Sustainability of RecSys

Emanuele Fontana

Università degli Studi di Bari Aldo Moro

- **Relatore**: Prof. Pasquale Lops
- **Relatore**: Prof. Cataldo Musto
- **Correlatore**: Dott. Giuseppe Spillo
- **Laurenado**: Emanuele Fontana

Indice

Sostenibilità

Sostenibilità

La sostenibilità è un concetto complesso e multidimensionale, emerso negli anni ’80, che ha acquisito un’importanza crescente nella società contemporanea. Essa si riferisce alla capacità di soddisfare i bisogni presenti senza compromettere quelli delle generazioni future, coinvolgendo la gestione responsabile delle risorse naturali, la tutela ambientale, lo sviluppo economico e sociale e la garanzia di un futuro migliore. La sostenibilità ambientale è uno degli aspetti più importanti della sostenibilità e riguarda la gestione responsabile delle risorse naturali, la tutela dell’ambiente e la prevenzione dell’inquinamento e del degrado ambientale.

Sostenibilità e AI

Nell'ambito dell'Intelligenza Artificiale (AI) e della sostenibilità ambientale, possiamo distinguere due tipi di AI sostenibile:

- **Sustainability of AI**: si concentra sulla misurazione della sostenibilità nello sviluppo e nell'uso dei modelli AI, come la carbon footprint e l'energia necessaria per addestrarli.
- **AI for Sustainability**: utilizza l'AI per affrontare le sfide della sostenibilità, come la previsione del cambiamento climatico e la gestione delle risorse naturali.

La Green AI si riferisce allo sviluppo di modelli AI che considerano il costo computazionale e l'impatto ambientale. In contrasto, la Red AI mira a creare modelli sempre più complessi senza considerare le risorse impiegate. La Green AI riduce parametri, complessità e operazioni computazionali per minimizzare l'uso di risorse energetiche.

Recommender Systems

RecSys

Un sistema
di raccomandazione è un software che suggerisce all'utente elementi di interesse (prodotti, servizi, contenuti) basandosi sulle preferenze e i comportamenti passati. Ampiamente usati in e-commerce, social network, servizi di streaming e piattaforme di informazione, questi sistemi migliorano l'esperienza utente, aumentano la soddisfazione e la fidelizzazione, e aiutano a scoprire nuovi contenuti. Utilizzano algoritmi di apprendimento automatico e intelligenza artificiale per analizzare i dati e generare raccomandazioni personalizzate.

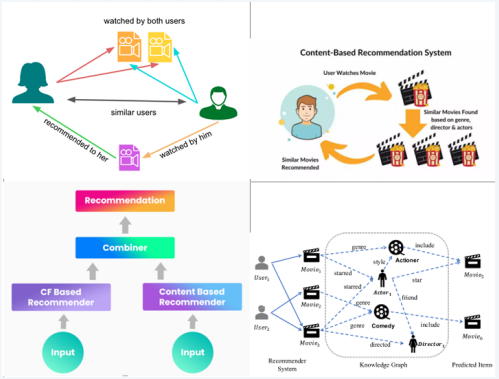


Figura: Tipologie di RecSys

Valutazione e problemi dei RecSys

Valutazione

Per valutare un RecSys si possono usare diverse metriche come il MAE (Mean Absolute Error) che misura la differenza tra il valore predetto e il valore reale, l'NDGC che misura la qualità delle raccomandazioni, la diversity che misura la varietà delle raccomandazioni, la coverage che misura la percentuale di elementi raccomandati, la recall che misura la capacità di un modello di raccomandare gli item rilevanti per un utente

Problemi

I RecSys possono presentare problemi di Cold Start, quando non si hanno informazioni sufficienti per fare raccomandazioni, di More of the Same, quando si raccomandano sempre gli stessi elementi, la vulnerabilità agli attacchi (come recensioni false).

Design degli Esperimenti

Design degli Esperimenti

Il lavoro svolto consiste nel:

- valutare e prevedere l'impatto ambientale di un sistema di raccomandazione (RecSys) in base alla sua sostenibilità
- cercare una soluzione per ridurre l'impatto ambientale di un RecSys senza però perdere di performance in modo significativo.

