Intensive, di conseguenza, la comunità ingegneristica, cerca di considerare, come per ogni specifica non funzionale, quali sono le tecniche e metodologie migliori per trattare la Fairness in ogni fase del ciclo di vita del software. Studi come quelli inerenti al Fairness Testing, oppure le tecniche Search-based per il bilanciamento dei requisiti di fairness tra definizioni strutturalmente e semanticamente contrastanti, sono solo alcuni degli esempi di soluzioni che gli studiosi di Ingegneria del Software hanno formulato negli ultimi anni, e molti altri saranno sicuramente proposti, considerando che la ricerca in ambito di Software Fairness, con tutta probabilità andrà avanti, in maniera molto pronunciata nei prossimi anni.

Ma guardare al futuro della ricerca, significa quindi capire in che direzione sarà opportuno far evolvere il concetto di Software Fairness, al fine di renderlo sempre più vicino al mondo delle aziende di sviluppo, ed in particolare, nel contesto AI, sempre più vicino al mondo dei Data Scientist e dei modelli AI-Intensive per problematiche reali. Ovviamente è molto complesso sistematizzare il concetto di Software Fairness nella pratica lavorativa, così come è stato fatto fin ora in modo teorico, molte sono le variabili in gioco:

- Quanto e come figure come Data Scientist, Ingegneri del Software in ambito AI e figure manageriali si approccino al problema?

- Quali sono le definizioni più adottate in ambito lavorativo per i problemi fair-critical negli ambiti in cui lo sviluppo AI-Intensive si concentra oggi giorno? Quali lo saranno in futuro?

- Esistono già best-practices da seguire per garantire alti livelli di fairness di un modello di machine learning? se si, secondo quali definizioni?

Di quesiti di questo tipo se ne possono fare tanti, e ovviamente rispondere a tutto in maniera definitiva è un qualcosa che va al di fuori degli obiettivi di questo lavoro di tesi. Però in un contesto così ancora inesplorato, il budget a disposizione, permette sicuramente di partire con l'interpellare i diretti interessati. Tramite tecniche empiriche mirate si tenterà, infatti, di capire se diretti interessati (e.g. Data Scientist e Project Manager) si approcciano attualmente al problema, se effettivamente le definizioni teoriche di fairness hanno effettivo riscontro sul campo e se e come processi ingegneristici fair-orinted vengono applicati durante il ciclo di sviluppo di una soluzione AI-Intensive con obiettivo ultimo di creare una vera e propria raccolta riassuntiva dello status della pratica che abbia un duplice obiettivo a lungo termine, ovvero:

- Provare a suggerire ai ricercatori quali sono i punti di forza e di debolezza per futuri lavori plausibilmente sempre più vicini al mondo del lavoro;

- Cercare di avvicinare sempre di più gli esperti del settore AI al contesto della fairness, in modo tale che adottino sempre di più processi di sviluppo fair-oriented.

CAPITOLO 4

Design

4.1 Quesiti di ricerca

4.1.1 RQ - Percezione del concetto di Fairness in azienda

In diretta continuazione con quanto osservato a chiusura del capitolo precedente (riflessioni sullo stato dell'arte), molti sono i quesiti che possono nascere al fine di avvicinare gli studi di ricerca inerenti lo sviluppo di soluzioni ML Fair-Oriented al reale contesto lavorativo. Al fine di fornire dettagli analitici inerenti lo stato della pratica aziendale, si è deciso di progettare la fase empirica del lavoro di tesi in modo da rispondere al seguente quesito di ricerca:

RQ: In che modo il concetto di Software Fairness è attualmente percepito nell'ambiente lavorativo ML-Intensive?

Questo macro quesito, mira a fornire alla ricerca un'overview analitica circa le attuali pratiche lavorative adottate da professionisti (quali Data Scientists o Ingegneri del Software) nello sviluppo di soluzioni ML-Intensive. Per fare ciò si è deciso di scomporre il macro quesito iniziale in 5 sotto obiettivi di ricerca, che in prima istanza guidano la successiva fase di studio empirico.

4.1.2 RQ1 - Come definire la fairness in ambito lavorativo

RQ1 - Quali sono i migliori approcci e definizioni per trattare la fairness in un contesto lavorativo?

Dall'analisi dello stato dell'arte, si è osservato che il concetto di software fairness, non è univocamente interpretabile, molte sono le definizioni e gli approcci formalizzabili a seconda delle specifiche esigenze connesse al dominio di uno specifico problema, ma nella pratica lavorativa, quali sono le sfaccettature del concetto di fairness che maggiormente aiutano i lavoratori nella formalizzazione degli obiettivi specifici, quali sono invece gli approcci di misurazione (tra quelle fornite dallo stato dell'arte)) che i professionisti adottano nel misurare i livelli specifici di Software Fairness di un modulo ML-Intensive. Per rispondere a questo quesito, si è deciso di rimodulare (in maniera semi-formale) alcuni approcci e definizioni formali analizzate precedentemente, al fine di capire sulla base del campione di indagine, quali possano essere effettivamente più affini e applicabili in pratica. Nel dettaglio i quesiti utili a rispondere a questo sub-goal di ricerca metteranno a confronto, 3 macro-sfaccettature del concetto di fairness:

- Equità nel formulare risultati di predizione;
- Non formulare predizioni sulla base di discriminazioni derivanti da feature sensibili;
- Assumere decisioni al fine di garantire risultati paritari per gruppi minoritari e maggioritari.

e tre macro approcci di misurazione, che sono alla base di diverse metriche, come osservato nell'analisi dello stato della pratica:

- Approcci basati su probabilità di predizione;
- Approcci basati su similarità matematica degli individui;
- Approcci basati su relazioni causali tra attributi sensibili e risultati.

4.1.3 RQ2 - Chi si occupa di fairness in ambito lavorativo

RQ2 - Come è composto generalmente un team lavorativo per lo sviluppo di moduli ML-Intensive Fair Critical?

Fornire un'analisi circa lo status lavorativo in termini di Software Fairness, necessità senz'altro di approfondire quali figure professionali hanno più impatto rispetto ad altre nell'analisi e nello sviluppo di soluzioni ML-Intensive fair-critical. Per rispondere a questo quesito, è necessario valutare che livello di impatto figure professionali quali Data Scientists, Ingegneri del Software, Data Engineer, Project Manager, Analisti, Architect o *Esperti specifici*, abbiano maggior impatto nello sviluppo di soluzioni Fair-Critical.

4.1.4 RQ3 - Fairness a confronto con altri aspetti non funzionali

RQ3 - Quanto il concetto di software fairness è importante se paragonato ad altri aspetti non funzionali?

Considerando che alcuni lavori di ricerca di tipo ingegneristico suggeriscono di trattare il concetto di software fairness come un vero e proprio requisito non funzionale prioritario di un sistema ML-Intensive [4], si è deciso di confrontare l'aspetto etico di un modulo di machine learning con altri aspetti non funzionali, maggiormente riconosciuti dagli standard ingegneristici (modello FURPS+), oppure ampiamente utilizzati nella pratiche di sviluppo di soluzioni ml-intensive, in modo tale da osservare, secondo il parere di esperti, quanto specifiche non funzionali più sistematiche nello sviluppo ML-intensive possano essere più o meno rilevanti rispetto il concetto di software fairness. Nell'ottica di formalizzare nel successivo processo di generalizzazione dei risultati dei veri e propri trade-off tra fairness e altri aspetti non funzionali, il quesito specifico con la successiva fase di investigazione empirica, mira a confrontare il concetto di Software Fairness, con:

- Usabilità;
- Affidabilità;
- Performance;
- Supportabilità;
- Accuracy;

- Sicurezza;
- Manutenibilità e retraining del sistema;
- Riusabilità e scalabilità.

4.1.5 RQ4 - Fairness come aspetto integrante di una Pipeline ML

RQ4 - In quali fasi di una tipica pipeline di Machine Learning è importante adottare strategie per garantire alti livelli di fairness?

Qualsiasi studio inerente lo stato della pratica che abbia riferimenti di tipo ingegneristico, non può prescindere dall'analizzare aspetti inerenti il ciclo di vita di un determinato target di prodotti software analizzati dall'indagine. In particolare, dall'analisi dei vari modelli di sviluppo standard, come osservato nella sezione di background di questo documento, il modello di sviluppo che si adatta meglio allo sviluppo di soluzioni ML-Intensive è senz'altro l'approccio basato tramite Pipeline di Machine learning, adottato da standard di sviluppo noti, quali il famoso ML-Ops [18].

Considerando quindi le fasi di una canonica pipeline di machine learning (immagine 2.1), quali sono le fasi su cui investire al fine di garantire un alto livello di fairness del sistema?. Partendo da questo quesito, obiettivo della successiva fase empirica, sarà quello di valutare l'utilità di adottare strategie e metodologie atte a preservare i livelli di fairness in ogni fase di sviluppo di un modulo ML-Intensive, così come formalizzate in una tipica Pipeline di machine learning.

4.1.6 RQ5 - Fairness e maturità aziendale

RQ5 - Quanto le compagnie di sviluppo ML-Intensive, sono mature nel trattare il concetto di fairness come un requisito non funzionale?

Per concludere la panoramica di analisi, si è deciso di proporre ai partecipanti all'indagine, un analisi critica circa il livello di maturità della propria azienda nel trattare software fairness all'interno dei propri progetti di sviluppo ML-Intensive. In particolare, facendo riferimento al noto standard di valutazione del livello di maturità aziendale CMM - *Capability Maturity Model* [30], è stata formalizzata una scala di misura, che permette di valutare a che livello di maturità è possibile classificare una generica azienda di sviluppo che produce Sistemi ML fair critical:

- Livello 0 - l'azienda non tratta software fairness;
- Livello 1 - L'azienda occasionalmente tratta software fairness, ma i processi a riguardo sono spesso disorganizzati e talvolta caotici;
- Livello 2 - L'azienda tratta la fairness e i relativi processi sono stabiliti, definiti e documentati;
- Livello 3 - L'azienda tratta regolarmente la fairness, e il suo sviluppo è gestito tramite processi standard per il management della fairness;
- Livello 4 - L'azienda tratta regolarmente fairness e il suo monitoraggio e controllo è gestito da processi specifici accompagnati da data collection e analisi;
- Livello 5 - L'azienda tratta regolarmente feirness e i relativi processi sono costantemente ottimizzati tramite feedback di monitoraggio.

Come osservato, la scala di valutazione riprende i livelli standard di classificazione del CMM, e nell'ottica specifica, ci si pone l'obiettivo di capire quante aziende (tra quelle coinvolte nella successiva fase di investigazione empirica) ritengono di lavorare in un contesto dove effettivamente la fairness sia trattata come un requisito non funzionale prioritario in termini di maturità, che appunto vanno dal trattare sporadicamente i livelli di fairness di un modulo di machine learning sviluppato, fino a trattarla con processi standardizzati (laddove possibile) con particolare attenzione all'ottimizzazione degli stessi.

4.2 Metodologia di ricerca

Al fine di ottenere contenuti informativi utili a rispondere agli obiettivi di ricerca formalizzati, si è deciso di è stato deciso di progettare la fase di raccolta dati per mezzo di un Survey da sottoporre ad esperti del dominio.

Perché un Survey di ricerca?

Progettare un Survey di ricerca che indaghi sullo status della pratica dello sviluppo si sistemi di machine learning fair, può avere un duplice vantaggio: innanzitutto interpellare esperti del dominio è un qualcosa che indirizzerà le future attività di ricerca verso la progettazione di strumentazioni e strategie che realmente possano rispecchiare le necessità di chi quotidianamente lavora nel mondo del machine learning secondo vincoli etici sempre più stringenti,

oltretutto fornire un overview iniziale delle pratiche più utilizzate può essere sicuramente d'aiuto anche agli esperti dell'ambito nel definire processi standard per trattare la fairness come altri aspetti di qualità già più sistematizzati.

4.2.1 Struttura e design del Survey

Al fine di bilanciare il bisogno di avere un survey ragionevolmente corto, con la necessità di renderlo abbastanza efficace da rispondere agli obiettivi di ricerca riportati nel precedente paragrafo, si è preso spunto dalla guida strutturale fornita da Andrews et al. [31], tra i principi più importanti di design da cui si è preso spunto, vale senz'altro la pena ricordare:

- La formulazione di risposte a scelta multipla o in scala (numerica o qualitativa), in modo tale da dare un carattere più analitico ai dati estraibili dalle risposte al questionario;
- L'utilizzo di un vocabolario chiaro, non ambiguo e coinciso al fine di eliminare ambiguità circa il significato della domanda;
- Specificare a priori quelli che sono gli obiettivi del Survey e chiarire da subito che le informazioni prese non saranno utilizzate per altri scopi:
- Raccogliere informazioni circa i partecipanti all'indagine, utili a categorizzare in maniera strategica i dati a disposizione;
- Garantire il rispetto della privacy specificando che i dati saranno trattati in forma anonima da un punto di vista di analisi e pubblicazione dei risultati, senza far riferimento alcuno a chi ha fornito le risposte;

Sulla base dei principi riportati, il survey di ricerca è stato formalizzato in 4 sezioni principali, ed al fine di garantire consistenza di contenuto, sono presenti due domande discriminanti, utili a comprendere se il partecipante è adatto o meno a trattare aspetti specifici di software Fairness nel contesto del machine learning. In fine, con l'obiettivo di verificare se l'intervistato abbia compilato il questionario in maniera consona e con la dovuta attenzione, sono stati definiti due Attenction Check Strategici che permetteranno in fase di analisi di scartare risposte non valide ai fini dell'indagine.

Per realizzare il Survey si è scelto di utilizzare la piattaforma *Google Form*. la quale in maniera nativa permette di:

- Formulare domande di vario tipo secondo le differenti esigenze di indagine;

- Formulare flussi alternativi di compilazione in base alle risposte;
- Suddividere le domande in differenti sottosezioni, coerenti con quanto progettato.

Si è scelto di adottare la strategia dei flussi alternativi, al fine di non collezionare risposte da figure senza esperienza nell'ambito dello sviluppo ML-Intensive e Fair oriented. La durata del questionario, onde evitare cali di attenzione prima della sottomissione è stata stimata attorno ai 10/15 minuti. Per reclutare i partecipanti in modo opportuno ed incentivarli alla compilazione, si è deciso di utilizzare la piattaforma specifica Prolific.

La figura 4.1 mostra una visione riassuntiva della struttura del Survey, identificando il flusso di domande principale in blu e i flussi alternativi in celeste ed in viola.

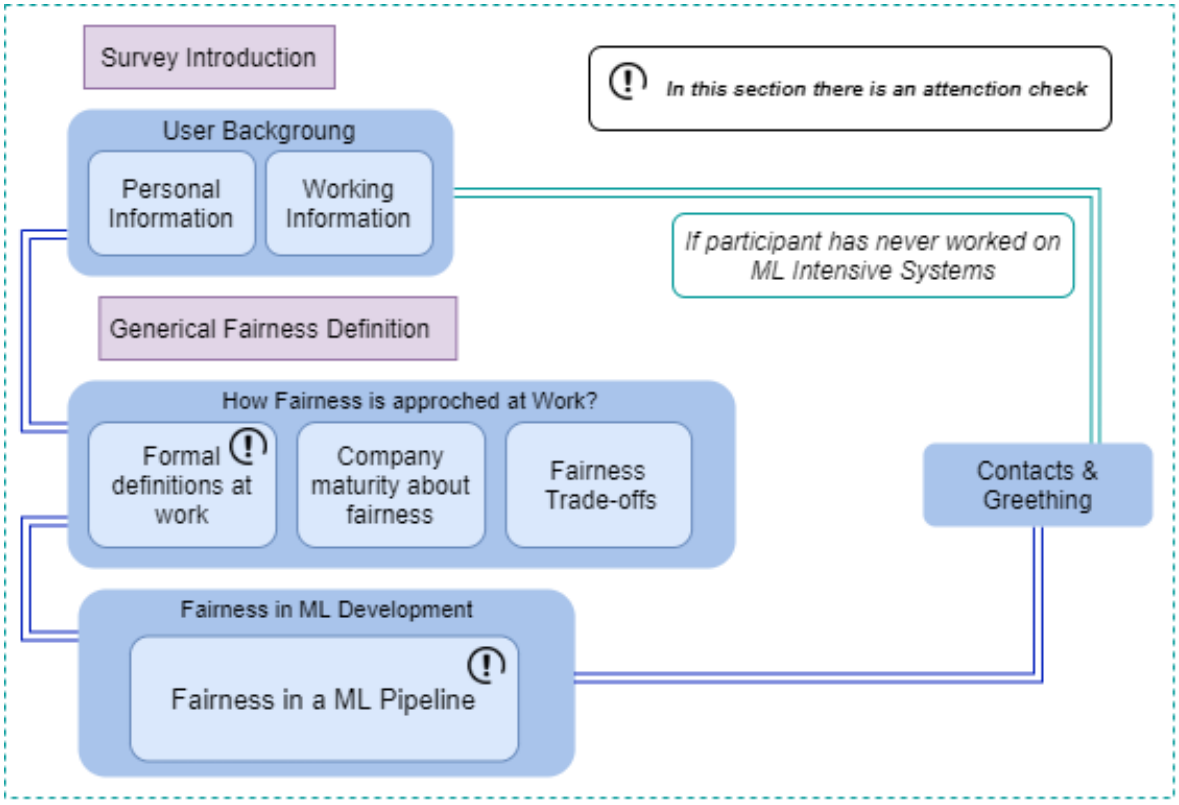

Figura 4.1: Diagramma di riepilogo della strutturazione del survey

Introduzione del Survey

Prima ancora di cominciare con la compilazione del Survey, si introduce velocemente il partecipante alla problematica di ricerca connessa alla software Fairness nei sistemi AI-Intensive, fornendo anche un piccolo esempio pratico, tra quelli definiti nel capitolo stato

dell'arte. Volutamente non si fornisce già all'inizio una caratterizzazione mirata del concetto, dato che è scopo dell'indagine capire se il concetto di Fairness sia approcciato in ambito lavorativo in maniera similare rispetto a quanto formalizzato in letteratura. Inoltre viene fornita qualche informazione circa i conduttori dell'indagine empirica e sul trattamento dei dati raccolti.

Background del partecipante

La prima sezione è mirata ad acquisire informazioni circa il background dei partecipanti, al fine di poter suddividere e manipolare successivamente le risposte al questionario sulla base delle informazioni sociali, culturali, lavorative dei partecipanti. La sezione è strutturata in modo tale da ricevere informazioni circa informazioni anagrafiche e etniche del partecipante, lo status lavorativo del partecipante, la sua esperienza lavorativa circa lo sviluppo e la realizzazione di sistemi ML-Intensive. Di seguito viene riportata una tabella riassuntiva della sezione con tutte le domande poste circa il background del partecipante.

In questa sezione, viene chiesto al partecipante se ha mai lavorato a sistemi di intelligenza artificiale o che includano moduli chi machine learning, qualora la risposta sia affermativa, il partecipante che decide di continuare con la compilazione viene indirizzato alla successiva sezione del questionario. Qualora il partecipante non avesse esperienza nello sviluppo di questi sistemi, alla pressione del tasto "Avanti", il partecipante viene condotto alla sezione di chiusura del Survey.

