UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN INFORMATICA

TESI DI LAUREA
IN
METODI DI RITROVAMENTO DELL'INFORMAZIONE

ANALISI DEL TRADE-OFF TRA FAIRNESS, ACCURATEZZA E SOSTENIBILITA' AMBIENTALE NEI RECOMMENDER SYSTEMS

RELATORE:
Prof. Cataldo Musto

LAUREANDO:
Andrea Franco

CORRELATORE:
Dott. Giuseppe Spillo

ANNO ACCADEMICO 2024 - 2025

Indice

Disclaimer

Tutti i marchi, nomi commerciali, prodotti e loghi menzionati in questa tesi sono di proprietà dei rispettivi titolari. L'autore non rivendica alcun diritto di proprietà su tali marchi o nomi commerciali e li utilizza solo a scopo informativo e descrittivo, senza alcun intento di violazione.

Abstract

Il concetto di fairness è diventato centrale nel dibattito sul machine learning, soprattutto da quando questi sistemi hanno iniziato a influenzare attivamente decisioni che, in passato, spettavano esclusivamente all'essere umano. Con l'adozione sempre più ampia di algoritmi in ambiti sensibili come giustizia, istruzione, lavoro e sanità, è diventato fondamentale interrogarsi su quanto tali sistemi siano davvero equi, trasparenti e affidabili. Oggi gli algoritmi, alimentati da enormi quantità di dati, supportano ogni tipo di scelta: dalla ricerca scientifica alla vita quotidiana, fino a modellare gusti e preferenze personali. Ma questa automazione crescente porta con sé importanti implicazioni etiche e sociali. Diversi studi hanno dimostrato come i sistemi algoritmici possano riprodurre o addirittura accentuare i pregiudizi della società. Il caso del sistema COMPAS, utilizzato negli Stati Uniti per stimare il rischio di recidiva degli imputati, è emblematico: nel 2016 fu evidenziato che classificava gli imputati afroamericani come ad alto rischio quasi il doppio delle volte rispetto ai bianchi, pur in assenza di nuovi reati. Episodi come questo mostrano chiaramente quanto l'uso acritico degli algoritmi possa avere conseguenze gravi, fino a compromettere la libertà individuale. Ma quanto siamo disposti a pagare affinché l'equità diventi una priorità nel machine learning? Questa domanda ci porta oltre l'etica, verso il tema più ampio della sostenibilità. Garantire l'equità algoritmica richiede spesso modelli più complessi, addestramenti più lunghi, metriche aggiuntive e una maggiore potenza di calcolo. Tutto questo ha un impatto diretto sul consumo energetico e sulle emissioni di CO_2 associate. In un contesto globale in cui il cambiamento climatico è una delle principali sfide, l'intelligenza artificiale deve essere valutata anche in termini di sostenibilità. Secondo il paradigma dei tre pilastri, ambientale, economico e sociale, in questa tesi ci concentriamo in particolare su quello sociale, ovvero l'equità nei sistemi di raccomandazione, e su quello

ambientale, legato all'impatto energetico delle soluzioni fairness-aware. A partire da queste considerazioni, la ricerca presentata in questa tesi analizza la relazione tra fairness e sostenibilità ambientale nei sistemi di raccomandazione. Attraverso l'uso del framework RecBole-FairRec e dello strumento CodeCarbon, abbiamo confrontato diversi modelli fairness-aware, misurando sia metriche di equità , di accuratezza e le emissioni di CO_2 generate in fase di addestramento. L'obiettivo della ricerca svolta è capire se, e a quale costo ambientale, si possa sviluppare un Recommender System che sia al tempo stesso equo e sostenibile.

Capitolo 1

Introduzione

1.1 Fairness: Definizioni e Prospettive

1.1.1 Definizioni di Fairness

Nello studio generale della Fairness, è ampiamente riconosciuto che non esista un'univoca definizione del termine, poiché il concetto è strettamente legato ai diversi scenari in cui applicarla [27]. È stato dimostrato teoreticamente che non è possibile soddisfare tutte le definizioni di equità contemporaneamente. In generale, le definizioni di equità possono essere suddivise in tre macro-categorie.
Fairness di gruppo richiede che i gruppi protetti ricevano un trattamento simile rispetto ai gruppi non protetti nei sistemi di Machine Learning [27].
Fairness individuale richiede che individui simili siano trattati in modo simile, valutando la somiglianza tramite caratteristiche sensibili o rappresentazioni latenti. È ortogonale alla Fairness di gruppo, poiché un metodo equo a livello individuale potrebbe non esserlo a livello di gruppo e viceversa [27].
Fairness ibrida mira a soddisfare più requisiti di equità contemporaneamente, riconoscendo che le esigenze cambiano in base al target e al contesto. Si evidenzia come alcune caratteristiche non possano essere soddisfatte simultaneamente[27].

1.1.2 Fondamenti teorici sulla giustizia (cenni)

Il concetto di giustizia, strettamente legato all'equità nel Machine Learning, varia a seconda del contesto applicativo. Sebbene spesso usati come sinonimi, giustizia ed

equità differiscono[28]: la prima riguarda l'aderenza a norme, la seconda la percezione di tale aderenza. Tre teorie principali emergono: l'Utilitarismo, massimizzazione del benessere collettivo, il Contrattualismo,accordi tra individui e l'Egualitarismo, distribuzione equa. Nel Machine Learning prevale l'approccio egualitario, con la sfida di definire criteri di similarità per individui/gruppi e determinare il trattamento equo appropriato al preciso contesto. Spesso nello studio dell'equità [27], si identificano quattro distinzioni fondamentali: Giustizia Conservatrice contro l'Ideale, dove la prima rispetta le norme esistenti mentre la seconda punta a una riforma radicale, con il Machine Learning spesso orientato all'approccio ideale nonostante i limiti pratici; Giustizia Correttiva contro la Distributiva, dove la prima riguarda la riparazione dei danni e la seconda l'allocazione delle risorse, quest'ultima predominante nei Recommender System; Giustizia Procedurale contro Sostanziale, mettendo a confronto i processi decisionali con i risultati, dove metriche come le Pari Opportunità adottano la seconda prospettiva; infine Giustizia Comparata contro la Non Comparata, che oppone valutazioni relative tra individui a valutazioni basate su caratteristiche intrinseche, entrambe presenti nel Machine Learning. L'applicazione della giustizia nel ML si concentra prevalentemente sugli esseri umani, con dibattiti tra visione relazionale e universale. I Recommender System devono bilanciare equità e prestazioni, considerando il loro impatto sociale crescente. La sfida principale è implementare principi di giustizia in algoritmi che influenzano risorse reali, mantenendo responsabilità verso utenti e società.

1.2 Fairness nella Classificazione

Approfondiamo ora il contesto della classificazione, fondamentale nel Machine Learning e ben sviluppato nella ricerca sulla Fairness. In questo contesto, le definizioni più comuni di equità si dividono in due principali categorie: fairness di gruppo e fairness individuale. Queste categorie rappresentano due diverse interpretazioni del concetto di trattamento equo e spesso portano a obiettivi contrastanti.

1.2.1 Fairness di gruppo

La fairness di gruppo richiede che le predizioni di un modello di ML siano distribuite in modo equo tra diversi gruppi protetti, definiti in base ad attributi sensibili

caratterizzanti.L'idea alla base è che i membri di un determinato gruppo protetto debbano avere il medesimo trattamento rispetto ai membri di un gruppo privilegiato.E' stato dimostrato nel tempo che è matematicamente impossibile soddisfare diversi criteri di fairness contemporaneamente, salvo in presenza di particolari condizioni statistiche.

1.2.2 Fairness individuale

La fairness individuale si basa sul principio che individui simili dovrebbero essere trattati in modo simile. Questo concetto si ispira all'uguaglianza di trattamento su base individuale piuttosto che su media di gruppo. Una delle definizioni più influenti è quella di Fairness Through Awareness [12], secondo cui è necessario definire una metrica di similarità tra individui, e garantire che due individui simili ricevano predizioni simili.Un'altra definizione importante è la Fairness controfattuale (Counterfactual Fairness) [24], che si basa su modelli causali: un predittore è considerato controfattualmente equo se la predizione per un individuo rimane invariata in un mondo ipotetico in cui il suo attributo sensibile, fosse stato diverso, mantenendo tutto il resto invariato. La fairness individuale è particolarmente importante in contesti dove la decisione ha un impatto significativo a livello personale, come nell'assunzione di personale, concessione di prestiti o ammissione a un'università. Tuttavia, applicarla richiede una conoscenza approfondita delle caratteristiche individuali e delle relazioni causali tra le variabili.

1.3 Fairness nei Ranking

Nei sistemi di ranking, la fairness è spesso studiata lato item, ovvero considerando come vengono trattati i contenuti, i candidati o i prodotti che appaiono nei risultati ordinati. Questo approccio è centrale nel paradigma del learning to rank, in cui l'obiettivo è ordinare gli elementi in base alla loro rilevanza stimata per un utente. Tuttavia, la distribuzione imparziale della visibilità e dell'accesso alle risorse può essere compromessa da bias strutturali come il position bias, che favorisce sistematicamente le posizioni più alte nella lista. Due delle principali definizioni operative di fairness in questo contesto sono la fairness probabilistica e la fairness basata sull'esposizione.

1.3.1 Fairness probabilistica

La fairness probabilistica, o statistical parity, impone vincoli sulla rappresentanza dei gruppi protetti nelle prime posizioni di una classifica. L'idea centrale è che la probabilità di comparire nella top-K debba essere proporzionale, o almeno non inferiore, alla quota di quel gruppo nella popolazione complessiva. Questo approccio si presta a scenari in cui l'obiettivo è garantire inclusione e diversità nei risultati, mantenendo comunque la qualità del ranking.Un esempio significativo è l'algoritmo FA*IR, che costruisce ranking top-K soggetti a vincoli di rappresentanza, cercando di preservare l'ordine originale quanto più possibile [31]. Questo tipo di tecnica si adatta bene a contesti in cui si desidera correggere squilibri strutturali o migliorare la visibilità di contenuti minori [27].

1.3.2 Fairness basata sull'esposizione

A differenza dell'approccio precedente, la fairness basata sull'esposizione considera esplicitamente il position bias, ovvero il fatto che gli item nelle prime posizioni di una classifica ricevono molta più attenzione da parte degli utenti. In questo caso, l'equità non si misura soltanto con la presenza nella top-K, ma con la quantità di esposizione ricevuta, che deve essere proporzionale alla rilevanza dell'item o del gruppo.In particolare, Singh e Joachims propongono un framework in cui l'esposizione assegnata a ciascun gruppo dovrebbe essere proporzionale alla sua utilità media. La disparità tra gruppi può essere formalizzata come differenza assoluta tra le esposizioni medie normalizzate. Questa visione è particolarmente adatta in contesti dinamici, dove l'esposizione ha un impatto cumulativo (ad esempio, sulla popolarità o il successo commerciale di un contenuto) [27, 31].

Problemi di Fairness

Nel seguente segmento si introduce la ricerca sulla Fairness nelle attività di classificazione binaria [27]. In un'attività di classificazione binaria si dispone di dati di addestramento $D_T = (x_i, y_i)_{i=1}^N$, dove $x \in \mathbb{R}^d$ sono i vettori di caratteristiche e $y \in \{-1,1\}$ le etichette di classe. L'obiettivo del problema di classificazione binaria è di prevedere l'etichetta $\hat{y}_i$ con una funzione di mappatura $f_{\boldsymbol{\theta}}(x_i)$ $\hat{y}_i = 1 \quad \text{se} \quad f_\theta(\mathbf{x}_i) > 0.5, \quad \text{altrimenti} \quad \hat{y}_i = -1$. Nella classificazione binaria ogni

utente ha una feature sensibile $s \in 0{,}1$ associata. L'obiettivo di una classificazione equa è evitare che s influenzi in modo non etico il processo decisionale. Lo spazio in cui si analizza tale situazione sono l'equità di gruppo e l'equità individuale.

1.4 Fairness Nei Recommender System

In questa sezione si introducono definizioni,metriche, metodi, valutazioni alla Fairness nella ricerca sui sistemi di raccomandazione. Si parte da una panoramica sui sistemi di raccomandazione e casi di unFairness, per poi presentare una tassonomia delle nozioni di Fairness, tecniche per promuoverla, valutazioni e dataset.

1.4.1 Nozioni generali e Unfairness

In un task di raccomandazione, si considera un insieme di utenti $U = \{u_1, u_2, \ldots, u_n\}$ e un insieme di item $V = \{v_1, v_2, \ldots, v_m\}$, dove n è il numero di utenti e m il numero di item. Le interazioni utente-item sono rappresentate da una matrice binaria $H = [h_{ij}]_{n \times m}$, dove $h_{ij} = 1$ se l'utente u_i ha interagito con l'item v_j, altrimenti $h_{ij} = 0$. L'obiettivo è prevedere i punteggi di preferenza S_{uv} per raccomandare a ogni utente u_i una lista top-N di item. I modelli moderni apprendono rappresentazioni di utenti e item, utilizzando funzioni di punteggio per generare raccomandazioni. I sistemi di Collaborative Filtering e Collaborative Reasoning sono addestrati direttamente dalla cronologia delle interazioni utente-item, mentre i modelli content-based possono anche sfruttare i profili utente/item come input o usare informazioni aggiuntive per aiutare ad addestrare il modello. Le considerazioni sulla Fairness nei sistemi di raccomandazione possono emergere da prospettive diverse. In particolare, si possono individuare due tipologie di ingiustizia: user-side e item-side.

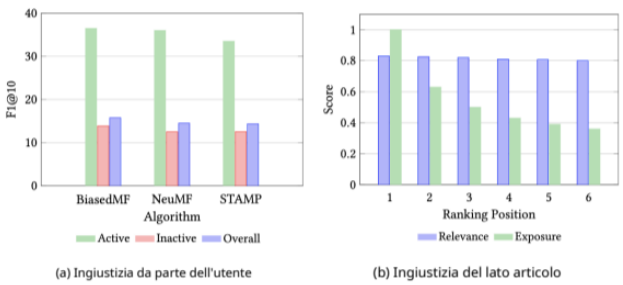
Unfairness user-side

L'unfairness lato utente si verifica quando diversi gruppi di utenti ricevono raccomandazioni di qualità o rilevanza significativamente diversa. Questa forma di disparità può compromettere l'esperienza dell'utente e generare effetti discriminatori, soprattutto nei casi in cui l'algoritmo apprende preferenze prevalentemente da

gruppi dominanti. Un problema noto è che gli utenti con comportamenti o interessi meno incisivi tendono a ricevere raccomandazioni meno personalizzate o meno pertinenti. Inoltre, gruppi con bassa attività, definiti cold users, vengono spesso trascurati a favore di utenti già ben profilati. Questo certamente porta a una disparità nell'utilità del servizio offerto dall'algoritmo, ovvero nella soddisfazione media ricevuta da gruppi diversi di utenti [27].

Unfairness Item-side

Nei sistemi di raccomandazione, gli elementi sono generalmente ordinati in base a un punteggio di rilevanza appreso dal modello, con quelli a punteggio più alto posizionati in cima alla lista. L'unfairness lato item si manifesta quando determinati contenuti, prodotti o candidati (individuati dall'algoritmo come item) ricevono una visibilità significativamente inferiore rispetto ad altri, nonostante siano comparabili in termini di rilevanza o qualità. Questo tipo di disparità è particolarmente critico in contesti come il lavoro o l'informazione o la divulgazione di notizie, dove la posizione e l'esposizione nella lista dei risultati può determinare successo o fallimento di un item. Questo fenomeno è analizzato attraverso una funzione di esposizione decrescente, comunemente utilizzata nella metrica Discounted Cumulative Gain (DCG), definita come: $\frac{1}{\log(1+i)}$, dove i rappresenta la posizione dell'elemento nella lista. Questa funzione evidenzia come gli elementi nelle prime posizioni ricevano molta più attenzione rispetto a quelli posizionati più in basso, causando un'esposizione ingiusta[27, 31].



(a) Ingiustizia da parte dell'utente (b) Ingiustizia del lato articolo

Figura 1.1: (a) Differenza tra gruppi di utenti inattivi e attivi sulla qualità della raccomandazione
(b) Differenza di esposizione tra elementi con punteggi di pertinenza ravvicinati [27].

1.4.2 Tassonomie e classificazioni

In questa sezione, è fornita una panoramica delle principali prospettive sulla Fairness nei sistemi di raccomandazione discutendo una tassonomia delle nozioni di Fairness nell'ambito dei Recommender System. Alcune di esse, come la User Fairness, la Multi-sided Fairness e la Dynamic Fairness, sono specifiche dei sistemi di raccomandazione, poiché tengono conto di aspetti rilevanti come l'interattività, la natura multi-side e la dinamicità dell'algoritmo. Altre nozioni, invece, come la Group Fairness, la Individual Fairness e la Causal Fairness, possono invece essere applicate più genericamente a diversi compiti di apprendimento automatico. Di seguito, oltre a fornire una definizione formale per ciascuna categoria, vi sarà un confronto relativo alle nozioni Fairness nei sistemi di raccomandazione.

Single-side Fairness vs. Multi-side Fairness

- **Single-side Fairness**: Un sistema di raccomandazione equo single-side considera le richieste di equità da una sola parte, che può essere il lato utente, il lato oggetto o il lato piattaforma.

- **Multi-side Fairness**: Un sistema di raccomandazione equo multi-side tiene conto delle richieste di equità provenienti da più parti contemporaneamente. Questo approccio affronta le problematiche di equità sia per gli utenti che per gli oggetti o anche per la piattaforma.

Alcuni approcci hanno esaminato come bilanciare l'equità tra diversi stakeholder. Ad esempio, in un contesto multi-lato, è importante ridurre le disuguaglianze di esposizione tra gli oggetti e garantire che gli utenti ricevano una distribuzione equa del valore delle raccomandazioni. In altri scenari, si propone di aggiornare gradualmente gli algoritmi di raccomandazione per evitare cambiamenti bruschi nell'esposizione degli oggetti, assicurando contemporaneamente un'utilità minima per ogni cliente. Alcuni studi si concentrano sul miglioramento della giustizia, sia a livello locale che globale, affrontando il bias di popolarità che può influire sulle raccomandazioni.

Users Fairness vs. Item Fairness

- **Users Fairness**: Un sistema di raccomandazione equo per gli utenti deve trattare in modo giusto gruppi predefiniti di utenti o utenti simili. Le richieste di equità dagli utenti riguardano principalmente la qualità delle raccomandazioni che ricevono.
- **Item Fairness**: Un sistema di raccomandazione equo per gli item deve trattare in modo giusto gruppi di item o singoli simili. Le considerazioni di equità per gli oggetti si concentrano principalmente sull'opportunità di esposizione degli oggetti nelle liste di raccomandazione.

Le problematiche di Fairness per gli utenti includono la disparità nelle prestazioni tra utenti attivi e inattivi, e la difficoltà di trattare equamente gli utenti che hanno interazioni limitate con il sistema, come nel caso degli utenti cold-start. L'equità per gli oggetti, del resto, si concentra sul risolvere i problemi di esposizione ingiusta dovuti a bias di popolarità, dove gli oggetti popolari ottengono maggiore visibilità a discapito di oggetti meno popolari ma di qualità equivalente o superiore. Una soluzione comune potrebbe essere quella di aumentare la presenza di oggetti meno popolari o garantire che il tasso di esposizione degli oggetti sia proporzionale alla loro qualità.

Group Fairness vs. Individual Fairness

- **Group Fairness**: Richiede che diversi gruppi predefiniti siano trattati equamente.
- **Individual Fairness**: Stabilisce che individui simili debbano ricevere trattamenti simili.