Nella prima parte dunque sono stati effettuati esperimenti per valutare l'impatto ambientale di diversi modelli di raccomandazione su diversi dataset. Questi dati sono stati utilizzati per addestrare un modello di regressione che permette di prevedere le emissioni prodotte da un modello di raccomandazione in base a diversi parametri. Nella seconda parte, invece, ci si è concentrati su trovare un trade-off tra emissioni e performance.

Base di partenza

In questo ambito sono stati già svolti diversi esperimenti che mostrano come spesso algoritmi più semplici riescono ad avere delle performance molto simili (se non migliori) a modelli più complessi, ma con un impatto ambientale decisamente minore.

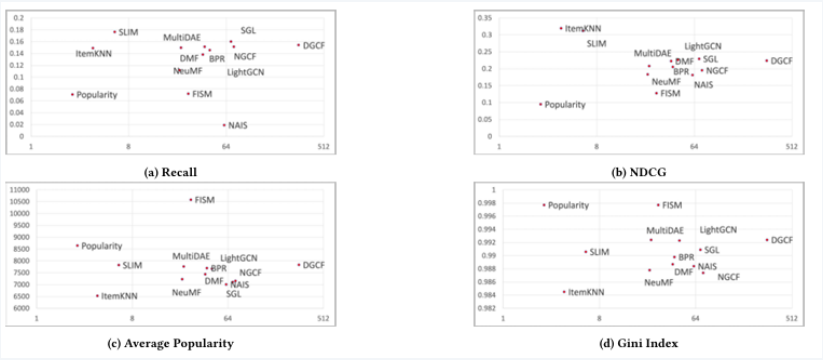


Figura: Trade-off tra emissioni e performance con dataset Mind

Una strategia comune per calcolare il CO_2eq è quella di moltiplicare tra loro il **carbon intensity(CI)** e l'**energia consumata(PC)** dall'attività (nel nostro caso l'esecuzione di algoritmi).

$$emission = CI \cdot PC$$

In particolare i valori di CI dipendono dalle diverse fonti di energia utilizzate durante la computazione (es. energia solare, energia eolica, etc.). Se s è la fonte di energia, e_s sono le emissioni per KW/h di energia e p_s è la percentuale di energia prodotta dalla fonte s, allora il CI è dato da:

$$CI = \sum_{s \in S} e_s \cdot p_s$$

Qui possiamo notare
la distribuzione delle emissioni prodotte dagli esperimenti iniziali. Il dataset del regressore è descritto dalle seguenti features di input:

- **n_users**: numero di utenti
- **n_items**: numero di item
- **n_inter**: numero di interazioni
- **sparsity**: sparsità del dataset
- **kg_entities**:
  numero di entità nel knowledge graph
- **kg_relations**: numero di relazioni nel knowledge graph
- **kg_triples**: numero di triple nel knowledge graph
- **kg_items**: numero di item nel knowledge graph
- **cpu_cores**: numero di core della CPU
- **ram_size**: quantità di RAM
- **is_gpu**: presenza di una GPU
- **model_name**: nome del modello
- **model_type**: tipo di modello

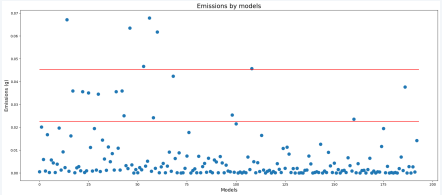


Figura: Dataset iniziale

Regressore - Introduzione

Sono stati utilizzati diversi modelli di regressione per prevedere le emissioni prodotte da un modello di raccomandazione in base alle features di input. I modelli utilizzati sono:

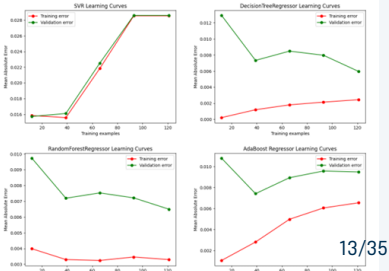
- **Random Forest** : sono modelli di regressione basati su alberi deci sionali. Vengono costruiti su più alberi decisionali che combinano le loro previsioni per ottenere una previsione pi‘ u accurata e stabile
- **Decision Tree**: sono modelli di regressione basati su alberi decisionali
- **AdaBoost**: L’AdaBoost Regressor è un modello basato sull’algoritmo di boosting AdaBoost. Questo algoritmo costruisce un modello di previsione combinando pi‘ u modelli di previsione più deboli.
- **SVG**: è un algoritmo che estende il concetto di Support Vector Machine (SVM) al caso della regressione