Domanda	***Tipo di Domanda***	***Obbligatoria***
Inserisci il tuo codice prolific	Testo Breve	No
Quanti anni hai?	Scelta multipla	No
In quale gender ti rispecchi maggiormente?	Caselle di controllo	No
Dove lavori?	Scelta multipla	No
qual è il maggior livello di istruzione che hai conseguito?	Scelta multipla	Sì

Domanda	*Tipo di Domanda*	*Obbligatoria*
qual è la tua posizione lavorativa attuale?	Caselle di controllo	Sì
In che settori lavori attualmente?	Caselle di controllo	Sì
qual è il tuo ruolo professionale?	Caselle di controllo	Sì
Quanti anni di esperienza hai in questo ruolo?	Scelta multipla	Sì
Hai mai lavorato allo sviluppo di soluzioni AI-Intensive o a sistemi che includono moduli di machine learning?	Scelta multipla	Sì
*** Per domanda obbligatoria si intende che il partecipante è obbligato a fornire una risposta**		

Tabella 4.1: Domande della sezione Background del Survey

Come la fairness è approcciatà a lavoro?

In questa sezione si cerca di investigare circa le attuali pratiche lavorative nello sviluppo di sistemi Fair-Critical, vengono fornite specifiche domande per fornire una visione specifica di come ed in che modo le aziende trattino la fairness in ambito lavorativo, si cerca quindi di identificare:

- Quali **aspetti specifici o definizioni formali** del concetto di fairness sono più utili durante lo sviluppo di soluzioni di machine learning;
- Quali sono i **ruoli professionali** connessi al concetto di fairness che doverebbero essere coinvolti nello sviluppo di soluzioni fair-critical;
- qual è il **livello di maturità** delle aziende nel trattare software Fairness nello sviluppo ML-Intensive in maniera sistematica e standardizzata - a tal proposito è stata formalizzata una specializzazione fair-oriented del CMM (Capability Maturity Model) [30].

- In che misura **altri specifici aspetti funzionali e non funzionali** siano da confrontare rispetto la software Fairness, in ottica tale da formalizzare una visione generale di quali potrebbero essere eventuali trade-off durante lo sviluppo di sistemi ML Fair-Critical.

Ovviamente questa sezione del Survey tocca tutta una serie di aspetti che meritano di essere approfonditi eventualmente con altri studi mirati, però sicuramente indagare i concetti, di cui sopra, è senz'altro un rilevante punto di inizio per la standardizzazione dei processi circa il trattamento della software Fairness.

La tabella 4.2 riporta nel dettaglio tutte le domande poste nella sezione del Survey circa l'esperienza lavorativa rispetto al concetto di Fairness.

Definizione-Domanda	***Tipo di Domanda***	***Obbligatoria***
Definizione generica di Software Fairness	Descrizione	–
Secondo te, quali dei seguenti aspetti rappresentano la definizione generica di fairness fornita in precedenza?	Griglia scelta multipla	Sì
Considerando la tua esperienza lavorativa, quanto i seguenti (approcci) sono trattati?	Griglia scelta multipla	Sì
Generalmente utilizzi altri approcci per lavorare con il concetto di Software Fairness?	Testo breve	No
Quale Bevanda(e) preferisci il sabato sera?	Caselle di controllo **	Sì
Considerando i seguenti ruoli (professionali), chi ha impatto sulle scelte inerenti la software fairness?	Griglia scelta multipla	Sì

Definizione-Domanda	*Tipo di Domanda*	*Obbligatoria*
In quale dei seguenti livelli di maturità, classificheresti il tuo ambiente lavorativo circa il trattamento della fairness?	Scelta multipla	Sì
Considerando i seguenti aspetti (funzionali e non funzionali) dello sviluppo software, quanto li ritieni importanti se comparati alla fairness?	Griglia scelta multipla	Sì
*** Per domanda obbligatoria si intende che il partecipante è obbligato a fornire una risposta**		
**** In questa sezione è presente un attenction check**		

Tabella 4.2: Domande della sezione Definizione Generale ed esperienza lavorativa del Survey

Fairness come aspetto integrante del ciclo di vita di un sistema ML-Intensive

Volendo condurre un indagine sullo stato della pratica dello sviluppo fair-oriented, è senz'altro necessario capire in quali fasi e processi dello sviluppo ML-Intensive la fairness necessiti di essere maggiormente attenzionata. Per porre quesiti che rispondano a questa macro-problematica è senz'altro necessario capire quale modello di sviluppo si adatti di più allo sviluppo di soluzioni ML-Intensive. A tal proposito, i quesiti sono stati formulati tenendo in considerazione una generica Pipeline per lo sviluppo di sistemi di machine Learning basata sulla filosofia di sviluppo MlOps (la sezione 2.3.2 del capitolo di background fornisce dettagli teorici in materia). Sulla base della pipeline, sono stati proposti dei quesiti mirati per capire effettivamente in quali fasi dello sviluppo ML-Intensive sia attualmente attenzionata e trattata come requisito primario la software Fairness nel contesto lavorativo del partecipante. Immaginando poi che le pratiche aziendali possano differire o convergere con le opinioni personali dell'intervistato, è stato previsto un quesito specifico circa l'opinione personale dell'intervistato. Progettando questa sezione, si è osservato che potrebbe essere anche utile capire quali tool commerciali o specifici possano essere particolarmente utili per trattare la Software Fairness in una pipeline di machine learning, perciò è stato posto un quesito mirato a riguardo.

La tabella 4.3 riporta nel dettaglio tutte le domande poste nella sezione del Survey circa il ciclo di vita di una soluzione ml-intensive.

Domanda	*Tipo di Domanda*	*Obbligatoria*
Considerando una generica pipeline di machine learning (come la seguente - figura 4.1), quanto consideri l'equità come un aspetto rilevante per ciascuna delle seguenti fasi nel tuo contesto lavorativo?	Griglia scelta multipla	Sì
Quali tool utilizzi (se previsti) per trattare la fairness in una pipeline di machine learning ?	Caselle di controllo	No
Contando indietro dal 5, quale numero viene dopo il 3?	Scelta multipla **	Sì
*** Per domanda obbligatoria si intende che il partecipante è obbligato a fornire una risposta**		
**** In questa sezione è presente un attenction check**		

Tabella 4.3: Domande della sezione ciclo di vita fair-oriented del Survey

Chiusura del survey

Dopo la compilazione intrinseca del questionario, è stata preparata una sezione di chiusura che consentisse al partecipante di lasciare il proprio recapito e-mail e il riferimento al proprio profilo linkedin, in modo tale da:

- Ottenere maggiori informazioni circa le risposte fornite qualora se ne riscontrasse la necessità;
- Richiedere maggiori informazioni circa il partecipante se necessario;
- Renderlo partecipe per future indagini quali follow-up interview di approfondimento.
- Fornirgli dettagli sui risultati dell'indagine qualora fosse interessato.

La sezione inoltre prevede che il partecipante possa fornire un opinione aggiuntiva personale circa la tematica affrontata al fine di formalizzare altre informazioni utili al trattamento

della software fairness nello sviluppo di soluzioni ML Intensive.

La tabella 4.5 riporta in maniera riassuntiva tutte le domande poste nella sezione di chiusura del documento.

Domanda	*Tipo di Domanda*	*Obbligatoria*
Se vuoi, puoi lasciarci maggiori informazioni circa la fairness. Qualsiasi informazione che non abbiamo considerato è importante.	Testo lungo	No
Se desideri restare aggiornato circa i risultati dello studio oppure essere contattato per partecipare ad interviste di approfondimento sul topic, gentilmente scrivi qui il tuo indirizzo e-mail.	Testo breve	No
*** Per domanda obbligatoria si intende che il partecipante è obbligato a fornire una risposta**		

Tabella 4.4: Domande della sezione di chiusura del Survey

4.2.2 Validazione del Survey

Ricordare che uno Survey troppo lungo può facilmente causare cali di attenzione, fattore che può notevolmente inficiare la validità delle risposte raccolte[31] In tal senso è stato deciso di realizzare una simulazione pilota con studenti magistrali, frequentanti un corso accademico specifico di Ingegneria del Software per l'intelligenza artificiale. I suddetti studenti sono stati invitati a compilare una copia del Survey elettronico in modo tale da stimare il tempo di esecuzione del survey su larga scala e verificare la presenza di di quesiti/definizioni da eventualmente semplificare a seguito del test pilota.

Prerequisiti di partecipazione al test pilota

- Laurea Triennale in informatica: tutti i partecipanti dispongono di tale titolo di studi;
- Nozioni generiche di Software engineering for Artificial intelligence e MLOps Pipeline - Lo status di avanzamento del suddetto corso all'avvio del test pilota copre tutte le conoscenze basilari necessarie per affrontare l'argomento;

- Panoramica introduttiva circa la Fairness in ambito ML-Intensive, l'attività di testing pilota sarà preceduta da una lezione teorica, circa tutte le definizioni necessarie alla fairness.

Attenzione - le risposte raccolte con il test pilota, non sono state utilizzate ai fini dell'analisi dei risultati, anche perché come osservato nella sezione successiva, il survey è rivolto a figure professionali e non accademiche. Inoltre l'aver acquisito nozioni teoriche di software fairness, basate sulle stesse fonti dello studio empirico, è una chiara minaccia alla validità dei risultati.

Risultati del test pilota

Il test pilota, nelle modalità di cui sopra, ha avuto luogo in data 09/05/2022, in totale, sono state raccolte 4 risposte di studenti, magistrali o dottorandi. Ne emerso che la durata media di compilazione del survey è stata di oscilla tra i 12 e 15 minuti, quindi del tutto accettabile con i criteri di accettazione prefissati. Contestualmente all'esecuzione del test pilota è stato richiesto agli studenti di fornire qualche feedback circa le loro impressioni sui contenuti del survey tramite l'utilizzo di una bacheca condivisa (nello specifico è stata creata un'istanza di bacheca tramite il tool web *Padlet*). Di seguito sono riassunti i principali cambiamenti apportati al Survey elettronico a seguito della chiusura del test Pilota:

Rimodulazione di contenuti forvianti: Dalle risposte ottenute al Survey pilota, ne è derivato che nonostante la stima dei tempi fosse soddisfacente, molte domande fossero state compilate con poca attenzione al contenuto, ipotesi poi confermata con i partecipanti al test pilota in una successiva riunione informale. Al fine di ridurre questo fenomeno nella più delicata fase di disseminazione del Survey, il team ha quindi deciso di rimodulare poi il questionario, rimuovendo quesiti ridondanti o non affini ai quesiti di ricerca (La strutturazione del survey presente in questo capitolo è consistente con le modifiche apportate al survey elettronico a seguito dell'esecuzione del test pilota).

Cambiamenti circa il trattamento delle informazioni sensibili: tramite il Padlet uno dei partecipanti al Survey, ha espresso delle perplessità circa il dover per forza rispondere a domande inerenti dati sensibili non di carattere lavorativo (quali sesso o età), nonostante fosse prevista la risposta "Preferisco non rispondere" per ciascuna di esse. Al fine di evitare che tale fattore possa arrecare disturbo anche durante la fase di propagazione del survey, si è deciso quindi di rendere queste domande non obbligatorie oltre a lasciare l'opzione

"Preferisco non rispondere", e di valutare successivamente risposte non ottenute in modo appropriato in fase di analisi.

Dopo aver apportato le opportune modifiche al survey, si ritiene che il test pilota nel suo piccolo, abbia dato riscontro positivo circa l'efficacia del survey progettato, per tanto si ritiene possibile procedere con la diffusione del survey su larga scala (nelle modalità di seguito espresse).

4.3 Reclutamento e diffusione del Survey

Lo studio empirico da condurre, è mirato per acquisire informazioni da figure professionali che abbiano lavorato all'interno dell'ambito dell'intelligenza artificiale con particolare focus su progetti fair-critical, in particolare il survey è rivolto figure professionali quali:

- Ingegneri del Software;
- Data Scientists;
- Data & Feature Engineers
- Programmatori Junior o Senior affini all'ambito ML-Intensive;
- Junior o Senior Manager aziendali affini all'ambito ML-Intensive;

4.3.1 Reclutamento dei partecipanti

Innanzitutto, una scelta chiave per la diffusione del Survey e l'acquisizione di risposte consiste nella scelta della piattaforma da utilizzare per raggiungere le figure professionali di nostro interesse. Fissando che il numero di risposte minimo da ottenere, per evitare minacce alla validità (come sarà successivamente discusso) deve necessariamente superare le 100 risposte, dopo aver effettuato un'attenta analisi delle piattaforme Social si è deciso di escludere a priori le principali piattaforme social non mirate al contesto lavorativo.

D'altra parte la crescente crescita professionale e l'enorme compatibilità con il mondo del Data Mining di **Linkedin** [32], ne fanno lo strumento ideale per ricercare ed identificare direttamente partecipanti interessanti all'indagine, nonostante la pratica possa essere più dispendiosa, si è deciso di verificare manualmente la presenza di figure di interesse a cui chiedere gentilmente di partecipare all'indagine tramite la condivisione del Survey a mezzo di e-mail. Oltre le principali piattaforme social, è stata identificata, sotto consiglio di esperti

di ricerca, la piattaforma di recruitment **Prolific**, la quale permette di formalizzare vincoli circa le categorie di destinatari a cui condividere l'indagine.

Al fine di garantire una corretta separazione tra i partecipanti all'indagine raggiunti a mezzo di Linkedin, rispetto quelli raggiunti con Prolific, si è deciso di condividere due copie distinte del Survey, ognuna specifica per piattaforma in modo tale da poter classificare le risposte più agevolmente, distinguendo anche la fonte di appartenenza. Le domande tra le due distinte copie del Survey elettronico sono identiche, a meno dell'inserimento del proprio ID Prolific, ovviamente specifica per quest'ultima piattaforma.

Considerazioni aggiuntive sull'utilizzo di Prolific

Come suggeriscono Reid et al. circa il 33% delle risposte sottomesse ad un Survey sottomesso con Prolific, statisticamente possono risultare invalide [33]. Al fine di ridurre al massimo le sottomissioni invalide, si è deciso di vincolare i partecipanti all'indagine Prolific tramite i seguenti filtri messi a disposizione dala piattaforma:

- Conoscenza fluente dell'inglese;
- Settore lavorativo: Informatica, Tecnologia, Ingegneria...;
- Completamento di un alto livello di studi: Diploma o superiori.

Non essendoci su Prolific un filtro apposito per selezionare facilmente il target di utenza di interesse, si è deciso di esplicitare testualmente i requisiti di partecipazione sovra riportati, invitando gentilmente partecipanti non qualificati ad ignorare la compilazione.

4.3.2 Disponibilità e diffusione del Survey

Si è deciso di rendere disponibile ai professionisti la compilazione del Survey solo a seguito dell'esecuzione del test pilota, e dopo eventuali modifiche di adattamento di issiue emerse a seguito dello stesso.
Compatibilmente alle tempistiche dettate da Prolific (21 giorni utili al fine di ottenere risposte utili ad una generica indagine e procedere con i relativi pagamenti) si è deciso di rendere disponibile per la compilazione Il Survey, dal 12/05/2022 fino al 03/06/2022.

In ogni caso, non appena raggiunto un livello significativo di risposte collezionate, o allo scadere del termine ultimo fissato, tali da poter passare alla successiva fase di pulizia e analisi

dei dati, la compilazione del Survey è stata disabilitata tramite apposita funzione presente sulla piattaforma Google Moduli.

4.3.3 Considerazioni Etiche

Considerando le norme vigenti in Italia, è strettamente necessario porre alcune considerazioni etiche. Dato che il questionario considera l'introduzione di figure terze è stato chiarito fin da subito che:

- Il partecipante sarà invitato a rilasciare informazioni che potrebbero essere soggette a vincoli di riservatezza aziendale, di conseguenza si rammenta che la compilazione può essere abbandonata in qualsiasi momento prima della sottomissione;
- Sarà garantito il rispetto della privacy, evitando di utilizzare le informazioni rilasciate, se non per gli scopi stessi stabiliti nella sezione introduttiva, e in ogni caso anonimizzando riferimenti diretti a dati sensibili nella rielaborazione e generalizzazione dei dati ottenuti;
- Il partecipante non vorrà comunicare tutti o alcuni dei dati sensibili richiesti, è stata prevista un opzione di risposta apposita, *Preferisco non comunicarlo*, che appunto consentirà al partecipante di restare neutrale circa la specifica informazione richiesta.
- A seguito del test pilota e prima della diffusione del survey su larga scala, è stato deciso di rendere non obbligatorie domande circa dati sensibili di carattere sociali quali sesso o etnia, per maggiori dettagli si rimanda alla sezione **Validazione del Survey**;

Qualora i partecipanti siano interessati ai futuri sviluppi dell'indagini, essi potranno in maniera facoltativa rilasciare la propria mail per ottenere un overview dei risultai e/o essere ricontattati per futuri approfondimenti. I partecipanti inoltre nella sezione di chiusura sono lasciati liberi di rilasciare qualsiasi informazione aggiuntiva che ritengano rilevante ai fini dell'indagine tramite apposito quesito aperto facoltativo.

4.4 Minacce alla validità

Differenti fattori potrebbero influenzare la validità dello studio empirico, le minacce possono essere differenti e impattare vari aspetti dello studio. A tale scopo vengono considerate di seguito le principali strategie adoperate per trattare gli aspetti di validità empirica dello studio, schematizzandole secondo le 4 principali categorie che vengono considerate

in maniera sistematica per questa tipologia di studi: validità di costrutto, validità interna, validità di conclusione e validità esterna.

4.4.1 Validità di Costrutto

La validità di costrutto, in uno studio empirico, riguarda tutte quelle possibili minacce che possono impattare la relazione che sussiste tra ipotesi e osservazioni. Per quanto riguarda lo studio specifico, la metodologia di ricerca selezionata è stata quella del Survey su larga scala, in modo tale da :

- Ottenere informazioni da esperti del dominio specifico (Data Scientists, Software Engineering etc.);
- Ottenere dati omogenei e mirati, facilmente analizzabili data la natura investigativa dello studio;

Il Survey, come viene successivamente approfondito nel capitolo di analisi dei dati, è stato progettato proprio tenendo conto della necessità di colmare quella che è la relazione causale tra ipotesi e risultati, infatti è possibile mappare ogni quesito di ricerca con uno o più domande del survey. È importante osservare come il mapping tra quesiti del Survey e obiettivi di ricerca non è stato reso pubblico ai partecipanti in fase di disseminazione, dato che anche il rendere noto quali sono gli obiettivi di ricerca e quali sono appunto le relazioni tra gli stessi e i risultati attesi può inficiare notevolmente la validità dello studio. I quesiti del survey, sono stati formulati in inglese, proprio al fine di garantire la più ampia comprensibilità da parte di tutti i partecipanti all'indagine. Oltretutto, si è deciso di applicare un lessico semplice e scorrevole nella formulazione dei quesiti, evitando periodi troppo lunghi e di difficile comprensione. Al fine di mitigare la validità di costrutto, è stato condotto, come osservato precedentemente, anche un test pilota con studenti che appunto simulassero la compilazione del Survey dal punto di vista di un individuo specifico della popolazione di interesse. Il Test pilota, ha evidenziato alcune criticità, poi risolte, circa i contenuti e la durata del Survey (ne viene data una panoramica di dettaglio nella precedente sottosezione di validazione del Survey). Con tali accorgimenti si può senz'altro dire che i partecipanti sono stati agevolati nella compilazione del survey, proprio chiedendo loro informazioni mirate e specifiche rispetto i vari obiettivi di ricerca, senza appunto andare ad intaccare o condizionare il grado di realismo del loro punto di vista.

