



Università degli Studi di Padova

---

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI CIVITA"

*CORSO DI LAUREA MAGISTRALE IN INFORMATICA*

# Applicazione di un sistema di raccomandazione in ambito BTB

*RELATORE*

FABIO AIOLLI

UNIVERSITÀ DI PADOVA

*LAUREANDO*

DANIEL ROSSI



# Ringraziamenti

*Ecco raggiunto un altro traguardo molto importante carico di felicità che vorrei condividere con coloro che mi sono stati vicino durante questi anni.*

*Prima di tutto vorrei porgere i miei ringraziamenti al Professor Aiolli e all'azienda Estilos per l'opportunità e il supporto fornitomi durante il periodo di tesi.*

*Ringrazio Anna per avermi aiutato emotivamente durante questi anni di università, grazie di aver condiviso con me i momenti belli ma soprattutto quelli brutti, inoltre ancora grazie per avermi aiutato con la revisione di questo documento.*

*Ringrazio la mia famiglia per lo sforzo fatto nello spronarmi a ricercare un futuro in cui mi sentissi realizzato.*

*Ai miei compagni di Università, per le giornate passate insieme in Torre Archimede e su zoom, grazie di avermi fatto sentire apprezzato e coinvolto, grazie per il tempo passato insieme Mattia, Davide e Giovanni.*

*Padova, Settembre 2021*

*Daniel Rossi*



# Abstract

*La seguente tesi è il risultato della collaborazione svolta nel periodo intercorso tra marzo e agosto 2021 tra il sottoscritto, l'Università di Padova, nella persona del Professor Aiolli, e l'azienda Estilos.*

*Nel mondo della vendita online, ma non solo, sentiamo parlare ormai sempre più spesso di sistemi di raccomandazione, in questa tesi andremo ad applicare diversi suoi approcci in un contesto non del tutto usuale, ossia quello di un e-commerce business to business (BTB) di un'azienda Cliente di Estilos. Si è dovuto lavorare sullo storico vendite relativo il canale online in quanto l'e-commerce non prevede la raccolta di valutazioni da parte degli utenti sui prodotti.*

*Inserito all'interno di un quadro più ampio gli obiettivi del progetto prevedono la rielaborazione dello storico vendite in modo d'avere i dati in forme più classiche, ossia rating discreti su una scala comune, a cui poi applicare gli approcci più popolari al momento nell'ambito dei sistemi di raccomandazione, come il collaborative filtering e il content based filtering.*

*Più nello specifico, il task che si persegue è quello di raccomandare a ciascun cliente una lista di prodotti che si ritiene possano interessarlo.*

*Ci si proponeva inoltre di trovare prodotti simili e correlati dato uno di partenza, anche utilizzando informazioni esterne. Si voleva poi vagliare approcci ibridi che permettessero di combinare informazioni che descrivono l'interesse di un cliente verso un prodotto rispetto diverse espressioni d'interesse, quali per esempio la quantità totale acquistata e la recentezza dell'acquisto. Data la natura dei dati, si voleva infine provare ad affrontare il problema come un next basket recommendation, andando a considerare le fatture come sessioni d'acquisto e studiando se questo approccio funzionasse meglio dei precedenti.*



# Indice

RINGRAZIAMENTI	ii
ABSTRACT	v
LISTA DELLE FIGURE	x
LISTA DELLE TABELLE	xiii
<b>1 INTRODUZIONE</b>	<b>1</b>
1.1 Contesto progetto . . . . .	1
1.2 L'idea di progetto . . . . .	2
1.3 Organizzazione del testo . . . . .	3
1.4 Convenzioni tipografiche . . . . .	3
<b>2 ANALISI DEI DATI</b>	<b>5</b>
2.1 Principali tabelle . . . . .	5
2.1.1 Tabella VBAK . . . . .	6
2.1.2 Tabella VBAP . . . . .	6
2.1.3 Tabella KNA1 . . . . .	6
2.1.4 Tabella MARA . . . . .	6
2.2 Prodotti . . . . .	7
2.2.1 Gerarchia prodotto (PRODH) . . . . .	7
2.2.1.1 Selezione categorie di 1° livello . . . . .	9
2.2.1.2 Overview categorie di 1° livello . . . . .	9
2.2.2 Gruppo merceologico (MATKL) . . . . .	11
2.2.3 Dimensione, volume e peso . . . . .	11
2.2.4 Preprocessing dati . . . . .	11
<b>3 SISTEMI DI RACCOMANDAZIONE</b>	<b>13</b>
3.1 Introduzione . . . . .	13
3.2 Preliminari . . . . .	13
3.2.1 Feedback impliciti / espliciti . . . . .	14
3.2.2 User-item interaction matrix . . . . .	14
3.2.3 Obiettivo . . . . .	14
3.3 Approcci . . . . .	15

3.3.1	Collaborative filtering . . . . .	15
3.3.1.1	User k-nearest neighbour . . . . .	16
3.3.1.2	Item k-nearest neighbour . . . . .	16
3.3.1.3	Matrix Factorization . . . . .	16
3.3.1.4	Variational Auto-Encoder for CF (VAECF) . . .	17
3.3.2	Content-based filtering . . . . .	19
3.4	Valutazione . . . . .	19
3.4.1	AUC . . . . .	20
3.4.2	nDCG@k . . . . .	20
4	PREPROCESSING STORICO VENDITE	<b>23</b>
4.1	Preprocessing matrici grezze . . . . .	23
4.1.1	Preliminari . . . . .	23
4.1.2	Tecnica product-based . . . . .	25
4.1.3	Tecniche group-based . . . . .	25
4.1.3.1	Normalizzazione Min-Max . . . . .	26
4.1.3.2	Tecnica ordered-based . . . . .	26
4.1.4	Approccio implicito . . . . .	27
4.2	Tecniche combinate . . . . .	27
4.2.1	Premesse . . . . .	27
4.2.2	Combinazione liste <i>TopN</i> . . . . .	27
4.2.3	Media matrici dei rating . . . . .	28
4.3	Approccio content-based . . . . .	28
4.3.1	Gerarchia prodotto . . . . .	29
4.3.2	Gruppo merceologico . . . . .	29
4.3.3	Nome prodotto . . . . .	29
4.3.4	Dimensione, volume, peso . . . . .	30
4.3.5	Profilo user . . . . .	30
4.4	Approccio next-basket . . . . .	30
4.4.1	Premesse . . . . .	31
4.4.2	Popolarità . . . . .	31
4.4.3	User Popularity-based CF (UP-CF) . . . . .	31
4.4.3.1	Predizione . . . . .	32
5	LIBRERIA CORNAC ED ESPERIMENTI	<b>33</b>
5.1	Dataset . . . . .	33
5.2	Esperimenti sulle matrici grezze . . . . .	34
5.2.1	Libreria Cornac . . . . .	34
5.2.2	Esperimenti sulle singole matrici grezze . . . . .	35
5.2.3	Esperimenti sulle matrici grezze combinate . . . . .	36
5.3	Esperimenti con approccio content-based . . . . .	37



5.4	Esperimenti con approccio next-basket . . . . .	37
6	RISULTATI SPERIMENTALI	<b>39</b>
6.1	Risultati preprocessing matrici grezze . . . . .	39
6.1.1	Risultati soglia . . . . .	40
6.1.1.1	MostPop . . . . .	40
6.1.1.2	VAECF . . . . .	40
6.1.2	Fase test preliminare . . . . .	41
6.1.2.1	Normalizzazione min-max gruppo globale . . . . .	42
6.1.2.2	Normalizzazione min-max gruppo user-based . . . . .	43
6.1.2.3	Normalizzazione min-max gruppo user-category-based . . . . .	44
6.1.2.3.1	Macchine . . . . .	44
6.1.2.3.2	Ricambi . . . . .	45
6.1.2.3.3	Totale . . . . .	45
6.1.2.4	Tecnica ordered-based gruppo globale . . . . .	47
6.1.2.5	Tecnica ordered-based gruppo user-based . . . . .	48
6.1.2.6	Tecnica ordered-based gruppo user-category-based . . . . .	49
6.1.2.6.1	Macchine . . . . .	49
6.1.2.6.2	Ricambi . . . . .	50
6.1.2.6.3	Totale . . . . .	50
6.1.2.7	Tecnica product-based . . . . .	52
6.1.3	Fase test avanzata . . . . .	53
6.2	Risultati matrici grezze combinate . . . . .	55
6.2.1	Combinazione liste <i>TopN</i> . . . . .	56
6.2.2	Media matrici dei rating . . . . .	57
6.3	Risultati con approccio next-basket . . . . .	58
6.4	Confronto critico . . . . .	59
7	CONCLUSIONI	<b>61</b>
7.1	Problemi critici risolti . . . . .	61
7.2	Problemi aperti . . . . .	62
7.3	Futuri sviluppi . . . . .	62
	GLOSSARIO	<b>63</b>
	ACRONIMI	<b>63</b>
	RIFERIMENTI BIBLIOGRAFICI	<b>67</b>



# Lista delle figure

2.1	Diagramma ER tabelle Mara, Kna1, Vbak e Vbap. . . . .	7
2.2	Diagramma ER tabelle Mara e T179. . . . .	8
3.1	Esempio user-item interaction matrix esplicita. . . . .	14
3.2	Rappresentazione visiva delle matrici R, P e Q. . . . .	17
3.3	Rappresentazione grafica del VAE. . . . .	18
6.1	Risultati normalizzazione min-max gruppo globale. . . . .	42
6.2	Risultati normalizzazione min-max gruppo user-based. . . . .	43
6.3	Risultati normalizzazione min-max gruppo user-category based dataset macchine. . . . .	44
6.4	Risultati normalizzazione min-max gruppo user-category based dataset ricambi. . . . .	45
6.5	Risultati normalizzazione min-max gruppo user-category based dataset totale. . . . .	46
6.6	Risultati ordered-based gruppo globale. . . . .	47
6.7	Risultati ordered-based gruppo user-based. . . . .	48
6.8	Risultati ordered-based gruppo user-category based dataset macchine. . . . .	49
6.9	Risultati ordered-based gruppo user-category based dataset ricambi. . . . .	50
6.10	Risultati ordered-based gruppo user-category based dataset totale. . . . .	51
6.11	Risultati tecnica product-based. . . . .	52
6.12	Risultati MF ottimizzato su test set con matrici selezionate. . . . .	54
6.13	Risultati UserKnn ottimizzato su test set con matrici selezionate. . . . .	54
6.14	Tabella risultati finali sul test set. . . . .	55
6.15	Informazioni sul modello e la tecnica dei risultati della tabella 6.14 . . . . .	55
6.16	Grafico composto metodo basato su combinazione liste <i>TopN</i> . . . . .	56
6.17	Grafico composto dei risultati metodo basato sulla media matrici grezze. . . . .	57



# Lista delle tabelle

2.1	Categorie (PRODH) 1° livello, unità totali e vendite. . . . .	9
2.2	Informazioni sulle categorie (PRODH) 1° livello selezionate . . . .	9
6.1	Risultati modello MostPop per tipo dataset sul validation set . .	40
6.2	Risultati modello MostPop per tipo dataset sul test set . . . . .	40
6.3	Parametri selezionati. . . . .	41
6.4	Risultati VAE CF ottimizzato per tipo dataset sul validation set. .	41
6.5	Risultati VAE CF ottimizzato per tipo dataset sul test set. . . . .	41
6.6	Riassunto tecniche che hanno superato la fase preliminare per dataset. . . . .	53
6.7	Tabella risultati finali combinazione liste <i>TopN</i> . . . . .	56
6.8	Tabella risultati finali del metodo basato su liste <i>TopN</i> sul test set.	57
6.9	Tabella parametri selezionati. . . . .	58
6.10	Tabella risultati approccio next-based sul validation set. . . . .	58
6.11	Tabella risultati approccio next-based sul test set. . . . .	58



# 1

## Introduzione

In questo capitolo introduttivo vedremo l'idea e il contesto del progetto, ne delineeremo gli obiettivi e riporteremo un elenco dei capitoli con una loro breve descrizione.

### 1.1 Contesto progetto

Nel mondo dei software Enterprise Resource Planning (ERP), ossia prodotti software pensati per le aziende che permettono la gestione e il controllo dei processi e delle funzioni aziendali, uno dei più famosi è di certo il gestionale SAP, il quale è sviluppato in moduli integrabili che, a seconda delle esigenze dell'azienda utilizzatrice, possono essere attivati in qualunque combinazione.

