Sustainability of RecSys

Emanuele Fontana

Università degli Studi di Bari Aldo Moro

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

Indice

Sostenibilità

Sostenibilità e AI

- Capacità di soddisfare i bisogni presenti senza compromettere quelli delle generazioni future.
- Coinvolge la gestione responsabile delle risorse naturali.
- Include lo sviluppo economico e sociale.
- Mira a garantire un futuro migliore (es. Agenda 2030 dell'ONU).
- **Sustainability of AI**: si concentra sulla misurazione della sostenibilità nello sviluppo e nell'uso dei modelli AI
- **AI for Sustainability**: utilizza l'AI per affrontare le sfide della sostenibilità, come la previsione del cambiamento climatico
- La Green AI sviluppo di modelli AI che considerano il costo computazionale e l'impatto ambientale.
- In contrasto, la Red AI sviluppo di modelli sempre più complessi senza considerare le risorse impiegate.

Recommender Systems

RecSys

- Software che suggerisce all'utente elementi di interesse basandosi sulle preferenze e i comportamenti passati.
- Migliorano l'esperienza utente, aumentano la soddisfazione e la fidelizzazione.
- Utilizzano algoritmi di apprendimento automatico e intelligenza artificiale.
- Diverse tipologie di RecSys: **Collaborative Filtering**, **Content-Based**, **Hybrid**, **Knowledge-Aware**.

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.

Base di partenza

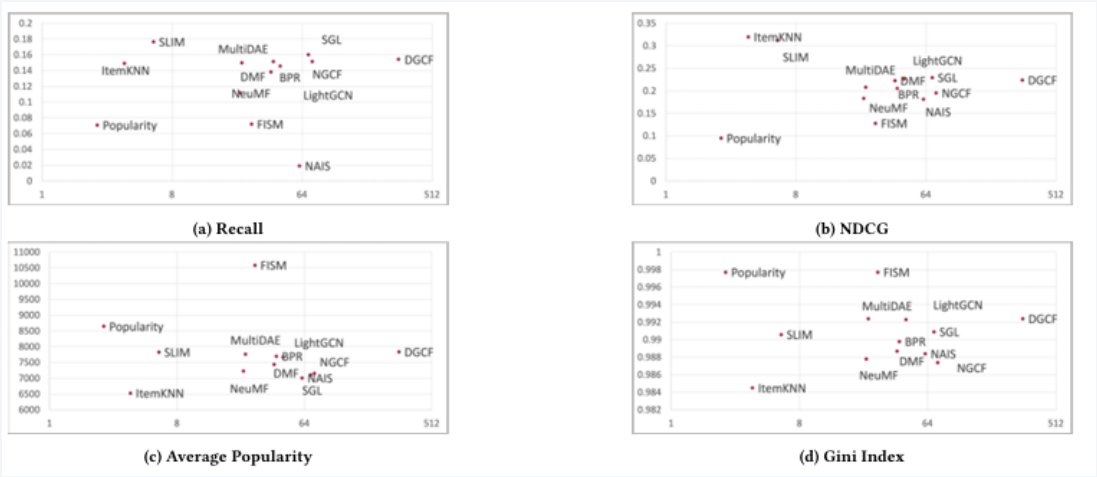


Figura: Trade-off tra emissioni e performance con dataset Mind

$$emission = CI \cdot PC$$

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

Regressore - Dataset e Modelli

Il dataset del regressore
è descritto dalle seguenti features di input:**n_users**, **n_items**, **n_inter**, **sparsity**, **kg_entities**, **kg_relations**, **kg_triples**, **kg_items**, **cpu_cores**, **ram_size**, **is_gpu**, **model_name**, **model_type**

I modelli utilizzati sono: **Random Forest**,**Decision Tree**,**AdaBoost**,**SVG**

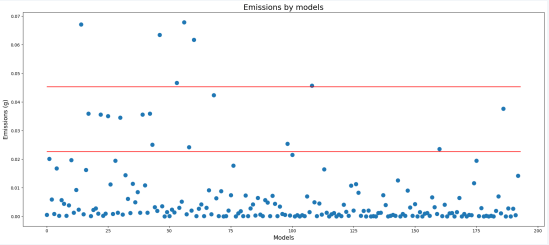


Figura: Distribuzione emissioni nel Dataset

Regressore - Analisi dei risultati

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

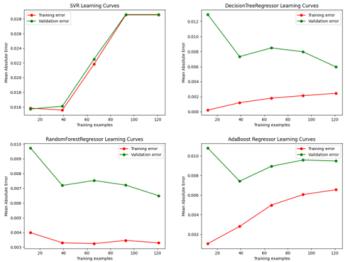


Tabella 4: Learning curve dei regressori

Benchmarking

Dataset - LFM_1b_artist

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

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

- **Filtraggio**: rimossi item e utenti con meno di 5 interazioni
- **Sampling**:sampling casuale di 50000 items e 20000 utenti
- **Stratificazione**: Per ogni utente : 75% , 50%, 25%