Infine è importante constatare, che ai fini della validità di costrutto, può essere senz'altro opportuno garantire la riproducibilità dello studio. In particolare si osserva come qualsiasi informazione o risorsa utilizzata ai fini dello studio sia stata resa disponibile pubblicamente.

4.4.2 Validità Interna

Per quanto concerne la possibilità di minacce allo studio dovute dalla presenza di un campione omogeneo o con poca esperienza nello specifico ambito richiesto è necessario considerare le varie scelte effettuate per la disseminazione del survey e le successive operazioni di pulitura prima di partire con la vera e propria procedura di analisi delle risposte raccolte. Prima di tutto, come osservato in precedenza, le piattaforme utilizzate, hanno permesso di interfacciarsi in maniera agevole direttamente con la popolazione target di riferimento. Per la disseminazione del Survey si è deciso di procedere con due piattaforme separate, quindi è doveroso esprimere qualche commento per ognuna di esse:

- Per quanto riguarda la piattaforma Linkedin, ovviamente la pubblicazione del link di compilazione del survey è stata fatta in maniera oculata rispetto alla popolazione target. Il materiale infatti è stato condiviso solo in specifici gruppi della piattaforma per comunità di professionisti affini ai candidati ideali per lo studio (Data Scientist, Manager etc.), Nonostante ciò, come osservato, l'utilizzo di Linkedin non ha portato ai risultati attesi, dato che appunto non è stato possibile in alcun modo incentivare gli esperti se non descrivendo la problematica e cercando di suscitare il proprio interesse;

- L'utilizzo di una piattaforma mirata alla disseminazione di studi di ricerca quale Prolific, ha permesso di discriminare individui in maniera molto più oculata, la piattaforma stessa permette di identificare filtri tecnici, linguistici o d'altra natura sui partecipanti ad un indagine, e per quanto riguarda lo studio specifico, l'utilizzo dei filtri sui partecipanti all'indagine è servito proprio a mitigare il più possibile la presenza di partecipanti all'indagine. Dettagli maggiori, sono riportati nel paragrafo 4.3.1 Reclutamento dei partecipanti, ma è senz'altro doveroso rammentare come effettivamente Prolific non metta a disposizione filtri per competenze specifiche, perciò onde evitare la presenza di figure non affini al mondo dell'ingegneria del software e-o all'intelligenza artificiale, è stato formulato un apposito messaggio da avviso nella sezione descrittiva dello studio Prolific. Di rilievo infine, è da osservare che i partecipanti all'indagine fossero incentivati alla partecipazione tramite compenso economico così come previsto dalla piattaforma stessa.

Considerando tutti i vincoli e le potenzialità delle piattaforme utilizzate, non è senz'altro garantibile a priori la presenza di un campione privo di bias prima di procedere all'effettiva analisi dei dati. A tale scopo, sono state progettate alcune strategie di Data Cleaning che trasformino i risultati di partenza in un vero e proprio campione analizzabile. Per lo studio specifico, si è deciso non considerare per la fase di analisi dei dati risposte che non rispettassero:

- Vincoli di esperienza esplicitamente richiesti dal Survey circa l'esperienza lavorativa nel campo ML-Intensive;
- Congruenza di risposta ai quesiti di Attenction Check introdotti nel Survey, ovviamente utili al fine di garantire maggior consistenza nelle risposte;
- Vincoli di utilizzo delle informazioni ricevute, sulla base dei vincoli di pagamento espressi dalle specifiche piattaforme di disseminazione.

Si è deciso in ogni caso di fissare la validità statistica del campione ad un numero non inferiore ai 100 individui conformi a tutte le specifiche richieste (vincolo poi rispettato a seguito della chiusura dell'indagine), maggiori dettagli sono esplicitati dalla sezione 5.1 del capitolo di analisi.

4.4.3 Validità di Conclusione

Altra importante categoria di minacce potenziali per un canonico studio empirico, riguarda senz'altro le metodologie utilizzate per l'analisi dei dati e la conseguente fase di formalizzazione dei risultati. Per la tipologia di studio condotta, in relazione agli obiettivi di ricerca fissati, è stato osservato come strumenti intrinseci di statistica descrittiva, potessero meglio adattabili alle specifiche esigenze rispetto ad altre metodologie differenti di analisi. I quesiti specifici del survey, sono principalmente di due tipologie:

- Quesiti strutturati, a risposta multipla univoca o plurivoca (più valori tramite caselle di controllo);
- Quesiti aperti che permettessero ai partecipanti di fornire in maniera diretta la loro opinione su specifici dello studio.

Per quanto riguarda l'analisi di gruppi di dati inerenti a quesiti strutturati in scala, è stata valutata di volta in volta la necessità di effettuare un analisi di tipo qualitativo oppure di tipo quantitativo. Per effettuare analisi di tipo numerico quantitativo, più agevoli da

un punto di vista grafico-visivo, e.g. l'analisi di rilevanza tra Fairness e altri aspetti non funzionali del quesito di ricerca RQ3, è stato necessario adoperare alcune trasformazioni di scala per mappare i valori qualitativi presentati ai partecipanti al survey in veri e propri valori quantitativi analizzabili. Altra necessità avuta, similare alle trasformazioni di scala, è senz'altro l'utilizzo di abbreviazioni e acronimi per riadattare in maniera visuale alcuni specifici quesiti, il paragrafo *abbreviazioni e trasformazioni di scala* del successivo capitolo di analisi, riporta nel dettaglio tutte le operazioni di pre-processing adoperate prima della fase di analisi.

Altra scelta di rilievo, inerente la validità di conclusione, sta nel definire quali siano ovviamente gli strumenti di analisi specifici per ogni goal di ricerca: come è infatti osservabile dalla sezione 5.3 del capitolo di analisi, per ogni osservazione effettuata sono stati presi in considerazione più strumenti grafici, che al meglio permettessero di aggregare i dati e descrivere i risultati ottenuti, in particolare tra i differenti strumenti di statistica descrittiva a disposizione, si è deciso di utilizzare a seconda delle specifiche esigenze:

- Diagrammi di dispersione (Scatter Plot);
- Box plot;
- Diagrammi a torta (Pie Chart);
- Diagrammi a barre (Bar Diagrams).

4.4.4 Validità Esterna

Per concludere infine, la panoramica di minacce alla validità dello studio, è senz'altro opportuno porre alcune considerazioni circa la validità esterna, che effettivamente comprende quello che è il livello di generalizzabilità dei risultati. Per lo studio specifico e per le metodologie utilizzate, è opportuno osservare che le considerazioni effettuate ed i risultati formalizzati, sono strettamente legati al campione di individui ottenuto a seguito della fasi di disseminazione del Survey e di Data Cleaning (ovvero 116 indivudui). Inoltre il livello di generalizzabilità dei risultati, va osservato anche in funzione a quella che è la distribuzione geografica o lavorativa del campione ottenuto, per porre un esempio (come osservabile nel successivo paragrafo 5.3.1 Composizione del Campione), la maggior parte dei partecipanti all'indagine si dichiara europea, quindi quanto emerso dall'indagine, potrebbe subire variazioni di rilievo se lo studio fosse ripetuto in con campioni di individui con diversa estrazione geografica. Infatti, per poter dare maggior validità esterna ai risultati, sarebbe opportuno

ripetere lo studio considerando campioni di individui che oltre a possedere un background lavorativo affine a quanto richiesto ai fini dell'indagine, siano di estrazione socio-culturale o geografica variegata.

Chiusura del survey

A seguito della ricezione di un totale di 203 risposte, in data 16/05/2022, si è osservato che il materiale a disposizione potesse essere più che disponibile al fine di procedere con le successive fasi di Data Cleaning, analisi e generalizzazione dei risultati, quindi si è deciso di terminare la fase di ricezione di risposte in tale data al fine di procedere con i successivi passaggi di studio.
In particolare, si osserva come il survey realizzato e pubblicato sulla piattaforma Prolific, abbia ottenuto un totale di 203 risposte. Il Survey pubblicizzato tramite piattaforma Linkedin, invece, ha prodotto soltanto 2 risposte entrambe poco utili ai fini dell'analisi, perchè ricevute da figure professionali non affini al mondo dell'intelligenza artificiale, per tale ragione, si è deciso di non considerare questo contributo per la successiva fase di Data Analysis.

CAPITOLO 5

Analisi dei dati e risultati di ricerca

5.1 Data Cleaning

Come osservato precedentemente, il survey Prolific, ha raccolto un totale di 203 risposte totali. Prima di partire con l'effettiva analisi dei dati però, si è reso necessario validare la consistenza e l'integrità delle risposte ricevute. Inizialmente 19 sottomissioni sono state considerate inattendibili e quindi rimosse dal dateset di partenza per i seguenti motivi:

- 17 sono state considerate inattendibili a causa di risposte errate alle domande poste come attenction check di verifica, quindi successivamente eliminate dal dataset originale.
- 2 risposte sono state eliminate, osservando inconguenze con l'identificativo Prolific immesso.

Delle 184 sottomissioni restanti, c'è inoltre da considerare che 68 partecipanti hanno dichiarato di non aver alcuna esperienza con lo sviluppo di moduli AI-intensive, quindi sono stati direttamente condotti alla sezione di chiusura del Survey elettronico, anche se mantenute nel dataset di partenza, tali risposte (per strutturazione intrinseca del survey) non possedevano informazioni utili per rispondere i quesiti di ricerca, ma data la presenza di quesiti aperti e contatti utili, esse sono state trasferite in un dataset secondario da utilizzare per successive fasi di approfondimento. I restanti 116 partecipanti all'indagine hanno invece dichiarato di avere effettivamente esperienza con lo sviluppo di soluzioni AI-Intensive,

quindi le risposte relative sono state ritenute utili al fine di elaborare i risultati dello studio in risposta agli obiettivi di ricerca prefissati precedentemente.

Da notare inoltre come 18 partecipanti dei 116 restanti (con esperienza nello sviluppo di soluzioni AI-Intensive), abbiano lasciato a disposizione un loro contatto email per eventuali interviste future, o per ottenere nuovi aggiornamenti circa l'elaborazione dei risultati.

5.2 Pre-processing

Prima di procedere con il vero e proprio lavoro di analisi, è stato necessario applicare alcune piccole trasformazioni ai dati grezzi, e consequenzialmente prima di illustrare il lavoro di analisi, si riportano riferimenti alle trasformazioni applicate al dataset, utili a interpretare i risultati ottenuti in maniera più agevole.

5.2.1 Mapping tra quesiti del survey e obiettivi di ricerca

Al fine di comprendere meglio quali quesiti, posti ai partecipanti all'indagine empirica, forniscano dati utili a rispondere ad uno specifico quesito di ricerca, il dataset di risposte iniziale, è stato diviso in 5 dataset più piccoli, ciascuno corrispondente ad uno specifico sub-goal di ricerca formalizzato precedentemente. La successiva tabella 5.1 riassume i dettagli dell'attività di mapping.

RQ1: Quali sono i migliori approcci e definizioni per trattare la fairness in un contesto lavorativo?
Secondo te, quali dei seguenti aspetti rappresentano la definizione generica di fairness fornita in precedenza?
Considerando la tua esperienza lavorativa, quanto i seguenti (approcci) sono trattati?
Generalmente utilizzi altri approcci per lavorare con il concetto di Software Fairness?
RQ2: Com'è generalmente composto un team lavorativo per lo sviluppo di moduli ML-Intensive Fair Critical?
Considerando i seguenti ruoli (professionali), chi ha impatto sulle scelte inerenti la software fairness?

RQ3: Quanto il concetto di software fairness è importante se paragonato ad altri aspetti non funzionali?
Considerando i seguenti aspetti (funzionali e non funzionali) dello sviluppo software, quanto li ritieni importanti se comparati alla fairness?
RQ4: In quali fasi di una tipica pipeline di Machine Learning è importante adottare strategie per garantire alti livelli di fairness?
Considerando una generica pipeline di machine learning (come la seguente - figura 4.1), quanto consideri l'equità come un aspetto rilevante per ciascuna delle seguenti fasi nel tuo contesto lavorativo?
Quali tool utilizzi (se previsti) per trattare la fairness in una pipeline di machine learning ?
RQ5: Quanto le compagnie di sviluppo ML-Intensive, sono mature nel trattare il concetto di fairness come un requisito non funzionale?
In quale dei seguenti livelli di maturità, classificheresti il tuo ambiente lavorativo circa il trattamento della fairness?

Tabella 5.1: Survey Question & Research goal mapping

5.2.2 Abbreviazioni e trasformazioni di scala

Considerando che molti concetti formalizzati sul questionario, sono stati espressi in forma discorsiva per garantirne una maggiore comprensione ai partecipanti all'indagine, si è reso necessario convertirli prima della fase di aggregazione dei dati, in valori facilmente compatibili agli strumenti automatici di analisi utilizzati. I quesiti specifici, per cui si è resa necessaria questa attività, sono state quindi formalizzate due tipologie di scale, una quantitativa o riassuntiva (composta da alias per il concetto espresso in forma discorsiva) e una qualitativa (contenente le vere e proprie opzioni di risposta alla domanda di riferimento).

Le successive tabelle riassumono quindi per intero le trasformazioni di scala applicate.

Definizioni di fairness e abbreviazioni

RQ1 - Quesito: Secondo te, quali dei seguenti aspetti rappresentano la definizione generica di fairness fornita in precedenza?

Valore Riassuntivo	***Valore qualitativo (risposte)***
Definizioni Probabilistiche	Treating similar individuals in a way that they are equally likely to receive a specific outcome
Definizioni basate su similarità matematica	Do not favor certain subjects over others on the basis of sensitive attributes, e.g., race, gender, etc.
Definizioni basate su casual reasoning	Taking decisions by protecting individuals and groups from mistreatments

Tabella 5.2: Mapping tra le tipologie di definizione di fairness e la loro forma discorsiva

Approcci al trattamento della fairness e abbreviazioni

RQ1 - Quesito: Considerando la tua esperienza lavorativa, quanto i seguenti (approcci) sono trattati?

Valore Riassuntivo	***Valore qualitativo (risposte)***
Approccio 1	We focus on guaranteeing high probability to obtain ethically correct outcomes regardless of sensitive features
Approccio 2	We focus on guaranteeing that machine learning predictions are not going to discriminate by sensitive features

Approccio 3	We model the relation between attributes and outcomes, verifying that the outcome does not depend on sensitive attributes

Tabella 5.3: Mapping tra le tipologie di approcci alla fairness e la loro forma discorsiva

Cambiamenti di scala per l'applicabilità di definizioni e approcci alla fairness

RQ1 - Quesito: Secondo te, quali dei seguenti aspetti rappresentano la definizione generica di fairness fornita in precedenza?

RQ1 - Quesito: Considerando la tua esperienza lavorativa, quanto i seguenti (approcci) sono trattati?

Scala quantitativa	***Valore qualitativo (risposte)***
1	Not at all
2	Slightly
3	Neutral
4	To a great extent
5	Extremely

Tabella 5.4: Scale qualitativa e quantitativa per la valutazione di definizioni e approcci

Cambiamenti di scala circa l'impatto professionale nel trattamento della fairness

RQ2 - Quesito: Considerando i seguenti ruoli (professionali), chi ha impatto sulle scelte inerenti la software fairness?

Scala quantitativa	***Valore qualitativo (risposte)***
1	Very low impact
2	Below average impact
3	Average impact
4	Above average impact
5	Very high impact

Tabella 5.5: Scale qualitativa e quantitativa per la valutazione dell'impatto professionale

Cambiamenti di scala circa la valutazione di importanza della fairness rispetto altri NFR

RQ3 - Quesito: Considerando i seguenti aspetti (funzionali e non funzionali) dello sviluppo software, quanto li ritieni importanti se comparati alla fairness?

Scala quantitativa	***Valore qualitativo (risposte)***
-2	Less important than fairness
-1	A bit less important than fairness
0	Neutral

1	A bit more important than fairness
2	More important than fairness

Tabella 5.6: Scale qualitativa e quantitativa per la valutazione dei fairness trade-offs

Cambiamenti di scala circa l'utilità di applicazione di strategie Fair-Oriented in una pipeline ML

RQ4 - Considerando una generica pipeline di machine learning (come la seguente - figura 4.1), quanto consideri l'equità come un aspetto rilevante per ciascuna delle seguenti fasi nel tuo contesto lavorativo?

Scala quantitativa	***Valore qualitativo (risposte)***
1	Not at all
2	Slightly
3	Neutral
4	Very
5	Extremelly

Tabella 5.7: Scale qualitativa e quantitativa per l'impatto di fairness su una Pipeline di Machine Learning

Livelli di maturità aziendale e spiegazine relativa

RQ5 - In quale dei seguenti livelli di maturità, classificheresti il tuo ambiente lavorativo circa il trattamento della fairness?

Livello	***Spiegazione***
Livello 0	We do not treat software fairness
Livello 1	We occasionally treat software fairness, but related processes are disorganized and even chaotic
Livello 2	We regularly treat fairness and related processes are established, defined and documented
Livello 3	We regularly treat fairness and it develops its own standard fairness management processes
Livello 4	We regularly treat fairness and it monitors and controls its own fairness related processes through data collection and analysis
Livello 5	We regularly treat fairness and fairness related processes are constantly improved through monitoring feedback

Tabella 5.8: Scale qualitativa e quantitativa per l'impatto di fairness su una Pipeline di Machine Learning

5.3 Analisi dei dati

Una volta realizzate le dovute trasformazioni di scala e la sistematizzazione dei concetti, il campione empirico, risulta quindi pronto per essere effettivamente analizzato. Ovviamente per la fase di analisi, è stato necessario selezionare un opportuno tool, che automatizzasse l'aggregazione statistica de dati e la formulazione di grafici descrittivi, tra le varie alternative disponibili, a tale scopo è stato selezionato il linguaggio statistico R, tramite il relativo tool di utilizzo RStudiio e la libreria ggplot2 per la formulazione di grafici descrittivi.

5.3.1 Composizione del campione

Prima di cominciare effettivamente con l'analizzare i risultati specifici di ogni sub-goal di ricerca, è opportuno effettuare qualche considerazione sul campione di 116 individui considerati validi dopo la fase di data cleaning.