Uno di questi moduli è l'e-commerce Hybris, utilizzato dalle aziende come canale di vendita online e alcune delle sue potenzialità sono: l'alto livello di personalizzazione e la possibilità di essere perfettamente integrato con i sistemi SAP, come per esempio con il modulo Customer Relationship Management (CRM), il quale si occupa di tutte le modalità di gestione delle relazioni con il cliente.

## 1.2 L'idea di progetto

Il progetto nasce, in un'ottica di innovazione del prodotto, all'interno di un progetto aziendale che mira all'ampliamento e miglioramento delle funzionalità di Hybris. Uno degli aspetti su cui si vuole lavorare è quello della personalizzazione dei prodotti mostrati agli utenti dell'e-commerce: si vuole quindi sperimentare raccomandazioni sui prodotti basate sullo storico vendite e non sui feedback lasciati dall'utente, in quanto la loro raccolta non è prevista dal sistema trattandosi di un e-commerce BTB (dove gli acquirenti sono *dealer*<sup>[g]</sup>, ossia aziende che a loro volta rivendono i prodotti). Partendo quindi dallo storico vendite di un'azienda Cliente, con Hybris configurato in versione BTB, l'obiettivo era quello di utilizzare i dati disponibili per raccomandare a ciascun cliente una lista di prodotti *TopN* che gli risultassero interessanti.

Inoltre per ciascun prodotto si vuole presentare una lista di prodotti simili ad esso, sempre interessanti per il cliente a cui viene mostrato quello specifico articolo.

Come detto, solitamente si parte da feedback impliciti/espliciti dati dagli utenti ai prodotti, ma non essendo disponibili si cercherà di estrarre informazioni relative all'interesse del cliente rispetto diversi punti di vista, quali può essere la quantità acquistata, la recentezza dell'acquisto, il numero di fatture in cui compare o la spesa totale per quello specifico articolo.

Una volta che le informazioni sono state organizzate in *matrici grezze user-item*<sup>[g]</sup>, si voleva eseguire una sorta di *preprocessing*<sup>[g]</sup> su di esse, andando a trasformarle in delle *matrici dei rating*<sup>[g]</sup> rispetto una scala comune che fornisse una misura d'interesse del cliente.

Sono state applicate le tecniche più popolari usate nei sistemi di raccomandazione, quali il collaborative filtering alle rating matrix ottenute dal preprocessing descritto precedentemente e il content-based filtering ai dati descrittivi dei prodotti. Data la non disponibilità di rating si è poi pensato di considerare il problema anche come una next basket recommendation, dove si vanno a considerare le sessioni d'acquisto e in base a queste si predice quella finale, questo approccio potrebbe funzionare nel caso in cui i clienti acquistino spesso gli stessi prodotti.



## 1.3 Organizzazione del testo

Di seguito viene riportata per ogni capitolo una piccola descrizione delle tematiche trattate:

- **Capitolo 2:** organizzazione dei dati, come sono stati trattati e quali informazioni si sono potute ricavare;
- **Capitolo 3:** breve riepilogo della teoria sui sistemi di raccomandazione, spiegando meglio gli approcci del collaborative filtering e del content based filtering, oltre che descrivendo il funzionamento degli algoritmi utilizzati e delle metriche;
- **Capitolo 4:** descrizione del preprocessing delle matrici grezze, dei metodi per combinarle, dell'approccio content-based e next-basket recommendation;
- **Capitolo 5:** descrizione di come si sono attuati i test per ciascun approccio;
- **Capitolo 6:** report sui risultati ottenuti e breve analisi dei risultati per ciascun metodo;
- **Capitolo 7:** conclusioni del lavoro, questioni aperte e possibili futuri sviluppi.

## 1.4 Convenzioni tipografiche

Il testo adotta le seguenti convenzioni tipografiche:

- ogni acronimo, abbreviazione, parola ambigua o tecnica viene spiegata e chiarita alla fine del testo;
- ogni parola di glossario alla prima apparizione verrà etichetta come segue *parola*<sup>[g]</sup>.



# 2

## Analisi dei dati

Nel seguente capitolo vedremo com'è composto lo storico ordini e quali solo le informazioni disponibili, inoltre descriveremo con maggior dettaglio alcuni 'campi'. Inoltre verranno descritte alcune operazioni di preparazione dei dati.

### 2.1 Principali tabelle

Lo storico vendite dell'azienda Cliente è stato estratto dal modulo hybris, questo è organizzato secondo le tabelle SAP, avremo quindi lo storico delle fatture, composto da una tabella per la testata della fattura (VBAK), ossia la parte descrittiva dove viene riportato l'acquirente, e una tabella per le posizioni della fattura (VBAP), ossia la parte dove vengono riportati i materiali acquistati. Abbiamo inoltre due tabelle che contengono rispettivamente l'anagrafica cliente (KNA1) e materiali (MARA), dove possiamo trovare informazioni aggiuntive che li descrivono. Mi è stato inoltre fornito un glossario che riportava per ciascuna tabella una breve spiegazione di ogni campo. Tutte queste tabelle sono in formato Excel. Vediamo ora per ciascuna di queste tabelle i campi annessi e alcune informazioni di natura statistica.

### 2.1.1 Tabella VBAK

La tabella VBAK contiene la testata di circa 35000 fatture, datate dall'anno 2016 fino a maggio 2021.

Ciascuna riga della tabella è la testata di una fattura e ne riporta il suo codice identificativo (**VBELN**) insieme con il codice del cliente a cui è associata (**KUNNR**). Inoltre ciascuna fattura riporta data e ora (**ERDAT**, **ERZET**) in cui è stata emessa, l'importo totale e la valuta corrispondente (**NETWR**, **WAERK**).

### 2.1.2 Tabella VBAP

La tabella VBAP contiene le posizioni delle fatture (circa 250000), riporta per ognuna di esse la lista di prodotti acquistati indicando diverse informazioni. Ciascuna riga della tabella riporta quindi il codice identificativo (**VBELN**) della fattura e il codice identificativo del prodotto acquistato (**MATNR**), poi vengono riportati per quel prodotto il prezzo unitario (**NETPR**), la quantità acquistata (**KWMENG**), la spesa totale con la valuta (**NETWR**, **WAERK**) e il codice gerarchia prodotto storico (**PRODH**), ossia quello salvato in MARA al momento dell'emissione della fattura.

### 2.1.3 Tabella KNA1

La tabella KNA1 riporta l'anagrafica cliente (circa 3000), per ciascuna riga abbiamo il codice cliente (**KUNNR**), il codice paese d'origine (**LAND1**), il nome dell'azienda (**NAME1**), la località (**ORT01**) e la regione (**REGIO**).

### 2.1.4 Tabella MARA

Come detto la tabella MARA è quella che riporta l'anagrafica dei materiali, nella nostra futura trattazione considereremo questi materiali come prodotti in quanto sono acquistabili all'interno dell'e-commerce.

Ciascuna riga della tabella riporta quindi un prodotto univoco composto dal suo usuale codice identificativo (**MATNR**), dal codice della gerarchia prodotto (**PRODH**) e del gruppo merceologico (**MATKL**) e una breve descrizione testuale (**MAKTX**), poi abbiamo delle informazioni su dimensione, volume e peso. Per le dimensioni abbiamo un campo (**GROES**) che le fornisce nel formato

lunghezza X larghezza X altezza, oppure altri (**LAENG**, **BREIT**, **HOEHE**), che indicano rispettivamente lunghezza, larghezza e altezza e l'unità di misura per entrambi i formati viene riportata nello stesso campo (**MEABM**).

Poi abbiamo due campi per volume e peso (**VOLUM**, **NTGEW**) e altri due per le loro rispettive unità di misura (**VOLEH**, **GEEWI**).

Vediamo ora nella Figura 2.1 il diagramma ER delle tabelle sopra descritte con le relative dipendenze.

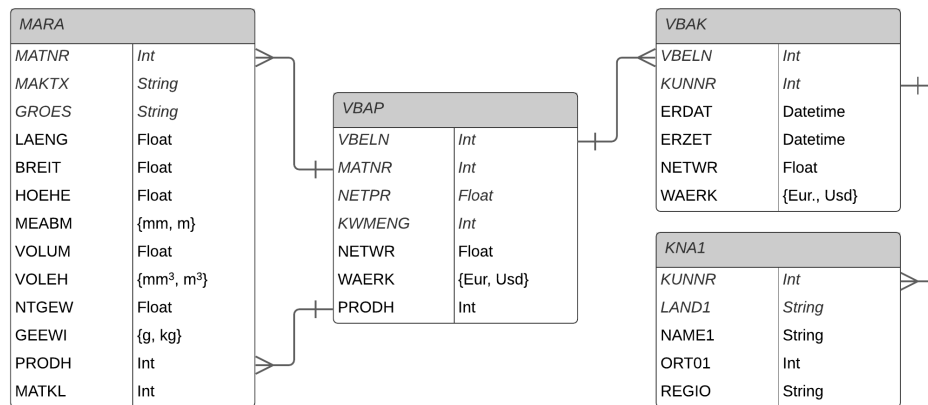


Figure 2.1: Diagramma ER tabelle Mara, Kna1, Vbak e Vbap.

## 2.2 Prodotti

Da questo momento in poi faremo riferimento ai materiali chiamandoli prodotti come detto in precedenza.

In totale nella tabella anagrafica materiali (MARA) sono presenti circa 75000 prodotti diversi, mentre i prodotti effettivamente venduti risultano essere molti meno attestandosi all'incirca verso gli 8000.

Abbiamo però due campi interessanti che riguardano la gerarchia prodotto (PRODH) e il gruppo merceologico (MATKL), questi due ci permettono di studiare la similarità dei prodotti.

### 2.2.1 Gerarchia prodotto (PRODH)

Il campo gerarchia prodotto (PRODH) è un campo numerico di 18 cifre utile per separare i prodotti rispetto le diverse categorie su più livelli. Nella tabella secon-

daria T179 vengono definiti i livelli di gerarchia e le diverse categorie, abbiamo il codice gerarchia prodotto (**PRODH**), il livello (**STUFE**) e il titolo (**VTEXT**), quindi la tabella Mara fa riferimento a questa come mostrato nella Figura 2.2.

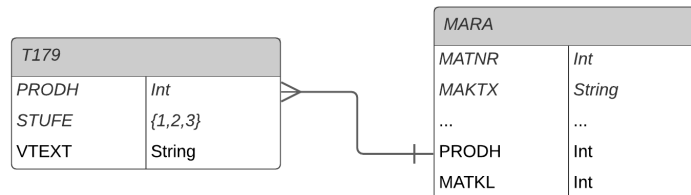


Figure 2.2: Diagramma ER tabelle Mara e T179.

Ciascun codice PRODH contenuto nella tabella T179 avrà rispettivamente il seguente numero di cifre in base al livello di gerarchia (STUFE):

- **STUFE = 1:** 1° livello della gerarchia, il codice sarà di 5 cifre;
- **STUFE = 2:** 2° livello della gerarchia, il codice sarà di 10 cifre, dove le prime 5 identificano la categoria di 1° livello a cui appartengono mentre le restanti 5 indentificano la sotto-categoria di 2° livello;
- **STUFE = 3:** 3° livello della gerarchia, il codice sarà di 18 cifre, dove le prime 10 identificano la categoria di 2° livello a cui appartengono mentre le restanti 8 indentificano la sotto-categoria di 3° livello.

Ciascun prodotto sarà quindi provvisto di un codice di 18 cifre che identificherà una categoria per ogni livello.

Nella tabella VBAP ci sono alcune posizioni dove a parità di codice prodotto (MATNR) si hanno codici PRODH diversi, questo è dovuto al diverso momento temporale in cui sono stati acquistati, infatti nella tabella VBAP il codice PRODH è storico, ho provveduto per semplicità ad aggiornarli tutti al codice PRODH più recente riportato nella tabella anagrafica materiali MARA. Il numero di prodotti interessati sono circa 100 su 10000 posizioni.

### 2.2.1.1 Selezione categorie di 1° livello

PRODH	#tot	#sold	titolo
00010	28	0	categoria1
00020	0	0	categoria2
00030	0	0	categoria3
00040	5	0	categoria4
00050	2	0	categoria5
00090	2	0	categoria6
00100	1117	173	CATEGORIA1
00200	645	130	CATEGORIA2
00250	31	11	CATEGORIA3
00300	405	92	CATEGORIA4
00400	525	36	CATEGORIA5
00500	1715	70	CATEGORIA6
00600	334	6	CATEGORIA7
00700	1	0	CATEGORIA8
00900	70441	7702	CATEGORIA9
00950	28	1	CATEGORIA10
09999	215	0	ALTRO

Table 2.1: Categorie (PRODH) 1° livello, unità totali e vendute.