L'equità di gruppo viene spesso misurata confrontando le discrepanze nelle prestazioni delle raccomandazioni tra gruppi definiti in base a caratteristiche sensibili. La ricerca dimostra che l'introduzione di tecniche di apprendimento consapevole dell'equità può migliorare sia la giustizia nella qualità delle raccomandazioni tra gruppi diversi sia le prestazioni complessive del sistema. Tuttavia, la disparità nell'accuratezza della raccomandazione può variare in base al contesto: è più evidente in scenari in cui l'interazione con il sistema richiede un costo marginale significativo,

mentre è meno rilevante in contesti con costi marginali ridotti . L'equità individuale, invece, si concentra sul garantire che ciascun utente riceva raccomandazioni proporzionate alle proprie preferenze. Un esempio è il caso delle raccomandazioni di gruppo, dove si cerca di massimizzare la soddisfazione di ogni membro minimizzando allo stesso tempo eventuali squilibri tra le loro utilità. Un'altra prospettiva riguarda l'equità controfattuale, che richiede raccomandazioni invariate anche se le caratteristiche sensibili dell'utente mutano in un mondo controfattuale, garantendo così un trattamento equo indipendentemente da tali attributi. Queste distinzioni evidenziano l'importanza del criterio scelto di equità più adatto al contesto di applicazione, bilanciando giustizia ed efficacia nelle raccomandazioni.

Associative Fairness vs. Causal Fairness

- **Equità Associativa**: Le metriche di equità associativa si basano sulla misurazione delle correlazioni tra gli esiti predittivi e le caratteristiche sensibili.
- **Equità Causale**: Le metriche di equità causale si basano sulla misurazione degli effetti causali delle caratteristiche sensibili sugli esiti predittivi.

Inizialmente, la ricerca sull'equità nell'apprendimento automatico si è concentrata su metriche basate sulle associazioni, con l'obiettivo di identificare differenze statistiche tra individui o "sotto-popolazioni". L'equità basata esclusivamente sulle associazioni non è sempre sufficiente, poiché non considera le relazioni causali tra le variabili; spesso, infatti, le decisioni discriminatorie di un modello derivano da una relazione causale tra caratteristiche sensibili e le decisioni del sistema. Per affrontare questo limite, sono stati introdotti approcci basati sulla causalità, che considerano anche la struttura del sistema. Questi si basano su interventi e concetti controfattuali, che presentano sfide applicative, poiché alcune proprietà non sono direttamente osservabili dai dati. L'integrazione della causalità nei sistemi di raccomandazione è stata essenziale per migliorarne l'equità, offrendo una concreta opportunità per ottenere modelli più trasparenti.

Static Fairness vs. Dynamic Fairness

- **Static Fairness**: Un sistema di raccomandazione staticamente equo considera un ambiente in cui le preferenze degli utenti e le caratteristiche degli oggetti

rimangono invariati durante l'intero processo di raccomandazione. La maggior parte degli studi sulla Fairness nei sistemi di raccomandazione adotta questo approccio, risolvendo il problema tramite ottimizzazione con vincoli adeguati. Tuttavia, tali sistemi non tengono conto dei cambiamenti che avvengono nel tempo.

- **Dynamic Fairness**: Un sistema di raccomandazione equo dinamicamente tiene conto dei cambiamenti delle preferenze e delle proprietà rispettivamente di utenti e oggetti durante il processo di raccomandazione. In un sistema dinamico, gli utenti possono diventare attivi da inattivi, e gli oggetti popolari pur essendo impopolari, alterando le dinamiche di equità.

Studi recenti dimostrano che applicare criteri di equità statici in modo poco chiaro può addirittura aggravare l'ingiustizia. Alcuni approcci hanno affrontato l'equità dinamica, considerando fattori come il cambiamento delle preferenze e degli attributi degli utenti durante l'interazione, e proponendo metodi come i processi decisionali di Markov con vincoli dinamici di equità o tecniche di apprendimento rinforzato per bilanciare lungo il tempo l'accuratezza e l'equità.

Centralized Fairness vs. Federated Fairness

- **Centralized Fairness**: Si basa su un algoritmo centrale che ha accesso ai dati di tutti gli utenti per migliorare l'equità del sistema.

- **Federated Fairness**: Si basa ,invece, su un algoritmo decentralizzato che migliora l'equità senza accedere a tutti i dati degli utenti, mantenendo le informazioni sui dispositivi locali.

L'accuratezza delle raccomandazioni dipende in larga misura dal livello di dettaglio delle informazioni raccolte sugli utenti, questa dipendenza potrebbe però sollevare problemi di privacy, specialmente nei casi in cui il sistema deve raccomandare contenuti a nuovi utenti senza dati storici. Per affrontare questa sfidasi applica l'apprendimento federato, che consente di addestrare il modello senza trasferire i dati utente su un server centrale, garantendo così la privacy. Tuttavia, questo approccio introduce nuove sfide per l'equità nei sistemi di raccomandazione. Da un

lato, alcuni obiettivi di equità user-based possono entrare in conflitto con l'approccio "federato", poiché l'equità spesso richiede la raccolta di caratteristiche sensibili degli utenti. Per risolvere questo problema, si possono modificare gli obiettivi di equità o integrare moduli di protezione della privacy. Dall'altro, i sistemi federati possono introdurre problemi di equità collaborativa, che riguardano l'uguaglianza del contributo tra i partecipanti. Questo problema è stato studiato in vari campi, come la sanità, la visione artificiale e l'elaborazione del linguaggio naturale. Allo stato attuale, la ricerca sull'equità nei sistemi di raccomandazione federati è ancora limitata, rendendo necessaria ulteriore ricerca per bilanciare equità, accuratezza e privacy.

1.5 Consumer-Fairness nei Recommender System

1.5.1 Introduzione e obiettivi dello studio

Il concetto di consumer fairness si riferisce alla necessità di garantire un trattamento equo a tutti gli utenti di un sistema di raccomandazione, evitando che i meccanismi interni generino discriminazioni, in particolare verso gruppi protetti. Lo studio di Boratto et al. [6] affronta in modo sistematico questo problema, riconoscendo come i sistemi raccomandativi influenzino sempre più le decisioni quotidiane, ma possano anche amplificare disuguaglianze preesistenti. L'analisi si concentra in particolare sulla group consumer fairness, ovvero sull'assenza di effetti discriminatori nelle raccomandazioni rivolte a utenti con caratteristiche demografiche sensibili, come genere ed età. Per valutare l'efficacia delle tecniche di mitigazione presenti in letteratura, sono stati analizzati quindici approcci su due dataset pubblici (MovieLens 1M e LastFM 1K), utilizzando due metriche di fairness: il disparate impact e il test di Kolmogorov-Smirnov. I risultati evidenziano che molte tecniche riescono a ridurre le disuguaglianze senza compromettere la qualità delle raccomandazioni, ma poche si dimostrano efficaci su entrambe le metriche, e il bias non colpisce sempre sistematicamente i gruppi minoritari.

1.5.2 Metodologie di mitigazione analizzate

Le tecniche analizzate coprono diversi approcci, che intervengono in varie fasi del processo raccomandativo:

- **Burke et al.** propongono una versione estesa di SLIM con regolarizzazione, per bilanciare la composizione del vicinato degli utenti [8].
- **Frisch et al.** adottano un modello di co-clustering con regressione ordinale, che integra gli attributi sensibili nella struttura [16].
- **Li et al.** propongono un riordinamento delle raccomandazioni, ottimizzando il trade-off tra utilità e equità tra utenti attivi e meno attivi [26].
- **Ekstrand et al.** applicano tecniche di resampling per riequilibrare la distribuzione delle interazioni utente [14].
- **Kamishima et al.** integrano vincoli di indipendenza statistica rispetto agli attributi sensibili nella funzione obiettivo del modello [23].
- **Rastegarpanah et al.** introducono utenti sintetici con valutazioni strategiche per compensare gli squilibri nei dati [33].
- **Ashokan e Haas** modificano i punteggi delle raccomandazioni, applicando strategie basate su valore o parità [1].
- **Wu et al.** utilizzano un approccio avversariale su grafi per eliminare informazioni sensibili dalle rappresentazioni latenti [39].

Per garantire coerenza nella valutazione, i modelli sono stati rieseguiti con i codici originali, mentre pre-processing e metriche sono stati standardizzati. L'accuratezza è stata misurata con RMSE e NDCG, mentre la fairness è stata valutata tramite disparate impact e test di Kolmogorov-Smirnov.

1.5.3 Risultati e conclusioni

I risultati sono stati eterogenei; nei task di raccomandazione Top-N, alcuni modelli si sono dimostrati equi secondo una metrica ma non secondo l'altra, a conferma che le due misure catturano aspetti distinti dell'equità. In particolare, su LastFM

1K l'iniquità di partenza era maggiore, e il resampling ha in alcuni casi peggiorato le prestazioni di fairness. Per quanto riguarda la predizione dei rating, gli effetti delle mitigazioni sono stati in genere modesti, salvo che per i metodi proposti da Kamishima e Ashokan [23, 1], che hanno prodotto miglioramenti più significativi.

1.6 Bias

1.6.1 Data Bias

La principale fonte di Bias nel Machine Learnig è rappresentata proprio dai dati di addestramento, che possono essere influenzati da distorsioni durante le fasi di generazione, raccolta e archiviazione. Si distinguono più tipologie di Data Bias:

- Bias Statistico: Si presenta quando vi sono dei difetti nel design sperimentale, nel processo di raccolta dei dati o se i dati sono poco accurati che non rappresentano la reale popolazione; [27]

- Bias Pre-esistente: Si presenta quando i dati di addestramento riflettono decisioni colme di pregiudizi, come discriminazioni di genere, razza o età, portando il sistema a perdere obbiettività ed equità;[27]

- Bias di Esposizione: Si manifesta quando gli utenti visualizzano solo un sottoinsieme di item, facendo sì che le interazioni non rilevate non corrispondano a preferenze negative; [11]

- Bias di Conformità: Si verifica quando gli utenti tendono ad allinearsi al comportamento del gruppo, anche contrastando il proprio giudizio personale, facendo sì che il feedback non rifletta sempre la realtà;[11]

- Bias di Posizione: Si verifica quando gli utenti tendono a interagire maggiormente con gli item posizionati in cima alla lista di raccomandazione, indipendentemente dalla loro effettiva rilevanza, facendo sì che gli item con cui si interagisce potrebbero non essere quelli più pertinenti ma quelli più popolari;[11]

1.6.2 Model Bias

L'unFairness nel Machine Learnig può essere causata da bias introdotti durante la progettazione, la formazione e la valutazione del modello. Si distinguono due tipologie di bias particolarmente rilevanti: una legata alla progettazione del modello (Bias da variabile omessa e induttivo) e l'altra legata alla sua valutazione (Bias di valutazione), entrambi in grado di compromettere l'equità e distorcere i risultati.

- Bias da variabile omessa: Si presenta quando alcune variabili o caratteristiche fondamentali, non vengono considerate durante la progettazione e l'addestramento del modello;[27]
- Bias induttivo: è l'insieme di assunzioni che guidano il modello nell'apprendimento della funzione obiettivo e nella generalizzazione oltre i dati di training;[11]
- Bias di valutazione: Si presenta solitamente quando si utilizzano parametri di riferimento inappropriati nella valutazione del modello.[27]

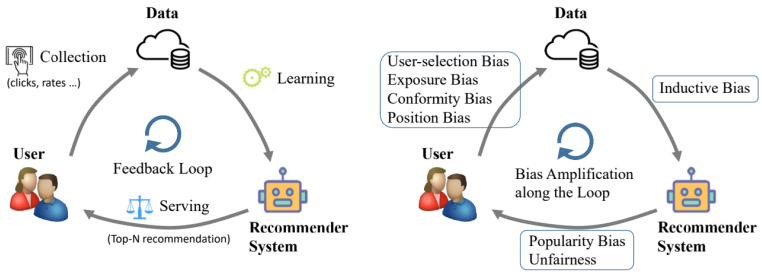


Figura 1.2: Esempio di feedback loop nei sistemi algoritmici: l'output di un sistema influenza i dati futuri.

I bias sono interconnessi a causa del feedback loop [1] Questo meccanismo può amplificare o correggere il comportamento del sistema, a seconda che il feedback sia positivo o negativo. nel caso in cui i dati di addestramento, spesso raccolti da comportamenti utente distorti, possano essere ulteriormente influenzati da algoritmi biased, aggraverebbero il problema. È quindi essenziale considerare queste interazioni per mitigare i bias in modo efficace.

[1]Feedback loop : processo in cui l'output di un sistema viene riutilizzato come input, creando un ciclo.

1.7 Tipologie di Approcci per il Debiasing

Generalmente, i metodi che mitigangarantiscono l'equità nel Machine Learnig rientrano in tre categorie: Pre-processing, In-processing e Post-processing.

- Pre-processing: Corregge i dati di addestramento per rimuovere possibili distorsioni prima dell'apprendimento, garantendo così maggiore Fairness. Tuttavia, questo metodo può ridurre l'accuratezza del modello e non sempre risulta efficace contro l'unFairness.
- In-processing: Modifica il processo di apprendimento per bilanciare accuratezza e correttezza, integrando metriche di Fairness nella funzione obiettivo. È un approccio flessibile, ma può provocare problemi legati all'ottimizzazione.
- Post-processing: Modifica l'output del modello per ridurre l'ingiustizia senza alterare i dati o l'addestramento. Anche questo metodo è flessibile, ma richiede l'accesso completo ai dati durante le fasi di test.

1.8 Overview sulla sostenibilità ambientale

Quando si parla di fairness nel machine learning, l'attenzione si concentra spesso sui suoi benefici sociali. Tuttavia, il costo ambientale associato allo sviluppo e all'addestramento di questi modelli necessita il focus della ricerca. In particolare, gli algoritmi fairness-aware richiedono risorse computazionali elevate e cicli di addestramento più lunghi rispetto ai modelli standard. E queste risorse non sono gratuite dal punto di vista ambientale. Dietro ogni esperimento apparentemente software, si nasconde una realtà hardware molto concreta: server e data center energivori, alimentati da migliaia di GPU, che consumano elettricità e generano emissioni di CO_2. Più un modello è complesso, più è lungo il suo training, più sarà alto il suo impatto ambientale. Secondo uno studio [35], l'addestramento di un modello di Natural Language Processing di grandi dimensioni può produrre fino a 284 tonnellate di CO_2, una quantità paragonabile a cinque volte le emissioni di un'auto durante l'intero ciclo di vita. Si è analizzato che il solo addestramento di GPT-3, da parte di Open-AI ha richiesto più di 1.200 MWh di energia elettrica, generando emissioni equivalenti a quelle annuali di una dozzina di famiglie americane [30]. Si

è stimato che l'AI contribuisce al 4% delle missioni prodotte, il doppio di quanto ne causa il traffico aereo mondiale.
I modelli fairness-aware non raggiungono questa scala, ma è importante notare che introducono un carico aggiuntivo non trascurabile: training iterativi, moduli avversari, regolarizzazioni sulla fairness. Tramite i test effettuati e analizzati nel cap.5 si è potuto osservare come le configurazioni con forti vincoli di equità generassero emissioni anche del 30–50% superiori, pur mantenendo la stessa accuratezza. Ma il problema è più ampio. Le grandi piattaforme cloud su cui si appoggiano la maggior parte dei modelli, AWS, Google Cloud, Azure, gestiscono data center che consumano decine di terawattora ogni anno, con un impatto globale significativo. Se queste infrastrutture non sono alimentate da fonti rinnovabili, ogni esperimento può contribuire, nel suo piccolo, all'aggravarsi della crisi climatica. Il rischio è di creare sistemi più giusti per le persone, ma più dannosi per l'ambiente. La sfida moderna è invece quella di sviluppare modelli che siano equamente sostenibili, per tutti i tre pilastri della sostenibilità.

Capitolo 2

Metriche di Valutazione della Fariness

2.1 Metriche nella Classificazione

Nel seguente segmento si introducono le metriche di valutazione della Fairness nelle attività di classificazione binaria [27]. In un'attività di classificazione binaria si dispone di dati di addestramento $D_T = (x_i, y_i)_{i=1}^N$, dove $x \in \mathbb{R}^d$ sono i vettori di caratteristiche e $y \in \{-1,1\}$ le etichette di classe. L'obiettivo della classificazione è di prevedere l'etichetta $\hat{y}_i$ con una funzione di mappatura $f_{\boldsymbol{\theta}}(x_i)$ $\hat{y}_i = 1 \quad \text{se} \quad f_\theta(\mathbf{x}_i) > 0.5, \quad \text{altrimenti} \quad \hat{y}_i = -1$. Ogni utente ha una feature sensibile $s \in 0,1$ associata; l'obiettivo è evitare che s influenzi in modo non etico il processo decisionale.

2.1.1 Fairness di gruppo

Richiede che i gruppi protetti siano trattati in modo simile ai gruppi avvantaggiati. Le nozioni di fairness di gruppo includono metriche basate sul tasso di predizioni positive, che richiedono la parità dei tassi di predizioni positive $\Pr(\hat{y} = 1)$ tra i diversi gruppi, e e metriche basate sulla matrice di confusione che considerano il Tasso di Veri Positivi (True Positive Rate, TPR), il Tasso di Veri Negativi (True Negative Rate, TNR), il tasso di Falsi Positivi (False Positive Rate, FPR) e il tasso di Falsi Negativi (False Negative Rate, FNR), in modo da catturare le differenze in

maniera più dettagliata [10]. Un esempio di metrica basata sul tasso di predizioni positive è:

Statistical Parity

La Parità Statistica, anche chiamata Parità Demografica, richiede che ogni gruppo abbia la stessa probabilità di essere classificato come positivo[9, 42].:

$$\Pr(\hat{y} = 1 \mid s = 0) = \Pr(\hat{y} = 1 \mid s = 1)$$

Il limite di questa nozione è che ignora le differenze tra i gruppi, quindi se un elemento dovesse essere impopolare per motivi legittimi, come ad esempio un prodotto di nicchia, sarebbe riconosciuto come un elemento non valido. Di conseguenza, imporre che tutti gli elementi debbano essere popolari alla stessa maniera potrebbe risultare poco ragionevole. Oltre alle metriche che si concentrano sulle varianti del tasso di predizioni positive $\Pr(\hat{y} = 1)$, la maggior parte dei criteri di fairness di gruppo si basano sulla matrice di confusione.