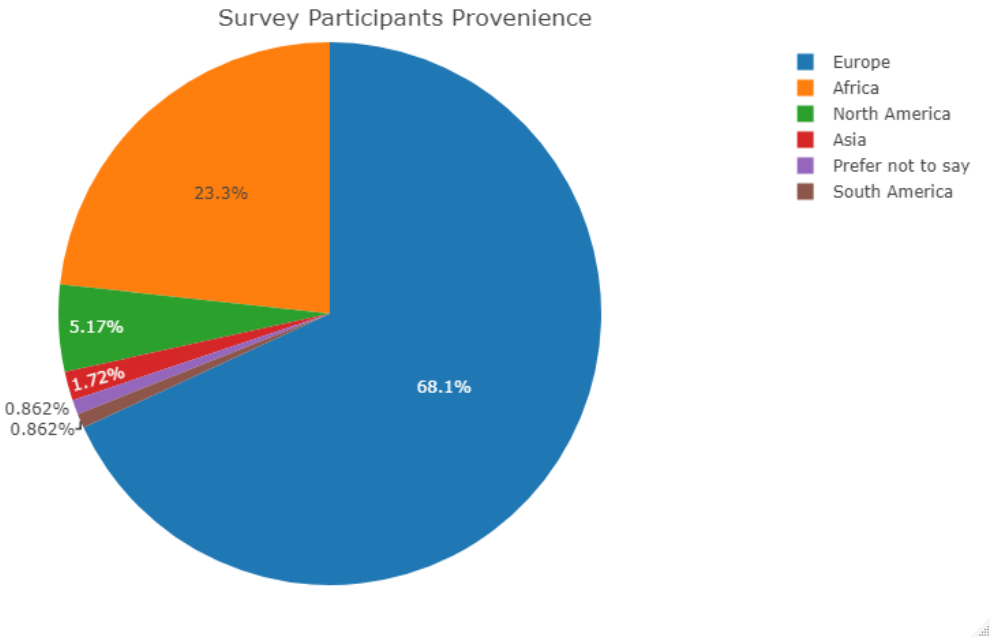

Figura 5.1: Distribuzione continentale del campione di analisi

Come osservabile dalla figura 5.1, il campione di analisi, è principalmente concentrato tra Europa e Africa, ciò significa appunto che le deduzioni successive, questo dettaglio non è trascurabile, dato che la generalizzabilità dei risultati, potrebbe essere messa in discussione, per aree geografiche rappresentate in maniera minore. Ad ogni modo, questo prima considerazione, è senz'altro da attribuire al fatto che Prolific è essenzialmente una piattaforma di origine britannica, quindi maggiormente pubblicizzata in Europa. Nel dettaglio, 79 parteci-

panti hanno dichiarato di provenire dall'europa, 27 dall'africa, 6 dal nord america, 2 dall'asia, 1 dal sud america, mentre 1 ha preferito non dichiarare la sua provenienza.

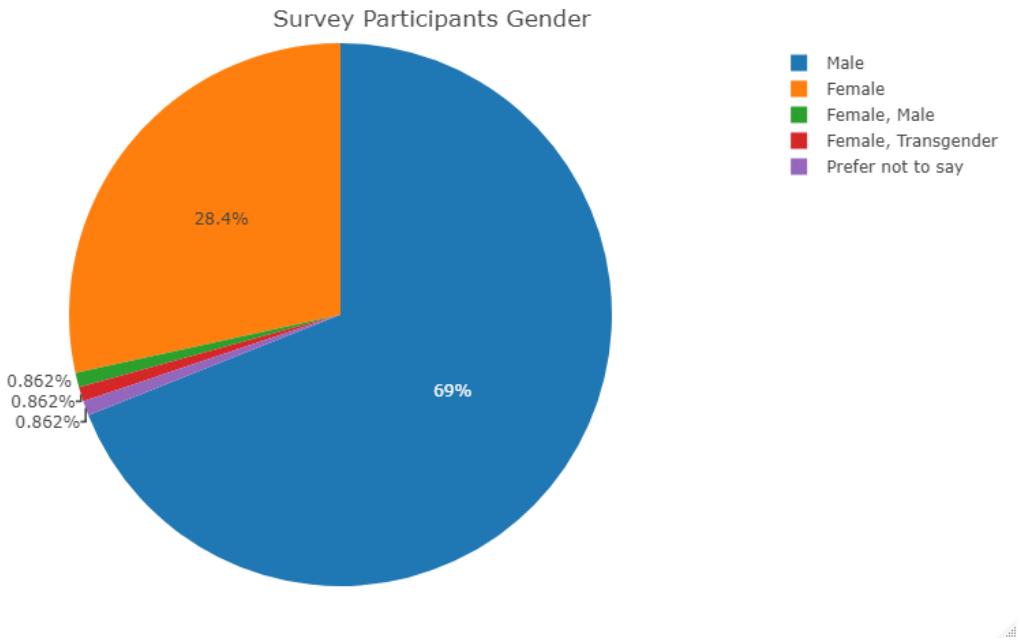

Figura 5.2: Distribuzione di gender nel campione di analisi

Analogamente alla provenienza, con la figura 5.2 è possibile effettuare qualche considerazione analoga, con l'identificazione di gender dei partecipanti. la presenza di tale quesito è stato ampiamente discussa in fase di progettazione del questionario, dato che essa è spesso un informazione da ritenere altamente discriminatoria se non non richiesta nel modo giusto, a tale scopo, è stata introdotta l'opzione di controllo Prefer Not To Say, che tuttavia è stata selezionata solo da un partecipante su 116. 80 invece si identificano nel gender maschile, 33 in in quello femminile, 1 in entrambi i precedenti, 1 come Transgender e donna contemporaneamente.

Considerando invece la figura 5.3 è possibile notare, come la maggioranza del campione, si concentri maggiormente tra i 18 e i 30 anni, più nello specifico 79 partecipanti si colloca in questo range, 34 invece sono i partecipanti appartenenti alla fascia di età tra i 31 e i 50 anni, portando quindi anche maggiore maturità professionale al campione, mentre infine 3 partecipanti dichiarano di possedere più di 50 anni.

Fin dall'inizio della fase di progettazione del Survey, al fine dell'attendibilità delle informazioni ricevute è stata considerato come fattore di rilevante importanza il livello di studio medio dei partecipanti, preferendo in particolare un alto livello di studio (laurea triennale e superiori), dalla figura 5.4 è osservabile come tale tendenza sia rispettata da quasi tutti

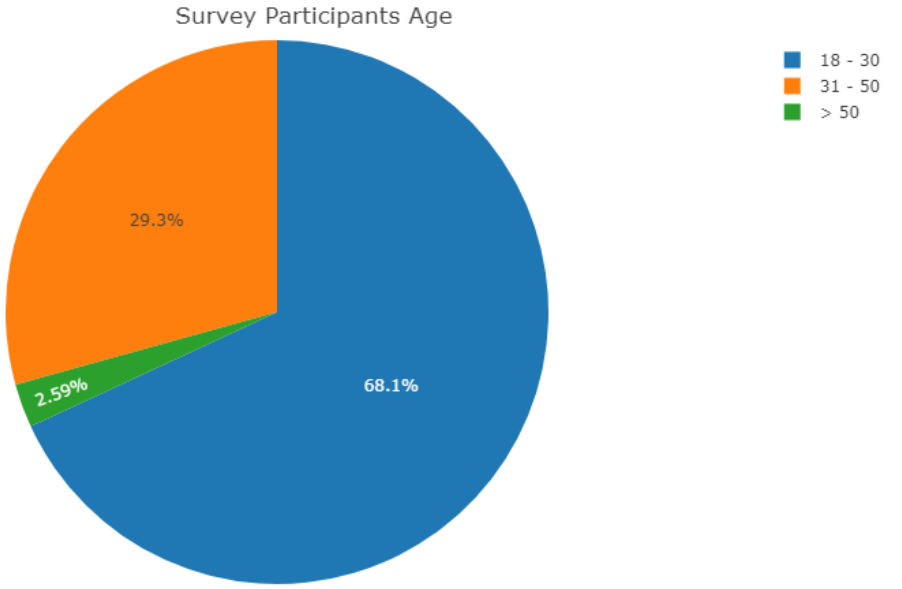

Figura 5.3: Distribuzione dell'età nel campione di analisi

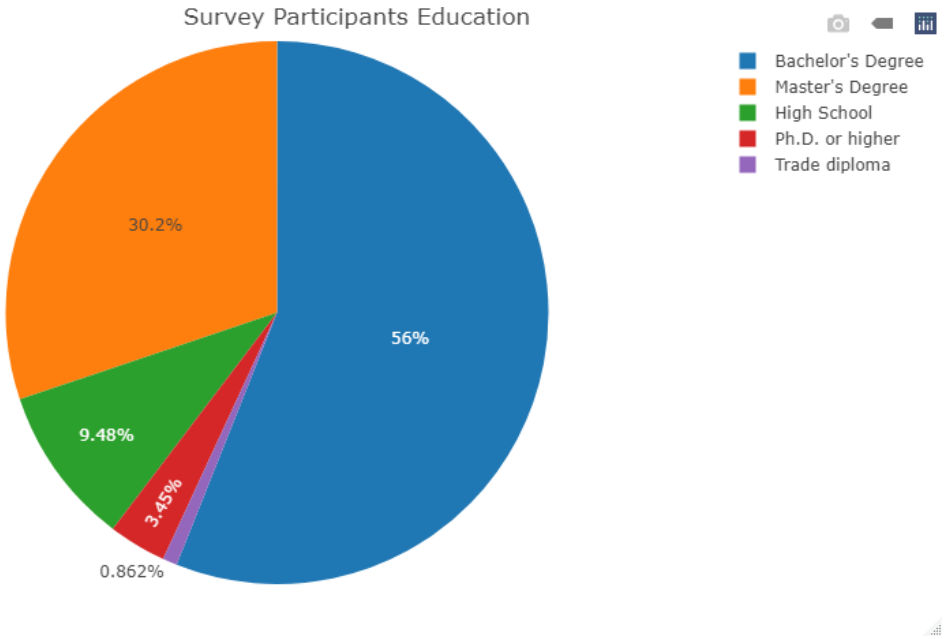

Figura 5.4: Distribuzione del livello di studi nel campione di analisi

i partecipanti all'indagine, nel dettaglio su 116 partecipanti totali, 65 dichiarano di possedere almeno un titolo di studi triennale, 35 invece dichiarano di aver conseguito un titolo magistrale, 4 risultano aver completato anche il dottorato di ricerca, mentre i restati 12 si dividono in 11 con un diploma di istruzione superiore e un singolo con diploma professionale.

Continuando poi con la posizione e il ruolo professionale dei partecipanti, è possibile notare come effettivamente, il campione degli intervistati sia abbastanza variegato e

corrispondente ai requisiti di appartenenza precedentemente fissati.

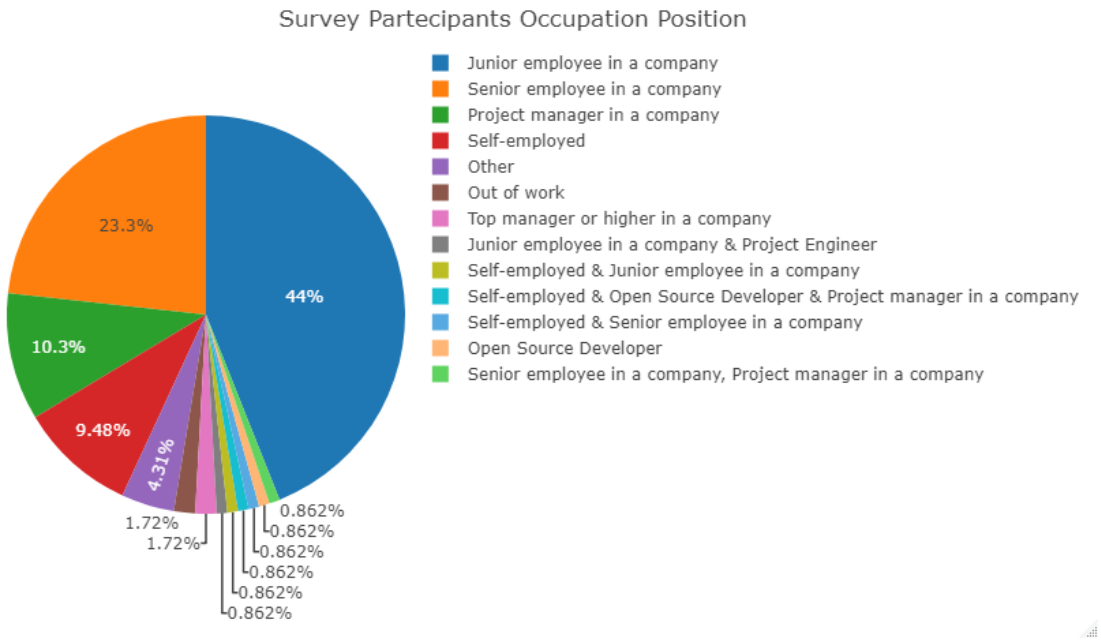

Figura 5.5: Livello di occupazione professionale dei partecipanti all'analisi

In particolare, come osserva la figura 5.5, la maggioranza dei partecipanti ricopre un ruolo attivo in una compagnia, nel dettaglio è possibile dividere questa fetta in 51 impiegati aziendali junior, 27 impiegati aziendali Senior, e 12 Project Manager e 2 come Top-Manager in azienda. Tra i restanti, 11 si dichiarano come lavoratori autonomi. Tra i restanti, ci sono figure professionali singole, che dichiarano di ricoprire più di una posizione professionale contemporaneamente o altri ruoli quali sviluppatore open source, infine risultano esserci 5 partecipanti che dichiarano di avere altra posizione altro rispetto a quelle specificate, mentre solo 2 risultano disoccupati al momento della sottomissione.

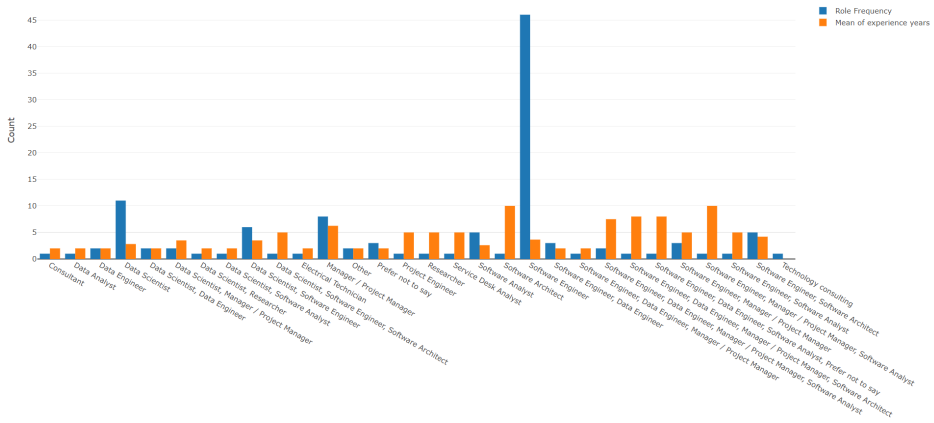

Figura 5.6: Ruoli professionali dei partecipanti all'analisi

Dando invece uno sguardo ai ruoli professionali, è possibile osservare come circa la metà

o poco più dei partecipanti, abbia affinità con l'ingegneria del software, l'ingegneria dei dati, e il management, leggermente meno ampia è la presenza di ruoli affini a posizioni quali Data Science o altri ruoli professionali, mentre in media. Mentre è di rilievo considerare che tutti i partecipanti hanno in media dai 2 ai 10 anni di esperienza nel ruolo indicato. Da un punto di vista empirico, l'eterogeneità del campione è senz'altro molto positiva dato che sia per via dei differenti ruoli, che per i variegati livelli di occupazione, è possibile dare più valenza statistica alle successive informazioni estratte dai dati grezzi.

5.3.2 Fairness in pratica, come rispondere al quesito generale?

RQ - In che modo il concetto di Software Fairness è attualmente percepito nell'ambiente lavorativo ML-Intensive?

Come più volte osservato più volte, cercar di raccogliere dati utili a rispondere in maniera formale a questo unico macro-quesito, può essere qualcosa di generalmente complesso e forse anche riduttivo da proporre per un concetto così variegato come quello di Software Fairness, sicuramente considerando l'andamento generale dell'analisi dei dati, così come viene riportato nei successivi paragrafi, è senz'altro facile capire che, come stabilito dalla ricerca Fairness, è un concetto estremamente variabile ed in evoluzione, quindi così come per la fase di progettazione, anche quella di analisi risulta essere più assimilabile se scomposta in sotto-punti speculari ai sub-goal di ricerca definiti.

5.3.3 Applicabilità di definizioni e approcci per fairness

RQ1 - Quali sono i migliori approcci e definizioni per trattare la fairness in un contesto lavorativo?

Data l'ampia variabilità del concetto, non è corretto porre un quesito di ricerca che voglia definire univocamente il concetto o la definizione *aziendale* di software fairness. Da un punto di vista analitico, la figura 5.7, dimostra infatti come per gli esperti, tutte le tipologie di definizioni e metriche teoriche, possono essere ampiamente applicate in casi reali di sviluppo, per tutte e tre i gruppi di definizione, i livelli 4 e 5 delle scale di applicabilità superano il 50% dei consensi, a seconda dei contesti di applicazioni e dei requisiti specifici del modulo ML. Da un'analisi più attenta, però si può osservare come i partecipanti all'indagine ritengano lievemente più applicabili i gruppi di definizione e metriche basati su **similarità matematica**,

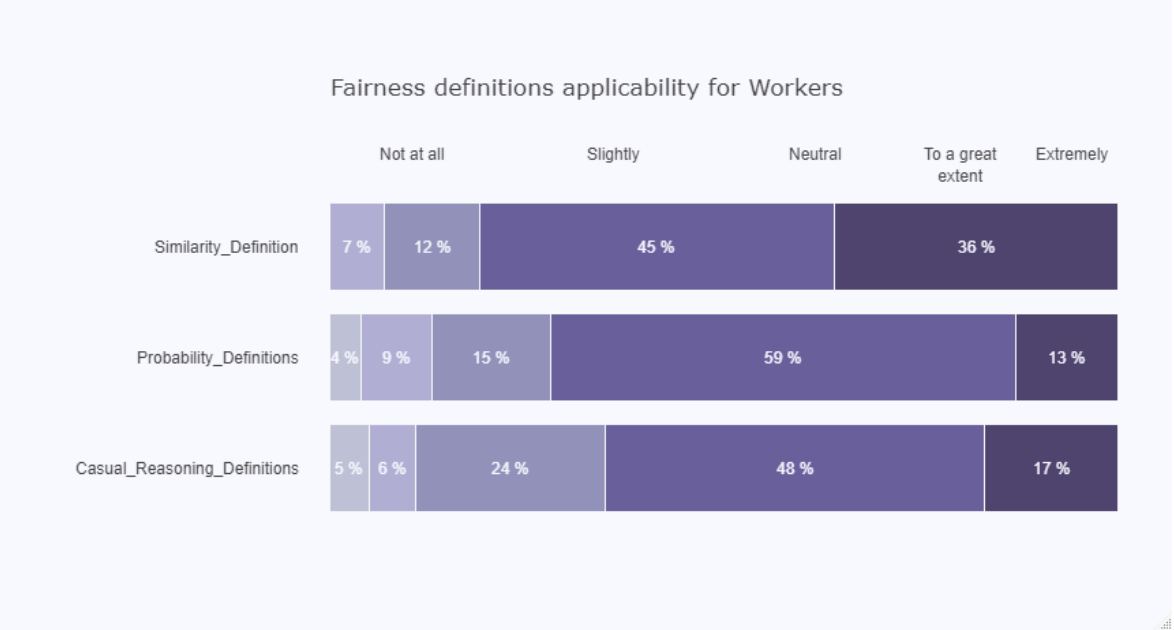

Figura 5.7: Definizioni di fairness in ambito lavorativo

rispetto a quelli definiti in **puri termini probabilistici** piuttosto che quelli basati su **relazioni causali tra features e outcome**.