Dataset - MovieLens10M

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

Tabella: Statistiche MovieLens10M

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

- **Filtraggio**: rimossi item e utenti con meno di 5 interazioni
- **Sampling**:sampling casuale di 50000 utenti e 10000 items

Emissioni - Risultati

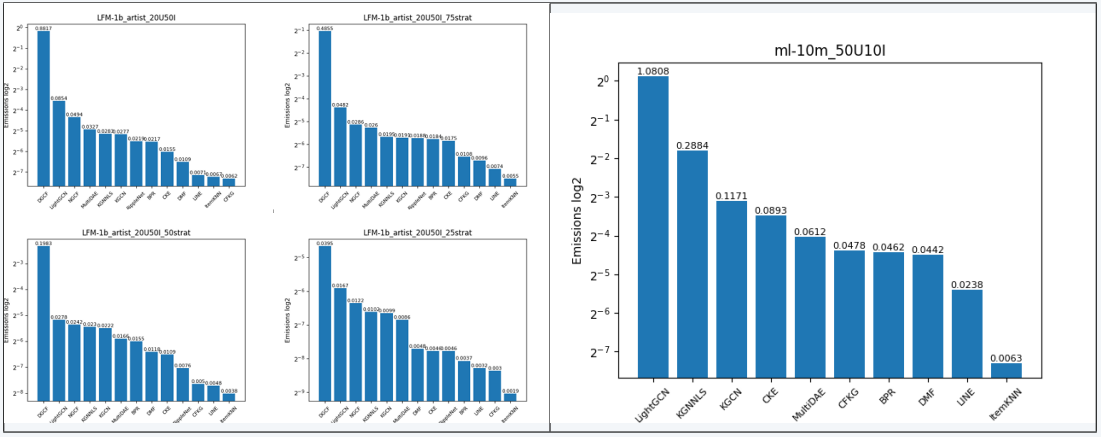


Tabella: Emissioni di CO2 per i vari modelli

Trade - Off

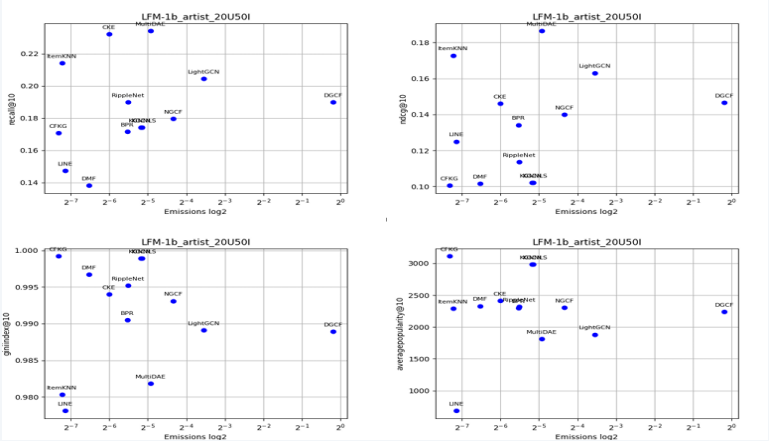


Figura: Esempio di trade-off tra emissioni e performance

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.

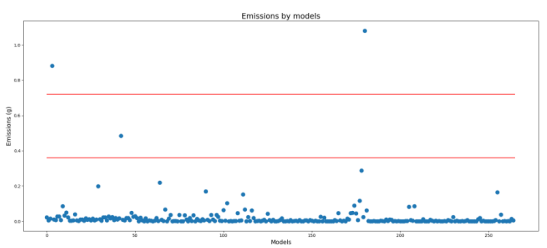


Figura: Nuova distribuzione dei dati

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

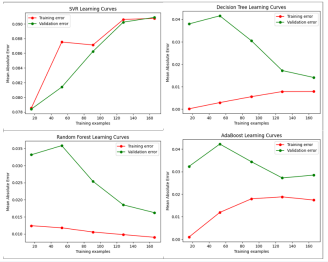


Figura: Risultati con dataset completo

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.

