

Migliorare la Business Intelligence con l'AI

Discussione di Laurea Triennale in Informatica

Riccardo Stefani
23 Luglio 2025



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- 1 Introduzione
- 2 Analisi delle vendite
- 3 Sistema di raccomandazione
- 4 Deploy e frontend
- 5 Ottimizzazione
- 6 Conclusioni e considerazioni finali

Startup innovativa fondata nel **2024** a **San Marino**

Mission

- Soluzioni **AI avanzate**
- Migliorare **efficienza aziendale**
- Focus su **LLM**
- **Agenti intelligenti**

Prodotti principali

- Automazione **processi aziendali**
- **AI Task Builder**
- **AI Chatbot Builder**



Scopo: Automatizzare la **Business Intelligence** per **e-commerce**

Analisi delle vendite

- **Input:** Dati di vendite di e-commerce (**elenco degli acquisti**)
- **Output:** Report automatico con **statistiche** e **grafici**

Sistema di raccomandazione

- **Input:** **Storico acquisti** clienti-prodotti, e **nomi significativi** dei prodotti
- **Output:** Raccomandazioni personalizzate di **prodotti** e **clienti**

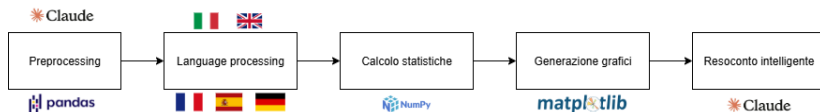
Motivazione personale

Approfondire soluzioni **AI** per **analisi dati** e **sistemi di raccomandazione**

Analisi delle vendite - Pipeline per la generazione del report



Pipeline di elaborazione per l'**analisi automatizzata** dei dati di vendita e la **generazione del report**:



Analisi delle vendite - Formati di output del report



Output dell'analisi: Il report generato viene presentato all'utente in **multipli formati**:

PDF

Formato **professionale** per archiviazione e condivisione

Generazione PDF
con ReporLab



HTML

Visualizzazione **interattiva** e responsiva nel browser

Generazione HTML
con Jinja2



Email

Invio automatico del report alla mail dell'utente

Invio di email tramite server
SMTP interno a Oribea

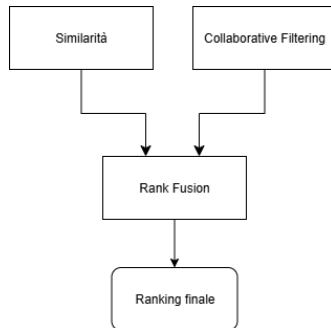


Obiettivo

Combinare i vantaggi del **Collaborative Filtering** con quelli della **similarità basata su contenuto** per ottenere raccomandazioni più accurate e robuste.

Approccio

Implementazione di un algoritmo di **Rank Fusion**, utilizzando in particolare il **Reciprocal Rank Fusion (RRF)**, per combinare efficacemente i risultati dei due sistemi, archiviati come **matrici**.



Sistema di raccomandazione - Collaborative Filtering



Un sistema di **Collaborative Filtering** sfrutta i comportamenti passati degli utenti per generare raccomandazioni.

Principio base

- Analizza **preferenze utenti simili**
- Identifica **pattern di acquisto**
- Predice **nuovi interessi**



Logica

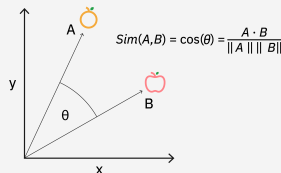
"Gli utenti con **gusti simili** nel passato avranno **preferenze simili** in futuro"

Un sistema di **Similarità** sfrutta le caratteristiche semantiche dei prodotti per generare raccomandazioni.

Principio base

- Analizza i **nomi descrittivi** dei prodotti
- Calcola la **Cosine Similarity**
- Identifica i **prodotti simili**

Cosine Similarity



Logica

"Se due prodotti hanno **caratteristiche simili**, le raccomandazioni per uno possono essere **utili anche per l'altro**"

Sistema di raccomandazione - Formato di archiviazione delle matrici



Confronto di 5 formati per ottimizzare l'archiviazione matrici:

Formati analizzati

- **CSV**: Semplice ma inefficiente
- **HDF5**: Form. binario strutturato
- **NPY**: Nativo NumPy veloce
- **Parquet**: Colonnare compresso
- **Zarr**: Array n-dimensionali **cloud**



Risultato

Zarr offre il miglior **compromesso** tra velocità e memoria

Il **Reciprocal Rank Fusion (RRF)** combina efficacemente i ranking di diversi sistemi di raccomandazione:

Processo RRF

- **Input:** Ranking da CF e Similarità
- **Calcolo:** Score reciproco per posizione
- **Fusione:** Somma dei punteggi
- **Output:** Ranking finale unificato