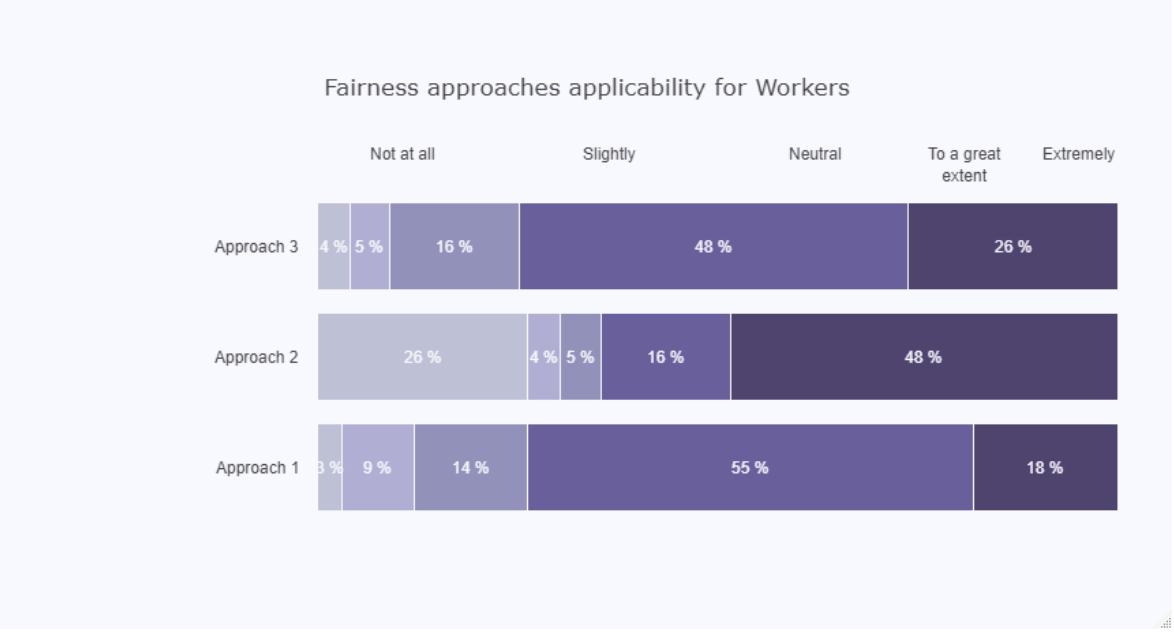

Figura 5.8: Approcci al concetto di fairness in ambito lavorativo

Contrariamente se si osservano gli approcci pratici proposti, si osserva facilmente come in azienda prevalga facilmente l'idea che **i livelli alti di Software Fairness**, vadano ricercati cercando di **ridurre le dipendenze tra gli attributi sensibili e i risultati predetti in fasi di data preparation** (approccio 1), e quindi specularmente nel **cercare di constatare che**

i risultati stessi non dipendano da feature sensibili in fase di validazione (approccio 3). Meno percepito come applicabile, risulta essere invece il secondo approccio proposto, ovvero concentrarsi nel garantire che il learner non effettui valutazioni discriminatorie sulla base di feature sensibili. Ciò è probabilmente da attribuire, alla complessità di questa seconda opzione proposta, non facilmente intuibile e difficilmente analizzabile rispetto le altre due.

Ovviamente anche in azienda, il concetto di fairness non si limita a quanto espresso in teoria, esso può essere visto a vari livelli di dettaglio, e soprattutto in riferimento al dominio di utilizzo. A dimostrazione di ciò, risultano interessanti molte risposte ottenute al terzo quesito (risposta aperta breve) progettato per questo primo sub-goal di ricerca, ovvero: Generalmente utilizzi altri approcci per lavorare con il concetto di Software Fairness?

Segue un breve report di alcune delle considerazioni più interessanti:

- Approcci Domain Specific - Provare a rendere i risultati disponibili a persone con disabilità, quindi in sostanza, renderli equi per tutti;
- Metodologie Empiriche - Condurre survey e ottenere in cosiderazioni opinioni diverse;
- Ottimizzazione nella gestione dei dati - Ottenere dei dati di training sensibili abbastanza affinchè i risultati siano il meno possibile discriminanti;
- Analisi di correlazione - Analizzare la correlazione forte-debole tra risultati e features;

5.3.4 Impatto professionale per il trattamento della fairness

RQ2 - Come è composto generalmente un team lavorativo per lo sviluppo di moduli ML-Intensive Fair Critical?

Osservando le figure 5.9 e 5.10, è facilmente intuibile, come **tutti i ruoli professionali** proposti,vengano considerati cruciali durante lo sviluppo di una soluzione fair critical, infatti il valore mediano per ogni figura professionale si attesta almeno a 4 (secondo valore di rilevanza nella scala precedentemente illustrata). Volendo entrare più nel dettaglio, è possibile osservare come **manager** ed **esperti specifici**, siano figure estremamente di rilievo e utili nella gestione degli aspetti di equità di un modulo di Machine Learning. Risultano molto simili sono invece attribuibili a figure come Data Scientist, Data Engineer e Ingegneri del software, a dimostrazione del fatto che, anche in un contesto reale, mantenere alta la sinergia tra queste due branche dello sviluppo ML-Intensive sia cruciale soprattutto in contesti molto

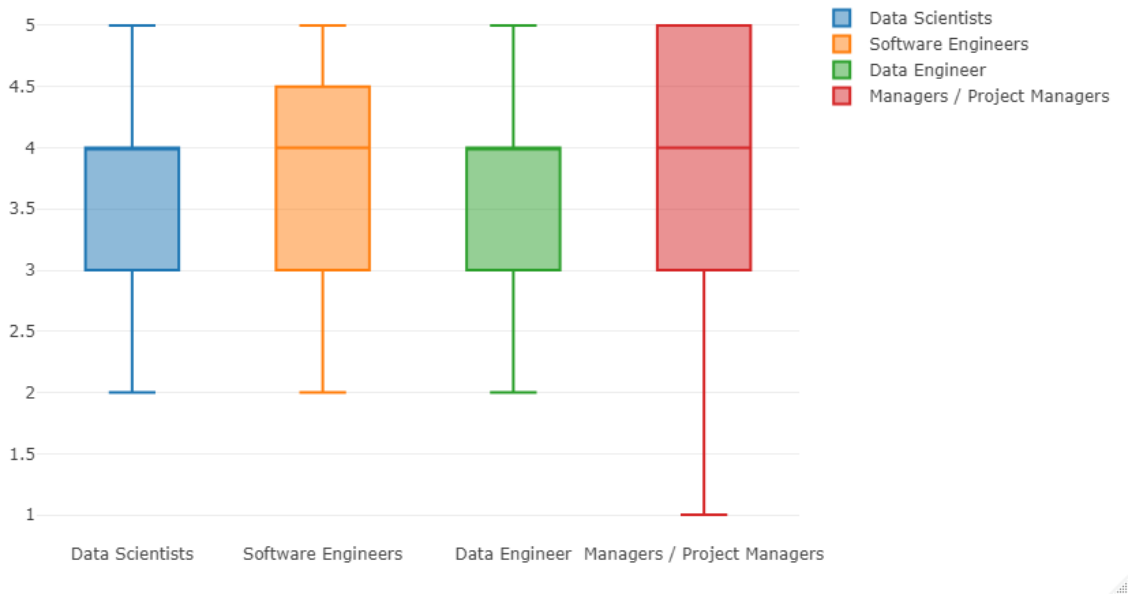

Figura 5.9: Ruoli professionali durante lo sviluppo Fair Oriented 1/2

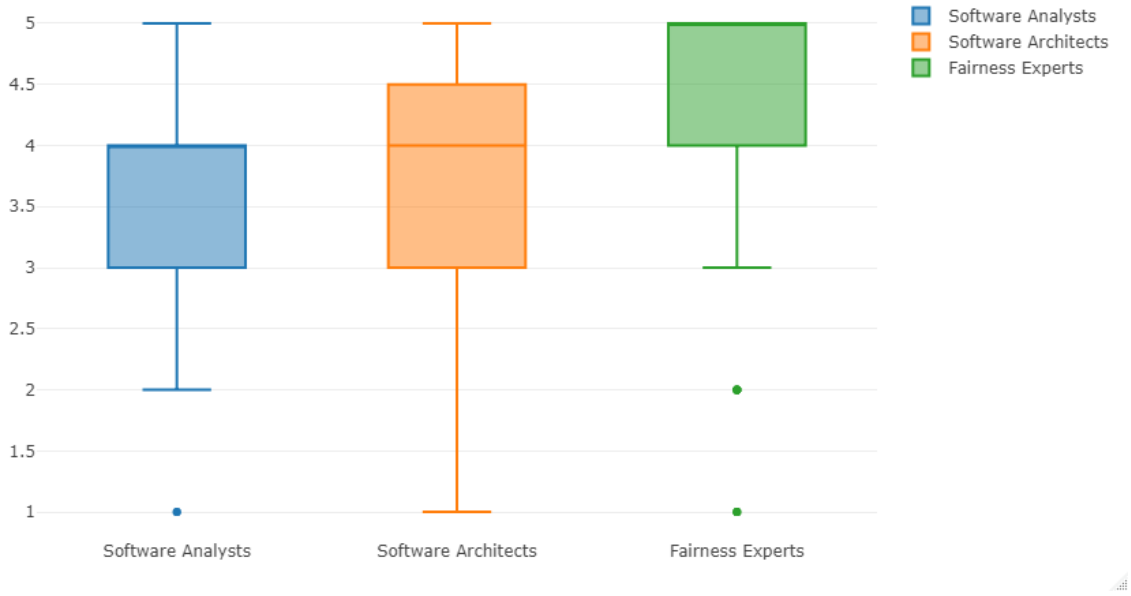

Figura 5.10: Ruoli professionali durante lo sviluppo Fair Oriented 2/2

specifici come quello della fairness. Da notare come ci sia una leggera presenza di outliers per ruoli come Analisti ed Esperti di fairness (basso impatto in controtendenza con il resto del campione), ma ciò è da considerare come fattore eccezionale, dato che tali figure possono essere comunque molto specifiche e magari non presenti in ogni ambiente lavorativo dei partecipanti all'indagine.

5.3.5 Confronto tra fairness con altre caratteristiche qualitative non funzionali

RQ3 - Quanto il concetto di software fairness è importante se paragonato ad altri aspetti non funzionali?

Come visto più volte, analizzare la fairness, come un vero e proprio attributo non funzionale di prima classe sia la chiave di lettura al fine di eliminare fault etici latenti in un modulo di machine learning [4]. Ovviamente chiedere richiedere tale sforzo ai professionisti è un qualcosa di molto ostico soprattutto mettendo a paragone Fairness, con requisiti molto più standardizzati, quali accuracy o sicurezza. Ma analizzando i risultati, è possibile provare ad osservare qualche deduzione interessante.

N.B. nei successivi diagrammi, sono state rappresentate le densità delle singole risposte dei partecipanti, in blu si può notare la fetta di persone che considera uno specifico requisito altamente meno importante rispetto a Fairness (-2 della scala quantitativa) , in verde è racchiusa la parte di campione che considera fairness leggermente più importate rispetto all'altro requisito analizzato dal diagramma (-1 della scala quantitativa), in giallo quella che considera fairness rilevante quanto il requisito confrontato (0 nella scala quantitativa), in arancione quella fetta di campione che considera il requisito in analisi lievemente più importante rispetto a fairness (+1), mentre in rosso gli individui che considerano il requisito specifico altamente più importante di fairness (+2).

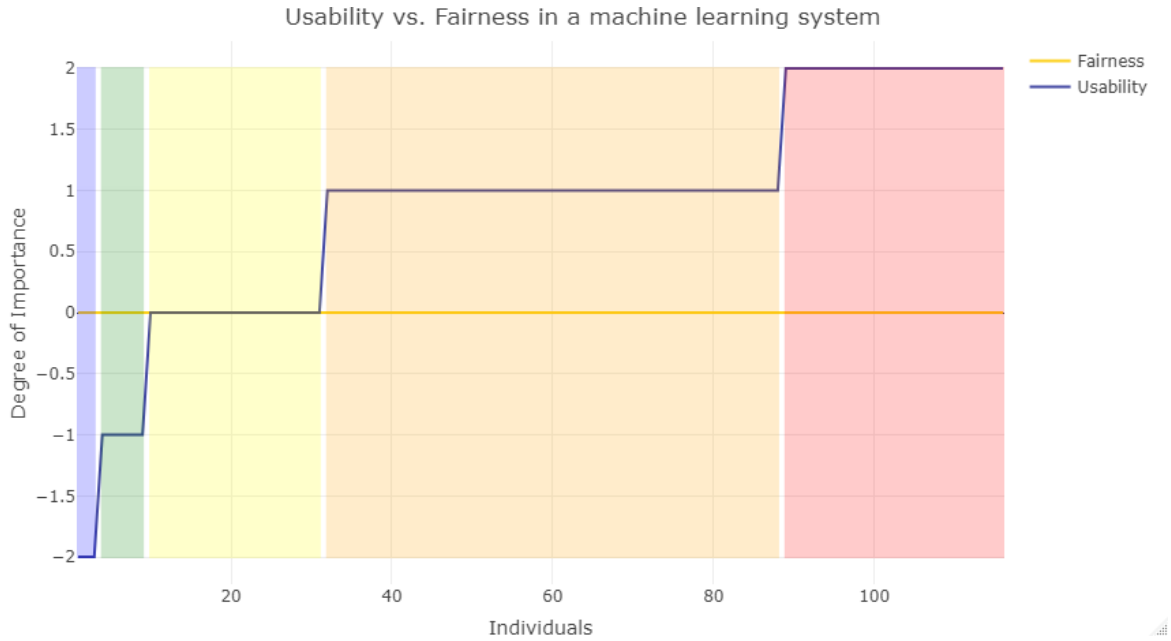

Figura 5.11: Fairness vs Usabilità

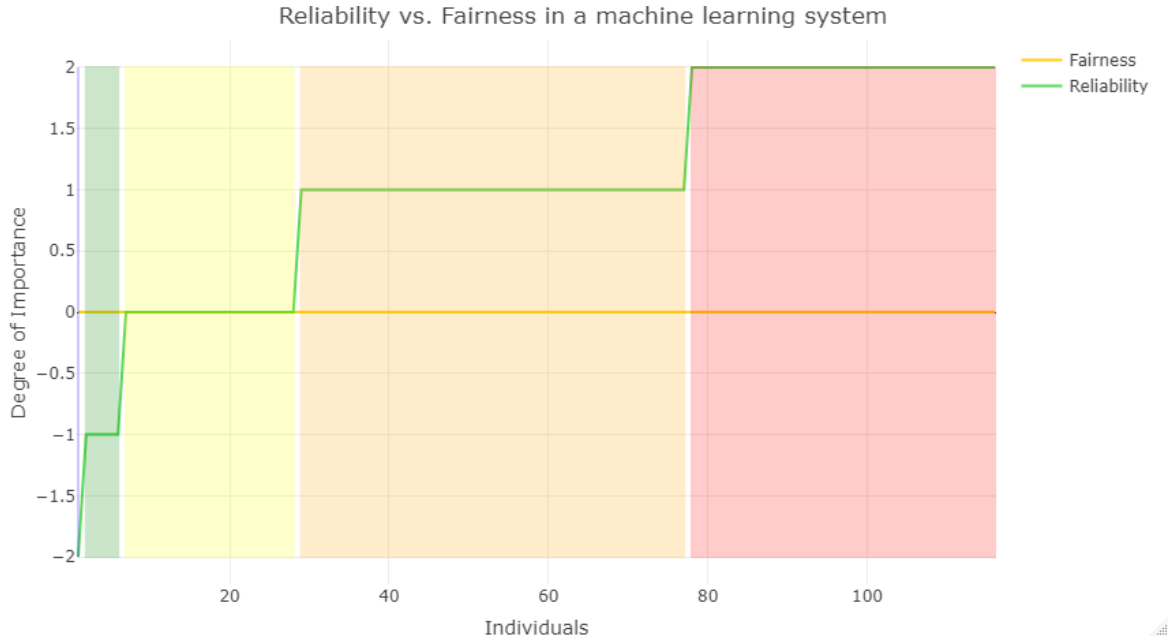

Figura 5.12: Fairness vs Affidabilità

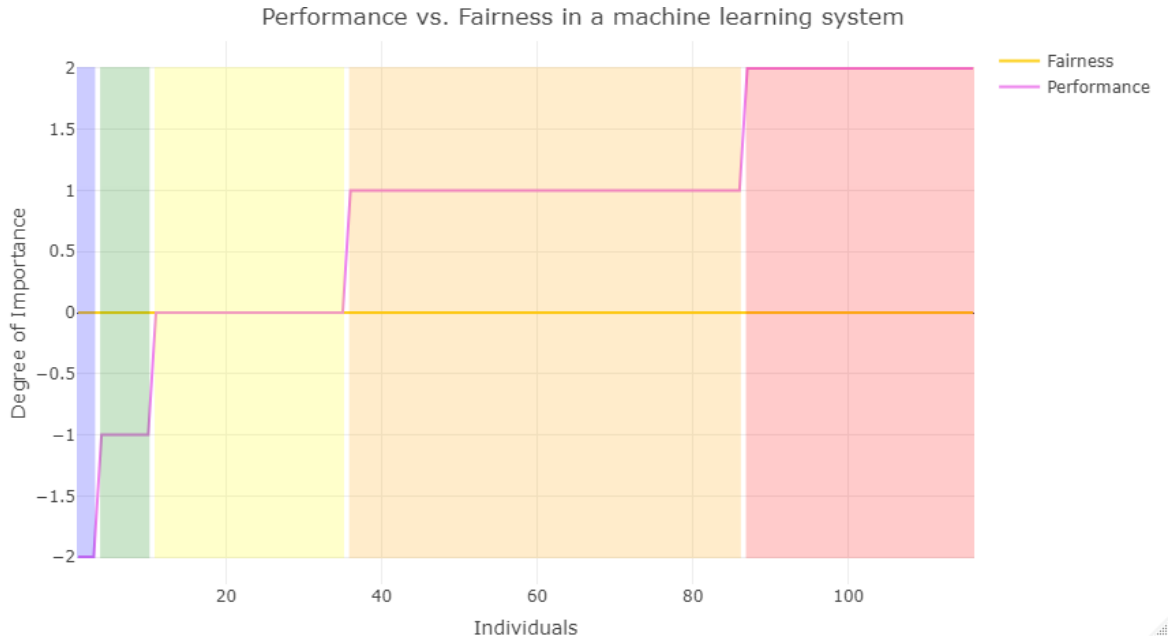

Figura 5.13: Fairness vs Performances

Come osservabile dalle figure 5.11 alla 5.14, aspetti non funzionali standard, del noto modello furps+ (usabilità, affidabilità, performance e supportabilità), sono aspetti che per la maggior parte del campione, risultano essere di misura più rilevanti di fairness, in particolare si osserva che per aspetti particolarmente critici per i moduli di intelligenza artificiale, quali **affidabilità e performances**, la fetta di persone che li considera **estremamente più importanti**, sale anche **sopra le 25/30 unità**, per diminuire leggermente negli altri aspetti riportati. È però da tener conto come quasi tutti gli aspetti del modello iniziale analizzato (Furps+),

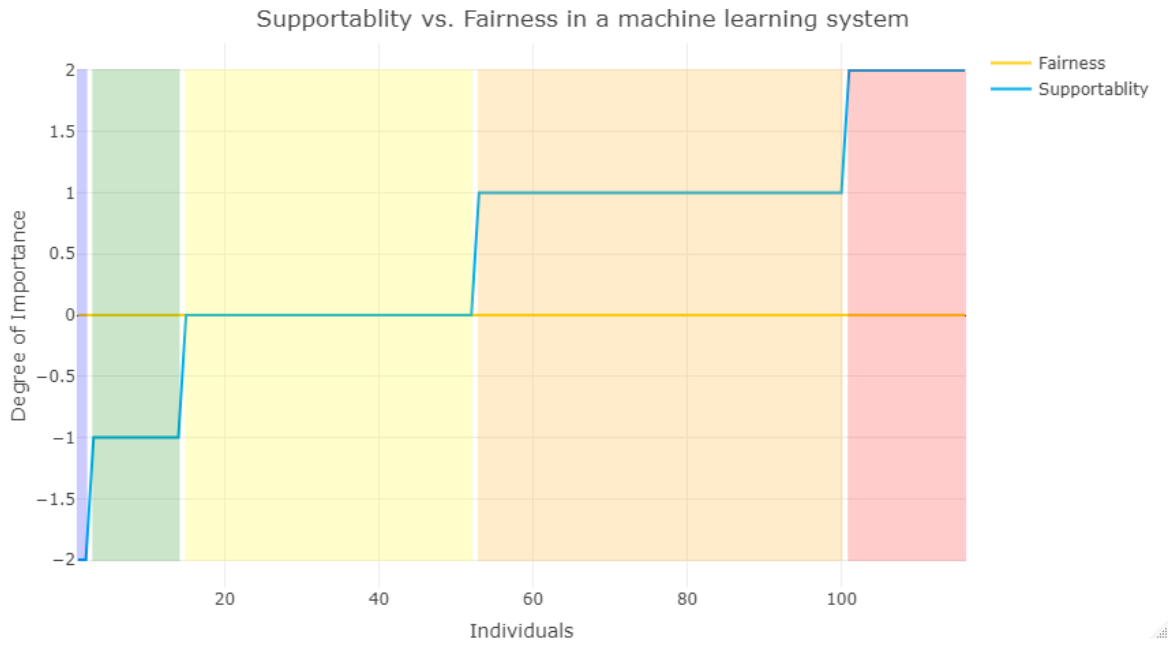

Figura 5.14: Fairness vs Supportabilità

come fairness, siano altamente variabili a seconda delle necessità del tool specifico, infatti, osservando come la maggior parte dei partecipanti consideri **usabilità e supportabilità** leggermente più impattanti nello sviluppo ml-intensive, rispetto fairness, si può sicuramente lasciare margine di confronto tra questi attributi e fairness a seconda delle specifiche esigenze.