Chiaramente il fatto che siano tutti a zero è dovuto all’aggiornamento dei codici PRODH di cui abbiamo parlato precedentemente, a prescindere da ciò il numero di posizioni che prima riportavano codici appartenenti alle categorie prese in considerazione non superava la decina, quindi non considerare queste categorie in quanto si è smesso di usarle sembra la scelta più logica.

La categoria ALTRO (09999) non è stata considerata in quanto riporta prodotti che non sono disponibili sull’e-commerce. Inoltre le categorie CATEGORIA8 (00700) e CATEGORIA10 (00950), dato il basso numero di prodotti presenti in MARA e le basse vendite, si è preferito non considerarle.

### 2.2.1.2 Overview categorie di 1° livello

PRODH	#posizioni	$\sum_{KWMENG}$	$\mathbb{E}_{KWMENG}$	$\mathbb{E}_{NETPR}$ (€)	$\sum_{NETWR}$ (€)	$\mathbb{E}_{NETWR}$ (€)	titolo
00100	5908	15461	2.61	1613.44	3865.97	22840167.61	CATEGORIA1
00200	1936	3219	1.66	5898.09	8640.59	16745496.87	CATEGORIA2
00250	333	2949	8.85	552.99	3772.04	1377422.72	CATEGORIA3
00300	745	1390	1.86	4353.24	5364.34	3996430.94	CATEGORIA4
00400	389	1651	4.24	708.17	2404.47	940777.84	CATEGORIA5
00500	1133	12984	11.46	175.75	1034.63	1172236.58	CATEGORIA6
00600	153	494	3.23	448.52	1338.80	83501.04	CATEGORIA7
00900	239740	1070334	4.46	26.67	66.61	15968039.56	CATEGORIA9
	250339	1108493.51	4.72	1706.86	3225.75	63124073.16	valori riassuntivi

Table 2.2: Informazioni sulle categorie (PRODH) 1° livello selezionate

Nella tabella 2.2 per ogni categoria *PRODH* di 1° livello possiamo vedere:

- $\#posizioni$ : numero di posizioni in cui compaiono prodotti di quella categoria in fattura;
- $\sum_{KWMENG}$ : quantità totale di prodotti acquistati appartenenti a quella categoria;
- $\mathbb{E}_{KWMENG}$ : quantità media per fattura di prodotti acquistati appartenenti a quella categoria;
- $\mathbb{E}_{NETPR}$ : prezzo medio per fattura di prodotti acquistati di quella categoria;
- $\sum_{NETWR}$ : spesa totale per prodotti di quella categoria;
- $\mathbb{E}_{NETWR}$ : spesa totale media per fattura di prodotti di quella categoria.

Dalla tabella possiamo vedere come la categoria RICAMBI & ACCESSORI riporti un prezzo medio per fattura molto più basso rispetto alle altre categorie, questo è dovuto al fatto che i pezzi di ricambio ed accessori non sono macchine o sistemi da usare per fornire un servizio quanto un prodotto per riparare ciò che già si possiede. Possiamo vedere che in termini di posizioni l'acquisto di pezzi di ricambio copra una cospicua parte delle posizioni in fattura, oltre che costituire un'importante parte del fatturato per l'azienda. Le altre categorie indicano macchine e sistemi per la pulizia quindi il prezzo medio per prodotto è molto maggiori e per i clienti finali questi rappresentano un investimento.

Da quanto detto finora si vengono a creare due macro categorie di prodotti:

- **Macchine**: questa macro categoria racchiude sette categorie di 1° livello (00100, 00200, 00250, 00300, 00400, 00500, 00600);
- **Ricambi**: questa macro categoria invece racchiude la sola categoria 00900.

Dobbiamo dare un'ultima precisazione, la maggior parte delle categorie di 2° e 3° livello risultano essere di prodotti appartenenti alla macro categoria delle macchine, quindi la gerarchia è molto più densa orizzontalmente per le macchine rispetto che i pezzi di ricambio, infatti per le macchine le categorie risultano essere i diversi modelli di macchinari disponibili e i prodotti a catalogo di quella categoria sono le varianti dello stesso modello di macchinario. Per i pezzi di ricambio abbiamo solo poche categorie contenitore che li raggruppano tutti insieme.



### 2.2.2 Gruppo merceologico (MATKL)

Il gruppo merceologico (MATKL) non è organizzato come una gerarchia, come lo è invece la gerarchia prodotto (PRODH), bensì come un insieme di prodotti, in totale abbiamo circa 160 gruppi, dove uno di questi contiene tutti i prodotti che prima abbiamo classificato come macchine. Rispetto il codice PRODH, il gruppo merceologico è più divisivo rispetto i ricambi, questo ci può aiutare in quanto ora siamo in grado di categorizzare anche i ricambi.

### 2.2.3 Dimensione, volume e peso

I campi riguardanti dimensione, volume e peso potrebbero essere utili per ricercare una similarità tra i prodotti.

Le informazioni sulle dimensioni, come lunghezza, larghezza e altezza sono praticamente ridondanti nei campi GROES e (LAENG, BREIT, HOEHE) se non per alcuni prodotti dove le informazioni sono esclusive di uno dei due formati.

Per peso e volume abbiamo i rispettivi campi numerici e altri due campi che riportano le unità di misura, per il volume possono essere i metri cubi o i millimetri cubi, per il peso i kilogrammi o i grammi. La criticità di queste misure riguarda la loro scarsità, infatti su 75000 prodotti abbiamo informazioni su volume e peso rispettivamente solo sul 20% e 39%, mentre sui prodotti acquistati almeno una volta sul 19% e 5%.

### 2.2.4 Preprocessing dati

In questa sezione andremo a vedere quali operazioni si sono rese necessarie per preparare i dati:

- posizioni con spesa totale nulla sono stati eliminati;
- posizioni con valuta in *Usd* sono state convertiti in *Eur*;
- prodotti con misure di dimensione e volume, con unità di misura diversa rispettivamente dai *metri* e *metri*<sup>3</sup>, sono stati convertiti negli stessi;
- prodotti con misure del peso con unità di misura diverse dal *kilogrammo* sono state convertite nello stesso;

- alcuni clienti non sono stati considerati in quanto appartenenti allo stesso gruppo dell'azienda Cliente.

# 3

## Sistemi di raccomandazione

In questo capitolo vedremo un riassunto della teoria riguardante i sistemi di raccomandazione, dei suoi diversi approcci e degli algoritmi presenti nella tesi.

### 3.1 Introduzione

Uno dei campi più popolari al momento verso cui si rivolge una particolare attenzione è quello dei sistemi di raccomandazione<sup>12</sup> (RS), in quanto l'attività online sta aumentando sempre più e nascono sempre più spesso nuovi servizi che permettono di scegliere oggetti, siano questi prodotti, video, musica, film o molto altro, da cataloghi vastissimi. I sistemi di raccomandazione permettono di navigare questi cataloghi andando a cercare gli oggetti che risultino più interessanti per l'utente.

### 3.2 Preliminari

In generale possiamo dire che un RS si compone di diversi elementi, in primo luogo abbiamo i cosiddetti 'attori' del problema, gli user e gli item, rispettivamente gli utenti del sistema (U) e gli oggetti che si vuole consigliare (I). Abbiamo a disposizione inoltre informazioni riguardo l'interazione tra user e item solitamente

sotto forma di feedback implicito o esplicito, questa misura viene definita rating. Questi vengono utilizzati dal RS, insieme con eventuali dati legati al contesto di user e item, per effettuare raccomandazioni.

### 3.2.1 Feedback impliciti / espliciti

Solitamente le informazioni che legano user e item, ossia i rating, possono essere di due tipi:

- Implicito: 1 se c'è stata interazione tra lo user e l'item, 0 se non c'è stata;
- Esplicito: valutazione numerica nell'intervallo  $[1, scale]$ , 0 se non c'è stata interazione.

Nel nostro caso di studio però ci ritroviamo a metà strada in quanto, se per esempio considerassimo la quantità come un dato esplicito ci troveremmo così ad avere un dato su una scala non continua, mentre se lo facessimo come se fosse implicito trascureremmo delle informazioni che possono in qualche modo fornire una misura di interesse.

### 3.2.2 User-item interaction matrix

		<i>Items</i>					
		<i>1</i>	<i>2</i>	...	<i>i</i>	...	<i>m</i>
<i>Users</i>	<i>1</i>	5	3		1	2	
	<i>2</i>		2				4
	:			5			
	<i>u</i>	3	4		2	1	
	:					4	
	<i>n</i>			3	2		

Figure 3.1: Esempio user-item interaction matrix esplicita.

I rating sono organizzati in matrici, dette user-item interaction matrix o semplicemente matrici dei rating ( $R$ ), dove sulle righe abbiamo gli utenti mentre sulle colonne abbiamo gli oggetti, nell'incrocio abbiamo riportato il rating. La matrice come detto può essere implicita o esplicita e le celle vuote corrispondono allo 0. Quando scriviamo  $r_{ui}$  intendiamo che  $u$  è lo user e  $i$  è l'item.

### 3.2.3 Obiettivo

L'obiettivo del sistema può essere quello di consigliare ad uno user una lista di  $N$  item, detta  $TopN^{[g]}$  che si ritiene possano interessargli, oppure dato un item si può trovare una lista di item che si considerino simili allo stesso in accordo con i 'gusti' dello user.

### 3.3 Approcci

Definito quindi il task abbiamo diversi modi per poter soddisfare il nostro obiettivo, in generale abbiamo due principali categorie di RS:

- **Non Personalizzato:** andiamo a consigliare i prodotti che globalmente risultano più popolari, ossia che abbiano complessivamente ricevuto più valutazioni, o quelli con rating più alto. Questo approccio non va a considerare le informazioni relative al singolo user;
- **Personalizzato:** ci sono diversi approcci che vedremo nelle sezioni successive, in generale si fanno raccomandazioni basate sulla similarità tra user. I due approcci più famosi sono il collaborative filtering, dove si cerca di consigliare item ad uno user basandosi su user simili, e il content-based filtering si cerca di raccomandare item simili a quelli con cui si ha già interagito.

Nelle sezioni successive andiamo a spiegare più nel dettaglio il collaborative e content-based filtering.

#### 3.3.1 Collaborative filtering

Il collaborative filtering<sup>11</sup> è un approccio agli RS basato sulla similarità, raccomandiamo ad uno user item interessanti per altri user simili ad esso, e viceversa item simili ad altri item per cui ha dimostrato interesse. La similarità può essere quindi di due tipi: item-based, basata quindi sulla similarità tra prodotti o user-based ossia su quella tra user. Ci sono due approcci possibili al collaborative filtering:

- Memory-based: utilizziamo la matrice dei rating per calcolare la similarità tra user e item, metodi basati sull'algoritmo K nearest neighbour;
- Model-based: utilizziamo dei modelli che attraverso degli algoritmi permettono di predire il rating su item non valutati.

### 3.3.1.1 User k-nearest neighbour

User k-nearest neighbour (UserKnn) è un metodo memory-based che fa uso della matrice dei rating, ogni user avrà quindi un proprio 'profilo', ossia la propria riga nella matrice dei rating. Per esempio immaginiamo di avere dei rating impliciti, l'idea è quella di calcolare la similarità tra tutti gli user, ad esempio tra due user  $u$  e  $v$  questa sarà calcolata con la funzione coseno  $s_{uv} = \frac{|I_u \cap I_v|}{\sqrt{|I_u||I_v|}}$ , dove  $I_u$  sono gli item valutati dallo user  $u$ . Dopo questa operazione è possibile calcolare il rating previsto per ogni item  $i$  non valutato rispetto ad uno user  $u$  con la seguente formula  $\hat{r}_{ui} = \frac{1}{k} \sum_{v \in N_i^k(u)} s_{uv}$ , dove  $N_i^k(u)$  è la lista dei  $k$  user, che hanno valutato l'item  $i$ , più simili allo user  $u$ .

Più in generale selezioniamo i  $k$  user con similarità più alta con il nostro user target e calcoliamo la media pesata dei loro rating usando come pesi la similarità. Fatto questo si procede ad ordinare per ciascuno user tutti i prodotti secondo i rating ottenuti e si ottiene così la lista *TopN* degli item più interessanti.

Questo metodo funziona senza informazioni relative alle caratteristiche degli user o item e può gestire rating espliciti o impliciti con formule leggermente diverse.