Svantaggio

Non considera i punteggi nei ranking di input, solo la loro posizione

Sistema di raccomandazione - Metriche di valutazione



Il sistema di raccomandazione è stato valutato utilizzando **metriche pre-filtro** e **post-filtro**:

Metriche pre-filtro

Valutano le raccomandazioni **prima** del filtro sui prodotti già acquistati:

- Recall@k
- Precision@k
- MAP@k
- MRR@k
- Unserendipity@k

Metriche post-filtro

Valutano le raccomandazioni **dopo** il filtro sui prodotti già acquistati:

- Average Item Similarity
- Mean Popularity@k

Explainability del sistema di raccomandazione per **trasparenza decisionale**, disponibile solo per l'**admin**:

Sistema di Explainability implementato

Logging di ranking e punteggio in tutte le classifiche generate, per ciascun prodotto o utente raccomandato:

- **RRF**: ranking e score finale
- **Collaborative Filtering**: ranking e punteggio
- **Similarità**: ranking e punteggio

```
> * 2025-06-23 09:35:18.176 CEST make_prediction.py - INFO - Product ranking (unfiltered, 4114 total)
> * 2025-06-23 09:35:18.176 CEST 1) The product Lelit PS25 ferro da stiro a caldaia 800 W 2,5 L Acciaio inossidabile Nero, Acciaio inossidabile is ranked 1 with score 1.3333 i...
> * 2025-06-23 09:35:18.176 CEST - It has rank 3 with score 1.2599 in the collaborative filtering-based ranking
> * 2025-06-23 09:35:18.176 CEST - It has rank 1 with score 12.8552 in the similarity-based ranking
> * 2025-06-23 09:35:18.176 CEST 2) The product Philips HI5922/30 Ferro generatore di vapore is ranked 2 with score 1.0010 in the fused ranking (Reciprocal Rank Fusion, RRF) b...
> * 2025-06-23 09:35:18.176 CEST - It has rank 1 with score 1.4782 in the collaborative filtering-based ranking
> * 2025-06-23 09:35:18.176 CEST - It has rank 1048 with score 8.6266 in the similarity-based ranking
> * 2025-06-23 09:35:18.176 CEST 3) The product Lelit PS21 ferro da stiro a caldaia 800 W 1,4 L Acciaio inossidabile Nero, Acciaio inossidabile is ranked 3 with score 0.7500 i...
> * 2025-06-23 09:35:18.176 CEST - It has rank 4 with score 1.2431 in the collaborative filtering-based ranking
> * 2025-06-23 09:35:18.176 CEST - It has rank 2 with score 12.6294 in the similarity-based ranking
```

Deploy e frontend - Integrazione con Google Cloud



Integrazione con **Google Cloud Platform** per il deployment scalabile e gestione dei dati:

Google Cloud Functions

Implementazione di due funzioni serverless:
sales-analysis e
recommendation-system

Google Cloud Storage

Archiviazione delle matrici di raccomandazione generate da **sales-analysis**, perchè siano prelevabili da **recommendation-system**



Sviluppo di **interfacce frontend** dedicate per entrambe le task, cioè sviluppo di **form** per l'invio dei dati:

Funzionalità

- **Integrazione diretta** nel sito dell'e-commerce
- **Validazione** dell'input
- Interfaccia **dedicata**, non generale
- Gestione **dinamica** del submit

Tecnologie

- React
- React Hook Form
- Zod
- Shadcn/ui



React Hook Form



Ottimizzazione del preprocessing per **migliorare le performance**:

Miglioramenti

- Introduzione delle **Pandas Vectorized Ops**
- Eliminazione delle operazioni **pandas.DataFrame.apply**
- Miglioramento dell'algoritmo delle **etichette**

Risultati nei dataset testati

- **Generale miglioramento** dei tempi di esecuzione
- Miglioramento **medio** del **20%**
- Miglioramento **massimo** vicino al **40%**

Dataset	T. pre-ott. (s)	T. post-ott. (s)	Miglioramento (%)
Swillm	110	68	38,2%
Dee	58	38	34,5%
Answer	181	120	33,7%
Orders_export	220	168	23,6%

Il periodo di **stage** presso Oribea si è concluso con **successo**, raggiungendo **tutti gli obiettivi** prefissati e **migliorando la Business Intelligence** delle aziende clienti delle task.

Risultati ottenuti

- Implementata task di **analisi vendite**
- Implementata task di **raccomandazione**
- Sviluppate **interfacce frontend** per collegamento diretto da sito e-commerce

Sviluppi futuri

- **Chatbot** per unire le due task
- Sistema di **logging avanzato**
- Introduzione della **serendipità** nelle raccomandazioni
- **Data reduction** per ridurre le dimensioni delle matrici

Grazie per l'attenzione!

Presentazione disponibile su:

[`https://github.com/Ricky03/`](https://github.com/Ricky03/)

`Presentazione-Discussione-LT-Informatica`

Ci sono domande?