Regressore - Analisi dei risultati

E' stato utilizzato uno split del 70% per il training e del 30% per il testing. Il dataset sembra essere troppo piccolo per poter generalizzare bene i modelli. Inoltre, il dataset è molto sbilanciato, con pochi valori per ogni feature. Per i risultati attuali, il Decision Tree Regressor è il modello migliore

Regressor	MAE	RMSE	MSLE
SVR	0.0288215	0.0008862	0.0008537
Decision Tree	0.0048531	0.0000969	0.0000918
Random Forest	0.0054369	0.0001088	0.0001026
AdaBoost	0.0071778	0.0001113	0.0001059

Tabella 3: Risultati ottenuti



Benchmarking

Dataset - LFM_1b_artist

Il primo dataset utilizzato è LFM_1b_artist, un dataset di ascolti musicali contenente informazioni riguardanti le interazioni tra utenti e artisti musicali

Feature	Valore
Numero di utenti	120322
Numero di item	3123496
Numero di interazioni	65133026
Sparsità	0.9998266933373666
avg_interactions	541.3226675088513

Tabella: Statistiche dataset LFM_1b_artist

Dataset - LFM_1b_artist

Questo risultava essere troppo grande per le risorse a disposizione, quindi così processato:

- **Filtraggio**: il dataset è stato filtrato eliminando tutte le interazioni in cui erano coinvolti utenti e/o item con meno di 5 interazioni
- **Sampling**:sampling casuale di 50000 items e 20000 utenti. Sono sate mantenute solo le interazioni che coinvolgevano questi utenti e items
- **Stratificazione**: Sono state effettuate le seguenti stratificazioni per utente in modo da mantenere la distribuzione delle interazioni: 75% , 50%, 25%

Dataset - MovieLens10M

E' un dataset di valutazioni di film, contenente informazioni riguardanti le valutazioni date dagli utenti ai film. In questo caso sono presenti 10 milioni di valutazioni. E' stato anche aggiunto un knowledge graph contenente informazioni riguardanti i film e gli attori.

Feature	Valore
Numero di utenti	69878
Numero di item	10677
Numero di interazioni	10000054
Sparsità	0.9865966722939162
avg_interactions	143.10732991785684

Tabella: Statistiche MovieLens10M

Dataset - MovieLens10M

Anche questo risultava essere troppo grande per le risorse a disposizione, quindi così processato:

- **Filtraggio**: il dataset è stato filtrato eliminando tutte le interazioni in cui erano coinvolti utenti e/o item con meno di 5 interazioni
- **Sampling**:sampling casuale di 50000 utenti e 10000 items. Sono sate mantenute solo le interazioni che coinvolgevano questi utenti e items

Emissioni - LFM_1b_artist

Si può subito
notare come DGCF è il modello che emette più CO_2 in assoluto, da 2 a 10 volte di più rispetto agli altri modelli a seconda del dataset. LightGCN e NCFG sono rispettivamente il secondo e il terzo modello che emettono più CO_2. Questi due modelli sono di tipo general, ma nonostante ciò emettono di più rispetto ad altri di tipo knowledge-aware, come per esempio il KGCN