### 3.3.1.2 Item k-nearest neighbour

Item k-nearest neighbour (ItemKnn) è un metodo memory-based molto simile al precedente, qui si va a considerare però gli item da raccomandare e il loro profilo è la propria colonna della rating matrix. Si calcola la similarità tra tutti gli item e dato uno user si procede a calcolare il rating stimato sugli item che non ha valutato, trovando per ciascuno di essi la lista di  $K$  item più simili ad esso, poi calcola la media pesata dei rating dei  $K$  item selezionati usando come peso la similarità. Le formule sono molto simili alle precedenti solo che al posto di usare il profilo dello user si usa quello dell'item.

### 3.3.1.3 Matrix Factorization

Nell'approccio model-based, Matrix Factorization (MF) è uno dei modelli più famosi, questo si basa sul concetto che si possa mappare user e item verso uno spazio delle feature comune di una certa dimensionalità  $k$ . Creiamo quindi due matrici  $P$  e  $Q$ , dove  $P$  è una matrice avente sulle righe gli user e come colonne le  $k$  feature, mentre  $Q$  è una matrice avente sulle righe le  $k$  feature e sulle colonne

gli item. Vogliamo ottenere una approssimazione della rating matrix attraverso la moltiplicazione di  $P$  e  $Q$ , ossia  $R \approx P \cdot Q^T = \hat{R}$ , come nella Figura 3.2.

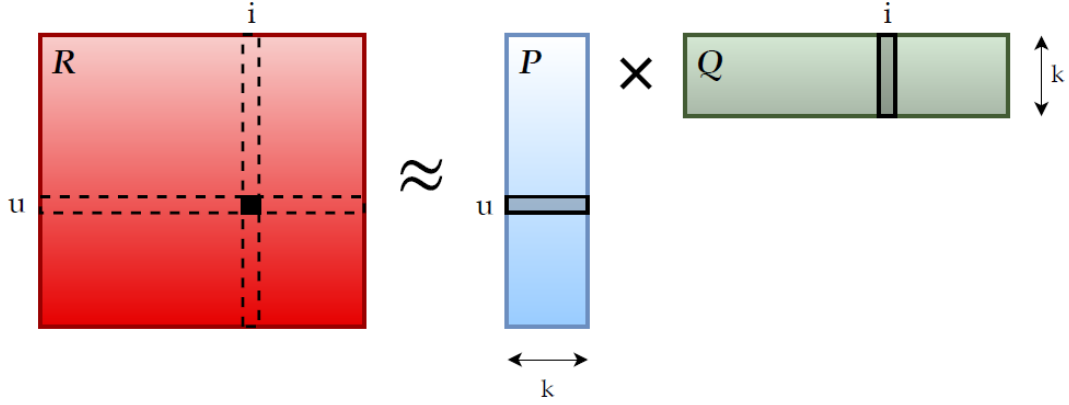


Figure 3.2: Rappresentazione visiva delle matrici  $R$ ,  $P$  e  $Q$ .

Quello che otteniamo è quindi un profilo sia per gli user che per gli item rispetto lo stesso spazio delle feature. Il problema maggiore risulta però essere quello di ottenere queste due matrici, possiamo farlo definendo la seguente funzione di loss  $L(P, Q) = \min_{P, Q} \|R - PQ^T\|^2 + \lambda(\|P\|^2 + \|Q\|^2)$ . Alleniamo il modello fissando in modo alternato le matrici  $P$  e  $Q$ , cercando quindi di ridurre dopo ciascuna iterazione la differenza tra  $R$  e  $PQ^T$ , la  $\lambda$  è un hyper-parametro utile per la regolarizzazione, infatti permette di penalizzare in base alla norma delle matrici  $P$  e  $Q$ , facendo sì che queste abbiano una norma vicina allo zero.

Una volta che il modello è allenato possiamo predire il rating dato da uno user  $u$  su un item  $i$  moltiplicando i profili corrispondenti  $\hat{r}_{ui} = P_u \cdot Q_i^T$ .

#### 3.3.1.4 Variational Auto-Encoder for CF (VAECF)

Il VAECF è un Variational Auto-Encoder<sup>5</sup> (VAE) che si basa sull'Auto-Encoder<sup>1</sup> (AE), una rete neurale composta da due parti: l'encoder (E) e il decoder (D). L'encoder si occupa di ridurre la dimensionalità della distribuzione iniziale, ottenendo quindi una sua rappresentazione nello spazio latente. Il decoder prende questa rappresentazione e cerca di ricostruire la distribuzione iniziale, possiamo dire che, data una distribuzione  $x$ , l'obiettivo sia quello di far sì che  $x \approx D(E(x))$ . L'AE usando questa funzione obiettivo è però molto soggetto all'overfitting, ciò che invece vorremmo è uno spazio latente regolarizzato da cui poter generare

nuove distribuzioni. Il VAE è infatti un modello generativo che apprendere le distribuzioni iniziali e le mappa verso una rappresentazione nello spazio latente, ciò che si vuole fare è regolarizzare questo spazio in modo da poter generare da questo nuove distribuzioni. Nel VAE l'encoder mappa la distribuzione iniziale  $x$  verso due valori  $(\sigma_x, \mu_x)$ , media e varianza di una distribuzione gaussiana  $N(\sigma_x, \mu_x)$ , da questa poi si estrae un sample  $z$  e attraverso il decoder si cerca di ricondurlo alla distribuzione iniziale  $x$ . Possiamo vedere queste operazione nella Figura 3.3.

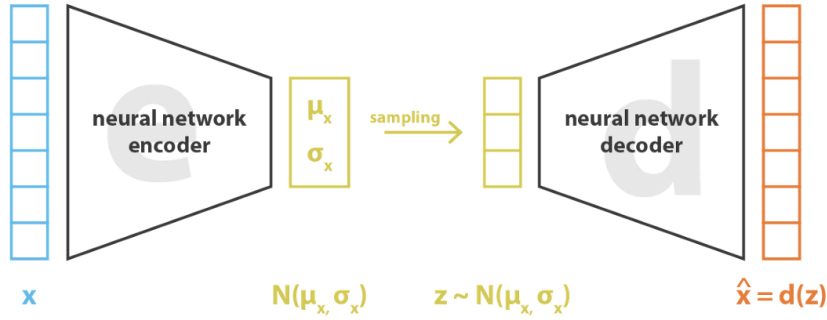


Figure 3.3: Rappresentazione grafica del VAE.

Per permettere di usare l'algoritmo *backpropagation*<sup>[g]</sup> si esegue il sampling da una distribuzione  $N(0, 1)$  e poi si riscalda  $z$  rispetto quella data dalla rappresentazione  $N(\sigma_x, \mu_x)$ , questo passaggio viene chiamato 'reparametrisation trick'.

La funzione obiettivo è definita come segue:

$$L(x) = \|x - D(z)\|^2 + KL[N(\sigma_x, \mu_x), N(0, 1)]$$

Si nota il trade-off tra funzione obiettivo dell'AE e la regolarizzazione attuata sulla rappresentazione delle distribuzioni interne grazie alla KL divergence, una funzione che misura la distanza tra la distribuzione  $N(\sigma_x, \mu_x)$  e  $N(0, 1)$ , cercando di ridurla il più possibile. Rispetto ai modelli precedenti che potevano funzionare sia con rating espliciti che impliciti, il VAE CF accetta solo quest'ultimi. Il modello prende in input la matrice dei rating impliciti andando a considerare ogni riga come una distribuzione associata allo user e per ciascuno di essi ritorna la distribuzione generata da cui poi si possono estrarre gli item consigliati andando a selezionare quelli che riportano valori più alti nella distribuzione data dal decoder.



### 3.3.2 Content-based filtering

Il content-based filtering<sup>8</sup> è un approccio che si basa sull'idea di consigliare item simili a quelli con cui si è già interagito.

Ciascun item possiede delle feature, per esempio nel caso si abbia come item dei film queste potrebbero essere i generi, l'insieme delle feature può essere mappata su di un vettore delle feature, per ogni item quindi si riporta nel vettore 1 se possiede quella feature, 0 altrimenti, questo permette di definire un profilo per l'item. Una volta fatto ciò si procede a creare anche un profilo per lo user, ci sono diversi modi per farlo ma per esempio si può considerare tutti i profili degli item con cui si è interagito e farne quindi la media pesata basata sui rating corrispondenti. Ottenuto un profilo anche per lo user si può calcolare la similarità tra di esso e quello degli item per poi ordinarli secondo similarità ottenendo così la lista *TopN*.

## 3.4 Valutazione

Quanto abbiamo visto finora sono metodi che ci permettono di effettuare le raccomandazioni, vogliamo trovare però anche il modo per poterle valutare. Per prima cosa dobbiamo dividere le matrici dei rating in training e test set.

Per fare ciò andiamo ad eseguire uno shuffle delle coppie (user, item) e si va ad assegnare al training set l'80% delle coppie e le restanti al test set, questo sistema non ci assicura che uno user sia presente nel training set. Date le raccomandazioni fornite dal RS vogliamo valutarle rispetto due aspetti principali: il rating e il ranking. Il primo aspetto riguarda semplicemente la diversità tra i rating stimati da quelli reali delle coppie (user, item) del test set, queste metriche non vengono solitamente usate in quanto non è un buon modo per valutare un RS perché non ci permette di capire se un'item consigliato sia rilevante. Andando invece a considerare il concetto di ranking ci rendiamo conto che sia più legato a ciò che vogliamo andare a valutare, le metriche annesse considerano rilevanti gli item presenti nel test set e vanno a verificarne la posizione nella lista *TopN*. In generale le metriche si applicano a ciascuno user e il risultato finale è la media dei singoli risultati.

### 3.4.1 AUC

L'AUC<sup>3</sup> (Area Under the Curve) è una metrica che permette di valutare un RS basandosi sul numero di coppie di item rivelanti e non presenti nella  $TopN$ , dove un item non rilevante ha uno score minore rispetto a quello di uno rilevante. Questa metrica assegna lo stesso peso a ciascun item rispetto il ranking, questo significa che non controlla la posizione in cui si trovano gli item più interessanti. Vediamo di seguito la formula:

$$AUC = \frac{1}{|N^+| \cdot |N^-|} \sum_{i \in N^+} \sum_{j \in N^-} [s(i) > s(j)]$$

Il termine  $N^+$  è l'insieme degli item presenti nel test set, mentre  $N^-$  sono tutti gli item rimanenti.  $s(i)$  è la valutazione data dal RS sull'item  $i$ , quello che si fa è andare a contare il numero di coppie di item in ordine nella  $TopN$  andando a vedere gli score assegnati loro. Utilizziamo la funzione  $[s(i) > s(j)]$  che restituisce 1 se lo score dell'item rilevante è maggiore, 0 altrimenti. Infine dividiamo il numero di coppie ordinate in modo corretto per il numero di coppie totali.

Il valore dell' $AUC \in [0, 1]$ , dove più il valore si avvicina ad 1, minori saranno le coppie in ordine sbagliato.

### 3.4.2 nDCG@k

La metrica nDCG@k<sup>13</sup> (normalized Discount Cumulative Gain) ci permette di calcolare una misura basata sulla posizione degli item rilevanti, ossia quelli del test set, tra i primi k item della lista  $TopN$ .

$$DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad IDC@k = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

Ed infine per calcolare la  $nDCG@k$  usiamo la seguente formula:

$$nDCG@k = \frac{DCG@k}{IDC@k}$$

Per prima cosa dobbiamo dire che la metrica lavora su una parte della lista  $TopN$ , ossia la parte contenente i primi K item che definiamo come  $TopK$ .

Gli item più rilevanti nel nostro caso sono gli item presenti nel test set.

Andiamo ora ad analizzare numeratore e denominatore per capire meglio la loro funzione.

- La  $DCG@k$  si basa sull'idea che gli item più rilevanti debbano trovarsi il più possibile in testa alla lista  $TopK$ , quindi si vuole penalizzare un item sempre di più via via questo si trovi nella coda della lista, per far ciò il denominatore aumenta all'aumentare della posizione seguendo una scala logaritmica.
- La  $IDCG@k$  è equivalente a calcolare la  $DCG@k$  sulla lista  $TopK$  ideale, ossia la lista riportante tutti gli item ordinati in modo ideale secondo score nella posizione corretta, questo equivale al valore massimo ottenibile dalla metrica.

Quindi andando a dividere il numeratore con il denominatore si attua una normalizzazione, il valore finale di  $nDCG@k \in [0, 1]$ , dove più si avvicina ad 1 più la lista  $TopK$  assomiglia a quella ideale.