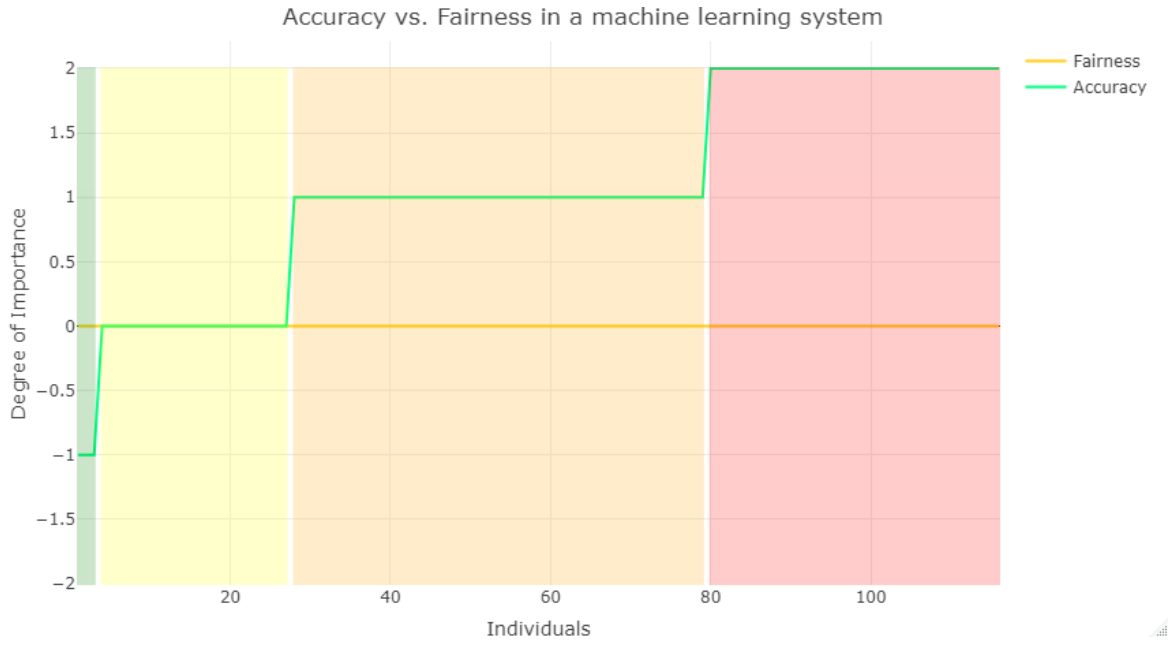

Figura 5.15: Fairness vs Accuracy

Il discorso cambia di misura per aspetti già più specifici per un canonico modello di

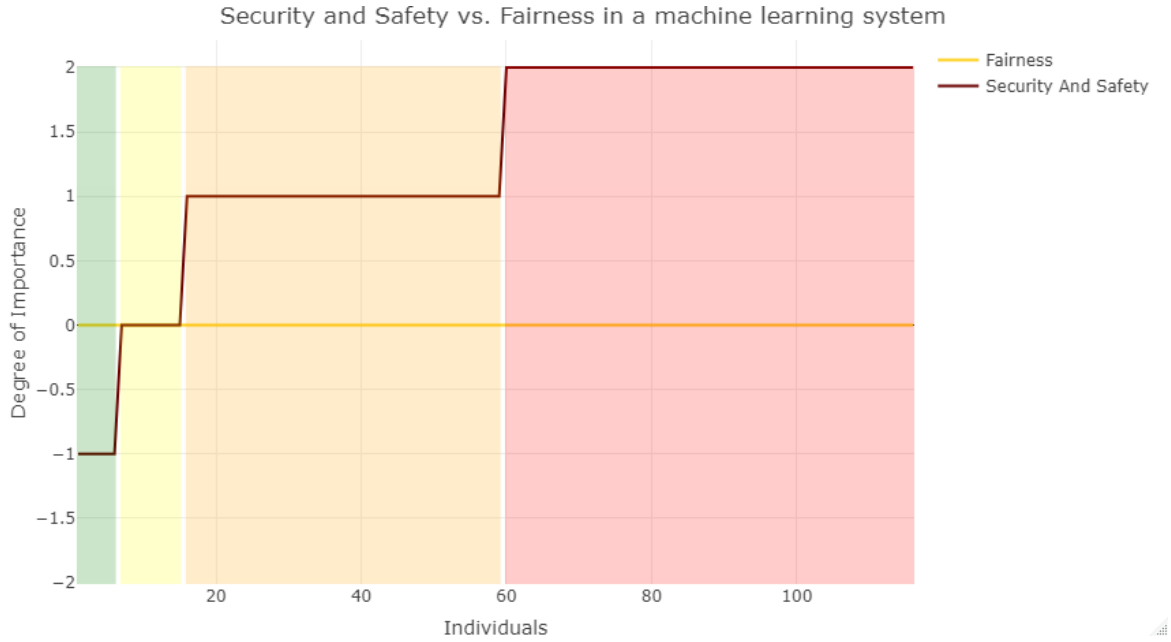

Figura 5.16: Fairness vs Sicurezza

machine learning, infatti sia per accuracy che sicurezza, nessuno dei partecipanti ha ritenuto opportuno indicarli come ampiamente meno importanti rispetto alla fairness, inoltre per questi due aspetti anche la finestra di voto che va da -1 a 0, risulta essere molto ristretta. Sia **l'accuracy che la Sicurezza, sono da considerare da lievemente ad estremamente più rilevanti** rispetto a fairness sulla base del campione. Ciò molto probabilmente è dovuto al fatto che chi lavora quotidianamente con questi sistemi, in maniera quasi automatica si troverà a lavorare a problematiche di accuracy o sicurezza del modello, cosa non sempre così semplice (almeno allo stato attuale delle cose) per fairness.

In maniera più semplificata (aggregando i concetti), è stato chiesto ai partecipanti di provare a fornire un parere similare alle altre specifiche non funzionali quali manutenibilità & retraining - figura 5.17 (in ottica evolutiva di un generico tool ML-Intensive) e scalabilità & riusabilità - figura 5.18. Per questa tipologia di requisiti, i report grafici, possono facilmente far dedurre considerazioni similari rispetto al modello Furps+, ma probabilmente per arrivare ad osservazioni più precise, si dovrebbe analizzare separatamente i singoli dati (scelta non adottata per evitare di rendere troppo complesso il quesito ai partecipanti).

In generale, le problematiche discriminatorie ed il concetto di software fairness, sono probabilmente concetti molto distanti dagli aspetti di qualità, già ampiamente studiati e sistematizzati. Allo stato della pratica, il generico lavoratore, sia esso data scientist, ingegnere

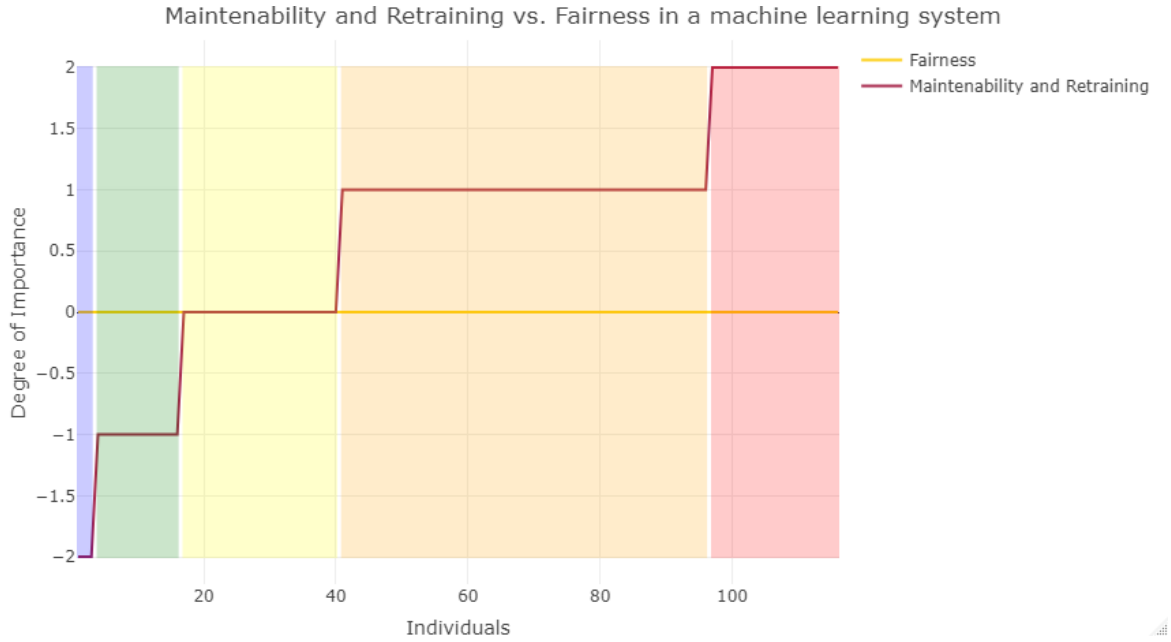

Figura 5.17: Fairness vs Manutenibilità e Retraining

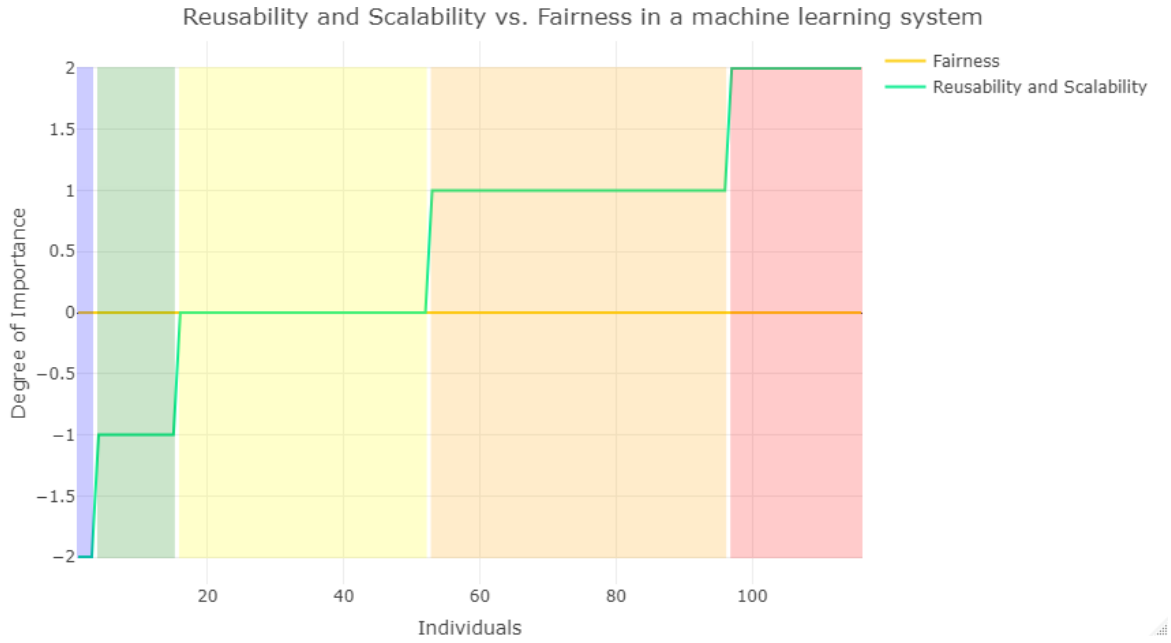

Figura 5.18: Fairness vs Riusabilità e Scalabilità

del software, manager o quant'altro, difficilmente darà più rilevanza in termini assoluti a fairness rispetto ad altre prerogative non funzionali (a maggior ragione per aspetti critici, quali accuracy, performance o sicurezza), e le cause possono essere molteplici, ad esempio:

- Probabilmente la rilevanza del concetto di fairness aumenterà mano a mano che la società percepirà le problematiche connesse come un problema prioritario, quindi tali confronti dovranno essere ripetuti in futuro;

- Fairness probabilmente necessita di essere analizzata in maniera diversa rispetto gli aspetti non funzionali standard, probabilmente in ragione alla specificità del dominio;

Avendo quindi osservato, come gli altri aspetti non funzionali, tendano ad essere considerabili da lievemente più rilevanti a nettamente più rilevanti rispetto la software fairness, considerando l'intero campione di analisi, è possibile provare a formalizzare qualche considerazione ulteriore, analizzando le risposte medie ottenute per singolo settore.

N.B: Il singolo valore di confronto tra fairness ed un altro generico NFR, è stato calcolato come media matematica delle singole risposte dello specifico settore, ad esempio se nella figura 5.19, per il settore Healthcare, è riportato valore 0 della scala quantitativa tra il confronto tra Fairness e Usability, è necessario considerare che *mediamente* per il settore specifico l'aspetto non funzionale Usability è egualmente importante rispetto l'aspetto di Fairness nell'utilizzo/sviluppo di una soluzione ml-intensive.

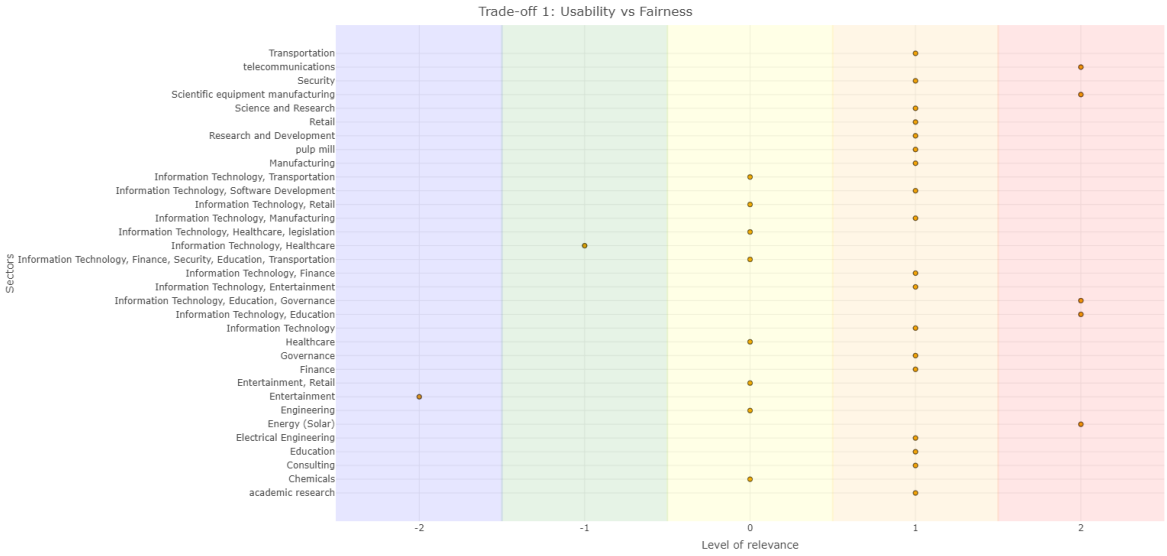

Figura 5.19: Fairness vs Usabilità per settore professionale

L'analisi di dettaglio per settore è disponibile online al link: `https://github.com/CFerrara98/Empirical-Investigation-On-Fairness-Development`. Per porre qualche considerazione ulteriore, sono state riportati: il diagramma inerente il confronto tra Usabilità e Fairness - figura 5.19 (rappresentativo della tendenza di risposta che hanno assunto i confronti specifici tra fairness e aspetti qualitativi più comuni tra i moduli Ml Intensive e altri sistemi IT, e il grafico di confronto per settore tra Sicurezza e Fairness - figura 5.20, rappresentativo della tendenza di confronto assunta per aspetti non funzionali più tecnici in un modulo ML-Intensive.