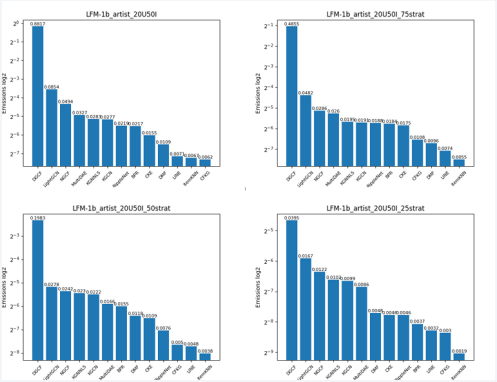


Figura: Emissioni LFM_1b_artist

Emissioni - MovieLens10M

E' possibile notare come molti modelli rispetto al dataset LFM-1b_artist_20U50I abbiano emissioni di CO2 molto più alte (a volte anche 10 volte tanto). Ciò sicuramente è dovuto al fatto che il numero delle interazioni presenti in questo dataset sono circa 10 volte quelle del dataset LFM-1b_artist_20U50I. E' possibile comunque notare come i modelli che le emissioni dei modelli seguono circa lo stesso ordine dei dataset precedenti, confermando le tendenze.

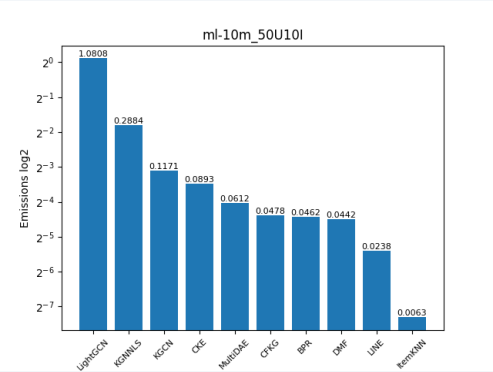


Figura: Emissioni MovieLens10M

Trade - Off

Per ogni dataset
sono stati creati grafici che mostrano
il trade-off tra emissioni e performance
simili a quello mostrato in figura. In
generale DGCF è il modello in cui il trade-off
mostra i risultati peggiori. In genere ItemKNN
risulta essere il migliore nel trade-off per
le metriche di ranking mentre LINE il migliore
per le metriche di popolarità e giniindex.

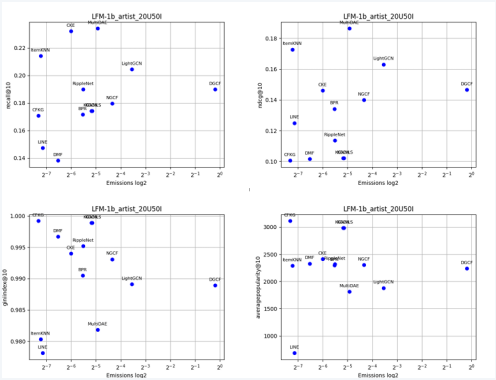


Figura: Trade-Off

Regressore - Dataset Completo

Come è possibile
notare i nuovi esperimenti hanno
portato a un'ulteriore sbilanciamento
nel dataset, in quanto tutti
gli esperimenti con DGCF e altri modelli
svettano sui risultati degli altri modelli in
emissioni. Questo si riflette nei risultati
ottenuti dai modelli di regressione.
Per quanto riguarda i modelli,
sono stati eseguiti addestramenti
con diversi split tra training e test set.

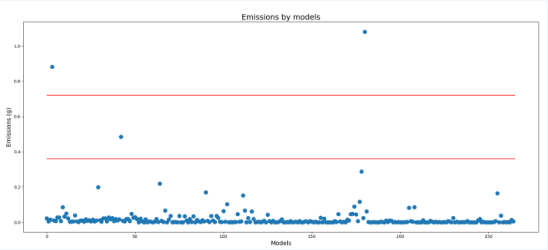


Figura: Nuova distribuzione dei dati

Regressore - Analisi dei risultati Dataset Completo

Sono stati
eseguiti addestramenti con i seguenti split:

- 50% training, 50% test
- 60% training, 40% test
- 70% training, 30% test
- 80% training, 20% test
- 90% training, 10% test

I migliori risultati sono stati ottenuti con
lo split 70-30, con il Decision Tree Regressor
che risulta essere il modello migliore.