Equal Opportunity[18]

L'Uguaglianza di Opportunità richiede che il Tasso di Veri Positivi (TPR) sia lo stesso tra i diversi gruppi [8]:

$$\Pr(\hat{y} = 1 \mid y = 1, s = 1) = \Pr(\hat{y} = 1 \mid y = 1, s = 0)$$

Equalized Odds[3]

Più stringente dell'Uguaglianza di Opportunità, il criterio delle pari probabilità considera anche il Tasso di Falsi Positivi (FPR) e richiede che i diversi gruppi abbiano lo stesso TPR e lo stesso FPR [**?**, 18]:

$$\Pr(\hat{y} = 1 \mid y = 1, s = 1) = \Pr(\hat{y} = 1 \mid y = 1, s = 0) \quad \text{e}$$
$$\Pr(\hat{y} = 1 \mid y = -1, s = 1) = \Pr(\hat{y} = 1 \mid y = -1, s = 0)$$

Overall Accuracy Equality[3]

Questo criterio richiede la stessa accuratezza tra i gruppi :

$$\Pr(\hat{y} = -1 \mid y = -1, s = 1) + \Pr(\hat{y} = 1 \mid y = 1, s = 1) =$$
$$\Pr(\hat{y} = -1 \mid y = -1, s = 0) + \Pr(\hat{y} = 1 \mid y = 1, s = 0)$$

Equalizing Disincentives [22]

Questo, invece, richiede che la differenza tra il Tasso di Veri Positivi (TPR) e il Tasso di Falsi Positivi (FPR) sia la stessa tra i gruppi :

$$\Pr(\hat{y} = 1 \mid y = 1, s = 1) - \Pr(\hat{y} = 1 \mid y = -1, s = 1)$$
$$= \Pr(\hat{y} = 1 \mid y = 1, s = 0) - \Pr(\hat{y} = 1 \mid y = -1, s = 0)$$

Treatment Equality[3]

Infine, questo criterio richiede che il rapporto tra il Tasso di Falsi Negativi (FNR) e il Tasso di Falsi Positivi (FPR) sia lo stesso tra i gruppi :

$$\frac{\Pr(\hat{y} = 1 \mid y = -1, s = 1)}{\Pr(\hat{y} = -1 \mid y = 1, s = 1)} = \frac{\Pr(\hat{y} = 1 \mid y = -1, s = 0)}{\Pr(\hat{y} = -1 \mid y = 1, s = 0)}$$

2.1.2 Fairness Individuale

Invece di considerare la fairness tra gruppi, le nozioni di fairness individuale richiedono che individui simili siano trattati in modo simile; tra gli approcci comuni ci sono:

Counterfactual Fairness

La Fairness Controfattuale è basata sulla causalità, la quale richiede che il risultato predetto dal sistema di apprendimento sia lo stesso sia nel mondo controfattuale che nel mondo reale per qualsiasi individuo[24]. Dato un insieme di variabili latenti di sfondo U, il predittore $\hat{Y}$ è equo controfattualmente se, in qualsiasi contesto $\mathbf{X} = \mathbf{x}$ e $S = s$, vale la seguente equazione per tutti i y e per qualsiasi valore s' raggiungibile da S:

$$\Pr\left(\hat{Y}_{S \leftarrow s}(U) = y \mid \mathbf{X} = \mathbf{x}, S = s\right) = \Pr\left(\hat{Y}_{S \leftarrow s'}(U) = y \mid \mathbf{X} = \mathbf{x}, S = s\right)$$

Tuttavia, le tecniche per raggiungere la fairness controfattuale possono essere diverse, alcune delle metodologie impiegate in tale ambito sono: rimuovere le caratteristiche sensibili e i loro discendenti dal modello e dalla funzione di predizione; autoencoder variazionali; apprendimento avversario; pre-elaborazione dei dati; regolarizzazione causale; aumento dei dati, ecc.

Fairness Through Awareness

Tale approccio richiede che due individui con caratteristiche non sensibili simili ricevano risultati predetti simili[13]. Prendiamo in considerazione due individui $\mathbf{x}_1$ e $\mathbf{x}_2$, La distanza tra i due individui è definita da $d(\mathbf{x}_1, \mathbf{x}_2)$, e la differenza tra i risultati predetti è calcolata tramite $F(\hat{y}_1, \hat{y}_2)$. La Fairness attraverso la consapevolezza richiede che:

$$F(\hat{y}_1, \hat{y}_2) \leq \alpha \cdot d(\mathbf{x}_1, \mathbf{x}_2)$$

ovvero che la differenza nel trattamento deve avere come limite superiore la differenza tra i due individui presi in considerazione.

2.2 Metriche Nei Ranking

Oltre agli studi sulla fairness nei task di classificazione, la ricerca ha affrontato il problema della fairness nei sistemi di ranking[27]. Gli algoritmi di raccomandazione possono essere visti come un tipo di problema di ranking, rendendo gli studi su questi ultimi rilevanti per migliorare l'equità nelle raccomandazioni. La ricerca si concentra principalmente sull'ingiustizia lato item (candidati da classificare), mentre nei sistemi di raccomandazione il concetto di fairness è stato esteso ai multiple stakeholder. Esistono due tipologie principali di attività di ranking: lo Score-based Ranking, dove il punteggio è ottenuto direttamente da una funzione predefinita, e il Learning to Rank, dove, invece, il punteggio è stimato addestrando un modello su esempi di training arricchiti con preferenze. Per discutere la fairness da una prospettiva di machine learning, la ricerca si concentra prevalentemente sul problema del Learning to Rank. In un'attività di Learning to Rank supervisionato, è dato un insieme di candidati $\{c_1, c_2, \ldots, c_n\}$, dove ciascun candidato è descritto da un insieme di feature X, che possono includere feature sensibili S. L'obiettivo di un algoritmo di ranking è generare una lista ordinata l che ordina i candidati in

base alla loro rilevanza prevista rispetto a una query di ricerca, I candidati con il punteggio più alto appariranno in cima alla lista, ottenendo maggiore visibilità. In questa sezione, per fornire una panoramica da una prospettiva ben definita, divideremo le tipologie di fairness in: Fairness basata sulla probabilità e Fairness basata sull'esposizione.

2.2.1 Fairness basata sulla probabilità

Le nozioni di fairness basata sulla probabilità richiedono una proporzione minima/massima di candidati protetti nella lista top-K[43], ad esempio la fairness è definita come:

$$L_{k\ell} \leq \sum_{1 \leq j \leq k} \sum_{i \in P_\ell} x_{ij} \leq U_{k\ell},$$

dove x è una matrice di assegnazione binaria e $x_{ij} = 1$ se l'item i è assegnato alla posizione j. I limiti $U_{k\ell}$(Superiore) e $L_{k\ell}$ (Inferiore) garantiscono che un certo numero di item con proprietà ℓ compaiano nelle prime K posizioni, perseguendo obiettivi di fairness di gruppo.

2.2.2 Fairness Basata sull'Esposizione

La fairness basata sull'esposizione distribuisce equamente le opportunità di esposizione, considerando il position bias[34].La fairness è definita come la differenza media di esposizione tra gruppi:

$$F(G_1, G_2) = \left| \frac{1}{|G_1|} \sum_{c \in G_1} \text{Exposure}(c) - \frac{1}{|G_2|} \sum_{c \in G_2} \text{Exposure}(c) \right|$$

Dove Exposure(c) rappresenta l'esposizione attesa del candidato c in base alla sua posizione nel ranking. In modo analogo, l'equità tra individui può essere stimata calcolando la differenza assoluta tra le esposizioni di due candidati:

$$F(c_1, c_2) = |\text{Exposure}(c_1) - \text{Exposure}(c_2)|$$

Questo approccio permette di valutare sia la fairness di gruppo che quella individuale, senza imporre limiti rigidi sulle posizioni.

2.3 Metriche Nei Recommender System

I sistemi di raccomandazione utilizzano varie metriche per valutare la fairness, ciascuna legata alla specifica definizione di equità considerata. La ricerca, [37], si concentra principalmente sulla outcome fairness nota anche come giustizia distributiva, in particolare sulle varianti: consistent fairness e calibrated fairness. Per la consistent fairness, le metriche principali sono:Absolute Difference, varianza e coefficiente di Gini. Queste sono versatili e adatte a valutazioni su gruppi o individui, sia per utenti che per item, anche se possono risultare imprecise in contesti specializzati. Per la calibrated fairness, si utilizzano principalmente la divergenza KL e la norma L1, che hanno un'ampia applicabilità ma poche implementazioni specifiche per due gruppi distinti. Tuttavia, queste metriche standard si basano su statistiche del primo ordine e non colgono le specificità contestuali di utenti e item, motivo per cui la ricerca ha proposto alternative come:

- Prove statistiche avanzate sulle distribuzioni
- Metriche specializzate per la fairness

Anche se più limitate, queste soluzioni offrono valutazioni più accurate per problemi specifici. La scelta delle metriche va quindi basata sulla definizione di fairness e sulle caratteristiche del sistema e del dominio.

2.3.1 Metriche di Consistent Fairness[41]

Absolute difference (AD)

L'AD quantifica la differenza assoluta di utilità tra gruppo protetto (G_0) e non protetto (G_1), dove $f(G)$ rappresenta il rating medio per gli utenti o l'esposizione totale per gli item.Più i valori tendono allo zero maggiore sarà la qualità della raccomandazione.

$$AD = |f(G_0) - f(G_1)|$$

Statistica KS

La statistica KS è un test non parametrico che valuta l'uguaglianza tra due distribuzioni confrontando le loro funzioni di ripartizione empiriche:

$$KS = \left| \sum_{i=1}^{T} I \cdot \frac{\mathcal{G}(R_0, i)}{|R_0|} - \sum_{i=1}^{T} I \cdot \frac{\mathcal{G}(R_1, i)}{|R_1|} \right|$$

In questo contesto, T rappresenta il numero di intervalli, mentre I indica l'ampiezza degli intervalli. La funzione $\mathcal{G}(R, i)$ si riferisce al conteggio nelle distribuzioni, ovvero il numero di occorrenze all'interno di ciascun intervallo. Rispetto alla Differenza Assoluta (AD), la statistica KS cattura anche inconsistenze di ordine superiore, non solo differenze nelle medie. Un valore KS più basso indica distribuzioni più simili tra i gruppi.

Metriche basate sull'esposizione

Le metriche rND, rKL e rRD si basano sull'esposizione degli item per un ranking τ. A differenza delle metriche precedenti, queste metriche prendono in considerazione la posizione di esposizione, calcolando l'ingiustizia cumulativa scontata e normalizzata in modo simile all'NDCG. Per queste metriche, più basso è il valore, più eque sono le raccomandazioni:

$$rND = \frac{1}{Z} \sum_{i=10,20,\ldots}^{N} \frac{1}{\log_2 i} \left| \frac{|S^+_{1\ldots,i}|}{i} - \frac{|S^+|}{N} \right|,$$

$$rKL = \frac{1}{Z} \sum_{i=10,20,\ldots}^{N} \frac{1}{\log_2 i} \left(\frac{|S^+_{1\ldots,i}|}{i} \log \frac{\frac{|S^+_{1\ldots,i}|}{|S^+|}}{N} + \frac{|S^-_{1\ldots,i}|}{i} \log \frac{\frac{|S^-_{1\ldots,i}|}{|S^+|}}{N} \right),$$

$$rRD = \frac{1}{Z} \sum_{i=10,20,\ldots}^{N} \frac{1}{\log_2 i} \left| \frac{|S^+_{1\ldots,i}|}{|S^-_{1\ldots,i}|} - \frac{|S^+|}{|S^-|} \right|.$$

Dove il normalizzatore Z rappresenta il valore massimo possibile delle corrispondenti misurazioni, $|S^+_{1\ldots,i}|$ è il numero del gruppo protetto nella top-i del ranking τ, S^+ è il numero del gruppo non protetto nell'intero ranking.

Pairwise Ranking Accuracy Gap (PRAG)

La PRAG quantifica l'equità negli item confrontando l'accuratezza del ranking tra coppie di item appartenenti a gruppi diversi (I_1, I_2)[4], calcolata come: dove $PairAcc$ misura la probabilità di corretta ordinazione dati i feedback reali y_i. Valori inferiori indicano maggiore equità.

$$PRAG = |PairAcc(I_1 > I_2|q) - PairAcc(I_1 < I_3|q)|$$

dove $PairAcc$ misura la probabilità di corretta ordinazione dati i feedback reali y_i; valori inferiori indicano maggiore equità.

Metriche di Unfairness

La Value Unfairness è stata proposta per misurare l'inconsistenza nell'errore di predizione con segno tra due gruppi di utenti[41]. Esistono tre varianti di Value Unfairness. L'Absolute Unfairness misura l'inconsistenza dell'errore di predizione assoluto, mentre l'Underestimation Unfairness e l'Overestimation Unfairness misurano rispettivamente l'inconsistenza nella sottostima e nella sovrastima delle valutazioni reali. Più basso è il valore, più eque sono le raccomandazioni.

$$U_{val} = \frac{1}{m}\sum_{i=1}^{m} |(E_0[f]_t - E_0[r]_t) - (E_1[f]_t - E_1[r]_t)|$$

$$U_{abs} = \frac{1}{m}\sum_{i=1}^{m} ||E_0[f]_t - E_0[r]_t| - |E_1[f]_t - E_1[r]_t||$$

$$U_{under} = \frac{1}{m}\sum_{i=1}^{m} |max(0, E_0[r]_t - E_0[f]_t) - max(0, E_1[r]_t - E_1[f]_t)|$$

$$U_{over} = \frac{1}{m}\sum_{i=1}^{m} |max(0, E_0[f]_t - E_0[r]_t) - max(0, E_1[f]_t - E_1[r]_t)|$$

Dove $E_0[f]_t$ rappresenta il punteggio predetto medio per l'item i-esimo del gruppo 0, e $E_0[r]_t$ è la valutazione media per l'item i-esimo del gruppo 0.
Le metriche sopra indicate sono applicabili per misurare l'inconsistenza tra due gruppi. Di seguito si elencano e analizzano le metriche per misurare l'equità tra più gruppi. È importante notare che, poiché possiamo considerare la fairness individuale come un caso speciale di fairness di gruppo cioè ogni individuo appartiene

a un gruppo unico, teoricamente queste metriche di fairness di gruppo potrebbero applicarsi anche alla fairness individuale.

2.3.2 Metriche per Gruppi Multipli

2.3.3 Differential Fairness

La Differential Fairness è una misura di equità ispirata alla differential privacy, proposta per garantire trattamenti simili tra gruppi demografici, anche se piccoli o intersezionali[15]. Un meccanismo M soddisfa la ϵ-differential fairness se, per ogni coppia di gruppi sensibili s, s' e ogni possibile output y, vale:

$$e^{-\epsilon} \leq \frac{P(M(x) = y \mid A = s)}{P(M(x) = y \mid A = s')} \leq e^{\epsilon}$$

dove ϵ controlla il grado di disparità ammessa tra i gruppi: più è piccolo, maggiore è l'equità. Questo approccio è particolarmente utile per affrontare le disuguaglianze che emergono dall'intersezione tra più attributi sensibili, superando i limiti delle metriche tradizionali.

Varianza

La varianza rappresenta una misura statistica di dispersione ampiamente utilizzata per valutare la fairness nei sistemi di raccomandazione, applicabile sia a livello di gruppo che individuale. Questa metrica quantifica la variabilità di diverse grandezze significative, tra cui l'errore di predizione dei rating, il livello di soddisfazione degli utenti e l'esposizione media degli item.

$$\textbf{Variance} = \frac{1}{|\mathcal{V}|^2} \sum_{\psi_i \neq \psi_j} (f(v_u) - f(v_g))^2$$

Min-Max Difference (MMD)

L'MMD , definita come $\max(f(v)) - \min(f(v))$, misura la discrepanza tra valori estremi di utilità allocate. Applicata ai sistemi di raccomandazione, quantifica sia l'inconsistenza nell'esposizione media degli item che il disaccordo tra utenti, dove valori inferiori indicano maggiore equità distributiva.

$$\textbf{MMD} = \max(f(v), \forall v \in \mathcal{V}) - \min(f(v), \forall v \in \mathcal{V})$$

F-statistic (ANOVA)

L'ANOVA unidirezionale analizza differenze significative tra le medie di tre o più gruppi. La $F - statistic$ risultante funge da indicatore di fairness: valori più bassi riflettono maggiore equità nelle raccomandazioni. Questo approccio valuta l'errore di predizione dei rating, confrontando l'utilità individuale ($f(Ind_j)$), quella di gruppo ($\overline{v_i}$) e quella globale ($\overline{v}$).

$$F = \frac{MST}{MSE}$$

$$MST = \frac{\sum_i |v| \times (\overline{v}_i - \overline{v})^2}{|\mathcal{V}|^2 - 1}$$

$$MSE = \frac{\sum_i \sum_{t \in v_i} (f(Ind_j) - \overline{v}_i)^2}{\sum_{t \in \mathcal{V}'} |v| - |\mathcal{V}|}$$

Gini Index

Il Gini Index è ampiamente utilizzato in sociologia ed economia per misurare il grado di disuguaglianza sociale. Nelle raccomandazioni, misura la fairness individuale consistente, dove l'utilità può essere la rilevanza predetta per un utente o l'esposizione per un item. Valori più bassi indicano maggiore equità.

$$Gini = \frac{\sum_{v_x, v_y \in \mathcal{V}} |f(v_x) - f(v_y)|}{2|\mathcal{V}| \sum_v f(v)}$$

Jain Index

Utilizzato originariamente in ingegneria delle reti, misura l'ingiustizia nella soddisfazione degli utenti o nell'esposizione degli item. Valori più alti corrispondono a maggiore fairness.

$$Jain = \frac{(\sum_v f(v))^2}{|\mathcal{V}| \cdot \sum_v f(v)^2}$$

Shannon Entropy

Misura l'incertezza del sistema e viene applicata alla disparità nell'esposizione degli item. Valori più bassi indicano maggiore equità.

$$Entropy = -\sum_{v \in \mathcal{V}} p(v) \cdot \log p(v)$$

Rapporto Min-Max

Calcola il rapporto tra l'utilità minima e massima allocata. Utilizzato per valutare la soddisfazione degli utenti in raccomandazioni di gruppo. Valori più elevati denotano maggiore equità.

$$MinMaxRatio = \frac{\min\{f(v), \forall v \in \mathcal{V}\}}{\max\{f(v), \forall v \in \mathcal{V}\}}$$

Least Misery

Corrisponde al valore minimo di utilità allocata. Metriche di fairness comune per raccomandazioni di gruppo. Valori più alti indicano maggiore equitàQueste metriche sono comunemente utilizzate per misurare la disparità distributiva nei recommender systems, sia a livello individuale che di gruppo, come discusso in [41, 5].

$$LeastMisery = \min\{f(v), \forall v \in \mathcal{V}\}$$

Si osserva che tutte le metriche presentate hanno una natura scalare, dove valori più bassi corrispondono a un maggiore livello di Fairness. In particolare, le metriche di Value Unfairness risultano specificamente progettate per confronti tra due gruppi, mentre strumenti come Varianza, Min-Max Difference e ANOVA dimostrano maggiore versatilità, essendo applicabili a scenari che coinvolgono tre o più gruppi. Questa distinzione risulta particolarmente rilevante nella scelta della metrica più appropriata in base alla complessità del sistema di raccomandazione e alla struttura dei gruppi considerati.

2.3.4 Metriche di Calibrated Fairness

La calibrated fairness richiede la definizione di una funzione di merito $Merit(\cdot)$ che misura il merito di un individuo o gruppo [5]. La distribuzione equa dell'allocazione è calcolata come: $p_f(v_i) = \frac{Merit(v_i)}{\sum_j Merit(v_j)}$ dove $p_f(v_i)$ rappresenta la proporzione ideale di allocazione. La proporzione effettiva è invece: $p(v_i) = \frac{f(v_i)}{\sum_j f(v_j)}$ Le metriche di calibrated fairness misurano generalmente la differenza tra la distribuzione delle utilità p e la distribuzione dei meriti p_f. Tutte le metriche di gruppo in questo contesto sono applicabili a gruppi multipli.