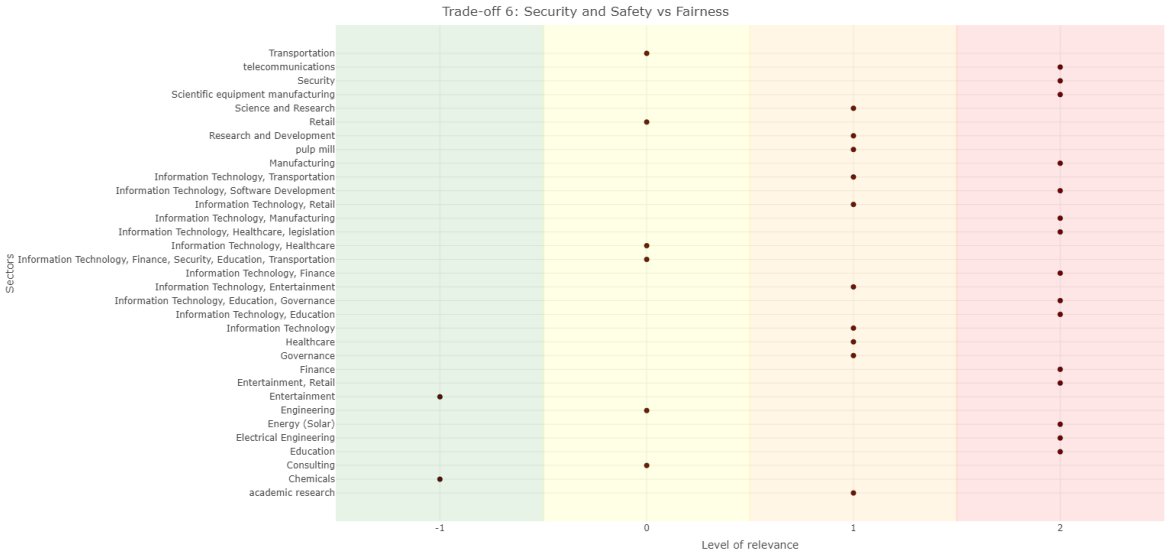

Figura 5.20: Fairness vs Sicurezza per settore professionale

Analizzando quindi i *trade-off* non funzionali raggruppando le risposte per settore, si evidenzia quindi come sia confermata la tendenza generale, ovvero che aspetti non funzionali più standardizzati tendono ad essere considerati dalla maggioranza del campione da lievemente più rilevanti ad estremamente più rilevanti, specialmente quando si parla di requisiti criticamente importanti per il buon funzionamento di un modulo ml-intensive e.g. l'aspetto di Sicurezza. Aumentando la granularità del confronto di rilevanza tra fairness e l'altro NFR considerato a livello settoriale, è però possibile porre qualche considerazione più rilevante, è il caso del confronto tra Sicurezza e Fairness, dove si evidenzia una criticità estremamente rilevante della prima rispetto la seconda per settori in cui lavorare in maniera priva di attacchi esterni è estremamente critico o addirittura vitale, e.g. il settore Governance, mentre lo stesso non può dirsi per settori dove gli aspetti di Sicurezza diventano estremamente meno rilevanti rispetto le discriminazioni dell'individuo, è il caso del settore dell'intrattenimento, dove Fairness è considerata addirittura lievemente più rilevante rispetto gli aspetti qualitativi di sicurezza, quasi in contro tendenza con gli altri settori specifici di analisi.

Ovviamente la tendenza generale dei risultati per questo specifico goal di ricerca resta la stessa rispetto quella osservata a livello generale, ma porre l'accento su esempi di questo tipo, fornisce senz'altro una visione alternativa che fa intendere come fairness possa essere considerato un aspetto prioritario o meno rispetto altri aspetti qualitativi a seconda dello specifico dominio di utilizzo del modulo di machine learning in sviluppo.

5.3.6 Fairness come aspetto intrinseco di una pipeline ML

RQ4 - In quali fasi di una tipica pipeline di Machine Learning è importante adottare strategie per garantire alti livelli di fairness?

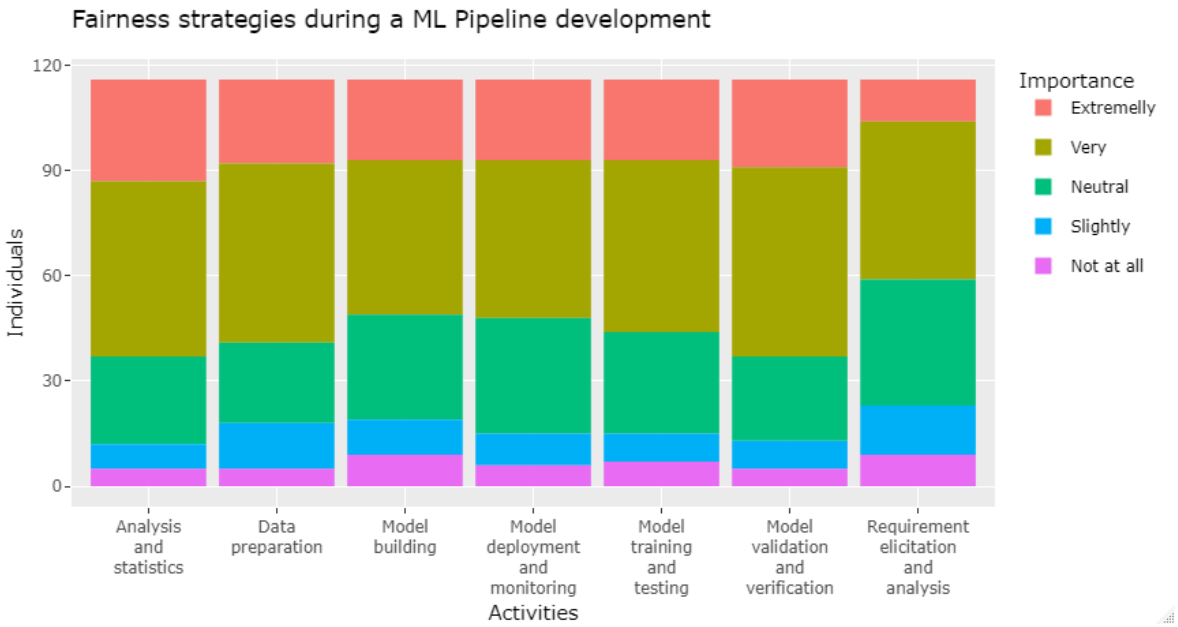

Figura 5.21: Applicabilità di strategie di Fairness improoving in una Pipeline di Machine learining

Come già osservato in fase di progettazione, questo specifico sub goal, mira a valutare l'utilità di adottare strategie di fairness level improoving in ogni fase di una generica pipeline di machine learning. Dai dati e dal report grafico di figura 5.21, è evidenziabile, come la **fairness di un sistema di machine learning è un aspetto di rilevanza in ogni fase di una generica pipeline**, ma da un occhiata più specifica, è facile dedurne come essa stessa può essere un aspetto qualitativo che migliora mano a mano che il sistema evolve sia dopo la fase di building, **dato che la fase di verifica e validazione viene considerata una delle più valide** per attestare il livello di fairness del sistema, **seguita dalla fase di analisi e statistica** (successiva al deploy del modello), e di conseguenza dalla **fase di preparazione dei dati**, che ovviamente va raffinata ad ogni ciclo di sviluppo di una tipica pipeline ML. Ciò significa che la pratica lavorativa, secondo i dati raccolti, probabilmente suggerisce che attualmente la Fairness è un aspetto di un modello di machine learning che va di pari passo con la sua evoluzione, ed è probabilmente lì che è necessario investire con soluzioni specifiche.

Per fornire altri dettagli circa l'utilizzo di approcci e strumenti utili al trattamento di fairness come aspetto integrante del ciclo di sviluppo di un modulo ml, si osserva (figura 5.22) letteralmente pochi partecipanti all'indagine abbiano fatto dichiarato di non fare uso di tool

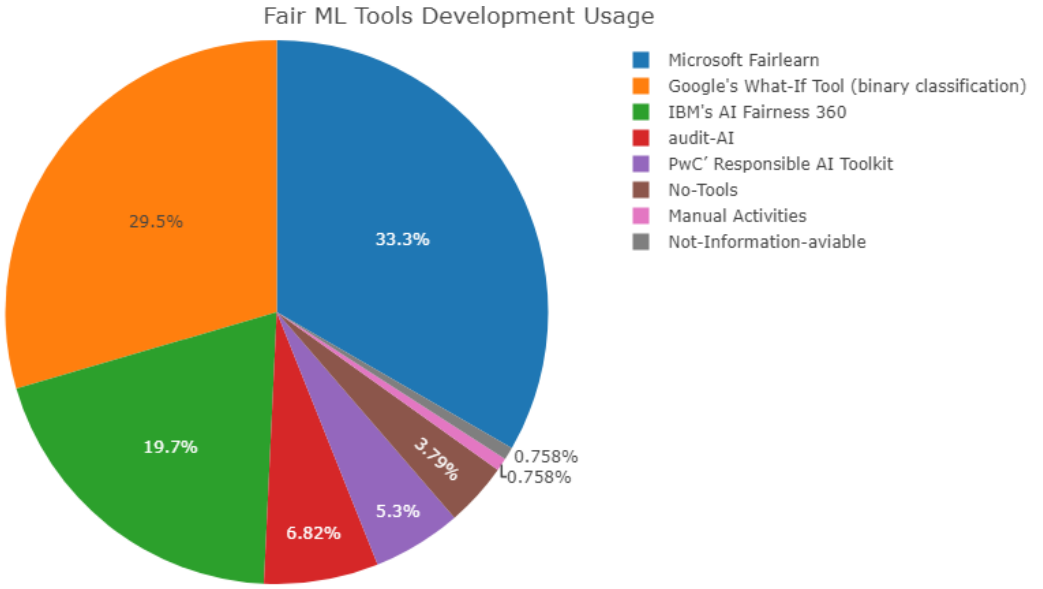

Figura 5.22: Utilizzo di tool noti per il Fair ML Development

specifici per il fairness improoving. la maggior parte di essi invece dichiara di aver utilizzato almeno una volta un tool proprietario tra i noti Microsoft Fairlearn, Google's-What-IF o IBM's AI Fairness 360. Ciò è sicuramente un buon segnale, dato che le aziende evidentemente, sono molto propense ad adottare strategie specifiche per progettare soluzioni ML Fair su larga scala.

5.3.7 Fairness e maturità aziendale

RQ5 - Quanto le compagnie di sviluppo ML-Intensive, sono mature nel trattare il concetto di fairness come un requisito non funzionale?

Ultimo punto conclusivo della panoramica analitica sullo stato della pratica lavorativa, è appunto cercare di capire se e come le aziende dei partecipanti all'indagine definiscono la problematica di fairness durante un generico progetto di sviluppo ML-Intensive e soprattutto quanto sono mature le politiche aziendali a riguardo.

Come risporta la figura 5.23, è facile fornire una risposta preliminare a questo questo di ricerca, il contesto attuale, provando a generalizzare quanto osservabile sulla base del campione, è costituito da circa un **50% delle aziende che praticano sviluppo di soluzioni ML-Intensive**, che tratta fairness ai **livelli 1 e 2**, ovvero, secondo la scala prefissata, trattano le problematiche connesse alla fairness in maniera sporadica o abitudinaria, ma senza l'utilizzo di standard specifici. Interessante anche come dell'altra metà dei partecipanti, soltanto

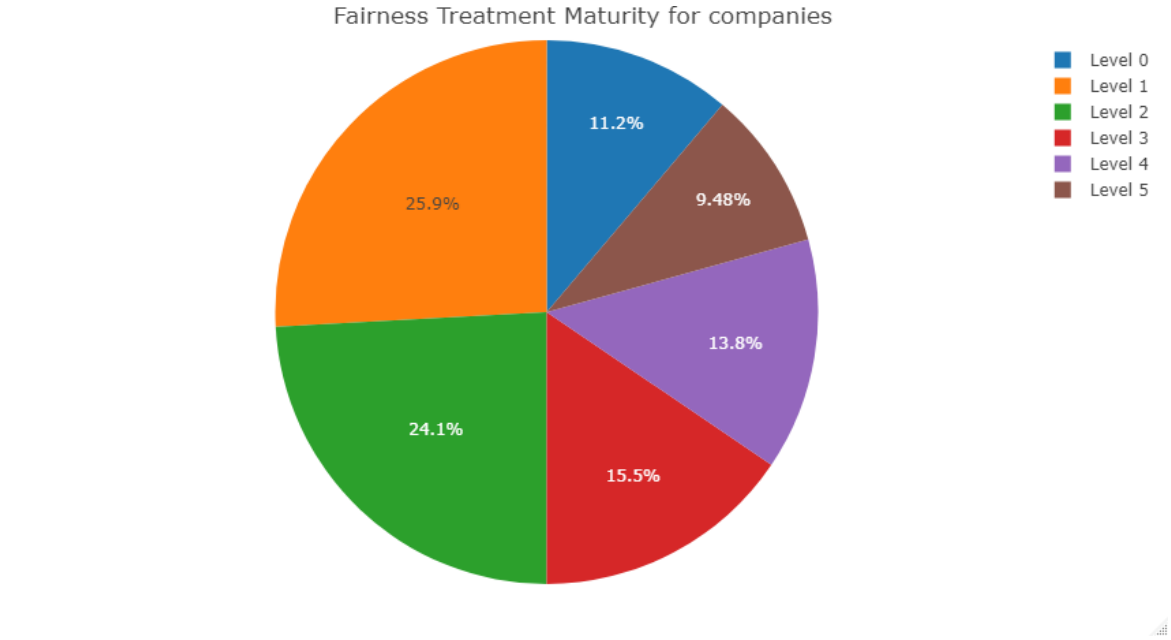

Figura 5.23: Risultati analitici del Fair Capability Maturity Model

l'11,2% dichiari che la sua azienda non tratti affatto fairness, infatti **la fetta restante di partecipanti colloca la propria azienda ai livelli 3, 4 e 5**, quindi dal trattare Fairness in maniera abitudinaria con specifici standard di sviluppo, fino all'applicazione di tecniche di process improoving sugli specifici processi aziendali.

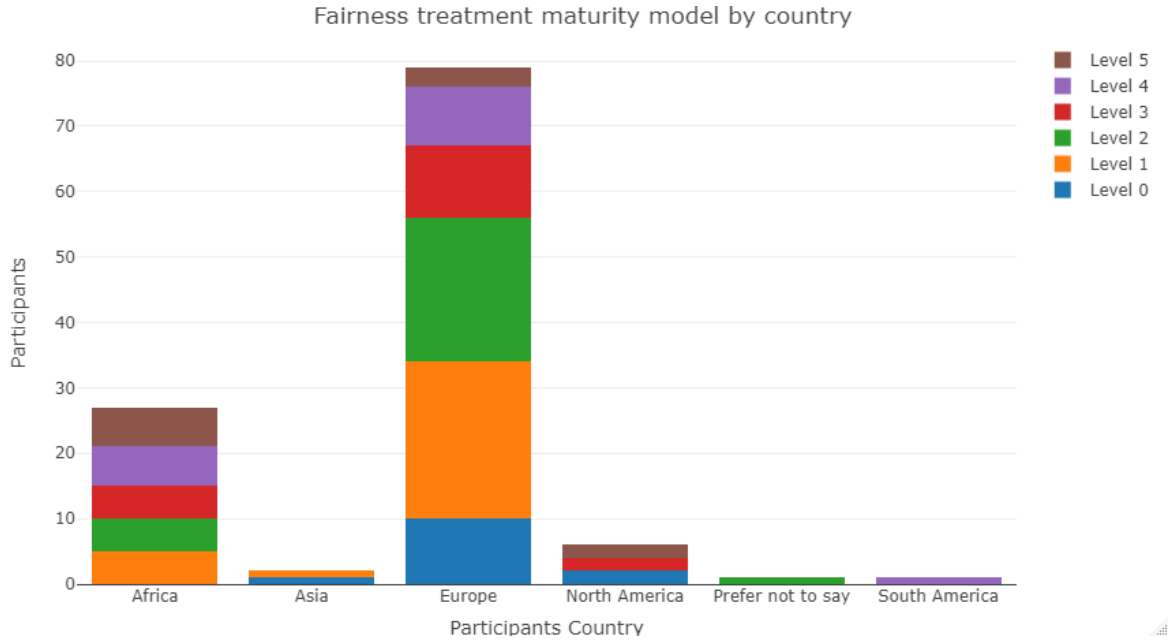

Figura 5.24: Risultati analitici del Fair Capability Maturity Model per continente

Spostando l'analisi su un ottica più mirata, come quella continentale - figura 5.24, si nota ancora meglio come ad esempio in Europa (la maggioranza del campione) le aziende

pratichino processi atti alla gestione della fairness. Ciò è senz'altro un fattore ampiamente positivo, dato che è facilmente osservabile come le aziende, ad esempio su scala europea, siano disposte a migliorare e ad applicare processi ingegneristici nell'ambito ml intensive, di pari passo all'evoluzione metodologica della ricerca e degli investimenti pubblici [34].

5.4 Discussioni e Implicazioni

5.4.1 Quanto fairness è *matura* nello sviluppo ML aziendale?

Partendo dall'analisi dello stato dell'arte, in questo lavoro di tesi si è osservato come effettivamente i ricercatori dell'ambito ingegneristico, evidenzino la necessità di trattare il concetto di software fairness come un vero e proprio aspetto qualitativo *primario* nello sviluppo ML intensive, affinché le elaborazioni dei moduli prodotti e i risultati di predizione degli stessi, siano immuni dal presentare problematiche connesse a vulnerabilità discriminatorie [4]. Sulla base dei dati raccolti dall'indagine, è senz'altro interessante notare, come la maggior parte delle compagnie coinvolte riconosca l'esistenza di questa problematica ed applichi strategie a riguardo, come osservabile infatti dai risultati del quinto quesito di ricerca, solo una minima parte dei partecipanti dichiara che la propria compagnia non tratta affatto Fairness (11,2 %) durante i quotidiani processi di sviluppo, mentre, dal lato opposto, oltre il 30 % dei partecipanti dichiara di applicare Fairness con i dovuti standard propri dell'azienda di appartenenza fino all'applicare strategie specifiche di process improoving nel trattamento di fairness. A conferma di ciò, l'analisi dei dati permette di osservare come tool commerciali per il trattamento di Fairness in un modulo di machine learning esistano e risultano essere estremamente utili nello sviluppo di soluzioni eticamente corrette, in particolare i tools più quotati dagli intervistati sono Microsoft Fairnlearn, che approccia lo sviluppo fair-critical con casi d'uso reali, e Google What If? che permette proprio di analizzare in maniera qualitativa e quantitativa i livelli di fairness di uno specifico modulo.

5.4.2 Quanto fairness è *immatura* nello sviluppo ML aziendale?

Concentrandosi invece sui dati rimanenti del quinto punto di ricerca analizzato, va osservato come il 50 % del campione di analisi, dichiara di trovarsi ai livelli 1 e 2 della scala di maturità formalizzata, quindi è opportuno chiedersi se effettivamente fairness è vista o meno a livello pratico come un aspetto prioritario rispetto altre specifiche non funzionali.

Dall'analisi generale dei confronti tra fairness con altri aspetti di qualità in un modulo ML-Intensive si è visto effettivamente come le opinioni raccolte tendano ad essere in ogni caso tendenti al considerare Fairness da lievemente meno rilevante, come per i confronti con aspetti qualitativi appartenenti al modello FURPS+, ad estremamente meno rilevante, come nel caso di specifiche estremamente più tecniche e standardizzate quali Security o Performance del modello. Vero che dall'analisi settoriale è riscontrabile come questa tendenza sia estremamente differente a seconda del dominio specifico, e.g. la differente rilevanza media di Fairness rispetto a Security nei settori di Governance ed Intrattenimento, ma dai dati a disposizione è senz'altro evidente come l'aspetto etico sia ancora poco maturo e standardizzato per essere considerato una vera e propria specifica non funzionale in un modulo di machine learning al specifiche *storiche* più documentate e standardizzate. Molto probabilmente a tal proposito la chiave di svolta è racchiusa nel supporto che la comunità scientifica può apportare allo stato della pratica, senz'altro incentrando i nuovi studi sulle necessità e percezioni che l'attuale stato della pratica evidenzia, ma anche verificando in maniera sistematica, se quanto già stato fatto fin ora possa essere applicato attivamente in contesti di sviluppo reali.