Regressor	MAE	RMSE	MSLE
SVR	0.1103392	0.0268736	0.0154947
Decision Tree	0.0419923	0.0139534	0.0067713
Random Forest	0.0410102	0.0179366	0.0081916
AdaBoost	0.0486434	0.0160941	0.0074143

Tabella 33: Risultati ottenuti con il nuovo dataset split 70/30

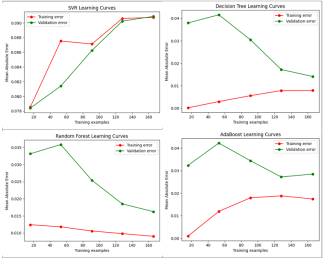


Figura: Risultati con dataset completo

Regressore - Dataset Azure

E’ stato creato un nuovo dataset con i risultati ottenuti solo sugli esperimenti eseguiti su Azure per avere un regressore specifico per gli esperimenti eseguiti su tale macchina.

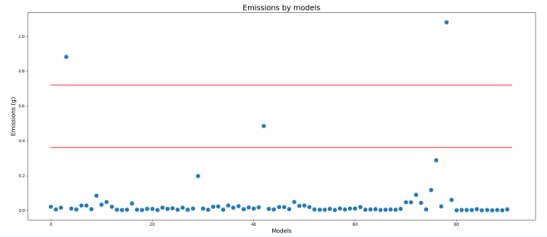


Figura: Nuova distribuzione dei dati

Regressore - Analisi dei risultati Dataset Azure

Sono stati
eseguiti addestramenti con i seguenti split:

- 50% training, 50% test
- 60% training, 40% test
- 70% training, 30% test
- 80% training, 20% test
- 90% training, 10% test

I migliori risultati sono stati ottenuti con
lo split 90-10, con il Decision Tree Regressor
che risulta essere il modello migliore.

Regressor	MAE	RMSE	MSLE
SVR	0.0896299	0.0081883	0.0072278
Decision Tree	0.0220335	0.0025806	0.0020781
Random Forest	0.0231489	0.0023014	0.0018764
AdaBoost	0.0384317	0.0063016	0.0047659

Tabella 47. Risultati ottenuti con il nuovo dataset Azure con split 90/10

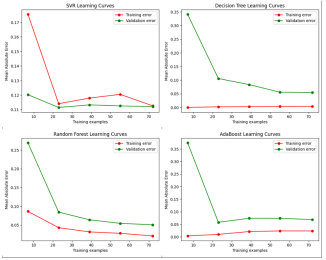


Figura: Risultati con dataset completo

Errori nelle classi

Il dataset è stato diviso in 3 classi, ovvero **low**, **medium** e **high**, in base ai valori di emissioni di CO2 (rispettivamente codificate con 0,1,2). Per ogni elemento del test set, sono stati calcolati gli errori e successivamente si è proceduto a calcolare la media degli errori per ogni classe. Gli errori sono stati calcolati come segue:

- **Errore assoluto**:

$$|y - \hat{y}|$$

- **Errore percentuale**:

$$\frac{|y - \hat{y}|}{y} \cdot 100$$

dove y è il valore reale e $\hat{y}$ è il valore predetto.

Errori nelle classi

Dataset Completo

- **low_bound**:0.36027045739563296
- **high_bound**: 0.7205403996360675.

Classe	Numero elementi	Errore assoluto medio	Errore percentuale medio
low	78	0.021805	813
medium	0	-	-
high	2	0.790026	80

Tabella: Errori delle classi per il dataset completo

Dataset Azure

- **low_bound**: 0.3610033427811713
- **high_bound**: 0.7203571782896829.

Classe	Numero elementi	Errore assoluto medio	Errore percentuale medio
low	78	0.044774	188.63
medium	0	-	-
high	2	0.588654	66.76

Tabella: Errori delle classi per il dataset Azure

Addestramento sostenibile

Addestramento sostenibile - Introduzione

L'idea alla base sarebbe quello di fermare l'addestramento studiando la derivata della curva. Siccome i valori sono discreti, si è deciso di approssimare la derivata con la differenza tra due punti consecutivi mediante la seguente formula:

$$\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

Quando la differenza tra due rapporti consecutivi è minore di una certa soglia, per un certo numero di epoche consecutive, si ferma l'addestramento del modello.