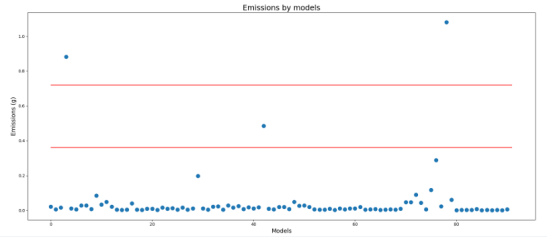


Figura: Nuova distribuzione dei dati

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

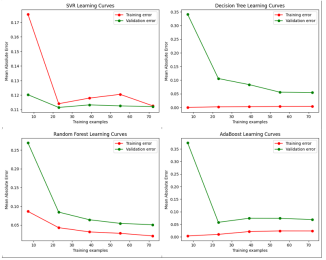


Figura: Risultati con dataset completo

Errori nelle classi

Gli errori sono stati calcolati come segue:

- **Errore assoluto**:

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

- **Errore percentuale**:

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

Dataset Completo

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

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

Approssimazione della derivata della curva:

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

L'addestramento sostenibile
si basa su una soglia di derivata e un numero di epoche consecutive in cui la derivata è sotto la soglia

Parte esplorativa

- 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
- Alcuni modelli (es. DGCF) sono molto
  sensibili al nuovo criterio, altri (es. DMF) meno

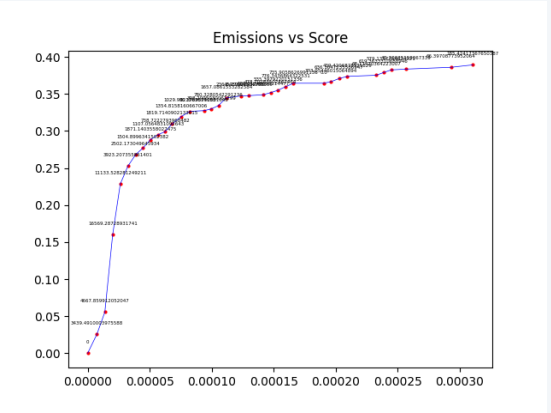
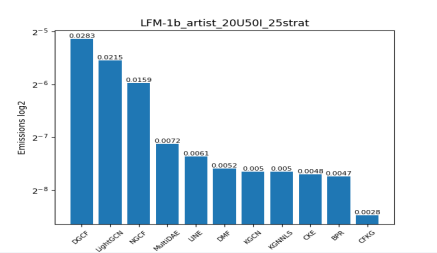
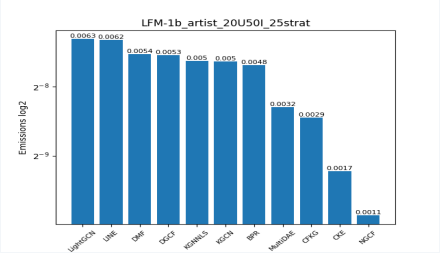
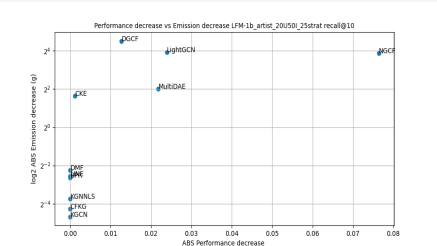


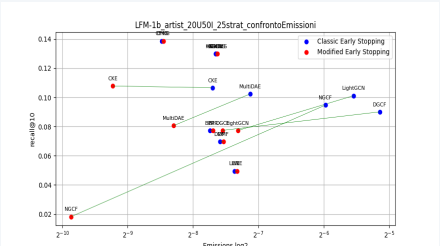
Figura: Andamento score e emissioni

Addestramento sostenibile – Esempi di risultati









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

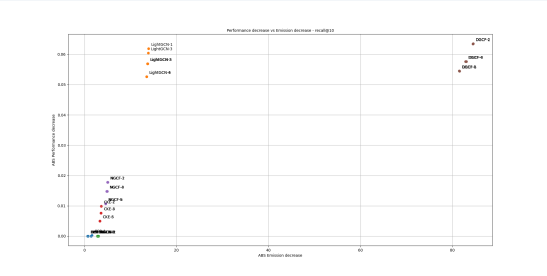


Figura: Sensibilità dei parametri con metrica Recall@10

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