MinSkew e MaxSkew

Lo skew (deviazione) per un gruppo v è definito come $\log(\frac{p_f(v)}{p(v)})$. MinSkew e MaxSkew sono definite come:

- MinSkew: valore più alto indica maggiore fairness

$$MinSkew = \min \left\{ \log \left(\frac{p_f(v)}{p(v)} \right), v \in \mathcal{V} \right\}$$

- MaxSkew: valore più basso indica maggiore fairness

$$MaxSkew = \max \left\{ \log \left(\frac{p_f(v)}{p(v)} \right), v \in \mathcal{V} \right\}$$

KL-divergence

Misura la differenza tra due distribuzioni di probabilità p e p_f. Valore più basso indica maggiore fairness.

$$D_{KL}(p, p_f) = \sum_{v \in \mathcal{V}} p(v) \log \frac{p(v)}{p_f(v)}$$

NDKL

Misura di unfairness basata su KL-divergence, normalizzata e scontata. Valore più basso indica maggiore fairness.

$$NDKL@K = \frac{1}{Z} \sum_{t}^{K} \frac{1}{\log(t+1)} D_{KL}^{l}$$

JS-divergence

Versione simmetrica della KL-divergence. Valore più basso indica maggiore fairness.

$$D_{JS}(p, p_f) = \frac{1}{2} \left(D_{KL} \left(p, \frac{p_f + p}{2} \right) + D_{KL} \left(p_f, \frac{p_f + p}{2} \right) \right)$$

Overall Disparity

Misura la disparità media tra proporzioni di utilità e merito. Valore più basso indica maggiore fairness.

$$OD = \frac{2}{|V|(|V|-1)} \sum_{t=0}^{|V|} \sum_{j=t+1}^{|V|} \left\| \frac{p(v_t)}{p_f(v_t)} - \frac{p(v_j)}{p_f(v_j)} \right\|$$

Generalized Cross-entropy

L'entropia incrociata generalizzata misura la differenza tra distribuzioni di probabilità. Valori più alti indicano maggiore fairness. La metrica è definita come:

$$GCE = \frac{1}{\alpha(1-\alpha)} \left[\sum_{v \in \mathcal{V}} p_f^{\alpha}(v) p^{(1-\alpha)}(v) - 1 \right]$$

dove α rappresenta un iperparametro di regolazione.

L1-norm

La norma L1 quantifica la distanza assoluta tra le distribuzioni osservate (p) e quelle ideali (p_f), utilizzabile sia a livello individuale che di gruppo. Un valore più basso indica un sistema più equo:

$$L1\text{-}norm = \sum_{v \in \mathcal{V}} |p(v) - p_f(v)|$$

La metrica è particolarmente efficace per valutare scostamenti globali tra allocazioni reali e ideali.

Capitolo 3

Metodi per Effettuare il Debiasing

3.1 Debiasing nella classificazione

Nel momento in cui si vuole lavorare sull'equità di un modello di classificazione, ci si può muovere lungo tre strade principali: pre-processing, in-processing e post-processing. Ognuna di queste opzioni cerca di ridurre il bias, ma lo fa in momenti diversi del flusso di lavoro. In generale, si parte dalla scelta di un concetto di fairness da rispettare, per poi adottare tecniche che aiutino a renderlo coerente.

3.1.1 Pre-processing

Il pre-processing si occupa direttamente dei dati che verranno dati in input all'algoritmo. Se i dati iniziali sono meno sbilanciati, allora anche il modello che li userà potrà essere più equo. Per esempio, si può modificare qualche etichetta, intervenire su alcune variabili, o trasformare il dataset in modo che le informazioni sensibili abbiano meno peso. Una tecnica piuttosto nota è il massaging, che modifica le etichette di alcuni soggetti con feature sensibili $S = s$ da "$-$" a "$+$", bilanciando contemporaneamente le etichette di un numero equivalente di soggetti con $S \neq s$ da "$+$" a "$-$", e il re-weighting, che assegna pesi diversi ai soggetti per ridurre la dipendenza. Entrambi i metodi risultano facili da implementare e flessibili, con il rischio di ridurre le prestazioni del modello a causa della modifica dei dati. Ideale quando il dataset è accessibile e si desidera adattare modelli esistenti senza grandi modifiche[9].

3.1.2 In-processing

Con l'in-processing, invece, si lavora all'interno del processo di addestramento del modello. Una tecnica piuttosto comune è l'introduzione di vincoli di fairness nella funzione obiettivo. Ad esempio, si può richiedere che la probabilità di classificazione positiva sia la stessa per tutti i gruppi, indipendentemente dall'attributo sensibile. Questo principio è noto come Statistical Parity[42]. Quest'ultima impone che la probabilità di assegnare una predizione positiva sia la stessa indipendentemente dall'attributo sensibile:

$$Pr(\hat{y} = 1 \mid s = 0) = Pr(\hat{y} = 1 \mid s = 1)$$

Per garantire l'equità, è possibile controllare la covarianza tra l'attributo sensibile s e la signed distance $d_\theta(x)$ rispetto al confine di decisione del classificatore. Poiché questa distanza determina il valore della predizione, l'obiettivo è renderla indipendente da s. La covarianza è definita come:

$$Cov_{SP}(s, d_\theta(x)) = E[(s - \bar{s})d_\theta(x)] - E[(s - \bar{s})]\bar{d_\theta(x)}$$

Affinché il modello soddisfi la parità statistica, la covarianza empirica stimata sui dati di addestramento deve essere approssimativamente zero. Il problema di ottimizzazione finale è formulato come segue:

$$\min_\theta L(\theta)$$

soggetto a:

$$\frac{1}{N} \sum_{(x,s)\in D_T} (s - \bar{s}) d_\theta(x) \leq c$$

$$\frac{1}{N} \sum_{(x,s)\in D_T} (s - \bar{s}) d_\theta(x) \geq -c$$

Dove $c \in \mathbb{R}^+$ è un valore soglia che consente di bilanciare l'accuratezza del modello con il rispetto dei vincoli di fairness. Per implementarlo, una strategia consiste nel monitorare la covarianza tra l'attributo sensibile e la distanza dell'istanza dal confine decisionale del classificatore. In pratica, se il classificatore tende a prendere decisioni diverse a seconda del gruppo di appartenenza, questa covarianza sarà alta. Obiettivo dell'ottimizzazione diventa quindi quello di minimizzarla, restando comunque entro una soglia predefinita. Questo tipo di vincolo forza una

relazione molto debole tra la decisione finale del modello e la variabile sensibile. Anche se può comportare un leggero calo in termini di accuratezza, diversi esperimenti dimostrano che riesce a migliorare significativamente l'equità delle classificazioni[42].

3.1.3 Post-processing

Il post-processing entra in gioco alla fine del processo, dopo che il modello è stato addestrato, cercando di correggere eventuali squilibri nelle sue decisioni. Una strategia possibile consiste nel modificare la soglia decisionale in modo differenziato per i vari gruppi, oppure nell'aggiustare il peso attribuito ad alcune feature che influenzano le predizioni[18]. Il metodo proposto è quello di usare un classificatore dotato di meccanismo di attenzione, che aiuta a capire quali variabili incidono di più sulla fairness. Se, ad esempio, una feature ha un impatto rilevante sull'equità ma non contribuisce molto all'accuratezza, si può decidere di ridurne l'importanza o persino di eliminarla. Questo tipo di approccio è molto flessibile, anche perché non richiede di modificare l'intero modello o di ripetere l'addestramento. È quindi adatto quando si lavora su modelli già pronti o su sistemi in produzione. Va però detto che, in termini di risultati, il post-processing tende ad essere meno efficace rispetto a interventi più profondi fatti prima o durante la fase di addestramento.

3.2 Debiasing nel Ranking

In questa sezione vedremo come viene affrontato il tema della fairness nei sistemi di ranking, utilizzando lo schema ormai consolidato che distingue tra pre-processing, in-processing e post-processing. Anche se questi approcci sono in parte simili a quelli già analizzati per la classificazione, nel contesto del ranking si nota una tendenza diversa: gran parte dei lavori si concentra soprattutto sulle fasi centrali e finali del processo, cioè durante l'addestramento del modello o subito dopo. Questo perché intervenire direttamente sulla classifica prodotta, o sulle regole che la generano, offre un controllo più mirato sulle disuguaglianze che possono emergere. Come vedremo, ogni approccio ha punti di forza e limiti, e la scelta dipende spesso dal tipo di dati e dagli obiettivi del sistema.

3.2.1 Pre-Processing

Un approccio comune consiste nel trasformare i dati in una rappresentazione low-rank che preservi a tutti gli effetti l'equità individuale[25]. Dati due campioni x_i e x_j, con $\tilde{x}_i$ e $\tilde{x}_j$ che rappresentano le loro caratteristiche non sensibili, si cerca una funzione di mappatura ϕ tale che:

$$|d\left(\phi(x_i), \phi(x_j)\right) - d\left(\tilde{x}_i, \tilde{x}_j\right)| \leq \epsilon$$

dove d è una misura di distanza e ϵ è un limite di tolleranza. Il problema viene formulato come un clustering probabilistico con K cluster, ciascuno rappresentato da un vettore prototipo v_k. Ogni campione x_i viene assegnato a un cluster con probabilità u_{ik}, che dipende dalla sua distanza dai prototipi. La rappresentazione trasformata $\tilde{x}_i$ è data da:

$$\tilde{x}_i = \phi(x_i) = \sum_{k=1}^{K} u_{ik} \cdot v_k$$

Per mantenere l'utilità dei dati, si minimizza la perdita di informazione:

$$L_{util}(X, \tilde{X}) = \sum_{i=1}^{M} \|x_i - \tilde{x}_i\|_2^2$$

Per garantire equità, si minimizza la differenza tra le distanze nelle rappresentazioni originali e trasformate:

$$L_{\text{fair}}(X, \tilde{X}) = \sum_{i,j=1}^{M} \left(d(\tilde{x}_i, \tilde{x}_j) - d(\tilde{x}_i, \tilde{x}_j)\right)^2$$

L'obiettivo finale combina i due termini con pesi λ e μ:

$$L = \lambda \cdot L_{util} + \mu \cdot L_{fair}$$

3.2.2 In-Processing

La maggior parte degli studi sul fair ranking adotta metodi di elaborazione in-process, i quali mirano a sviluppare direttamente un modello di ranking equo partendo da zero[29]. Questi metodi affrontano il problema della fairness pairwise, dove si considera un insieme di query S estratte da una distribuzione D, e ogni

candidato è associato a un vettore di feature $x \in X$ e un'etichetta di rilevanza $y \in Y$. L'algoritmo impara una funzione di punteggio $f : X \to \mathbb{R}$ per ordinare i candidati. Si definisce l'accuratezza pairwise dipendente dal gruppo $A_{G_i>G_j}$ come la probabilità che un candidato cliccato del gruppo G_i sia classificato sopra un candidato non cliccato ma rilevante del gruppo G_j:

$$A_{G_i>G_j} := P\left(f(x) > f(x') \mid y > y', (x, y) \in G_i, (x', y') \in G_j\right).$$

La fairness pairwise cross-group richiede che $A_{G_i>G_j} = \kappa$ per tutti i gruppi $i \neq j$, dove κ è una costante. Il problema effettivo consiste nel massimizzare l'accuratezza complessiva (AUC) sotto il vincolo di fairness:

$$\max_{f \in \mathcal{F}} \text{AUC}(f) \quad \text{s.t.} \quad |A_{G_i>G_j}(f) - A_{G_k>G_l}(f)| \leq \epsilon.$$

La ricerca mostra una riduzione delle violazioni di fairness, ma con una riduzione dell'AUC .

3.2.3 Post-Processing

Nel caso del post-processing, l'obiettivo è riordinare la lista prodotta dal ranking iniziale per migliorare l'equità, senza modificare il modello originale. Questo approccio è noto anche come re-ranking fairness[34]. Dato un insieme di documenti $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ associati a una query q, si parte da una classifica prodotta dal modello e si cerca di riassegnare le posizioni in modo da distribuire in modo equo l'esposizione, cioè la visibilità data ai vari documenti.

L'esposizione di un documento d_i dipende dalla probabilità $\mathbf{P}_{ij}$ di essere mostrato nella posizione j, ponderata per l'importanza della posizione $\mathbf{v}_j$:

$$\text{Exposure}(d_i|\mathbf{P}) = \sum_{j=1}^{N} \mathbf{P}_{ij} \cdot \mathbf{v}_j$$

Per ciascun gruppo G_k, si calcola l'esposizione media dei suoi documenti:

$$\text{Exposure}(G_k|\mathbf{P}) = \frac{1}{|G_k|} \sum_{d_i \in G_k} \text{Exposure}(d_i|\mathbf{P})$$

Il vincolo di parità statistica impone che tutti i gruppi ricevano la stessa esposizione:

$$\text{Exposure}(G_k|\mathbf{P}) = \text{Exposure}(G_{k'}|\mathbf{P})$$

Nel concreto, questo significa che i documenti di tutti i gruppi devono avere le stesse possibilità di essere visualizzati in alto nella classifica. Gli esperimenti mostrano che questo tipo di correzione riduce in modo efficace le disuguaglianze tra gruppi, anche se può comportare un piccolo compromesso in termini di accuratezza predittiva.

3.3 Debiasing nei Recommender System

La ricerca sulla Fair Reccomendation si concentra sul definire metriche di equità e sviluppare tecniche appropriate per promuovere tali metriche. Le tecniche utilizzate sono varie e possono differire significativamente a seconda delle definizioni di equità adottate. In questa sezione, si propone una categorizzazione, evidenziando alcune tecniche tipiche e comunemente utilizzate, al fine di facilitare la comprensione di come l'equità venga tecnicamente raggiunta nei sistemi di raccomandazione.

3.3.1 Regolarizzazione e Ottimizzazione Vincolata

Le tecniche per promuovere l'equità nei sistemi di raccomandazione si basano principalmente su metodi di regolarizzazione e ottimizzazione vincolata, che possono essere applicati sia in-processing che post-processing. I criteri di equità possono essere formulati come regolatori o vincoli per guidare il processo di ottimizzazione del modello. Gli obiettivi possono includere: massimizzazione dell'utilità delle raccomandazioni sotto vincoli di equità,massimizzazione dei requisiti di equità entro limiti di utilità, o l'ottimizzazione congiunta di entrambi gli obiettivi con un compromesso ragionevole. Un esempio significativo è la proposta di quattro nuove metriche di equità per la previsione delle preferenze nei sistemi di raccomandazione basati su collaborative filtering. Queste metriche misurano la discrepanza nella qualità delle previsioni tra utenti svantaggiati e avvantaggiati, divisi in base a caratteristiche binarie come il genere[41]. Le metriche confrontano le valutazioni medie e i punteggi predetti per ciascun item tra i due gruppi. Alcuni approcci misurano la Fairness nei sistemi di raccomandazione attraverso confronti a coppie, utilizzando

regolatori che incentivano il miglioramento della metrica durante l'addestramento del modello, aumentando così l'equità nei ranking generati. Nel contesto delle raccomandazioni di gruppo, dove gli item sono suggeriti a utenti con preferenze diverse, vengono proposte metriche specifiche per garantire un trattamento equo tra i membri del gruppo. Questo problema viene affrontato tramite ottimizzazione multi-obiettivo, in cui l'equità è trattata come un regolatore e risolta secondo i principi dell'Efficienza Paretiana. [1]

3.3.2 Adversary Learning

L'adversary learning è una tecnica basata su un gioco min-max tra il predittore principale e un classificatore avversario[2]. L'obiettivo dell'attaccante è dedurre informazioni relative agli attributi privati degli utenti, mentre il sistema di raccomandazione si propone di apprendere gli interessi degli utenti sotto la regolarizzazione imposta dall'attaccante. Il metodo proposto scompone l'embedding dell'utente in due parti: un'embedding bias-aware, che cattura le informazioni relative ai bias legati alle caratteristiche sensibili e un'embedding bias-free, privo di pregiudizio per effettuare un ranking delle notizie consapevole della fairness. L'apprendimento avversario viene utilizzato per rimuovere le informazioni, che creano pregiudizio, dalle caratteristiche sensibili degli utenti, promuovendo una raccomandazione più equa.

3.3.3 Reinforcement learning

Alcune ricerche sostengono che la raccomandazione non sia solo un problema di previsione, ma anche un problema decisionale sequenziale, e suggeriscono di modellare il problema di raccomandazione come un Processo Decisionale di Markov, risolvendolo attraverso tecniche di reinforcement learning [17]. L'equità dinamica nell'esposizione degli item è stata studiata considerando come la popolarità degli

[1] **Efficienza Paretiana**: Condizione in cui non è possibile migliorare il benessere di un individuo senza peggiorarne quello di un altro. In contesti di ottimizzazione multi-obiettivo, come nei sistemi di raccomandazione, una soluzione è Pareto-efficiente quando non esiste un'altra soluzione che migliori un obiettivo senza comprometterne un altro.

item cambi nel tempo in base alle raccomandazioni e al feedback degli utenti, formulando il problema come un Processo Decisionale di Markov Vincolato. Un altro approccio utilizza l'apprendimento per rinforzo multi-obiettivo per bilanciare equità e utilità, apprendendo politiche di raccomandazione ottimali. Nei sistemi interattivi, le preferenze degli utenti e lo stato di equità, che evolvono nel tempo, sono integrati negli stati del modello MDP attraverso un framework RL per garantire equità a lungo termine in modo dinamico. Una delle sfide principali nell'utilizzo di tale metodologia è il rischio di "manipolazione dell'utente", un fenomeno in cui un sistema di raccomandazione potrebbe cercare di aumentare il coinvolgimento degli utenti a lungo termine manipolando le loro opinioni, preferenze e convinzioni attraverso le raccomandazioni. La potenziale ingiustizia, derivante dal fatto che diversi utenti sono influenzati da manipolazioni diverse, richiede particolare attenzione da parte dei ricercatori e degli sviluppatori di sistemi.

3.3.4 Metodi Causali

L'obiettivo principale di questi metodi è indagare le relazioni sottostanti i dati e il modello, inclusi gli effetti causali tra variabili sensibili e decisioni, nonché le dipendenze tra variabili sensibili e non sensibili. Per affrontare il problema dell'equità nei reccomender system i metodi causali sfruttano diverse strategie. Un approccio utilizza grafi causali per identificare e rimuovere discriminazioni dirette e indirette, ricostruendo classifiche eque[24, 44]. Un altro studio, invece, separa l'interesse dell'utente dalla popolarità degli item utilizzando due tipi di embedding, generando raccomandazioni basate solo sull'interesse eliminando il bias di popolarità. Un ulteriore metodo incorpora l'inferenza causale nei bandit, sviluppando algoritmi equi che soddisfano vincoli di equità controfattuale. Infine, le tecniche di Inverse-Propensity-Scoring correggono bias come quelli di popolarità o selezione riponderando le istanze, seppur nonostante possano avere limiti nel gestire grandi variazioni nelle probabilità osservazionali. Generalmente i metodi causali presentano problemi pratici, come la definizione di nozioni basate su interventi e controfattuali, che sono quantità non osservabili e non sempre calcolabili da dati osservazionali.

3.3.5 Altri Metodi

Diverse altre tecniche sono utilizzate dalla ricerca per promuovere la fairness nelle raccomandazioni [32, 7, 40]. Di seguito alcuni esempi :

- **Metodi di data augmentation**: Propone una strategia per migliorare proprietà socialmente rilevanti, come l'equità individuale o di gruppo, aggiungendo dati "antidoto" a un sistema di raccomandazione pre-addestrato basato su fattorizzazione di matrici. Questo approccio non richiede la modifica dei dati di input originali o dell'algoritmo di raccomandazione, ma aggiunge valutazioni di nuovi utenti selezionati per migliorare l'equità.

- **Metodi basati su Variational Autoencoders (VAE)**: Introduce casualità nel funzionamento dei VAE per mitigare il bias di posizione in più round di raccomandazione. Vengono proposte quattro diverse distribuzioni di rumore, dimostrando che l'aggiunta di rumore durante il campionamento dei valori dalla rappresentazione latente del VAE può garantire equità a lungo termine, con un compromesso accettabile tra equità e qualità delle raccomandazioni.