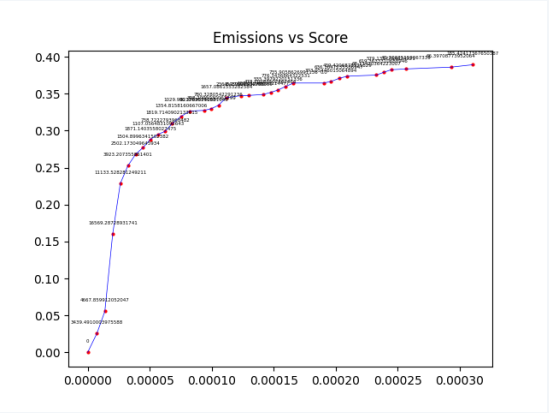


Figura: Andamento score e emissioni

Addestramento sostenibile - Esplorazione

La parte esplorativa è stata effettuata su diversi dataset utilizzando diversi valori di soglia e di epoche per cercare di capire la relazione tra i parametri, il trade-off e la dimensione del dataset. Gli esperimenti svolti sono stati:

- MovieLens1M con soglia 50 e 5 epoche
- LFM-1b_arist_20U50I_25strat con soglia 30 e 7 epoche
- Amazon_Books con soglia 40 e 6 epoche

I risultati mostrano come per alcuni modelli (DGCF esempio) il nuovo criterio di stop influenzi di molto le emissioni e il punteggio, mentre per altri modelli (DMF ad esempio) non influenzi molto. In generale all'aumentare delle dimensioni del dataset bisogna rendere il criterio meno stringente.

Addestramento sostenibile – Esempi di risultati

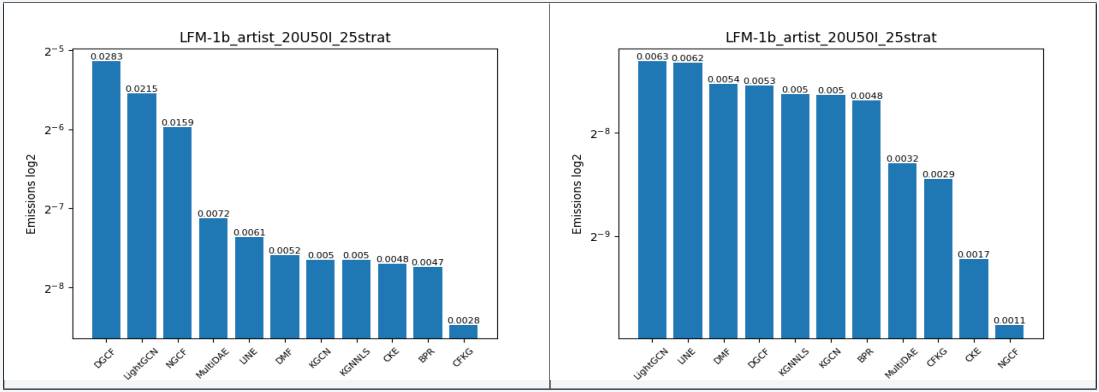


Tabella: Emissioni con criterio classico e modificato

Addestramento sostenibile - Esempi di risultati

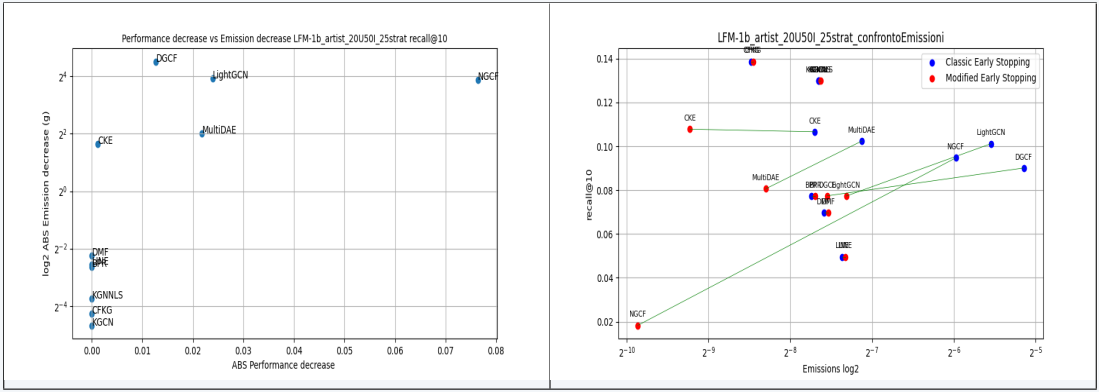


Tabella: Trade-Off

Addestramento sostenibile – Confronto criteri

Lo
step successivo è stato quello di confrontare diversi criteri di early stopping per lo stesso dataset per cercare di capire la sensibilità di questi ultimi. Abbiamo un totale di 6 esperimenti con dataset MovieLens1M:

- Soglia 40, 5 epoche
- Soglia 30, 5 epoche
- Soglia 40, 6 epoche
- Soglia 30, 6 epoche
- Soglia 40, 7 epoche
- Soglia 30, 7 epoche

Lo scopo è trovare un compromesso tra performance e sostenibilità. Analizzando i risultati grafici (come quelli prima visti) e i grafici di sensibilità, si può capire quale criterio è più adatto per un certo modello

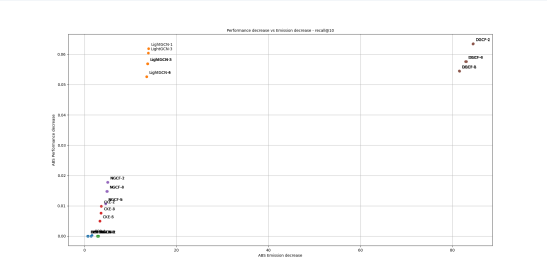


Figura: Sensibilità dei parametri con metrica Recall@10

Addestramento sostenibile – Risultati modelli

Modello	Parametro più impattante	Migliori risultati
BPR	Soglia	Soglia 40 e 6 epoche
CFKG	Soglia	Soglia 40 e 6 epoche
CKE	Epoche consecutive	Soglia 40 e 6 epoche
DMF	Nessuno predominante	Soglia 40 e 7 epoche
KGCN	Epoche consecutive	Soglia 40 e 5 epoche
KGNNLS	Soglia	Soglia 40 e 5 epoche
LINE	Soglia	Soglia 40 e 7 epoche
MultiDAE	Soglia	Soglia 40 e 7 epoche
LightGCN	Soglia	Soglia 40 e 6 epoche
NGCF	Epoche consecutive	Soglia 40 e 5 epoche
DGCF	Epoche consecutive	Soglia 40 e 6 epoche

Tabella: Parametri più impattanti e migliori risultati per ciascun modello

Tipo di Modello	Parametro predominante	Numero di Modelli	Modelli
Collaborative Filtering	Soglia	5	BPR, DMF, LightGCN, MultiDAE, LINE
Collaborative Filtering	Epoche	2	NGCF, DGCF
Knowledge Aware	Soglia	2	CFKG, KGNNLS
Knowledge Aware	Epoche	2	CKE, KGCN

Tabella: Riassunto dei parametri dominanti per tipo di modello

Conclusioni e sviluppi futuri

Conclusioni

Benchmarking

Vengono confermate le ipotesi iniziali per cui spesso i modelli più complessi hanno emissioni maggiori non giustificate da un miglioramento delle performance elevato.

Regressore

Il nuovo dataset è più ricco del precedente ma anche più sbilanciato

Addestramento sostenibile

E' possibile ridurre le emissioni di un modello di raccomandazione senza perdere in modo significativo di performance

Sviluppi futuri

Benchmarking

E' necessario effettuare più esperimenti variando dataset,modelli e hardware per avere una visione più completa del problema.

Regressore

Con più dati a disposizione si potrebbero creare modelli più complessi (come reti neurali) per cercare di migliorare le performance.

Addestramento sostenibile

Eseguire più esperimenti con altri dataset e altri hardware per confermare o meno i risultati ottenuti.

Iperparametri

Tutti gli esperimenti sono stati effettuati con iperparametri di default. Dunque tutta la fase di benchmarking e di addestramento sostenibile potrebbe essere rivista anche in termini di ricerca degli iperparametri migliori.

Grazie per l’attenzione!

Emanuele Fontana

Università degli Studi di Bari Aldo Moro