# 4

## Preprocessing storico vendite

In questo capitolo vedremo una descrizione dettagliata di tutte le tecniche applicate allo storico vendite con l'obiettivo di raccomandare a ciascuno user gli item per lui più interessanti.

### 4.1 Preprocessing matrici grezze

In questa sezione andremo a vedere diverse tecniche che sono state utilizzate per trasformare le *matrici grezze*<sup>[9]</sup> user-item in matrici dei rating.

#### 4.1.1 Preliminari

Definiamo l'insieme degli user  $U$ , l'insieme degli item  $I$  e le matrici grezze user-item  $RG$ . Ciascuna tecnica lavora andando a considerare le matrici  $RG$  come un vettore di triplette  $V = [(u, i, RG_{(u,i)} \neq 0) | \forall (u \in U, i \in I)]$  con  $RG_{(u,i)} \in \mathbb{R}^+$ .

Facciamo inoltre riferimento a  $V_c$  come il vettore delle coppie (user,item),  $V_{(u,i)}$  come il valore della tripletta di user  $u$  e item  $i$ ,  $V_u$  il vettore dei valori delle triplette con user  $u$  e  $V_i$  il vettore dei valori delle triplette con item  $i$ .

Ciascuna tecnica implementa una diversa funzione  $f$  biettiva di trasformazione

che possiamo riassumere come segue:

$$f : [(u, i, V_{(u,i)}) | \forall (u, i) \in V_c] \rightarrow [(u, i, r \in [1, scale]) | \forall (u, i) \in V_c]$$

Queste tecniche si propongono di trasformare il valore  $V_{(u,i)}$  di ciascuna tripletta in un rating  $r \in [1, scale]$ , con  $scale$  sempre dispari.

Alcune tecniche faranno riferimento ad una distribuzione dei rating uniforme discreta o gaussian-like su gruppi di elementi. Quando ci si troverà ad applicare queste distribuzioni avremo un vettore di elementi ordinati secondo un certo criterio.

Vediamo come vengono assegnati i rating secondo queste due distribuzioni:

- uniforme discreta: divide il vettore in modo tale che ogni valore nella scala dei rating compaia lo stesso numero di volte, assegnandoli in modo crescente, dal capo alla coda del vettore;
- gaussian-like: si va a definire una distribuzione normale  $N(0, scale/3)$ , poi si generano una quantità sufficiente di numeri secondo la suddetta distribuzione. Fatto questo si convertono tutti i numeri decimali in interi, si selezionano solo gli interi nell'intervallo  $[-scale/2, scale/2]$  e si traslano nell'intervallo  $[1, scale]$ .

Infine calcoliamo la probabilità per ciascun numero intero nella scala.

Per assegnare i rating al vettore non si fa altro che iterare sugli interi dell'intervallo  $[1, scale]$ , andando ad eseguire in sequenza le seguenti operazioni:

1. moltiplico la probabilità di quell'intero per la lunghezza del vettore;
2. converto il valore risultante ad intero, ottenendo quindi il numero di elementi che dovranno avere quel rating;
3. partendo dall'inizio del vettore assegno quel rating a quello specifico numero di elementi e poi una volta raggiunto l'ultimo procedo col successivo intero della scala a partire dall'elemento seguente.

L'assegnazione dei rating secondo tali distribuzioni è implementato da due funzioni che restituiscono un vettore di coppie, formate dall'elemento e dal rating corrispondente.

### 4.1.2 Tecnica product-based

La tecnica *product-based* permette di analizzare gli item da un punto di vista globale. Si procede considerando gli item in termini assoluti, vediamo di seguito le operazioni per applicarlo:

1. otteniamo il seguente vettore  $p = [(i, \sum V_i) | \forall i \in I]$ ;
2. dopo aver ordinato il vettore  $p$  basandoci sul secondo termine delle coppie, conserviamo solo il primo elemento di ciascuna di esse;
3. andiamo ad applicare la funzione uniforme discreta / gaussian-like a tale vettore, ottenendo per ogni item un rating;
4. per ogni tripletta si va a recuperare il rating corrispondente al suo item e crea una nuova tripletta col suddetto valore.

Questa tecnica porta ad avere, indipendentemente dallo user, la stessa valutazione per ogni item ed è quindi molto sensibile alla popolarità dello stesso nello storico vendite.

### 4.1.3 Tecniche group-based

Le tecniche presenti in questa sezione permettono di dividere il vettore delle triplette  $V$  in diversi gruppi, applicare separatamente a ciascuno di essi il metodo ed infine unire insieme i vettori risultanti.

Va rispettata la condizione che l'intersezione tra tutti i gruppi deve essere nulla. Vediamo le possibili divisioni in gruppi delle triplette:

- un unico gruppo con tutte le triplette;
- un gruppo per ogni user contenente solo le sue triplette;
- per ogni user e per ogni categoria un gruppo contente tutte le triplette di quello user dove gli item appartengono a quella categoria;

Vediamo ora i diversi metodi applicati ad un singolo gruppo.

#### 4.1.3.1 Normalizzazione Min-Max

Una delle prime tecniche usate per il preprocessing dei dati è quella della normalizzazione min-max, per applicarla andiamo a considerare un gruppo  $G \subseteq V$  e applichiamo a ciascuna tripletta la funzione min-max, in generale otteniamo il seguente vettore risultante:

$$G = [(u, i, \frac{G_{(u,i)} - \min(G_r)}{\max(G_r) - \min(G_r)} \in [0, 1]) | \forall (u, i) \in G_c]$$

Ora tutti i valori delle triplette di  $G$  si troveranno in un intervallo  $[0, 1]$ , per portarlo invece nell'intervallo  $[1, scale]$  dobbiamo applicare la seguente formula:

$$[(u, i, (scale - 1) \cdot \frac{G_{(u,i)} - \min(G_r)}{\max(G_r) - \min(G_r)} + 1 \in [1, scale]) | \forall (u, i) \in G_c]$$

Inoltre una volta applicata la formula, oltre che tenere i rating così come sono nel dominio dei numeri reali, si è provato anche a convertirli in numeri interi, verranno chiamate rispettivamente *continous* e *rint*.

Si voleva provare in questo modo a capire se gli user avessero volumi d'acquisto diversi e se prodotti delle stesse coppie avessero logiche d'acquisto simili. Inoltre dobbiamo puntualizzare che se guardiamo per esempio la distribuzione della quantità totale rispetto i prodotti, noteremo che risulta assumere il comportamento di una curva discendente, quindi ci sono molti prodotti acquistati in bassa quantità e pochi in grande quantità. Applicando questo metodo, che non va a cambiare la distribuzione iniziale dei valori ma solo a scalarli, otterremo molti rating bassi.

#### 4.1.3.2 Tecnica ordered-based

Il seguente metodo prevede di lavorare su un gruppo di triplette  $G \subseteq V$  e di eseguire le seguenti operazioni:

1. ordiniamo il vettore  $G$  secondo il terzo valore delle triplette;
2. applichiamo la funzione uniforme discreta / gaussian-like a tale vettore;
3. sostituiamo al valore della tripletta quello del rating assegnatogli.



Questa tecnica permette di confrontare le triplette attraverso l'ordinamento e consente una migliore distribuzione dei rating rispetto la normalizzazione min-max, ma è da verificare se questa ci fornisca risultati sperimentalmente migliori.

#### 4.1.4 Approccio implicito

Tutti gli approcci che abbiamo visto producono matrici dei rating esplicite, chiaramente un tentativo sarà quello di usare una versione della matrice grezza implicita.

## 4.2 Tecniche combinate

Nelle sezioni precedenti abbiamo visto diverse tecniche di preprocessing per ottenere dei rating dalle matrici grezze, in questa andremo invece a descrivere due approcci utili per cercare di combinare insieme rating provenienti dalle diverse *espressioni d'interesse*<sup>[g]</sup>.

### 4.2.1 Premesse

Come riportato nel capitolo dell'analisi dei dati, le informazioni disponibili sugli item ci permettono di valutare l'interesse dello user verso di essi secondo diversi *aspetti*, quali la quantità acquistata, la spesa totale, il numero di fatture in cui compaiono e la recentezza dell'ultimo acquisto, definiremo questi aspetti da ora in poi come *espressioni di interesse*. Queste sono organizzate in matrici grezze, a cui nella sezione precedente, abbiamo applicato diverse tecniche di preprocessing andando a trasformarli in rating. L'idea di unire insieme queste *espressioni di interesse* sembra essere un buon modo per migliorare la qualità delle raccomandazioni finali. I metodi combinati prendono in input le matrici grezze e vi applicano una delle tecniche di preprocessing illustrate nella sezione precedente. Ci sono due modi per combinarle insieme che appronfiremo nelle seguenti sezioni.

### 4.2.2 Combinazione liste *TopN*

Il primo metodo si propone di ottenere per ogni user una lista *TopN* di item per ciascuna espressione di interesse, queste poi andranno combinate insieme

attraverso l'uso del borda count, un sistema di voting basato sulla posizione. Vediamo ora quali sono le operazioni da attuare:

1. applicare la stessa tecnica di preprocessing a tutte le matrici grezze delle espressioni di interesse ottenendo le corrispettive matrici dei rating;
2. usare uno degli approcci del collaborating filtering sulle matrici dei rating ottenendo così le liste *TopN*;
3. combinare insieme le liste *TopN* secondo un sistema di voting, il borda count, nel quale ogni item della lista riceve uno score in base alla posizione, questi si sommano e gli item vengono riordinati in base al punteggio risultante.

#### 4.2.3 Media matrici dei rating

Mentre il precedente metodo prevedeva di applicare il collaborative filtering separatamente a ciascuna matrice dei rating, in questo calcoliamo una media delle matrici grezze, da cui, dopo aver eseguito una delle tecniche di preprocessing su di essa, ricaviamo una sola matrice dei rating.

A questa andiamo poi ad applicare uno degli approcci del collaborative filtering e così otteniamo la lista *TopN*.

### 4.3 Approccio content-based

In questa sezione vedremo un'applicazione dell'approccio content-based utilizzando le informazioni descrittiva disponibili sugli item.

Abbiamo diverse fonti d'informazioni:

- categorie rispetto i diversi livelli;
- gruppo merceologico;
- nome del prodotto;
- dimensioni;
- volume;

- peso.

In generale ogni informazione descrittiva, può essere classificata in due modi:

- Binaria: un'item può avere o meno questa caratteristica;
- Continua: l'item può avere o meno questa caratteristica, ma se ce l'ha l'informazione è un valori in un intervallo.

L'idea di base è di creare un vettore dove ogni posizione corrisponda ad una specifica caratteristica, nel caso l'informazione descrittiva sia binaria si va ad assegnargli una cella dove vi sarà 1 se l'item la possiede, 0 altrimenti. Nel caso questa sia continua verrà divisa in sotto-intervalli gestiti come caratteristiche binarie.

#### 4.3.1 Gerarchia prodotto

A ciascuna categoria della gerarchia abbiamo assegnato una cella del vettore. Come detto in precedenza, per la gerarchia prodotto abbiamo diverse categorie per ogni livello, per ciascuna di esse abbiamo assegnato una cella del vettore delle caratteristiche, se un'item apparteneva ad una di queste categorie andavamo ad impostare quella specifica cella a 1, altrimenti 0.

#### 4.3.2 Gruppo merceologico

Per il gruppo merceologico si è fatto come nel caso precedente, quindi per ogni codice merceologico si è andato ad assegnargli una cella del vettore delle caratteristiche.

#### 4.3.3 Nome prodotto

Il nome del prodotto poteva darci informazioni aggiuntive sulla similarità tra di essi, osservando l'anagrafica materiali notiamo che i nomi sono assegnati in modo organizzato: prodotti simili in cui cambiano solo alcune componenti riportano lo stesso nome con le specifiche diverse.

Per ogni nome prodotto sono state eseguite le seguenti operazioni:

1. eliminazione di simboli e informazioni ridondanti, per esempio alcuni nomi avevano riportata anche la dimensione, già disponibile in altri campi;
2. separazione delle restanti parole;
3. inserimento delle suddette in un dizionario contenente la totalità delle parole e il rispettivo numero di occorrenze.

Dopo un'ulteriore pulizia a mano siamo andati a selezionare le parole con almeno 10 occorrenze, ottenendo alla fine circa 600 parole. Ad ognuna di esse è stata assegnata ad una cella del vettore delle caratteristiche, se un item nel suo titolo conteneva una di queste parole, si impostava a 1 la cella corrispondente.