- **Metodi basati su self-distillation**: Utilizzano le previsioni del modello sui dati originali come "insegnante" per regolarizzare le previsioni su dati aumentati con comportamenti utente casualmente rimossi. Questo approccio affronta il problema delle prestazioni ingiuste per i cold users, dimostrando che la self-distillation può aiutare il modello a catturare equamente le distribuzioni di interesse sia degli utenti attivi che di quelli "cold".

Capitolo 4

Algoritmi Utilizzati Nella Ricerca

In questo capitolo introdurremo i quattro algoritmi principali impiegati nella fase sperimentale del progetto di ricerca: FairGo, NFCF (Neural Fair Collaborative Filtering), FOCF (Fairness Objectives for Collaborative Filtering) e PFCN (Personalized Fairness based on Causal Notion). Si tratta di algoritmi sviluppati per integrare meccanismi di controllo del bias nei sistemi di raccomandazione, ciascuno con un approccio differente in termini di architettura, strategia di addestramento e tipo di fairness perseguita.
FairGo sfrutta la struttura del grafo utente-item per apprendere rappresentazioni fair attraverso un sistema di filtri e apprendimento avversario[39];
NFCF (Neural Fair Collaborative Filtering) basato su una fase di pre-training su dati non sensibili e una fase di fine-tuning con tecniche di debiasing e penalizzazione della disparità;
FOCF (Fairness Objectives for Collaborative Filtering), introduce direttamente nella funzione obiettivo del modello termini specifici per ridurre diverse forme di unfairness;
PFCN (Personalized Fairness based on Causal Notion), infine, adotta un approccio controfattuale per garantire che gli attributi sensibili non influenzino le raccomandazioni.
I paragrafi seguenti sono dedicati all'analisi di ciascun modello, con l'obiettivo di illustrarne la logica, l'architettura, le componenti principali e le tecniche adottate per ridurre o controllare l'influenza degli attributi sensibili.

4.1 FairGo

4.1.1 Architettura generale

FairGo è un modello che introduce la fairness personalizzata nei sistemi di raccomandazione, partendo dal concetto che le raccomandazioni non dovrebbero dipendere da attributi sensibili come genere, età o professione. Per farlo, sfrutta la struttura a grafo insita nei dati di interazione utente-item e agisce direttamente sulle loro proiezioni vettoriali, trasformandole in rappresentazioni da cui tali attributi non siano più facilmente ricostruibili [39]. Il modello è progettato per essere indipendente dall'algoritmo di base, quindi può essere applicato a diversi sistemi senza richiedere modifiche profonde. A partire dalle embedding generate, FairGo apprende dei filtri associati agli attributi sensibili, con l'obiettivo di rimuoverne l'influenza residua. L'elemento principale è un meccanismo di Adversary learning: per ogni attributo da proteggere, un discriminatore cerca di predirlo a partire dall'embedding filtrato, mentre i filtri cercano di impedirglielo. Si instaura così un equilibrio tra queste due forze, che porta a rappresentazioni più neutrali rispetto agli attributi sensibili, senza sacrificare la qualità delle raccomandazioni.

4.1.2 Struttura multi-livello e funzione obiettivo

Una delle innovazioni più interessanti introdotte da FairGo è la modellazione della fairness su due livelli distinti. Il primo è la node-level fairness: qui si cerca di garantire che l'embedding finale di ciascun utente, dopo il filtraggio, non contenga più informazioni rilevanti sull'attributo sensibile [39]. Il secondo livello è invece la ego-centric fairness, che si concentra sulla struttura locale del grafo, cioè sui nodi vicini all'utente. Queste informazioni, ovvero le caratteristiche aggregate del vicinato, non dovrebbero permettere di inferire attributi sensibili. Questo aspetto è particolarmente importante nei sistemi di tipo collaborative filtering, dove le raccomandazioni si basano proprio sulle somiglianze tra utenti. La funzione obiettivo del modello riflette questo schema, combinando le due componenti in un'unica formulazione avversaria:

$$\max_{F} \min_{D} \; \mathbb{E}\left[\log q_R(r \mid f_u, f_v) - \lambda \log q_D(x_u \mid f_u, p_u)\right]$$

In questa equazione, f_u e f_v sono gli embedding filtrati dell'utente e dell'item, mentre p_u rappresenta la struttura ego-centrica dell'utente. Le funzioni q_R e q_D indicano rispettivamente la predizione del rating e quella dell'attributo sensibile. Il parametro λ controlla quanto peso dare alla fairness rispetto all'accuratezza. Per costruire p_u, cioè la rappresentazione ego-centrica, si può usare un meccanismo di aggregazione dei nodi vicini. Questo può essere fatto in modo manuale, ad esempio calcolando una media pesata sui vicini di diversi livelli (con pesi λ_l), oppure in maniera più sofisticata tramite una rete neurale che combina queste rappresentazioni in modo non lineare. Questo passaggio è fondamentale per rendere equa anche la struttura locale dell'utente, non solo il suo embedding diretto [39].

4.1.3 Penalizzazione della fairness e risultati

Per valutare la qualità del modello in termini di equità, FairGo utilizza due metriche principali:

- **Differential Fairness (DF)**: mira a garantire che le raccomandazioni non cambino significativamente tra gruppi demografici diversi;
- **Absolute Unfairness (Uabs)**: misura quanto l'errore medio nelle predizioni differisce tra gruppi svantaggiati e avvantaggiati.

In conclusione, FairGo si dimostra un approccio solido ed efficace per affrontare il problema della raccomandazione equa, specialmente in contesti basati su dati reali, e ha il vantaggio di poter essere integrato facilmente anche in sistemi già esistenti [39].

4.2 PFCN

4.2.1 Architettura generale

PFCN introduce il concetto di fairness adottando una prospettiva controfattuale che permette di analizzare l'influenza degli attributi sensibili. Alla base c'è un'intuizione: un sistema di raccomandazione equo dovrebbe generare gli stessi risultati per un utente anche nel caso in cui si modifichi uno dei suoi attributi sensibili, lasciando inalterate tutte le altre informazioni. Per realizzare questo principio,

PFCN interviene direttamente sulle embedding degli utenti, con l'obiettivo di eliminare l'effetto degli attributi sensibili attraverso un meccanismo di apprendimento avversario. L'algoritmo è progettato in modo da essere indipendente dal tipo di algoritmo di base, quindi può essere facilmente applicato ad architetture diverse come PMF, BiasedMF, Wide&Deep e DMF[19].

4.2.2 Fairness controfattuale e apprendimento avversario

Il funzionamento di PFCN si ispira alla cosiddetta fairness controfattuale; in pratica, si prende un utente u, con un certo attributo sensibile a_u e relativa embedding p_u, e si immagina un utente gemello u' identico in tutto, tranne che in quell'attributo. L'obiettivo è che le raccomandazioni generate per u e u' – cioè $\hat{y}_u$ e $\hat{y}_{u'}$ – siano il più simili possibile. Per ottenere questo risultato, PFCN utilizza una rete neurale di tipo filtro, indicata con F, che trasforma l'embedding originale p_u in una versione modificata $\tilde{p}_u = F(p_u)$, idealmente priva di tracce dell'attributo sensibile. L'addestramento di questa trasformazione avviene con un Adversary Learning: da una parte, un discriminatore cerca di indovinare l'attributo sensibile a partire da $\tilde{p}_u$; dall'altra, il filtro F viene ottimizzato per rendere questa predizione il più difficile possibile. La funzione di perdita tiene quindi conto di due obiettivi opposti, ovvero, da un lato mantenere la capacità del sistema di fare buone raccomandazioni, dall'altro ridurre il più possibile l'informazione sull'attributo protetto [19]. E' importante notare che questa procedura viene applicata durante una fase di preprocessing: una volta addestrato il filtro, le embedding trasformate vengono poi usate per addestrare il modello di raccomandazione vero e proprio. Questo rende PFCN compatibile anche con architetture più complesse, senza richiedere modifiche dirette al modello finale.

4.2.3 Applicazione su modelli e risultati ottenuti

L'algoritmo è stato testato in combinazione con diversi modelli di base, tra cui PMF, BiasedMF, Wide&Deep e DMF. I risultati ottenuti sono stati molto promettenti rispetto sia agli approcci standard che ad altri metodi orientati alla fairness, come FOCF, PFCN ha mostrato prestazioni mediamente superiori dal punto di vista dell'accuratezza, mantenendo allo stesso tempo alti livelli di equità. Un altro

punto a favore è che PFCN non richiede modifiche all'algoritmo di raccomandazione sottostante, risultando quindi facilmente integrabile anche in sistemi già esistenti [19].

4.3 NFCF

4.3.1 Architettura generale

NFCF (Neural Fair Collaborative Filtering) è un algoritmo che nasce con l'obbiettivo di estendere i modelli neurali di collaborative filtering, cercando di ridurre il bias di genere nelle raccomandazioni, soprattutto quando si tratta di item sensibili come corsi universitari o carriere professionali. La sua struttura si basa sul classico approccio pre-training/fine-tuning, che permette di usare dati abbondanti ma non sensibili – ad esempio like, valutazioni su film, preferenze su blog, per imparare buone rappresentazioni utente, che poi vengono adattate in modo mirato per gestire le raccomandazioni più critiche dal punto di vista dell'equità [21]. L'architettura vera e propria riprende quella dei modelli NCF (Neural Collaborative Filtering): si usano le embedding dell'utente p_u e dell'item q_i, che vengono concatenate e passate attraverso una rete neurale multilivello. Questa rete applica trasformazioni non lineari e produce, in uscita, una stima della probabilità che l'utente interagisca con l'item in questione:

$$\hat{y}_{ui} = \sigma(h^\top \phi_L(z_{L-1}))$$

dove ϕ_L rappresenta la sequenza di attivazioni nei vari strati della rete [21].

4.3.2 Pre-training, debiasing e fine-tuning

L'addestramento avviene in tre fasi, ognuna con un obiettivo specifico. La prima è il pre-training: qui il modello viene addestrato su un insieme di dati non sensibili D_n, usando una log-loss standard. Questa fase serve a costruire embedding utente di buona qualità, ma non è priva di rischi: dato che le preferenze possono essere correlate con il genere, si rischia di incorporare involontariamente dei bias. Per affrontare questo problema, nella seconda fase, il debiasing, viene stimato un vettore di bias $\vec{v}_B$, calcolato come la differenza media tra le embedding di utenti maschi e femmine. Una volta ottenuto, si procede a rimuovere da ogni embedding la

componente associata a questo vettore, con una proiezione ortogonale:

$$p'_u = p_u - (p_u \cdot \vec{v}_B)\vec{v}_B$$

Questo procedimento è ispirato a tecniche simili usate per ripulire i word embeddings da tratti sensibili, che potrebbero risultare discriminanti in seguito. Infine, nella terza ed ultima fase, il fine-tuning, si riparte dalle embedding debiased e dai pesi pre-addestrati per allenare un nuovo modello, stavolta sui dati sensibili D_s. In questa fase si usa una funzione di perdita arricchita con un termine legato alla fairness, pensato per bilanciare accuratezza e equità:

$$\min_W [L(W) + \lambda R(\varepsilon_{\text{mean}})]$$

Qui, $\varepsilon_{\text{mean}}$ misura quanto la fairness varia tra gruppi diversi, e λ serve per regolare quanto peso dare a questa penalizzazione [21].

4.3.3 Penalizzazione della fairness e risultati

Per valutare la fairness ottenuta, NFCF impiega due metriche principali:

- Differential Fairness: serve a verificare che la distribuzione delle raccomandazioni sia simile tra gruppi diversi, senza grandi variazioni dovute all'appartenenza demografica;
- Absolute Unfairness: misura la differenza media dell'errore di predizione tra utenti svantaggiati e avvantaggiati.

Secondo quanto riportato dagli autori durante la ricerca [21], sia il debiasing che la penalizzazione sono fondamentali: eliminarne anche solo una peggiora sensibilmente le metriche di equità. L'algoritmo si rivela quindi un approccio pratico ed efficace per trattare la raccomandazione equa in contesti reali.

4.4 FOCF

4.4.1 Architettura generale

FOCF è un algoritmo sviluppato con l'intento di integrare aspetti legati alla fairness direttamente nella fase di addestramento dei modelli di collaborative filtering.

L'idea è quella di lavorare direttamente sulla funzione obiettivo, aggiungendo termini che penalizzano eventuali disuguaglianze nelle predizioni, invece di modificare l'architettura del modello avendo il vantaggio di poter essere applicato senza stravolgere la struttura già esistente. Tra l'altro, FOCF è compatibile con diversi tipi di architetture di raccomandazione, ad esempio Matrix Factorization, NCF, DMF o Wide&Deep, e si integra bene con librerie già esistenti come RecBole-FairRec. Il suo scopo è quello di trovare un equilibrio tra accuratezza e fairness, cosa non banale, dando comunque all'utente la possibilità di scegliere quali metriche di equità usare, e con che peso [19].

4.4.2 Obiettivi di fairness e integrazione nei modelli

La discriminante di FOCF è l'uso di obiettivi di fairness sotto forma di regolatori da aggiungere alla loss. In sostanza, si possono inserire penalizzazioni che misurano forme diverse di unfairness. Le principali sono:

- value unfairness
- absolute unfairness
- under- and over-unfairness
- non-parity unfairness

Le metriche possono essere usate da sole oppure combinate. La funzione di perdita risultante ha la struttura seguente:

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \sum_i \lambda_i \cdot \mathcal{R}_i \tag{4.1}$$

dove $\mathcal{L}_{\text{base}}$ è la loss del modello standard, mentre i vari $\mathcal{R}_i$ rappresentano i termini di unfairness scelti. I coefficienti λ_i permettono di regolare l'importanza di ciascuno. Questo rende FOCF piuttosto flessibile e adatto a contesti diversi [19].

Capitolo 5

Studio del trade-off tra fairness, accuratezza e sostenibilità ambientale

Questo capitolo analizza lo studio sperimentale condotto con l'obiettivo di analizzare il trade-off tra accuratezza , fairness e sostenibilità ambientale nei Recommender System. L'analisi si concentra non solo sulle prestazioni dei modelli selezionati, ma anche sull'impatto che diverse configurazioni possono avere sulle tre dimensioni in esame. L'indagine è guidata dalle seguenti domande di ricerca:

- **RQ1**: È possibile delineare un trade-off tra accuratezza, fairness e sostenibilità ambientale nei RS?
- **RQ2**: Qual è l'impatto del training nei Recommender System su fairness e sostenibilità ambientale?
- **RQ3**: Qual è l'impatto del training nei Recommender System su accuratezza e sostenibilità ambientale?

Per rispondere a queste domande, sono stati eseguiti esperimenti su diversi algoritmi fairness-aware ,che hanno rispettivamente un diverso approccio alla fairness, utilizzando il framework RecBole-FairRec e integrando il monitoraggio dei consumi energetici mediante la libreria CodeCarbon.

5.1 Dataset - Movielens_1M

Gli esperimenti sono stati condotti sul dataset MovieLens_1M, utilizzato ampiamente nella letteratura sui sistemi di raccomandazione. MovieLens_1M contiene circa un milione di valutazioni espresse da 6.040 utenti su 3.706 film. L'attributo sensibile considerato è il gender (genere) dell'utente, variabile binaria (maschio/femmina), che viene utilizzata per valutare l'equità delle raccomandazioni fornite.

Caratteristica	**Valore**
Numero di utenti	6.040
Numero di item	3.706
Numero di valutazioni	1.000.209
Valori di rating	Da 1 a 5
Attributo sensibile	Gender (maschio/femmina)
Percentuale maschi	71.4%
Percentuale femmine	28.6%

Tabella 5.1: Statistiche principali del MovieLens_1M

5.2 Metriche di valutazione utilizzate

Per valutare le prestazioni dei modelli considerati abbiamo utilizzato un insieme di metriche articolato sulle tre dimensioni interessate:

Accuratezza

- **NDCG@10** (Normalized Discounted Cumulative Gain): misura la qualità del ranking delle raccomandazioni, penalizzando le posizioni errate nella top-10.

Fairness

- Differential Fairness: valuta la disparità di trattamento tra gruppi.
- Value Unfairness, Non-Parity Unfairness, Underestimation Unfairness: misurano diversi aspetti dell'iniquità predittiva tra gruppi, come lo scostamento

nella sovra/sottostima delle valutazioni attese, utilizzate per il modello FOCF che non implementa la differential fairness.

Sostenibilità ambientale

- **Emissioni di CO_2**: stimate in grammi attraverso CodeCarbon [**?**], che calcola l'energia utilizzata in base al consumo della GPU e al mix energetico locale.

Queste metriche, combinate, permettono di valutare l'effetto delle scelte di configurazione sul comportamento complessivo dei modelli. Nei paragrafi successivi, presentiamo l'analisi dettagliata dei framework e delle librerie utilizzate nell'analisi e i risultati ottenuti per ciascun algoritmo testato, confrontando le configurazioni focalizzate su accuratezza, fairness o entrambe, in relazione all'impatto ambientale.

5.3 Relazione tra Fairness e Produzione di CO_2

5.3.1 Complessità computazionale della fairness

Quando si parla di fairness nei sistemi di raccomandazione, si pensa giustamente agli aspetti affrontati in questa tesi: evitare discriminazioni, trattare gli utenti in modo equo, correggere i bias nei dati. Ma dietro queste buone intenzioni, si nasconde anche un lato meno visibile, quello computazionale. Rendere un algoritmo più equo richiede modelli più sofisticati, addestramenti più lunghi ed esperimenti ripetuti innumerevoli per trovare la giusta combinazione di parametri. Tutta questa complessità ha un prezzo in termini di energia consumata, e non si tratta solo di tempo di calcolo, ma anche di potenza elettrica, server attivi per ore, e conseguenti emissioni di anidride carbonica. In altre parole, l'equità ha un costo soprattutto ambientale.

5.4 Framework RecBole-FairRec

Per condurre gli esperimenti di questa ricerca, è stato utilizzato il framework RecBole-FairRec, estensione del framework RecBole pensata specificamente per l'analisi della fairness nei sistemi di raccomandazione.

5.4.1 RecBole

RecBole è una libreria modulare e altamente estensibile, sviluppata in PyTorch, che fornisce un ambiente unificato per la sperimentazione di algoritmi di raccomandazione. Supporta numerosi modelli state-of-the-art e consente una facile configurazione degli esperimenti tramite file YAML. La sua architettura modulare permette di separare chiaramente modelli, ottimizzatori, schemi di valutazione e pre-processing dei dati [38].

5.4.2 FairRec: focus sulla fairness

La versione FairRec introduce nel framework una serie di modelli fairness-aware, loss function specifiche, metriche di equità e configurazioni pensate per analizzare l'impatto di fattori sensibili come genere, etnia o ruolo. La versione modificata FairRec è disponibile pubblicamente su GitHub [36].

5.4.3 Vantaggi per la sperimentazione

L'utilizzo di RecBole-FairRec ha permesso di confrontare in modo sistematico modelli standard e fairness-aware, mantenendo costanti dataset, metriche e pipeline. Inoltre, la possibilità di settare l'ambiente su GPU ha reso compatibile il framework con strumenti di tracciamento energetico come CodeCarbon.

5.5 CodeCarbon

Rimarcando il concetto di una società sempre più attenta alle emissioni, il progetto CodeCarbon [20] si propone come strumento open-source progettato per misurare e monitorare le emissioni di CO_2 generate dall'esecuzione di codice; questo è particolarmente rilevante nei compiti che richiedono alta intensità di calcolo, tra cui vi sono anche gli addestramenti di modelli di machine learning, dove il dispendio energetico può essere significativo. Il calcolo delle emissioni si basa sulla seguente formula:

$$\text{Emissioni} = \text{Energia consumata} \times \text{Intensità di carbonio locale}$$

L'intensità di carbonio viene determinata a partire da dati pubblici relativi al mix energetico nazionale o regionale, rendendo il calcolo attendibile [20].