5.4.3 Definire e misurare fairness in un caso reale di sviluppo ML

Come osservato dalla letteratura, molti sono le definizioni e le metriche formalizzate al fine di trattare fairness durante lo sviluppo ML-Intensive [5]. Dai risultati del primo quesito di ricerca, si può notare come gli esperti tendano ad apprezzare in larga misura tutti i gruppi di definizioni proposte, però se si cerca qualche dettaglio in più, è facile osservare come approcci formali basati su similarità matematica, siano leggermente preferiti rispetto ai più formali aspetti basati su puro calcolo probabilistico o su relazioni di dipendenze causali. D'altro canto però, osservando i risultati inerenti il quesito inerente l'applicazione di approcci pratici nel trattamento di fairness, gli stessi intervistati dichiarano di trovarsi più a proprio agio a ragionare intuitivamente sulle dipendenze causali tra feature sensibili e risultati, piuttosto che su aspetti più tecnici, quali possibili attività di tuning dei parametri fair-oriented. A prima vista ciò potrebbe essere quasi un contro senso, ma effettivamente quanto le attuali definizioni basate su dipendenze causali, sono intuibili dai lavoratori se considerate nella loro forma più formale? Quanto i differenti gruppi di definizioni attualmente conosciuti sono applicabili in maniera generale tenendo conto delle svariate esigenze dei domini d'utilizzo? Intersecare ciò che i risultati di analisi ottenuti, si evince anche praticamente l'applicabilità dell'una o l'altra definizione di fairness, oppure l'utilizzo di una o più approcci pratici, dipenda in se

per se dal dominio di utilizzo, ciò non solo per l'enorme variabilità di rilevanza riscontrata analizzando a livello settoriale i dati relativi al terzo quesito di ricerca, ma anche perché molti partecipanti facciano presente in maniera esplicita, che oltre agli approcci noti in letteratura, essi stessi facciano uso di tecniche diverse da quelle proposte, ad esempio l'applicazione di strategie empiriche o di indicatori statistici come osservato dall'analisi dei dati. Quindi dove la ricerca può intervenire in tal senso? In definitiva non è probabile che l'una o l'altra strategia sia univocamente condivisibile o applicabile data l'estrema specificità della tematica, ma da un analisi più specifica degli standard di sviluppo effettivamente sarà possibile individuare quali processi formali possano essere o meno applicabili a seconda delle più svariate esigenze di dominio.

5.4.4 Processi di sviluppo ML Fair-Oriented, una visione a lungo termine

Constatato che fairness secondo l'attuale stato della pratica continua ad essere un aspetto estremamente dipendete dal dominio applicativo, e che gli studi a riguardo necessitino ancora di essere espansi tenendo conto in primo luogo dei bisogni pratici evidenziati dai professionisti, resta doveroso cercare di capire dove effettivamente la ricerca possa intervenire affinché il trattamento di fairness possa essere sistematizzato al pari di altre specifiche qualitative di un modulo ML-Intensive. È oramai noto alla comunità scientifica, ma anche allo stato della pratica che approcci di sviluppo evolutivo quali il noto standard MLOps, siano una strategia vincente nel trattare e migliorare le specifiche di qualità di un modulo intelligente, soprattutto nel contesto di MLOps, sono nate strategie specifiche che consentono di monitorare i livelli di un particolare aspetto non funzionale, ma ciò è possibile anche nel caso dell'etica? dall'analisi dei dati ricevuti in risposta ai quesiti inerenti le fasi di una canonica pipeline di machine learning e delle figure di rilievo in un team orientato allo sviluppo fair-oriented probabilmente ciò è assolutamente un buon punto su cui riflettere per futuri lavori di ricerca.

Partendo proprio dall'analisi delle fasi di una canonica pipeline di machine learning, si è visto come gli esperti del settore propongano come più critico il trattamento di fairness nelle fasi di gestione e manipolazione dei dati di training, validazione del modello oppure nelle statistiche di monitoraggio del modello in esecuzione nel contesto d'utilizzo. Proprio in questo contesto la ricerca offre già numerose soluzioni, basti pensare alle tecniche specifiche di selezione dei dati basate su diversità statistica [26] oppure alle specifiche tecniche di testing fair-orinted e ribilanciamento del campione [28]. Ma quanto queste strategie specifiche, sono

conosciute ed applicate dai professionisti? Quanto invece nuove strategie come nuovi modelli di validazione fair-specific sono necessarie per sistematizzare questi processi? tutto questo, quasi sicuramente necessita di ulteriori approfondimenti, e nuovi lavori quali l'analisi di bad & best practices specifiche per il trattamento di fairness in un modulo ML-Intensive o la ricerca di maggiori cause di discriminazione partendo ad esempio dall'analisi delle principali feature sensibili comuni alla maggior parte dei datasets di addestramento, possono essere senz'altro un buon punto di partenza. In connessione all'applicabilità di strategie, in un qualsiasi processo di sviluppo che si rispetti, va senz'altro definito chi debba assumere l'una o l'altra responsabilità. Dai dati ottenuti si evidenziano due principali osservazioni:

- Ingegneri del software e data scientist restano a pari merito figure cruciali e di simili rilevanza anche nel trattamento di specifiche problematiche quali i livelli di fairness nei moduli;
- La definizione di linee guida di management o il coinvolgimento di esperti specifici per il trattamento di software fairness sono aspetti che maggiormente fanno la differenza.

Ma da queste osservazioni possono anche riscontrarsi nuovi quesiti da porsi ed analizzare, ad esempio:

- Quando risulta essere cruciale che ruoli diversi quali data scientist e ingegneri del software lavorino in sinergia se si considera una pipeline di machine learning?
- Quali possono essere i punti di partenza per la definizione di standard di management specifici?
- Cosa si intende effettivamente per esperti nell'ambito Fairness? Esperti di socio-culturali di etica? Esperti del dominio specifico o quant'altro?

Oltre ciò, anche altri dati possono essere di ispirazione per nuovi studi, non ultimo la minore rilevanza data dai partecipanti ad altri aspetti critici di una pipeline di machine learning come l'ingegnerizzazione dei requisiti o la preparazione del modello. In generale il principale contributo che si evince da questo lavoro di tesi sta proprio nella consapevolezza che l'etica di un modulo di machine learning necessita di maturare ed evolvere in maniera sistematica proprio perché le cause e i fattori che impattano tale tematica oggigiorno sono sempre più evidenti e dai risultati ottenuti è senz'altro intuibile dove sia più utile intervenire in funzione di questi obiettivi.

CAPITOLO 6

Conclusioni

In questo lavoro di tesi, si è osservato come la progettazione e lo sviluppo aziendale di un canonico modulo di machine learning, non possa più prescindere dall'analizzare problematiche di carattere etico. Si è osservato inizialmente, tramite un'attenta analisi dello stato dell'arte, che l'aspetto etico, meglio conosciuto con la sua traduzione fairness, di un modulo di machine learning è notoriamente considerato in letteratura come un aspetto variegato e difficile da generalizzare, e per questo la comunità scientifica, sia nell'ambito dell'intelligenza artificiale, che dell'ingegneria del software, ha orientato gli studi di ricerca al fine di progettare e testare soluzioni che possano essere di supporto a chi sviluppa soluzioni intelligenti di trattare l'etica di un modulo ML-Intensive in modo più accurato e concreto. Volendo quindi verificare se e come i progressi della ricerca fossero percepiti allo stato della pratica aziendale, si è deciso di condurre un'indagine empirica, a mezzo di un Survey su larga scala, che appunto coinvolgesse gli esperti del settore per rispondere in maniera puntuale e il più completa possibile ad un principale obiettivo di ricerca, per l'appunto: *In che modo il concetto di Software Fairness è attualmente percepito nell'ambito lavorativo ML-Intensive?*. Specializzando quindi l'obiettivo principale in 5 obiettivi specifici che riguardassero appunto:

- Definizioni ed approcci pratici per il trattamento della fairness;
- Composizione ottimale di un canonico team di sviluppo ML-Intensive Fair Critical;
- Analisi di rilevanza dei livelli di Fairness rispetto altre specifiche non funzionali;

- Applicabilità di strategie specifiche per il trattamento di Fairness in una canonica pipeline ML;
- Livello di maturità aziendale nel trattamento della fairness;

si è passati quindi alla disseminazione su larga scala, che in pochi giorni ha prodotto 203 risposte, poi analizzate e ridotte a 116 prima (per controlli ed operazioni di data cleaning) di passare poi all'analisi dei dati ed alla conseguente generalizzazione dei risultati.

Dai dati osservati si è principalmente osservato che allo stato attuale della pratica, i professionisti concordano principalmente come il concetto di moduli di machine learning eticamente corretti, sia altamente dipendenti dal dominio di applicazione, e che l'una o l'altra definizione dipenda effettivamente dal contesto applicativo. In particolare poi, dai dati a disposizione, si evince come Fairness sia un concetto che necessita di ulteriori studi, prima di essere considerato a tutti gli effetti un aspetto di qualità maturo al pari di altre specifiche non funzionali. In particolare l'analisi dei dati evidenzia come:

- L'applicazione di strategie o approcci vada effettivamente sistematizzata ed a seconda del dominio d'utilizzo, ma contestualmente sia necessario comprendere bene la rilevanza di ciascun approccio nel contesto d'uso per sfruttarne a pieno le potenzialità;
- Il trattamento della fairness nello sviluppo ML-Intensive necessiti effettivamente di un intensiva collaborazione tra Data Scientists ed Ingegneri del Software, e capire in che fasi di una pipeline siano critiche queste figure è attualmente ancora un punto da sistematizzare;
- Durante lo sviluppo sia critico coinvolgere manager di progetto o *esperti* nel trattamento di fairness per garantire che le specifiche etiche siano rispettate, anche in questo caso sarà necessario approfondire bene le responsabilità o le competenze delle due figure;
- Fairness necessiti di strategie specifiche durante l'intero ciclo di vita di un modulo ML-Intensive, e se si prende come riferimento gli specifici moduli di machine learning, con particolare riferimento alle fasi evolutive del modello ovvero preparazione dei dati e delle feature, analisi e validazione del modello e reporting di statistica con il modello in esecuzione.

Dall'analisi ad ampio raggio condotta, quello che si evince quindi che il concetto di Fairness in un modulo ML-intensive, è un qualcosa da sistematizzare e approfondire con

studi mirati che tengano conto effettivamente di quanto già sia stato fatto e di cosa è necessario osservando con occhio critico le pratiche aziendali. In particolare si osserva come possano essere particolarmente critici studi che:

- Analizzino nello specifico quali siano le principali cause di discriminazioni o le opportune pratiche da adottare o meno nello sviluppo sviluppo Pipeline oriented, in modo tale da definire dei veri e propri protocolli di sviluppo fair-specific come già si fa con altre specifiche non funzionali;
- Guidino l'apprendimento delle nozioni basilari di fairness, in ambito accademico o aziendale, tenendo conto della stretta correlazione con il dominio applicativo, magari applicando definizioni o approcci a casi di studio noti, ed analizzandone vantaggi e svantaggi;
- Aiutino, di conseguenza, i professionisti nel discriminare a seconda dei casi quali approcci adoperare, in questo caso potrebbe essere anche utili approfondire l'applicabilità di altri metodi emersi dalle risposte ottenute come l'analisi condotta tramite metodologie empiriche o l'utilizzo di indicatori statistici di correlazione in fase di preparazione dei dati;
- Definiscano protocolli gestionali e di composizione del team sistematici che guidino lo sviluppo ML-Intensive Fair-Critical, definendo per l'appunto ruoli e responsabilità specifiche in ogni fase di sviluppo.

Va infine osservato, come l'attuale studio ribadisca che quanto stato già fatto sia effettivamente utile al fine degli obiettivi di lungo raggio che la ricerca si pone. Infatti, è facilmente osservabile che strategie specifiche analizzate a priori dell'investigazione empirica, abbiano fatto si che le aziende incrementassero notevolmente il loro grado di consapevolezza in merito il fattore etico delle soluzioni progettate. Quello che effettivamente si cerca di suggerire è che tramite il coinvolgimento di esperti professionali, oltre la progettazione di nuove strategie mirate, sarà anche possibile sistematizzare e applicare concretamente quanto già è stato proposto dalla ricerca, con il fine ultimo di rendere i sistemi di machine learning ampiamente robusti rispetto qualsiasi tipo di problematica di carattere etico/discriminatorio.

Ringraziamenti

In poche righe sicuramente non riuscirei a ringraziare in modo adeguato tutte le persone che hanno caratterizzato questi miei 5 anni di percorso, quindi mi limito a farlo a modo mio, con una semplice ma sincera riflessione: «Il mio cammino universitario è stato come una lunga lunga gita in un'ampia foresta, in questa gita sono tante le persone che mi hanno accompagnato tra momenti difficili ed esperienze fantastiche! Ora mi trovo esattamente qui, alla fine della foresta ad osservare un orizzonte immenso che ancora non conosco, spero davvero che chi continuerà questo cammino con me o lontano da me, possa avere il meglio che questa fantastica veduta ha da offrire».

Un sincero grazie a tutti coloro che mi hanno accompagnato fin ora!

Questo lavoro di tesi è dedicato a tutti voi!

Bibliografia

[1] J. Rech and K.-D. Althoff, "Artificial intelligence and software engineering: Status and future trends," *KI*, vol. 18, no. 3, pp. 5–11, 2004. (Citato alle pagine 1, 11 e 12)

[2] A. Burkov, *Machine learning engineering*, vol. 1. True Positive Incorporated, 2020. (Citato alle pagine 1, 14 e 15)

[3] J. Horkoff, "Non-functional requirements for machine learning: Challenges and new directions," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 386–391, 2019. (Citato alle pagine 2, 12, 13 e 18)

[4] Y. Brun and A. Meliou, "Software fairness," in *Proceedings of the 2018 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, pp. 754–759, 2018. (Citato alle pagine 2, 5, 6, 19, 26, 30, 33, 38, 74 e 84)

[5] S. Verma and J. Rubin, "Fairness definitions explained," in *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pp. 1–7, 2018. (Citato alle pagine 2, 20, 21, 22, 23, 24, 25, 29 e 85)

[6] B. Bruegge and A. Dutoit, "Object-oriented software engineering using uml, patterns, and java," 2009. (Citato alle pagine 5, 6 e 7)

[7] R. Mall, *Fundamentals of software engineering*. PHI Learning Pvt. Ltd., 2018. (Citato a pagina 5)

[8] P. Tripathy and K. Naik, *Software evolution and maintenance: a practitioner's approach*. John Wiley & Sons, 2014. (Citato a pagina 7)

[9] I. Sommerville, *Software Engineering*. Harlow, England: Addison-Wesley, 9 ed., 2010. (Citato a pagina 8)

[10] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016. (Citato a pagina 8)

[11] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128–138, 2017. (Citato a pagina 9)

[12] M. T. Islam, A. Fariha, and A. Meliou, "Through the data management lens: Experimental analysis and evaluation of fair classification," *arXiv preprint arXiv:2101.07361*, 2021. (Citato alle pagine 9, 18 e 23)

[13] L. Haldurai, T. Madhubala, and R. Rajalakshmi, "A study on genetic algorithm and its applications," *International journal of computer sciences and Engineering*, vol. 4, no. 10, p. 139, 2016. (Citato a pagina 10)

[14] J. Shaw, "Artificial intelligence and ethics," *Harvard Magazine*, vol. 30, 2019. (Citato alle pagine 11 e 12)

[15] F. Thung, S. Wang, D. Lo, and L. Jiang, "An empirical study of bugs in machine learning systems," in *2012 IEEE 23rd International Symposium on Software Reliability Engineering*, pp. 271–280, 2012. (Citato a pagina 11)

[16] P. Jain, "Interaction between software engineering and artificial intelligence-a review," *International Journal on Computer Science and Engineering*, vol. 3, no. 12, p. 3774, 2011. (Citato a pagina 12)

[17] H. Belani, M. Vukovic, and Car, "Requirements engineering challenges in building ai-based complex systems," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pp. 252–255, 2019. (Citato a pagina 13)

[18] Y. Zhou, Y. Yu, and B. Ding, "Towards mlops: A case study of ml pipeline platform," in *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 494–500, IEEE, 2020. (Citato alle pagine 15, 16 e 39)

[19] S. Alla and S. K. Adari, "Beginning mlops with mlflow: Deploy models in aws sagemaker, google cloud, and microsoft azure," *Beginning MLOps with MLFlow*, 2021. (Citato alle pagine 16 e 17)

[20] J. Dastin, "Amazon scraps secret ai recruiting tool that showed bias against women." `http://www.shorturl.at/agvCO`, 2018. (Citato a pagina 19)

[21] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," *Science*, vol. 366, no. 6464, pp. 447–453, 2019. (Citato a pagina 19)

[22] J. Angwin and J. Larson, "Machine bias - there's software used across the country to predict future criminals. and it's biased against blacks.." `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`, 2016. (Citato a pagina 20)

[23] R. Tatman, "Gender and dialect bias in YouTube's automatic captions," in *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, (Valencia, Spain), pp. 53–59, Association for Computational Linguistics, Apr. 2017. (Citato a pagina 20)

[24] J. Chakraborty, S. Majumder, and T. Menzies, "Bias in machine learning software: why? how? what to do?," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 429–440, 2021. (Citato alle pagine 20, 26 e 28)

[25] J. M. Zhang and M. Harman, ""ignorance and prejudice" in software fairness," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 1436–1447, 2021. (Citato alle pagine 26, 27 e 29)

[26] Z. Moumoulidou, A. McGregor, and A. Meliou, "Diverse data selection under fairness constraints," *arXiv preprint arXiv:2010.09141*, 2020. (Citato alle pagine 26, 30 e 86)

[27] A. Finkelstein, M. Harman, S. A. Mansouri, J. Ren, and Y. Zhang, ""fairness analysis" in requirements assignments," in *2008 16th IEEE International Requirements Engineering Conference*, pp. 115–124, IEEE, 2008. (Citato alle pagine 27 e 32)

[28] S. Galhotra, Y. Brun, and A. Meliou, "Fairness testing: testing software for discrimination," in *Proceedings of the 2017 11th Joint meeting on foundations of software engineering*, pp. 498–510, 2017. (Citato alle pagine 27, 33 e 86)

[29] S. Vasudevan and K. Kenthapadi, "Lift: A scalable framework for measuring fairness in ml applications," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2773–2780, 2020. (Citato a pagina 27)

[30] P. P. Kuver, "Sei cmm or iso 9000: Which is right for your organization?," in *New Directions in Project Management*, pp. 131–138, Auerbach Publications, 2001. (Citato alle pagine 39 e 44)

[31] D. Andrews, B. Nonnecke, and J. Preece, "Conducting research on the internet:: Online survey design, development and implementation guidelines," 2007. (Citato alle pagine 41 e 48)

[32] R. Sumbaly, J. Kreps, and S. Shah, "The big data ecosystem at linkedin," in *Proceedings of the 2013 acm sigmod international conference on management of data*, pp. 1125–1134, 2013. (Citato a pagina 50)

[33] B. Reid, M. Wagner, M. d'Amorim, and C. Treude, "Software engineering user study recruitment on prolific: An experience report," *arXiv preprint arXiv:2201.05348*, 2022. (Citato a pagina 51)

[34] C. Ritson, "17 reflections on food ethics," *Practical Ethics for Food Professionals: Ethics in Research, Education and the Workplace*, vol. 52, 2013. (Citato a pagina 84)