#### 4.3.4 Dimensione, volume, peso

Per le informazioni riguardanti lunghezza, larghezza, altezza, volume e peso si volevano creare per ciascuno di essi 20 intervalli aventi circa lo stesso numero di item. Si è assegnato al vettore delle caratteristiche una cella per ogni intervallo e se un'item aveva una delle misure all'interno dello stesso lo si impostava ad 1.

#### 4.3.5 Profilo user

Una volta creata la matrice avente sulle righe gli item e sulle colonne le caratteristiche, si è proceduto a calcolare i profili degli user facendo la media delle righe corrispondenti agli item acquistati dallo stesso.

### 4.4 Approccio next-basket

In questo capitolo opereremo sullo storico vendite considerando le fatture come sessioni d'acquisto a cui applicheremo la *User Popularity-based CF*<sup>4</sup>.

Ragionando in termini di logica d'acquisto è possibile che i dealer acquistino sempre lo stesso gruppo di item, questo ci porta a parlare del concetto di popolarità e di come questa possa variare nel tempo. Dato che questa soluzione è stata provata dopo le altre precedentemente descritte, si è notato come non fosse così facile "battere" il modello basato sulla popolarità, quindi si è pensato di muoversi in questa direzione per vedere se i risultati migliorassero.

L'obiettivo era quello di predire per ciascuno user gli item dell'ultima fattura.

#### 4.4.1 Premesse

Definiamo l'insieme degli user  $U$ , l'insieme degli item  $I$  e consideriamo ciascuna fattura come una transazione  $b_t^u$ , dove  $t$  indica la posizione ordinale nell'insieme delle transazioni di uno user di cardinalità  $B_u$  definito come  $\mathcal{B}_u = \{b_u^t | t \in 1, \dots, B_u\}$ . Definiamo  $\mathcal{B}_u^i = \{b_u^t | b_u^t \in \mathcal{B}_u \wedge i \in b_u^t\}$ , ossia l'insieme di tutte le transazione dello user  $u$  in cui compare l'item  $i$ .

#### 4.4.2 Popolarità

Possiamo calcolare la popolarità di un item rispetto ad uno user, detta *popularity user-wise*, con la seguente formula:  $\pi_i^u = \frac{|\mathcal{B}_u^i|}{B_u}$ . Dato che la popolarità può variare nel tempo il paper introduce il concetto di recentezza, concludendo che per predire l'ultima transazione potrebbe non essere efficace considerare quelle più vecchie, quindi attraverso una finestra temporale sulle quelle più recenti si va a calcolare la *recency aware user-wise popularity*, un modo di stimare la popolarità di un item per uno user solo su un ristretto numero di transizioni, di seguito la formula:

$$\pi_u^i @r = \frac{\sum_{t=\max(B_u-r, 0)}^{B_u} [i \in b_u^t]}{\min(r, B_u)}$$

Con il parametro  $r$  si definisce la finestra di transazioni, a partire dall'ultima, da tenere in considerazione, la funzione  $[i \in b_u^t]$  ritorna 1 se l'item  $i$  è presente nella transazione  $b_u^t$ , 0 altrimenti. Se  $r \geq B_u$  allora questa formula diventa equivalente alla *popularity user-wise*.

#### 4.4.3 User Popularity-based CF (UP-CF)

Questa soluzione, che si basa sul collaborative filtering, permette di trovare item interessanti per uno user andando ad osservare user simili ad esso, questa similarità tra due user  $u$  e  $v$  si concretizza nella funzione coseno asimmetrica:  $w(u, v) = \frac{|I_u \cap I_v|}{|I_u|^\alpha |I_v|^{1-\alpha}}$  con  $\alpha \in [0, 1]$ , dove il parametro  $\alpha$  permette di bilanciare la probabilità  $P(u|v)$  e  $P(v|u)$ .

#### 4.4.3.1 Predizione

Si vuole ora combinare la similarità tra user e il concetto di popolarità di un item per uno user, per fare ciò usiamo la formula:

$$\hat{r}_i^u = \sum_{v \in U} w(u, v)^q \pi_u^i$$

Dove il termine  $q$  è un parametro operante sulla località degli user, per un alto valore di  $q$  considereremo solo user molto simili a quello target.

# 5

## Libreria cornac ed esperimenti

In questo capitolo vedremo le modalità con cui sono stati svolti gli esperimenti e la libreria utilizzata per gli stessi.

### 5.1 Dataset

Le informazioni dello storico vendite sono state organizzate basandosi sulle due macrocategorie individuate nell'analisi: macchine e ricambi. Gli item presi in esame sono quelli acquistati almeno una volta, mentre gli user sono quelli che hanno effettuato almeno un acquisto. Le matrici dei rating verranno definite di un certo tipo di dataset in base all'insieme di item che contengono, sarà di tipo macchina se contiene solo item della macrocategoria delle macchine, ricambi se sono item di tipo ricambio, totale se ne contiene di entrambi i tipi.

In generale ciascuna matrice sarà di uno dei seguenti tipi di dataset:

- **Macchine:** 254 user e 518 item;
- **Ricambi:** 319 user e 7699 item;
- **Totale:** 322 user e 8217 item.

## 5.2 Esperimenti sulle matrici grezze

In questa sezione vedremo come sono stati condotti i test sulle matrici grezze, dato un tipo di dataset avremo una matrice grezza per ognuna delle quattro espressioni di interesse, che sono state calcolate tra uno user  $u$  e un item  $i$  come segue:

- **Quantità:** somma dei campi quantità (*KWMENG*) di tutte le posizioni delle fatture di  $u$  in cui è presente  $i$ ;
- **Spesa totale:** somma dei campi spesa totale (*NETWR*) di tutte le posizioni delle fatture di  $u$  in cui è presente  $i$ ;
- **Numero di fatture:** conta del numero di fatture di  $u$  in cui compare  $i$ ;
- **Recentezza:** ricerca della posizione di  $i$  nelle fatture di  $u$ , riportante la data più recente di acquisto. Se l'item è stato acquistato avremo quindi una data a cui andremo a sottrarre quella della fattura più vecchia, la differenza temporale viene trasformata in giorni (più sarà elevata più recente sarà l'acquisto).

### 5.2.1 Libreria Cornac

La libreria *Cornac*<sup>9</sup> gestisce completamente gli esperimenti, dall'acquisizione dei dati fino alla verifica dei risultati ed è stata scelta per la presenza di molti *modelli* e metriche per la loro valutazione.

Nello specifico i modelli utilizzati sono stati:

- MostPop: modello basato sulla popolarità, dove un item è più popolare in base al numero di user che lo hanno valutato, usato per ottenere un risultato di base;
- UserKnn<sup>2</sup>: implementazione dell'approccio memory-based del collaborative filtering basato su user, di default prende in considerazione i 20 user più simili ad uno target, usa la funzione di similarità coseno;
- ItemKnn<sup>10</sup>: come il precedente, ma basato sugli item. Non è stato infine utilizzato in quanto durante le fasi iniziali di test non ha mai riportato risultati superiori a quelli di base e richiedeva troppo tempo per la valutazione;



- MF<sup>6</sup>: implementazione del matrix factorization, metodo model-based del collaborative filtering, di default utilizza un vettore di dimensionalità 10 per lo spazio delle feature;
- VAE<sup>7</sup>: modello per la versione implicita, usato per avere un risultato di base avanzato.

I dati presi in input sono nel formato *user, item, rating* e volendo potevano essere divisi in training, validation e test set direttamente dalla libreria. Si è preferito però definirli esternamente con le rispettive percentuali: 70% al training set, 15% per validation e test set, ciascuno di essi è un insieme di indici, ciascuno dei quali punta ad una specifica tripletta che si trova sempre nella stessa posizione in quanto l'insieme delle triplette finali è ordinato allo stesso modo. Nella fase di valutazione dei risultati dei *modelli* veniva richiesto un parametro detto *rating\_threshold*, il quale serviva a binarizzare gli item rilevanti e irrilevanti del test set basandosi sul rating corrispondente. Non andava in alcun modo ad intaccare i rating nella fase di training. Il *rating\_threshold* è stato fissato a 1, in modo che le valutazioni su scale diverse fossero sempre confrontabili e si andassero a considerare tutte le triplette del test set.

### 5.2.2 Esperimenti sulle singole matrici grezze

Come riportato nel capitolo relativo al preprocessing delle matrici grezze, non è stata definita un'unica scala dei rating in quanto se ne volevano provare diverse:  $scale \in \{3, 5, 7, 9, 13, 19, 25\}$ . Inoltre per ogni tecnica dove fosse previsto l'utilizzo di un certo numero di funzioni (per esempio per normalizzazione min-max erano disponibili *rint* e *continuous*, mentre per *ordered-based* si potevano usare le distribuzioni *uniforme discreta* e *gaussian-like*), si è proceduto a testarle tutte. Considerando tutte le combinazioni di fattori, si sono ottenute in totale 2184 matrici dei rating. Per eseguire i test si è proceduto come segue:

1. ottengo un risultato di base con il modello MostPop sul validation set;
2. procedo con la fase preliminare, dove ogni matrice dei rating viene testata secondo le seguenti operazioni:
  - (a) 'alleno' i modelli (*MF*, *UserKnn*) con valori di default con il training set;

- (b) valuto i risultati del modello sul validation set, se questi valori risultano migliori di quelli di base (dati dal *MostPop*), allora questa matrice accede alla fase avanzata;
- 3. ottengo un risultato di base del test set con il modello *VAECF* ottimizzato sul validation set;
- 4. segue la fase avanzata, dove per ogni matrice selezionata vengono eseguite le seguenti operazioni:
  - (a) si effettua il tuning dei parametri dei modelli (*MF*, *UserKnn*) usando il validation set per ciascuna matrice e si calcola il risultato finale col test set;
  - (b) si confronta la valutazione finale basata sul test set con quella di base ottenuta con il *VAECF*;
- 5. tra tutte quelle che superano quest'ultimo passaggio scegliamo la migliore, una per ogni dataset.

Il confronto dei risultati è matematico quindi se anche un metodo dovesse avere una sola matrice che risulta selezionata nella fase preliminare, si procederà a testare anche quella nonostante il generale trend negativo del metodo.

### 5.2.3 Esperimenti sulle matrici grezze combinate

Per prima cosa si è proceduto a dividere in gruppi tutte le matrici dei rating aventi stesso metodo e stessa scala, si sono ottenuti tutti gruppi aventi 4 matrici, una per ogni *espressione d'interesse*. Per ognuno di questi gruppi si è proceduto ad applicare i due metodi combinati basati su liste *TopN* e media delle matrici dei rating. Per gli esperimenti si sono usati lo stesso training, validation e test set usati per gli esperimenti precedenti. Gli esperimenti miravano a capire se questo metodo potesse fornire risultati migliori rispetto a quelli di base dati da *VAECF* sul test set.

### 5.3 Esperimenti con approccio content-based

Per questo approccio non si è arrivati alla fase sperimentale in quanto dai primi test è risultato chiaro che non si avessero abbastanza informazioni sugli item.

I profili degli user erano molto simili tra loro e, nonostante diverse modifiche, per ogni user venivano restituiti circa gli stessi item. Probabilmente le misure continue, essendo disponibili per un numero limitato di prodotti, sono risultate poco influenti nella media finale, mentre la gerarchia prodotto e i nomi hanno fatto sì che venissero raccomandati i prodotti più popolari. Concludendo non si è voluto sviluppare ulteriormente l'approccio preferendo concentrarsi su altro.

### 5.4 Esperimenti con approccio next-basket

Con questo approccio si è usato un dataset diverso e quindi non è possibile confrontare i risultati con quelli precedenti. Sono stati selezionati tutti gli user che avessero almeno 5 fatture, che ricordiamo in questo approccio vengono viste come transazioni, riducendo il numero di user rispettivamente per il tipo dataset *macchine* da 254 a 224, per i *ricambi* da 319 a 257 e per il *totale* da 322 a 257.

Possiamo notare come siano stati eliminati un numero considerevole di user che avevano acquistato ricambi. L'insieme delle ultime transazioni  $\{b_u^{Bu} | \forall u \in U\}$  sono andate a formare rispettivamente randomicamente al 50% validation e test set, mentre le restanti sono entrate a far parte del training set.

Gli hyper-parametri disponibili sono i seguenti:

- asimmetria  $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ ;
- località  $q \in \{1, 5, 10, 50, 100, 1000\}$ ;
- finestra di recentezza  $r \in \{1, 5, 25, 50, \infty\}$ .