5.5.1 Funzionalità tecniche

CodeCarbon è in grado di stimare il consumo energetico delle principali risorse hardware, nel mio caso una GPU NVIDIA, tramite la libreria python pynvml. Al termine dell'esecuzione, i dati vengono salvati in un file tabellare .csv contenente informazioni dettagliate su consumi ed emissioni.

5.5.2 Obbiettivo etico

L'obiettivo centrale di CodeCarbon è quello di fornire strumenti concreti a sviluppatori e ricercatori per quantificare, confrontare e ridurre l'impatto ambientale delle proprie attività computazionali. Promuove così una programmazione consapevole e sostenibile, in linea con la filosofia del progetto: conoscere le proprie emissioni è nell'effettivo il primo passo per poterle ridurre[20].

5.6 Configurazione degli addestramenti

Per la valutazione sperimentale degli algoritmi fairness-aware è stato definito un protocollo di configurazione e addestramento, basato sull'uso di file di configurazione in formato YAML e sul monitoraggio integrato di prestazioni ed emissioni ambientali. L'intero processo è stato realizzato utilizzando il framework RecBole-FairRec, che consente la gestione modulare di dataset, modelli, e metriche, e i consumi energetici e le emissioni di CO_2, tramite la libreria CodeCarbon. Al fine di garantire un tracciamento accurato delle emissioni, il codice di lancio degli esperimenti è stato modificato per includere l'invocazione esplicita dei metodi "emissions_tracker.start()" e "emissions_tracker.stop()". I parametri utilizzati per ciascun modello sono riportati in dettaglio nella tabella 5.2, che sintetizza le configurazioni adottate per tutti gli esperimenti effettuati.

Protocollo di Addestramento

Il protocollo seguito per ogni modello è stato strutturato in tre fasi principali:

1. Definizione del file YAML: ogni esperimento è stato descritto da un file .yaml, contenente i parametri specifici dell'algoritmo, le impostazioni di addestramento e gli obiettivi di fairness.

2. Esecuzione controllata del training: ogni modello è stato addestrato per 100 epoche su GPU, mantenendo costanti i parametri comuni, variando soltanto i pesi delle penalizzazioni o i parametri legati alla fairness.

3. Raccolta e analisi dei risultati: al termine di ciascun addestramento, sono stati salvati i log delle metriche di accuratezza , fairness e i consumi stimati tramite CodeCarbon e organizzati in file .xlsx per una visone globale. I dati così raccolti sono stati successivamente analizzati per confrontare le performance dei modelli.

5.6.1 Parametri generali

Al fine di garantire un equilibrio tra qualità dell'addestramento, tempi computazionali e impatto ambientale, il numero di epoche è stato fissato a 100 per tutti i modelli. Questo valore è stato scelto in quanto sufficiente per permettere la convergenza nella maggior parte delle configurazioni senza introdurre overfitting o aumenti ingiustificati di emissioni. Anche gli altri iperparametri sono stati selezionati secondo criteri di stabilità ed efficienza di calcolo: una dimensione di embedding pari a 128 ha offerto un buon compromesso tra capacità espressiva e complessità; un batch size di 2048 ha garantito rapidità di calcolo su GPU, mentre un learning rate pari a 0.001 ha assicurato una discesa stabile della funzione di perdita durante l'ottimizzazione.

Algoritmo	fair_weight	vs_weight	dis_weight	Fair_objective
Fairgo	1.00E-04	[4,4]	—	—
Fairgo	1.00E-05	[1,1]	—	—
Fairgo	0.0	[1,1]	—	—
Fairgo	0.0	[1,50]	—	—
Fairgo	100.0	[1,4]	—	—
Fairgo	100.0	[1,1]	—	—
Fairgo	100.0	[1,50]	—	—
Fairgo	1000.0	[1,1]	—	—
Fairgo	1000.0	[1,50]	—	—
NFCF	0.0	—	—	—
NFCF	0.01	—	—	—
NFCF	0.1	—	—	—
NFCF	1.0	—	—	—
NFCF	100.0	—	—	—
PFCN	—	—	0	—
PFCN	—	—	1.0	—
PFCN	—	—	100	—
PFCN	—	—	1000	—
FOCF	0.0	—	—	nonparity
FOCF	0.0001	—	—	nonparity
FOCF	0.001	—	—	nonparity
FOCF	0.01	—	—	nonparity
FOCF	0.1	—	—	nonparity
FOCF	1.0	—	—	nonparity
FOCF	10.0	—	—	nonparity
FOCF	0.0	—	—	under
FOCF	0.0001	—	—	under
FOCF	0.001	—	—	under
FOCF	0.01	—	—	under
FOCF	0.1	—	—	under
FOCF	1.0	—	—	under
FOCF	10.0	—	—	under
FOCF	0.0	—	—	value
FOCF	0.0001	—	—	value
FOCF	0.001	—	—	value
FOCF	0.01	—	—	value
FOCF	0.1	—	—	value
FOCF	1.0	—	—	value
FOCF	10.0	—	—	value

Tabella 5.2: Pesi usati per ciascun algoritmo di raccomandazione

5.6.2 Esempi di file YAML

FairGo

FairGo richiede la modifica di due parametri chiave: vs_weight, per bilanciare il peso dei nodi più o meno vicini all'utente in base ai suoi embeddings e fair_weight, come peso globale della penalizzazione.

```
# fairgo_config.yaml
model: FairGo_GCN
dataset: movielens-1m
learning_rate: 0.001
epoch: 100
vs_weight: [1,1]\[1,50]\[4,4]
fair_weight: 100.0
```

NFCF

Per NFCF si è variato fair_weight durante la fase di fine-tuning, mantenendo fissi gli altri parametri.

```
# nfcf_config.yaml
model: NFCF
dataset: movielens-1m
fair_weight: 1.0
epoch: 100
```

FOCF

FOCF consente di specificare sia il peso della fairness tramite fair_weight che la tipologia di metrica da applicare tramite fair_objective.

```
# focf.yaml
model: FOCF
dataset: movielens-1m
fair_objective: value/under/nonparity
fair_weight: 10.0
```

PFCN

Nel caso di PFCN è necessario definire dis_weight, che regola l'intensità dell'apprendimento avversario, più è alto dis_weight, più il sistema si impegna a nascondere l'attributo sensibile dalle rappresentazioni ed evitare quindi penalizzazioni legate a quest'ultimo.

```
# pfcn_dmf.yaml
model: PFCN_DMF
dataset: movielens-1m
dis_weight: 100.0
epoch: 100
```

5.7 Analisi e discussione dei risultati ottenuti

Per ciascun modello sono stati condotti esperimenti variando i parametri di regolarizzazione della fairness, con l'obiettivo di osservare come la modifica dell'importanza attribuita alla fairness influenzi le prestazioni complessive del sistema. I risultati numerici sono organizzati in tabelle, che riportano, per ogni configurazione, i valori delle metriche sopra citate. A supporto dell'analisi, tali risultati sono stati ulteriormente elaborati e rappresentati tramite:

- Heatmap, impiegate per visualizzare l'interazione tra fairness, accuratezza ed emissioni in modo compatto e comparabile;

- Scatter plot, utilizzati per evidenziare correlazioni e compromessi (trade-off) tra le metriche, nonché per confrontare le dinamiche dei diversi modelli.

L'integrazione tra rappresentazione tabellare e visuale consente di cogliere con maggiore immediatezza i comportamenti emergenti, i punti di svolta tra le variabili analizzate e le configurazioni che rappresentano potenziali compromessi ottimali tra accuratezza, equità e sostenibilità.

fair_weight	vs_weight	emissions (g)	ndcg@10	Differential Fairness
0.0001	[4,4]	133.705	0.1554	15.05
0.00001	[1,1]	138.756	0.1562	15.029
0.0	[1,1]	140.350	0.1590	15.033
0.0	[1,50]	146.731	0.1608	15.025
100.0	[1,4]	133.343	0.1116	22.771
100.0	[1,1]	132.729	0.1263	22.514
100.0	[1,50]	137.660	0.0799	16.206
1000.0	[1,1]	137.259	0.0850	15.769
1000.0	[1,50]	137.518	0.1094	15.947

Tabella 5.3: FairGo

fair_weight	emissions (g)	ndcg@10	Differential Fairness
0.00	10.063	0.2962	14.504
0.01	18.868	0.2994	14.506
0.10	21.777	0.3000	14.489
100.00	10.245	0.1035	14.459
1.00	10.157	0.1035	14.459

Tabella 5.4: NFCF

dis_weight	emissions (g)	ndcg@10	Differential Fairness
0.0	15.060	0.2320	14.597
1.0	12.902	0.2315	14.432
100.0	8.898	0.1777	14.456
1000.0	12.253	0.2208	14.475

Tabella 5.5: PFCN

emissions (g)	**fair_objective**	**fair_weight**	**ndcg@10**	**Value Unfairness**
13.6472	value	0.0000	0.1520	0.1293
12.0382	value	0.0001	0.1520	0.1293
12.0294	value	0.0010	0.1520	0.1293
12.0610	value	0.0100	0.1520	0.1293
11.9500	value	0.1000	0.1520	0.1295
20.2055	value	1.0000	0.1628	0.1299
17.8940	value	10.0000	0.1601	0.1319

Tabella 5.6: FOCF – value unfairness

emissions (g)	fair_objective	fair_weight	ndcg@10	NonParity Unfairness
11.4417	nonparity	0.0000	0.1520	0.0006
10.9063	nonparity	0.0001	0.1520	0.0006
10.9251	nonparity	0.0010	0.1520	0.0006
10.9998	nonparity	0.0100	0.1520	0.0007
16.0778	nonparity	0.1000	0.1614	0.0031
11.8794	nonparity	1.0000	0.1601	0.0031
11.9854	nonparity	10.0000	0.1677	0.0021

Tabella 5.7: FOCF - nonparity unfairness

emissions (g)	fair_objective	fair_weight	ndcg@10	Underestimation Unfairness
12.3329	under	0.0000	0.1520	0.0365
12.9488	under	0.0001	0.1520	0.0365
12.2226	under	0.0010	0.1520	0.0365
12.3089	under	0.0100	0.1520	0.0364
12.1292	under	0.1000	0.1524	0.0361
11.7597	under	1.0000	0.1538	0.0351
13.1140	under	10.0000	0.1546	0.0337

Tabella 5.8: FOCF - Underestimation unfairness

5.7.1 Confronto tra i modelli: tra accuratezza, sostenibilità e giustizia

Per rispondere alla RQ1 è stata realizzata una serie di heatmap che mettono in relazione tre elementi chiave: la metrica ndcg@10, il peso attribuito alla fairness (fair_weight o dis_weight) e le emissioni (espresse in grammi).
Il modello FairGo_GCN 5.1 mostra un comportamento molto chiaro: quando non si applica alcun vincolo di fairness, l'accuratezza è raggiunge il massimo livello, ma le emissioni sono le più elevate. Man mano che si dà più importanza alla fairness, l'accuratezza diminuisce e le emissioni migliorano, la situazione ottimale è con il fair_weight=100.0, perché si riesce ad ottenere un ottimo compromesso tra tutti e tre i valori.

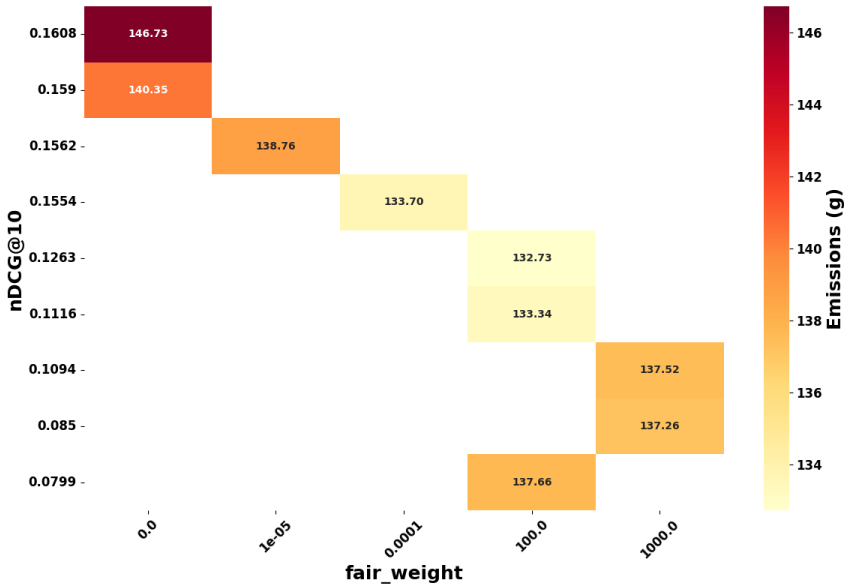


Figura 5.1: FAIRGO Heatmap

PFCN_DMF , in figura 5.2,aumentando l'importanza del peso della fairness, l'accuratezza peggiora mentre le emissioni migliorano, almeno fino al valore di fair_weight=100.0 un certo punto. Questo significa che, almeno nei dati considerati, è possibile ottenere un sistema più giusto e allo stesso tempo più sostenibile, senza sacrificare troppo la qualità, come il modello precedente, anche qui la configurazione più equilibrata si ottiene con dis_weight=100.0. Con FOCF, in figura 5.3, è stato adottato un approccio visivo diverso: nella heatmap, in alternativa ai

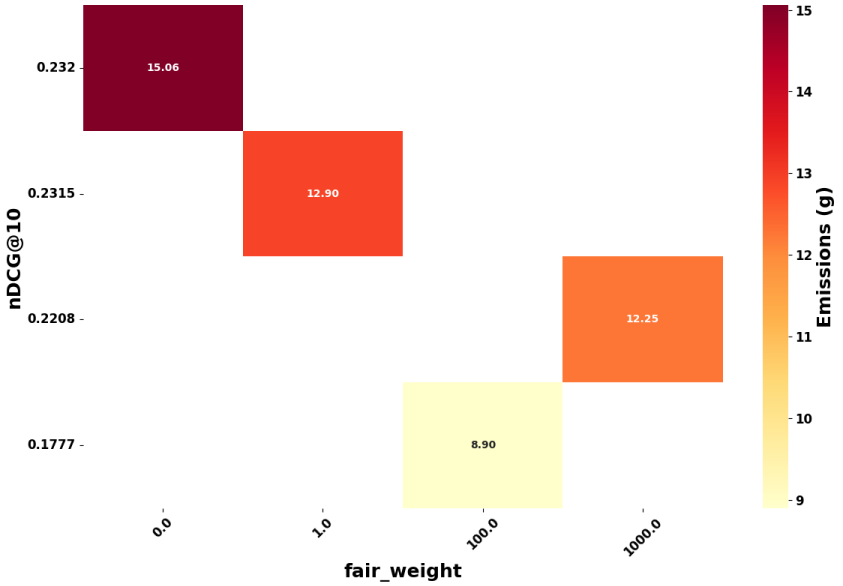


Figura 5.2: PFCN heatmap

valori numerici, sono state inserite etichette come nonparity, under e value, che indicano quale criterio di unfairness ha avuto la quantità di emissioni migliore. E' un modello che sorprende per il suo comportamento: aumentando l'importanza della fairness, sia l'accuratezza (*ndcg*@10) che le emissioni migliorano fino a raggiungere valori ottimali con fair_weight = 10.0. Questo suggerisce che, nei dati considerati, è possibile ottenere un sistema più giusto e allo stesso tempo più sostenibile, senza sacrificare la qualità. Da notare come con il massimo peso si ottenga un risultato ottimo al contrario di valori intermedi che faticano a raggiungere un ottimizzazione efficiente su tutti e tre i parametri, si può evincere che la mi gliore configurazione del modello ha come parametri fair_objective = nonparity e fair_weight = 10.0. Con

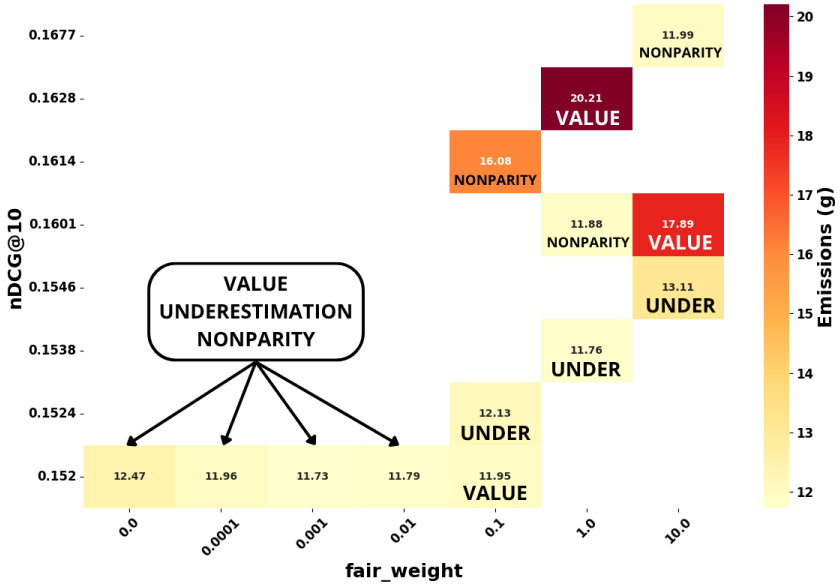


Figura 5.3: FOCF heatmap

NFCF, in figura 5.4, la situazione è più complessa: l'accuratezza parte nella media con emissioni contenute poi cresce fino a un al valore di fair_weight = 0.1, ma proprio nel punto di massima performance si registra anche il picco di emissioni. Se si forza ulteriormente la fairness, le emissioni tornano a calare ma l'accuratezza del risultato ne risente. In questo caso, trovare un buon equilibrio tra efficienza, equità e sostenibilità richiede molta attenzione, e rischierebbe di dover compromettere uno tra acuratezza, fairness e produzione di CO_2.