Si è testata ogni combinazione di hyper-parametri sul validation set e, selezionati quelli migliori in accordo al tipo di dataset, si è calcolata la valutazione finale del test set su ciascun dataset.



# 6

## Risultati sperimentali

In questo capitolo vedremo i risultati sperimentali per ogni metodo, vedremo prima quelli applicati alle matrici grezze, poi quelli basati sulla combinazione delle stesse, i risultati dell'approccio next-based per concludere infine con un confronto critico dei risultati.

### 6.1 Risultati preprocessing matrici grezze

Nelle sezioni successive saranno riportati i risultati delle valutazioni di ciascun modello rispetto la metrica  $AUC$  e la  $NDCG$  con  $k \in \{5, 10, 25, 100\}$ . La metrica che riteniamo sia più significativa è la  $NDCG$  in quanto si basa sul ranking degli item nella lista  $TopN$ , per i confronti abbiamo scelto di usare  $k = 25$ .

Come riportato nel capitolo precedente la valutazione sarà condotta in due fasi:

1. preliminare: confrontiamo i risultati dei modelli (MF, UserKnn) con parametri di default applicati alle matrici dei rating con il valore soglia dato dal modello MostPop, il tutto usando il validation set;
2. avanzata: dopo aver effettuato il tuning dei modelli con le matrici rimanenti, si confrontano i risultati con il bound dato dal VAE CF usando il test set.

### 6.1.1 Risultati soglia

In questa sezione vedremo i risultati del modello MostPop e del VAE CF, che verranno utilizzati come bound per valutare quelli successivi.

#### 6.1.1.1 MostPop

Come detto il modello MostPop restituisce per ogni user la stessa lista di item più popolari e nella tabella 6.1 ne riportiamo i risultati sul validation set.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.7644	0.0920	0.1263	0.1647	0.2414
ricambi	0.3427	0.0000	0.0299	0.0358	0.0732
totale	0.2810	0.0360	0.0733	0.0627	0.0811

Table 6.1: Risultati modello MostPop per tipo dataset sul validation set

Useremo questi risultati come soglia per la fase preliminare, testando i risultati delle matrici applicate ai modelli MF e UserKnn. Nel caso in cui questi siano migliori, si procederà al tuning dei parametri del modello stesso e al calcolo finale dei risultati sul test set.

Quelli riportati nella tabella 6.2 sono i risultati del MostPop sul test set.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.7897	0.1001	0.1364	0.1875	0.2610
ricambi	0.3924	0.0619	0.0753	0.0793	0.1365
totale	0.3101	0.0944	0.0807	0.0866	0.1281

Table 6.2: Risultati modello MostPop per tipo dataset sul test set

#### 6.1.1.2 VAE CF

Il modello VAE CF è un modello che funziona su rating impliciti, la fase preliminare ha richiesto il tuning dei parametri riportati di seguito:

- $n_{int}$  è la dimensione della rappresentazione latente interna, i possibili valori sono  $\{5, 10, 15, 20, 50\}$ ;

- $n_{hid}$  è il numero di neuroni del layer dell'encoder e del decoder, i valori possibili sono  $\{10, 20, 30, 50, 100, 200\}$ ;
- $f_{act}$  è la funzione di attivazione applicata nei layer nascosti, che può essere una delle seguenti  $\{sigmoid, tanh, elu, relu, relu6\}$ .

<i>dataset</i>	$n_{int}$	$n_{hid}$	$f_{act}$
macchine	5	30	relu
ricambi	5	30	sigmoid
totale	5	30	sigmoid

Il tuning dei parametri ha portato a considerare tutte le possibili combinazioni degli stessi, vediamo per ogni dataset quali valori hanno restituito i risultati migliori sul validation set.

Table 6.3: Parametri selezionati.

Vediamo ora i risultati suddivisi per dataset sul validation set con il modello ottimizzato nella Figura 6.4.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.8201	0.1325	0.1768	0.2282	0.3032
ricambi	0.4773	0.0506	0.0452	0.0643	0.1052
totale	0.4190	0.1208	0.0980	0.0741	0.0941

Table 6.4: Risultati VAE CF ottimizzato per tipo dataset sul validation set.

Dato che si utilizzeranno i risultati del VAE CF ottimizzato sul test set come soglia nella fase avanzata, questi sono presenti in Figura 6.5.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.8269	0.1473	0.1948	0.2487	0.3301
ricambi	0.4930	0.0801	0.0919	0.1075	0.1401
totale	0.4343	0.0988	0.0874	0.0964	0.1398

Table 6.5: Risultati VAE CF ottimizzato per tipo dataset sul test set.

### 6.1.2 Fase test preliminare

In questa sezione vedremo per ogni tecnica di preprocessing applicata sulle matrici grezze i risultati ottenuti. Cominceremo prima con la tecnica basata su min-max,

poi con quella ordered-based per concludere con quella product-based. Spesso si farà riferimenti ai risultati dei modelli, intendendo il risultato della metrica dato dall'applicazione del modello alla matrice dei rating. Come spiegato nei capitoli precedenti i metodi min-max e ordered-based sono applicabili a diversi gruppi di triplette, vedremo i risultati per ciascuno di essi.

### 6.1.2.1 Normalizzazione min-max gruppo globale

In questa sezione vedremo la normalizzazione min-max applicata al gruppo globale, ossia quello contenente tutte le triplette. Nel grafico composto 6.1 possiamo vedere sulle righe i tipi di dataset (macchine, ricambi, totale), mentre sulle colonne le *espressioni d'interesse*. Ciascun grafico poi mostra sulle ascisse la *scale* della matrice dei rating e sulle ordinate il risultato ottenuto da tale matrice rispetto la metrica  $NDCG@25$ .

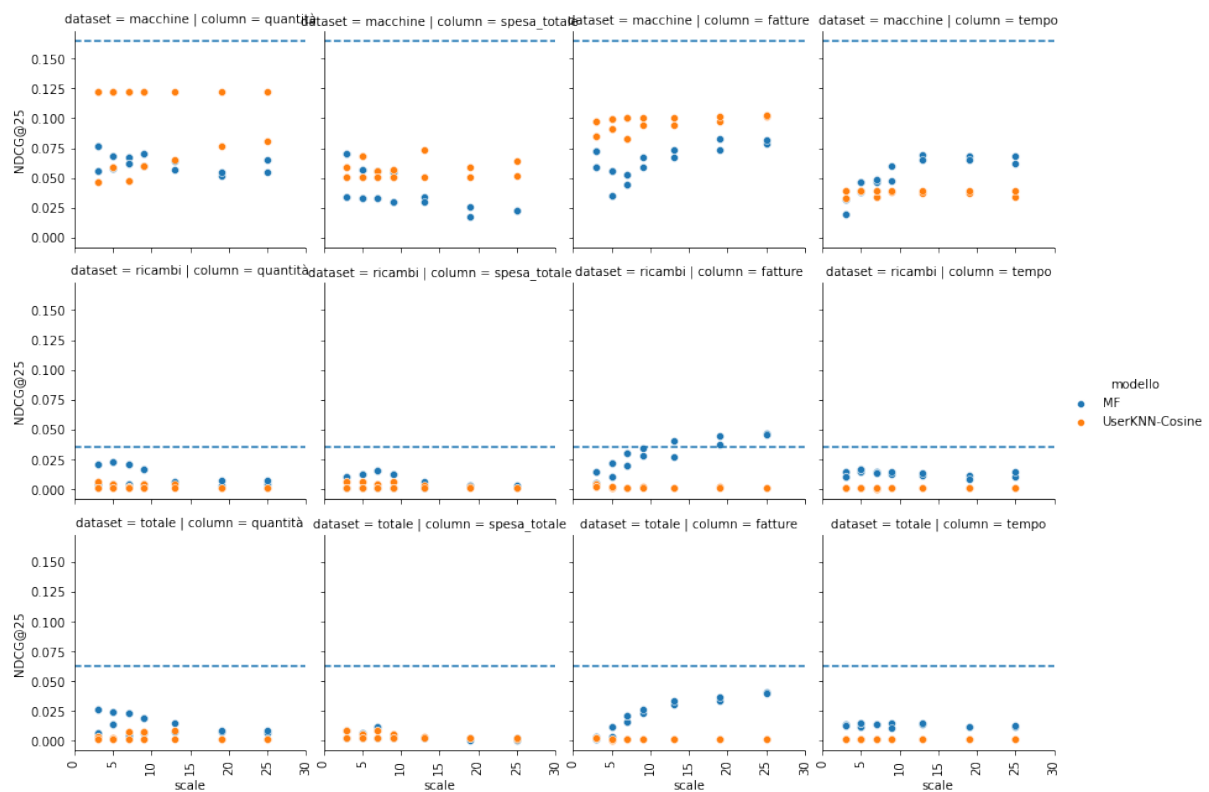


Figure 6.1: Risultati normalizzazione min-max gruppo globale.



La linea tratteggiata blu rappresenta il risultato del modello MostPop che viene usato come bound. In questo grafico composto per ogni valore della scala vengono riportati quattro puntini, per questa tecnica ogni matrice dei rating è stata tenuta sia in versione continua che in versione intera. A queste due matrici vengono applicati i modelli MF e UserKnn, da cui otteniamo quattro risultati, quelli di colore blu sono i risultati forniti dal modello MF, mentre quelli arancioni quelli del modello UserKnn.

### 6.1.2.2 Normalizzazione min-max gruppo user-based

In questa sezione vedremo la normalizzazione min-max applicata al gruppo user-based, ossia quello dove ciascuno di essi contiene solo le triplette di un singolo user. Il grafico composto 6.2 di seguito possiede la stessa struttura di quello (6.1) del paragrafo precedente. Sono presenti risultati positivi nell'incrocio tipo dataset ricambi - espressione d'interesse (numero fatture).

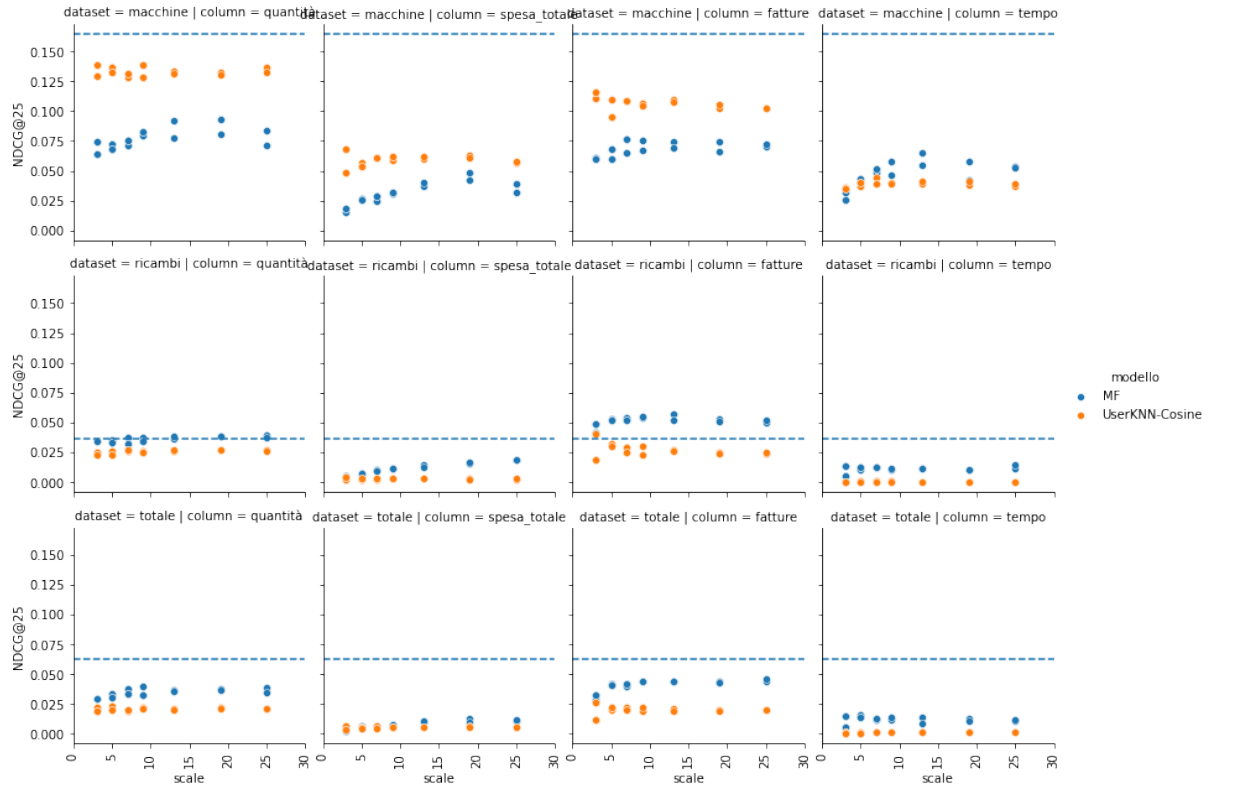


Figure 6.2: Risultati normalizzazione min-max gruppo user-based.