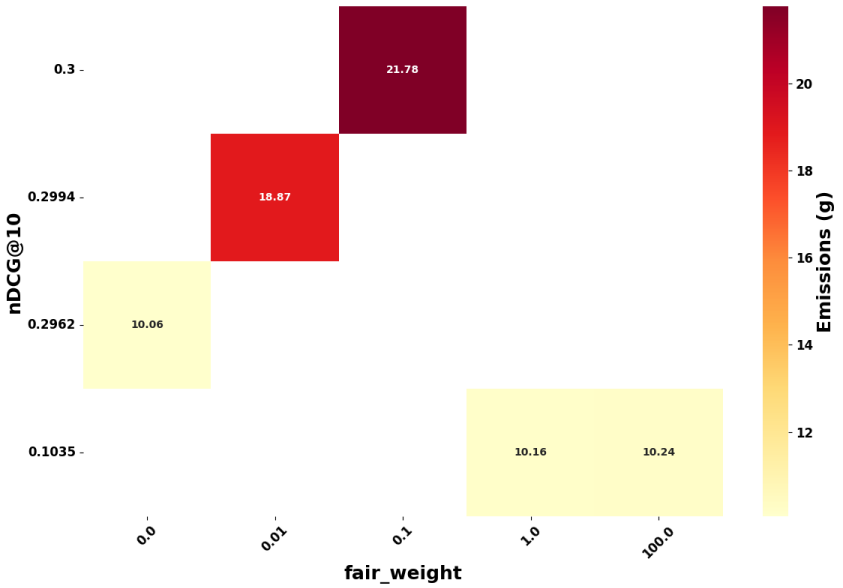


Figura 5.4: NFCF heatmap

5.7.2 Analisi di emissioni (g) Vs. Differential Fairness

Attraverso questi grafici si possono confrontare le performance dei modelli FAIRGO, NFCF e PFCN_DMF in relazione al trade-off tra fairness differenziale e emissioni di CO_2, in funzione dei principali iperparametri di debiasing. La zona ottimale nei grafici si colloca in basso a destra, dove la fairness è elevata e l'impatto ambientale contenuto. Dai risultati emerge, in relazione alla RQ2, che:
Nel modello FAIRGO, la combinazione dei parametri FW (Fairness Weight) e VS (Fairness Depth) evidenzia una correlazione favorevole: aumentando il peso e la profondità del vincolo di fairness al livello medio, si ottiene una migliore equità (DF fino a 23) e una riduzione delle emissioni (133g). La configurazione FW: medium, VS: medium rappresenta il compromesso ottimale. Per il modello NFCF,

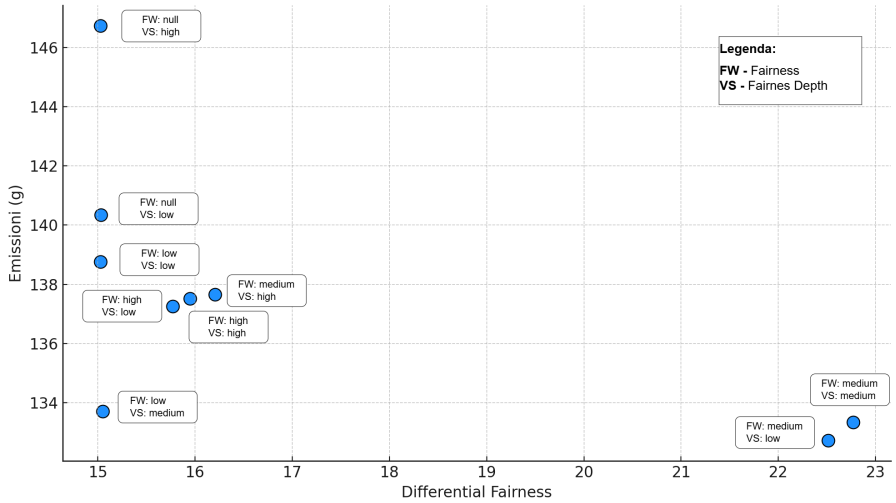


Figura 5.5: FAIRGO: emissions(g) Vs. Fairness

variando FW, la fairness differenziale resta pressoché costante (DF = 14.45–14.51), ma le emissioni variano significativamente, con picchi anche doppi rispetto ai valori migliori. FW: high garantisce emissioni basse (10g), ma senza un reale miglioramento nella fairness, suggerendo un comportamento instabile e poco scalabile. Nel caso di PFCN_DMF, l'aumento di DW (Debiasing Strength) porta a un leggero miglioramento della fairness e a una notevole riduzione delle emissioni. Il valore DW: medium risulta il più efficace (DF = 14.45, emissioni 9g), indicando che una forza di debiasing intermedia è sufficiente per ottenere un buon compromesso. FAIRGO si dimostra il modello più bilanciato e prevedibile. PFCN_DMF ha una

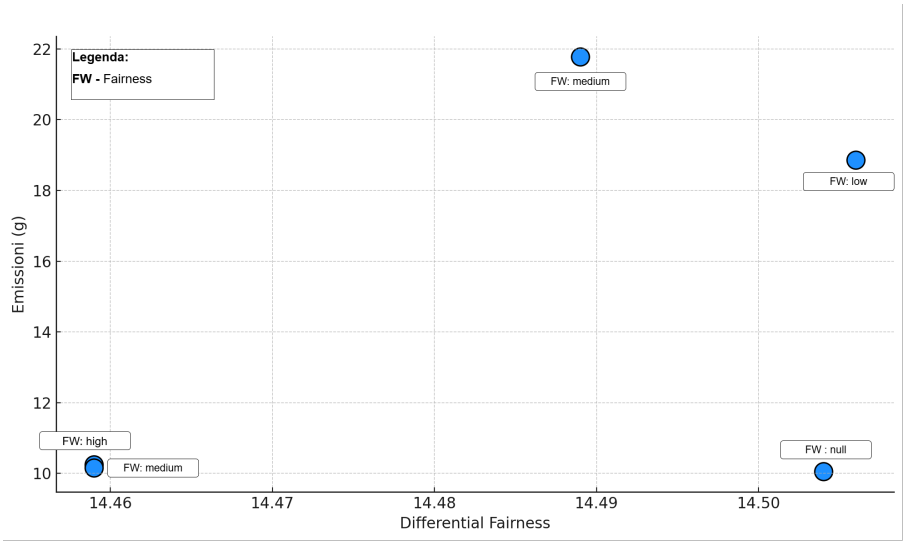


Figura 5.6: NFCF: emissions(g) Vs. Differential fairness

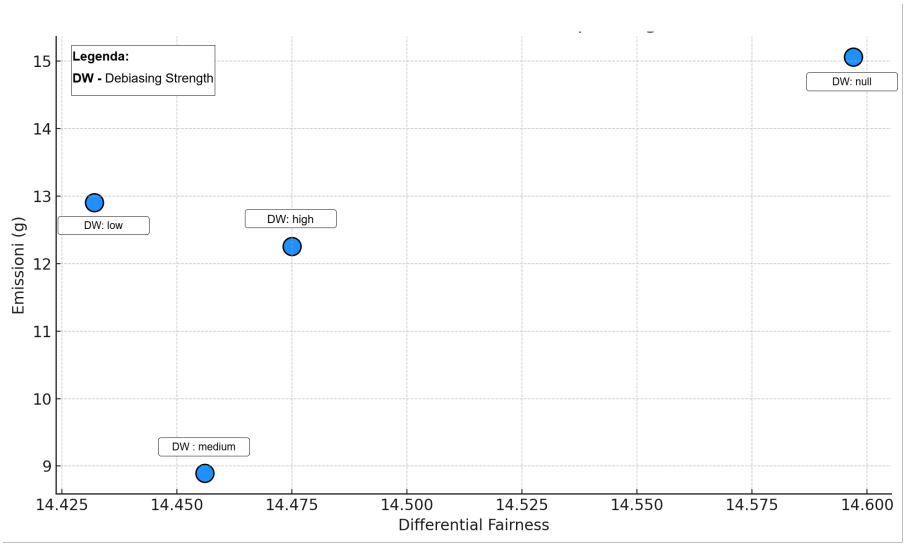


Figura 5.7: PFCN: emissions(g) Vs. Differential fairness

buona reazione con un tuning moderato, mentre NFCF mostra un andamento più irregolare, con scarsa sensibilità della fairness ai parametri e variabilità nelle emissioni. In relazione all'algoritmo FOCF Sono stati analizzati tre scenari, ciascuno ottimizzato rispetto a un diverso fairness objective (FO): underestimation, value e nonparity. I grafici riportano sull'asse X il valore finale dell'unfairness residua, cioè quanto il vincolo di fairness ha avuto effetto, e sull'asse Y le emissioni in grammi di CO_2. Ogni punto è etichettato con il livello del fairness weight (FW) utilizzato. Da tener conto che la sezione migliore in questi grafici è in basso a sinistra, che rappresenta minore emissioni ma anche minore disuguaglianza.

- **Underestimation**: i valori di unfairness più bassi (a sinistra), raggiunti con FW: medium, sono associati a emissioni più contenute (11,8g). FW: high non migliora l'equità e presenta emissioni superiori (13,0g), suggerendo un'inefficienza computazionale.

- **Value**: il vincolo FW: high porta a emissioni elevate (17,9g), ma senza un sostanziale miglioramento del valore FO. Le condizioni di fairness più efficaci si ottengono con FW: low/null, con impatti ambientali minori ($<$13,5g).

- **Nonparity**: i risultati migliori in termini di equità ed efficienza si ottengono con FW: low o null, che riduce la disparità a valori minimi con emissioni contenute (10,9g). L'aumentare di FW non garantisce una significativa riduzione dell'unfairness e comporta un costo computazionale più alto.

Possiamo affermare che un vincolo eccessivo (FW High) non sempre produce benefici concreti in termini di fairness e può aumentare significativamente la produzione di CO_2. In questi scenari, valori di FW medium o low risultano più efficaci nel bilanciare equità e sostenibilità.

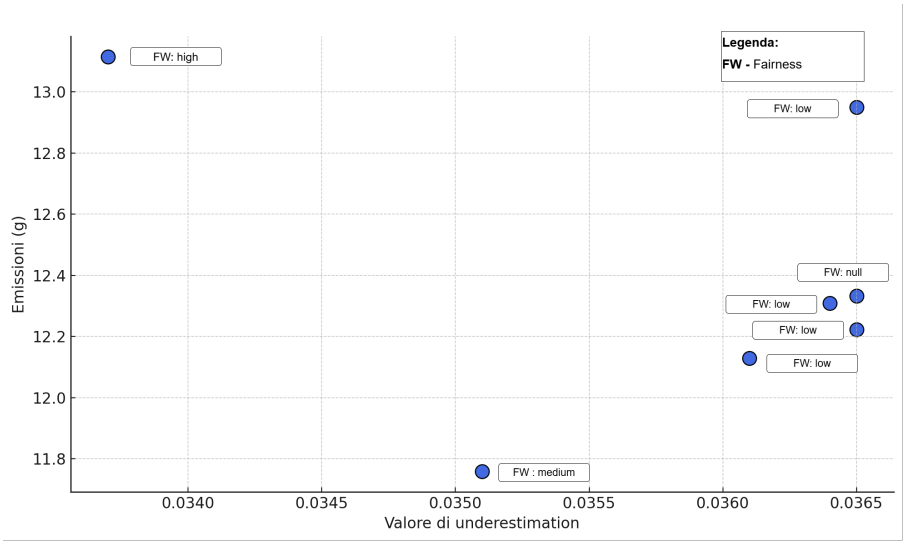


Figura 5.8: FOCF Underestimation: emissions(g) Vs. Underestimation unfairness

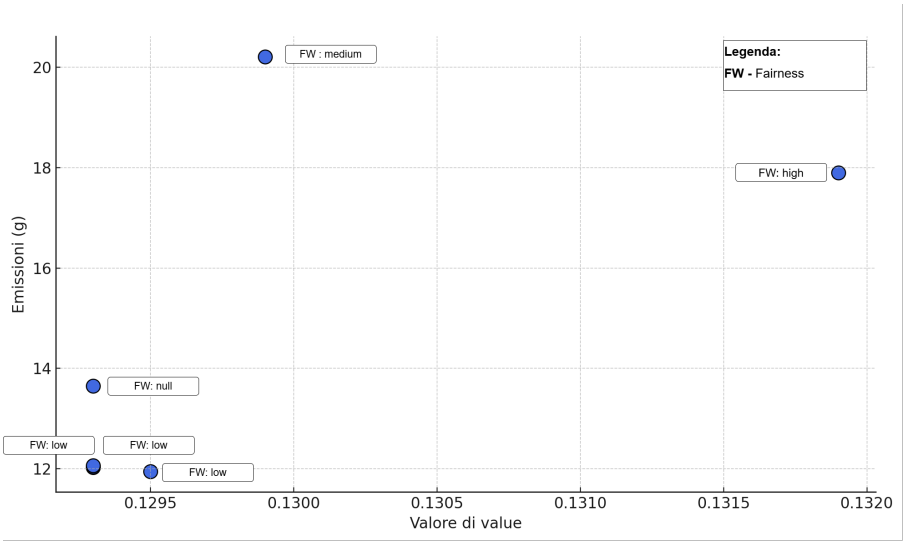


Figura 5.9: FOCF Value: emissions(g) Vs. Value unfairness

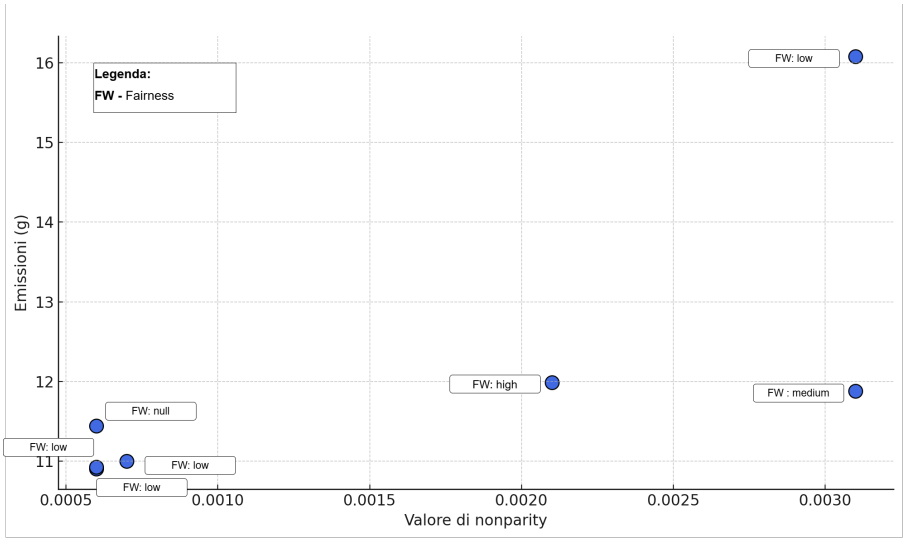


Figura 5.10: FOCF Non Parity: emissions(g) Vs. Nonparity unfairness

5.7.3 Analisi di emissioni (g) Vs. accuratezza

L'RQ3 trova riscontro nei risultati rappresentati nei seguenti scatter plot, dove sull'asse X è rappresentata la metrica ndcg@10, mentre sull'asse Y sono riportate le emissioni di CO_2 in grammi. Ogni punto del grafico è etichettato con il livello del vincolo applicato, espresso tramite FW (fairness weight) o DW (debiasing weight), e in alcuni casi con la profondità del vincolo stesso (VS). Nel modello PFCN_DMF, si può osservare un comportamento particolarmente bilanciato: l'uso di un vincolo medio di fairness (DW: medium) ha permesso di ridurre significativamente le emissioni (fino a circa 9 grammi), con una perdita contenuta in termini di accuratezza. Al contrario, la configurazione con fairness nulla raggiunge il massimo ndcg@10 (0,23), ma con un picco nelle emissioni (15g). Questo suggerisce che sia possibile trovare un buon compromesso, evitando le configurazioni estreme. Per il modello

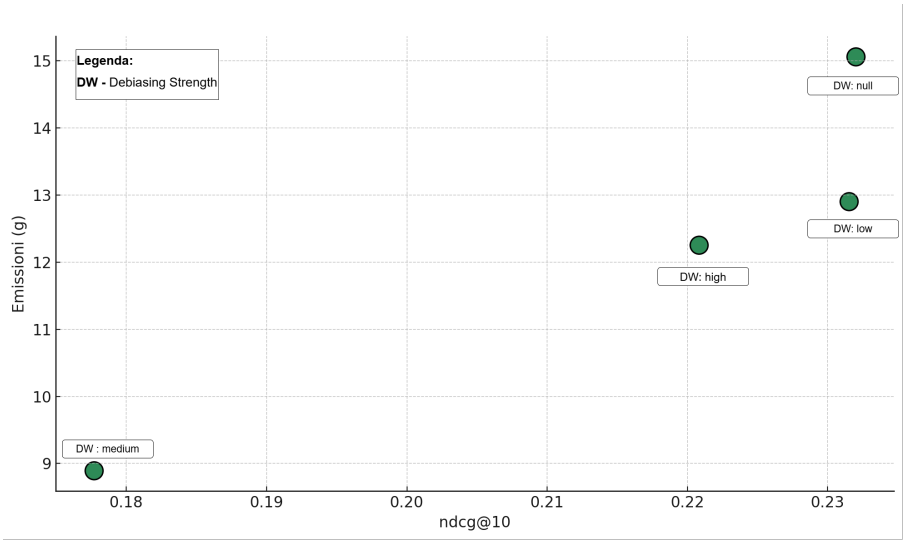


Figura 5.11: PFCN: emissions (g) Vs. NDCG

FairGo_GCN, i risultati confermano un trade-off molto marcato. Quando non viene imposto alcun vincolo di fairness (FW: null), l'accuratezza si mantiene elevata (fino a 0,16), ma le emissioni raggiungono i livelli più alti (oltre 146g). Aumentando FW e VS, l'impatto ambientale migliora (scendendo sotto i 138g), ma l'accuratezza crolla fino a valori inferiori a 0,1. Questo rende evidente come FairGo_GCN sia molto sensibile alla forza dei vincoli, richiedendo una calibrazione attenta a seconda degli obiettivi. Anche nel caso di NFCF, si riscontra un compromesso netto. La configurazione con FW: null permette di ottenere il massimo ndcg@10 (0,3), ma

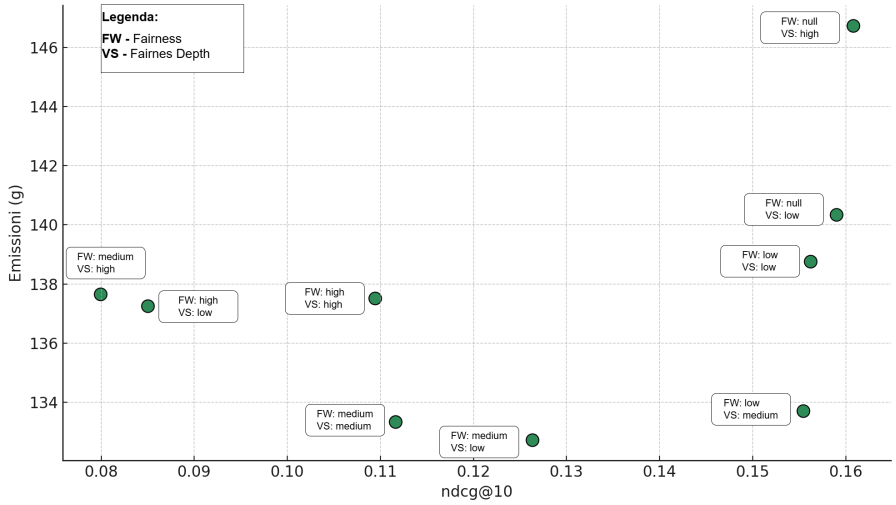


Figura 5.12: FAIRGO: emissions (g) Vs. NDCG

a discapito delle emissioni, che superano i 22 grammi. Le configurazioni con FW: high e FW: medium abbassano le emissioni fino a circa 10g, ma comportano un importante calo della qualità del ranking. Anche qui, bilanciare performance ed equità richiede attenzione e consapevolezza. Infine, sul modello FOCF si è con-

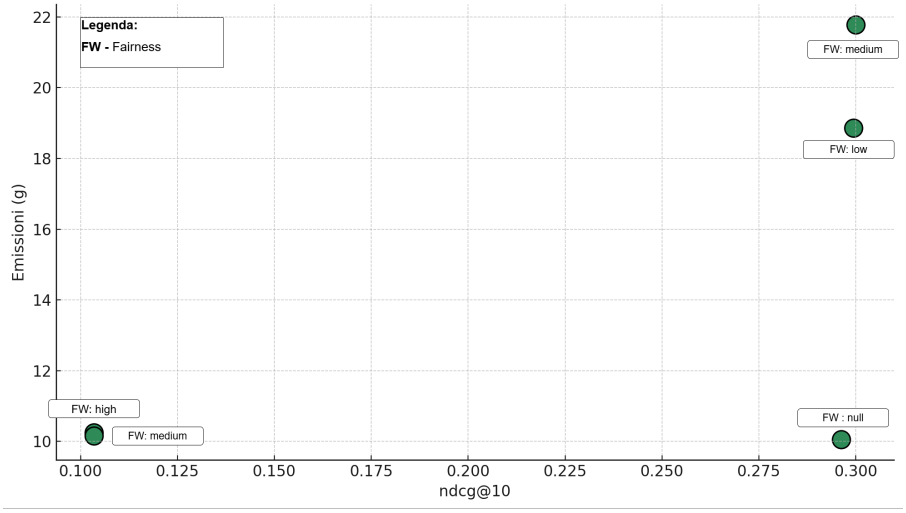


Figura 5.13: NFCF: emissions (g) Vs. NDCG

dotta un'analisi più approfondita, includendo anche la variabile relativa al tipo di fairness ottimizzata (Fairness Objective, FO). In particolare, si è messo in relazione tre approcci di unfairness: underestimation, value e nonparity. Per ciascun tipo di FO, è stato prodotto uno scatter plot in cui l'asse X rappresenta il valore finale dell'unfairness residua (cioè quanto il modello è riuscito a essere equo), mentre l'asse Y mostra le emissioni di CO_2. I punti sono etichettati in base al livello di FW. Questa analisi ha evidenziato differenze importanti:

- Con nonparity, si ottiene il miglior equilibrio: accuratezza stabile, buon livello di fairness e emissioni contenute (sotto i 12g), soprattutto con FW medium o high.

- Con underestimation, l'aumento eccessivo del vincolo porta a un peggioramento ambientale senza guadagni significativi in equità, mentre i valori medi di FW risultano più efficienti.

- Con value, si osserva il comportamento meno sostenibile: anche in presenza di FW: high, le emissioni salgono oltre i 20g senza una corrispondente riduzione dell'unfairness.

Per evidenziare ulteriormente queste differenze, infine sono stati aggregati i risultati in un unico scatter plot categorizzato per tipo di fairness. Questo mi ha permesso di visualizzare chiaramente che non solo la forza del vincolo (FW), ma anche la natura stessa dell'obiettivo di fairness influenza l'impatto ambientale del modello. In altre parole, migliorare l'equità non significa solo quanto fairness applico, ma anche su quale tipologia fairness scelgo di andare ad operare.

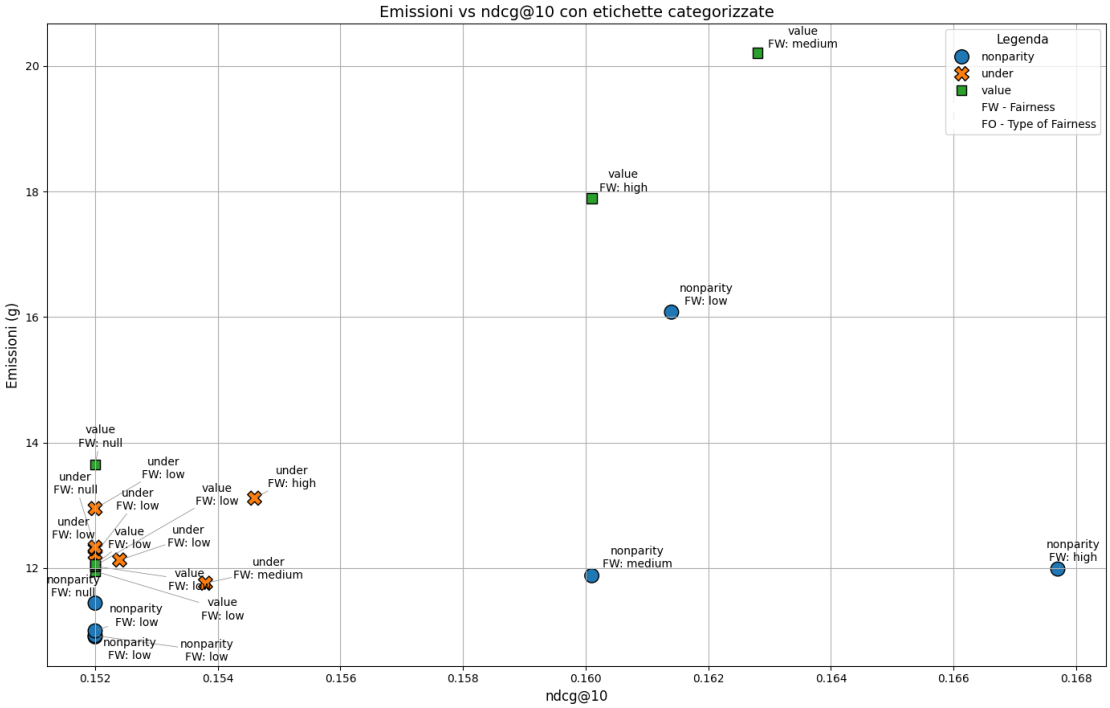


Figura 5.14: FOCF: emissions (g) Vs. NDCG

5.8 Confronto finale delle configurazioni

Al termine dell'analisi è stato realizzato un grafico tridimensionale (Figura 5.15) che confronta le migliori configurazioni fairness-aware dei modelli FOCF, FairGo, NFCF e PFCN_DMF rispetto a tre dimensioni: accuratezza (asse X), fairness normalizzata (asse Y) e impatto ambientale (asse Z). La fairness è stata ottenuta

come inverso della media tra value unfairness e nonparity unfairness, poichè uniche metriche calcolate comuni in tutti gli algoritmi, e normalizzata secondo la formula:

$$\text{fairness}_i = 1 - \frac{\text{unfairness}_i - \min(\text{unfairness})}{\max(\text{unfairness}) - \min(\text{unfairness})}$$

In questo modo, un valore di fairness pari a 1 indica la massima equità raggiunta tra i modelli testati, mentre un valore prossimo a 0 indica la minima equità. Il modello PFCN_DMF si distingue per aver raggiunto il miglior compromesso: massima fairness (1.00), emissioni minime (8.9 g CO_2) e una buona accuratezza (NDCG = 0.178). NFCF ottiene la massima accuratezza ma con equità solo intermedia, mentre FOCF mostra un equilibrio valido su tutte le metriche. FairGo, infine, risulta il modello meno vantaggioso. Il grafico consente di visualizzare chiaramente i compromessi tra le tre dimensioni, grazie all'ausilio di emoticon esplicative.

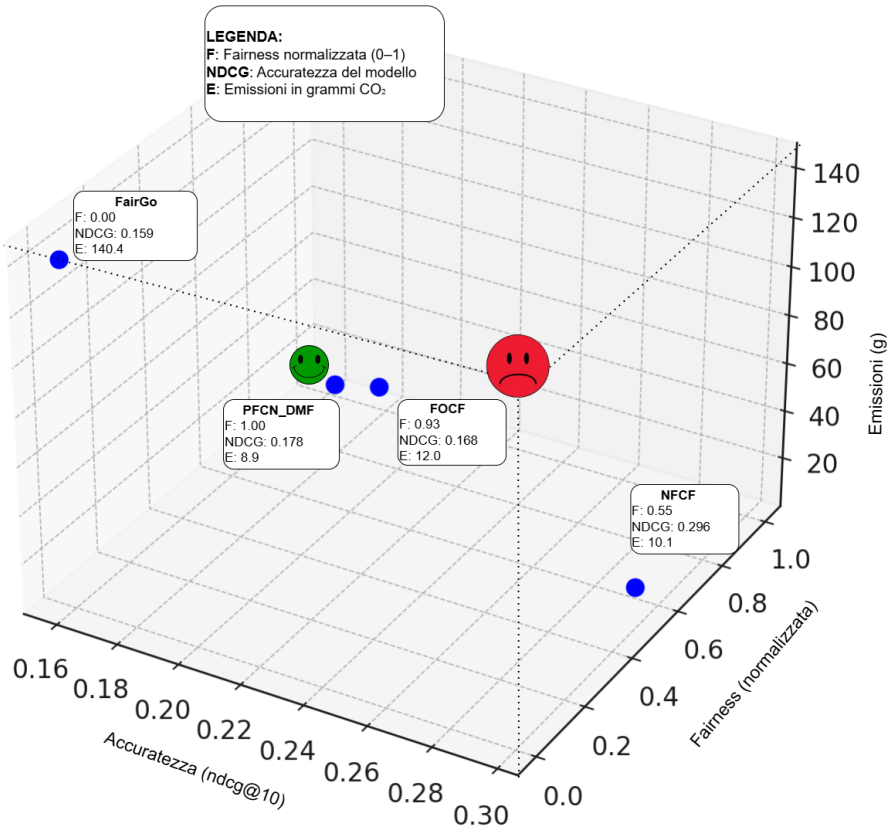


Figura 5.15: Confronto tridimensionale tra le configurazioni fairness-aware ottimali di ciascun modello. Ogni punto rappresenta una combinazione di accuratezza (asse X), fairness normalizzata (asse Y) e impatto ambientale (asse Z).

Capitolo 6

Conclusioni

In questo progetto di tesi, si è condotto uno studio sperimentale finalizzato ad analizzare la relazione tra fairness ed emissioni di CO_2 nei sistemi di raccomandazione. L'obiettivo principale è stato quello di comprendere se, e in quale misura, sia possibile progettare modelli in grado di garantire equità nelle raccomandazioni senza compromettere eccessivamente l'accuratezza o aggravare l'impatto ambientale. Per farlo, sono stati selezionati quattro algoritmi fairness-aware: FairGo, NFCF, PFCN e FOCF, ciascuno rappresentativo di approcci diversi alla mitigazione dei bias nei recommender system. Tali modelli sono stati testati all'interno del framework RecBole-FairRec, che ha permesso di monitorare in modo coerente prestazioni predittive e metriche di fairness. Per stimare le emissioni di CO_2 associate a ciascun addestramento, è stata integrata nel processo la libreria CodeCarbon, strumento open-source in grado di stimare il consumo energetico in funzione dell'hardware utilizzato e del mix energetico locale. La fase successiva ha riguardato la configurazione e l'addestramento dei modelli, effettuata mantenendo costanti dataset e pipeline sperimentale. Ogni modello è stato testato variando i pesi delle penalizzazioni legate alla fairness (es. fair_weight, dis_weight), così da osservare l'impatto delle scelte di configurazione sui risultati ottenuti. Le metriche utilizzate per valutare le predizioni includevano ndcg@10 per la qualità delle raccomandazioni e misure come Differential Fairness e Value Unfairness per stimare la giustizia distributiva. I risultati sono stati analizzati secondo due prospettive: da un lato, attraverso una valutazione quantitativa supportata da grafici (heatmap e scatter plot), che ha permesso di identificare trend e compromessi tra accuratezza, equità

ed emissioni; dall'altro, tramite una lettura qualitativa del comportamento dei modelli rispetto ai diversi criteri di fairness adottati. Dall'analisi condotta è emerso che:

- L'utilizzo esclusivo di tecniche fairness-aware, come nel caso di FairGo con elevati fair_weight, comporta una riduzione delle emissioni ma anche un peggioramento delle performance predittive.
- Alcuni modelli, come PFCN, sono riusciti a mantenere un buon equilibrio tra accuratezza e sostenibilità, mostrando che l'applicazione di vincoli moderati può migliorare il bilanciamento complessivo.
- Altri, come NFCF, hanno invece mostrato una maggiore variabilità, con emissioni che cambiano in modo significativo anche in assenza di miglioramenti sostanziali in termini di fairness.
- Infine, FOCF ha messo in luce quanto sia rilevante la scelta del criterio di equità da ottimizzare: tra gli obiettivi testati, il vincolo nonparity ha offerto il miglior compromesso tra giustizia ed efficienza energetica.

A sintesi del lavoro svolto, il grafico tridimensionale finale (Figura 5.15) ha permesso di visualizzare chiaramente le relazioni tra accuratezza, equità e sostenibilità, confermando che alcuni modelli, come PFCN, riescono effettivamente ad avvicinarsi a un punto di equilibrio virtuoso tra le tre dimensioni fondamentali. Complessivamente, le valutazioni ottenute permettono di concludere che la combinazione di approcci e configurazioni differenti è in grado di produrre sistemi di raccomandazione più equi e sostenibili, soprattutto se calibrata attentamente rispetto agli obiettivi applicativi. Sebbene l'introduzione della fairness comporti inevitabilmente un costo computazionale, i risultati ottenuti dimostrano che tale costo può essere contenuto, e in alcuni casi persino compensato, se si adottano soluzioni tecniche adeguate.

Bibliografia

[1] Ashwathy Ashokan and Christian Haas. Fairness metrics and bias mitigation strategies for rating predictions. *Information Processing & Management*, 58(5):102646, 2021.

[2] Ghazaleh Beigi, Kai Shu, Yanchi Zhang, and Huan Liu. Fairness in recommendation: A survey. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*, pages 39–48, 2020.

[3] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *arXiv preprint arXiv:1703.09207*, 2017.

[4] Alex Beutel, Jilin Chen, Zhe Zhao, Paul Covington, Sagar Jain, and Ed H Chi. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2212–2220, 2019.

[5] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 405–414, 2018.

[6] Ludovico Boratto, Gianni Fenu, Mirko Marras, and Giacomo Medda. Consumer Fairness in Recommender Systems: Contextualizing Definitions and Mitigations. volume 13185, pages 552–566. 2022. arXiv:2201.08614 [cs].

[7] Karine Borges and Kostas Stefanidis. Fairness-aware recommender systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2173–2176, 2019.

[8] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. Balanced Neighborhoods for Multi-sided Fairness in Recommendation.

[9] Toon Calders and Sicco Verwer. Building classifiers with independent label and sensitive feature distributions. *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, pages 51–58, 2009.

[10] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053*, 2020.

[11] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and Debias in Recommender System: A Survey and Future Directions, December 2021. arXiv:2010.03240 [cs].

[12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012.

[14] Michael Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. Scripts for All The Cool Kids, How Do They Fit In. Institution: Boise State University.

[15] James R Foulds, Rawad Islam, Pranay Keya, Sida Pan, Jennifer Richie, and Suresh Venkatasubramanian. Differential fairness: An intersectional fairness criterion for learning. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 675–685, 2020.

[16] Gabriel Frisch, Jean-Benoist Leger, and Yves Grandvalet. Co-clustering for Fair Recommendation. In Michael Kamp, Irena Koprinska, Adrien Bibal, Tassadit Bouadi, Benoît Frénay, Luis Galárraga, José Oramas, Linara Adilova, Yamuna Krishnamurthy, Bo Kang, Christine Largeron, Jefrey Lijffijt, Tiphaine Viard, Pascal Welke, Massimiliano Ruocco, Erlend Aune, Claudio Gallicchio, Gregor Schiele, Franz Pernkopf, Michaela Blott, Holger Fröning, Günther Schindler, Riccardo Guidotti, Anna Monreale, Salvatore Rinzivillo, Przemyslaw Biecek, Eirini Ntoutsi, Mykola Pechenizkiy, Bodo Rosenhahn, Christopher Buckley, Daniela Cialfi, Pablo Lanillos, Maxwell Ramstead, Tim Verbelen, Pedro M. Ferreira, Giuseppina Andresini, Donato Malerba, Ibéria Medeiros, Philippe Fournier-Viger, M. Saqib Nawaz, Sebastian Ventura, Meng

Sun, Min Zhou, Valerio Bitetta, Ilaria Bordino, Andrea Ferretti, Francesco Gullo, Giovanni Ponti, Lorenzo Severini, Rita Ribeiro, João Gama, Ricard Gavaldà, Lee Cooper, Naghmeh Ghazaleh, Jonas Richiardi, Damian Roqueiro, Diego Saldana Miranda, Konstantinos Sechidis, and Guilherme Graça, editors, *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, volume 1524, pages 607–630. Springer International Publishing, Cham, 2021. Series Title: Communications in Computer and Information Science.

[17] Yikun Ge, Xiting Liu, Harald Steck, Yongfeng Zhang, and Ji-Rong Wen. Towards long-term fairness in recommender systems. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 445–453, 2021.

[18] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. Barcelona, Spain.

[19] Blake Huebner, Thomas Elmar Kolb, and Julia Neidhardt. Evaluating group fairness in news recommendations: A comparative study of algorithms and metrics. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization (UMAP Adjunct '24)*, pages 337–346, Cagliari, Italy, 2024. ACM.

[20] Machine Learning CO2 Impact. Codecarbon: a tool to estimate the carbon emissions of computing. `https://github.com/mlco2/codecarbon`, 2023. Accessed: 2025-06-20.

[21] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. Debiasing career recommendations with neural fair collaborative filtering. In *Proceedings of The Web Conference (WWW)*, pages –, Ljubljana, Slovenia, 2021.

[22] Christopher Jung, Sampath Kannan, Changhwa Lee, Mallesh Pai, Aaron Roth, and Rakesh Vohra. Fair prediction with endogenous behavior. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, page 677–678, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Recommendation Independence.

[24] Matt Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[25] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. ifair: Learning individually fair data representations for algorithmic decision making. In *Proceedings of the 2019 International Conference on Machine Learning (ICML)*, 2019.

[26] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. User-oriented fairness in recommendation. In *Proceedings of the Web Conference 2021*, WWW '21, page 624–632, New York, NY, USA, 2021. Association for Computing Machinery.

[27] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. Fairness in Recommendation: Foundations, Methods and Applications, July 2023. arXiv:2205.13619 [cs].

[28] David Miller. Justice. `https://plato.stanford.edu/archives/spr2025/entries/justice/`, 2025. The Stanford Encyclopedia of Philosophy (Spring 2025 Edition), Edward N. Zalta & Uri Nodelman (eds.).

[29] Harikrishna Narasimhan, Jennifer Gillenwater, and Alekh Agarwal. Pairwise fairness for ranking and regression. In *International Conference on Artificial Intelligence and Statistics*, pages 2222–2230, 2020.

[30] David Patterson and et al. Carbon emissions and large neural networks. Google AI Blog, 2021.

[31] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: an overview. *The VLDB Journal*, 31(3):431–458, May 2022.

[32] Abbas Rastegarpanah, Krishna P Gummadi, and Mark Crovella. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *Proceedings of the 2019 World Wide Web Conference*, pages 231–240, 2019.

[33] Bashir Rastegarpanah, Krishna P. Gummadi, and Mark Crovella. Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 231–239, Melbourne VIC

Australia, January 2019. ACM.

[34] Amit Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 2018 ACM SIGIR Conference*, pages 82–91, 2018.

[35] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *Proceedings of the ACL*, 2019.

[36] Jiakai Tang. Recbole-fairrec: Fairness-aware recommender system library. `https://github.com/TangJiakai/RecBole-FairRec`, 2021. Accessed: 2025-06-23.

[37] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A Survey on the Fairness of Recommender Systems. *ACM Transactions on Information Systems*, 41(3):1–43, July 2023.

[38] Xiang Wei, Zihan Zhang, et al. Recbole: Unified, comprehensive and efficient recommendation library. `https://recbole.io/subpackage.html`, 2020. Accessed: 2025-06-23.

[39] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. Learning fair representations for recommendation: A graph-based perspective. In *Proceedings of the Web Conference 2021*, WWW '21, page 2198–2208, New York, NY, USA, 2021. Association for Computing Machinery.

[40] Xueqi Wu, Lei Wu, Hao Wang, and Yiqun Liu. Learning fair representations with self-distillation for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference*, pages 1835–1845, 2022.

[41] Sen Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*, pages 2921–2930, 2017.

[42] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

[43] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578, 2017.

[44] Lei Zheng, Vahid Noroozi, and Philip S Yu. Counterfactual recommendation for fair ranking. In *Proceedings of the 27th ACM SIGKDD Conference on*

Knowledge Discovery & Data Mining, pages 2440–2449, 2021.