### 6.1.2.3 Normalizzazione min-max gruppo user-category-based

Vediamo la normalizzazione min-max applicata al gruppo user-category-based, dove le triplette vengono divise per user e per categoria. Le macchine hanno a disposizione solo la divisione in categorie secondo la gerarchia prodotto, mentre i ricambi possono essere divisi secondo le categorie di 2° e 3° livello della gerarchia prodotto e dal gruppo merceologico. Per il tipo dataset totale si proverà ad usare prima solo le categorie della gerarchia prodotto, poi le categorie della gerarchia prodotto solo per le macchine mentre per i ricambi le categorie del gruppo merceologico.

#### 6.1.2.3.1 Macchine

Come possiamo vedere dal grafico composto 6.3, sulle righe abbiamo i livelli della gerarchia prodotto (1°, 2°, 3° livello), mentre sulle colonne abbiamo le espressioni d'interesse. Non sono presenti risultati che superino la soglia critica.

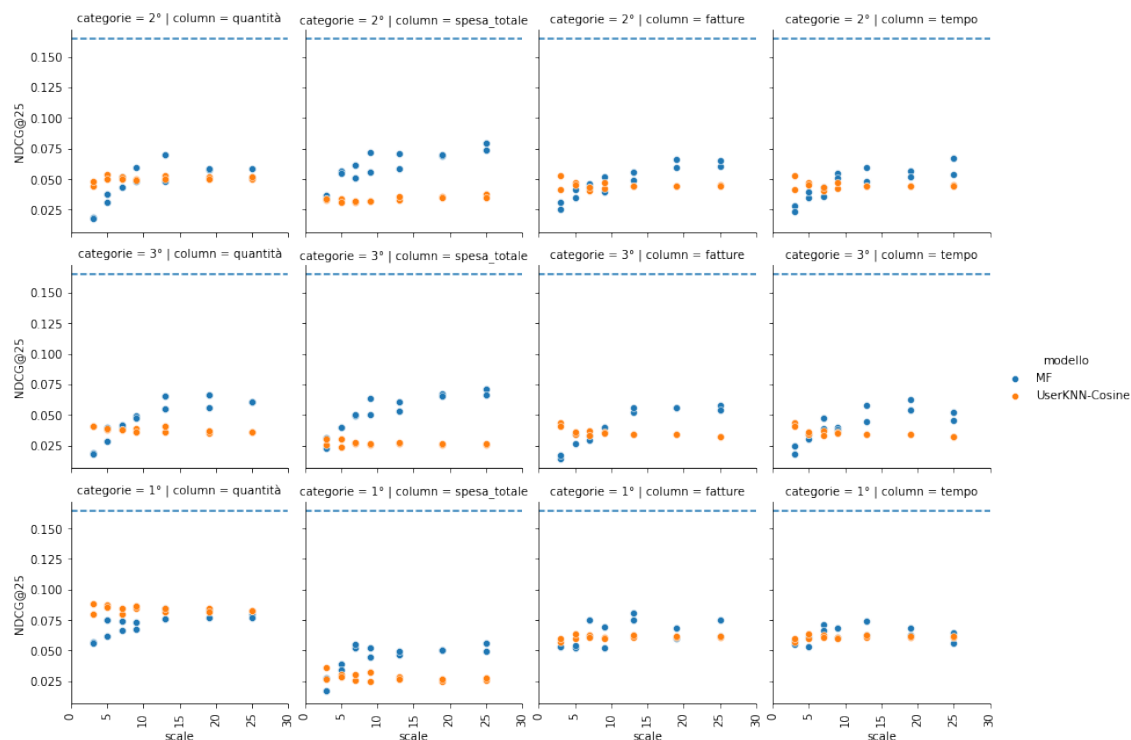


Figure 6.3: Risultati normalizzazione min-max gruppo user-category based dataset macchine.

### 6.1.2.3.2 Ricambi

Come possiamo vedere dal grafico composto 6.4, sulle righe abbiamo il 1° e 2° livello della gerarchia prodotto e la divisione del gruppo merceologico (*MATKL*), mentre sulle colonne abbiamo le espressioni d'interesse. Abbiamo risultati che superano la soglia critica dividendo le triplette secondo le categorie di 2° livello della gerarchia prodotto con le espressioni numero di fatture e recentezza.

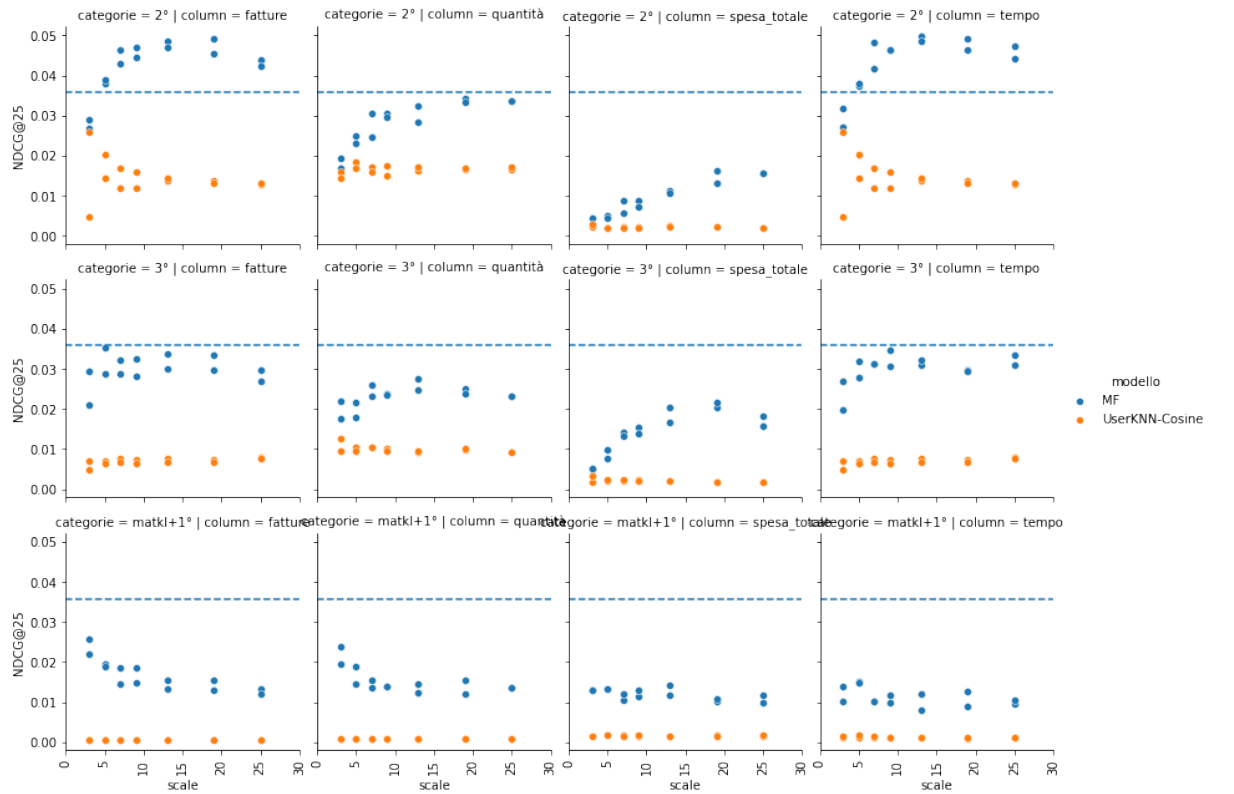


Figure 6.4: Risultati normalizzazione min-max gruppo user-category based dataset ricambi.

### 6.1.2.3.3 Totale

Come possiamo vedere dal grafico composto 6.5 sulle righe abbiamo il 1°, 2° e 3° livello della gerarchia prodotto e successivamente le precedenti categorie applicate solo alle macchine, mentre i ricambi si sono divisi secondo il gruppo merceologico. Sulle colonne abbiamo le espressioni d'interesse. Non abbiamo alcun risultato sopra la soglia.

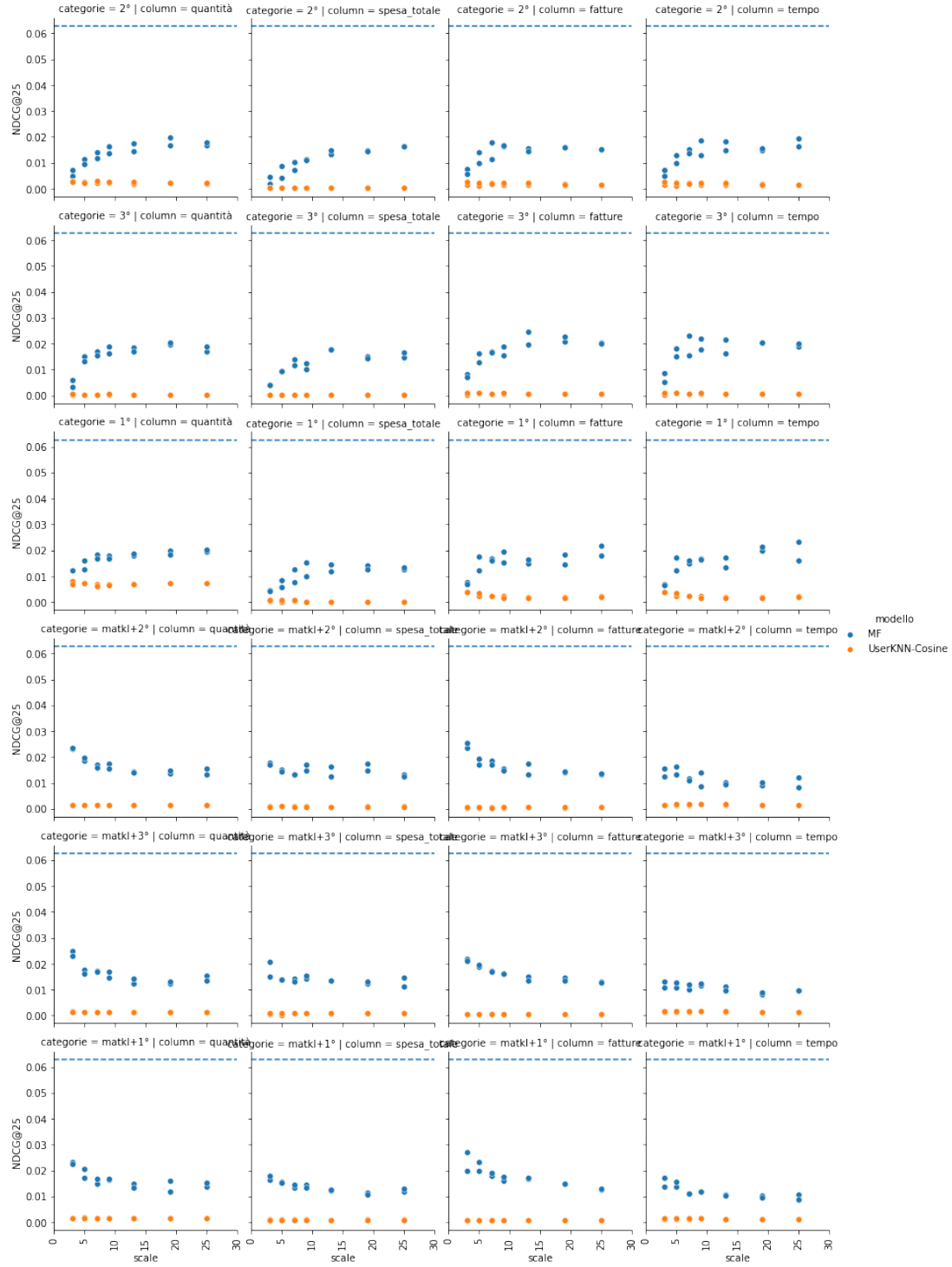


Figure 6.5: Risultati normalizzazione min-max gruppo user-category based dataset totale.

#### 6.1.2.4 Tecnica ordered-based gruppo globale

In questa sezione vedremo la tecnica di preprocessing ordered-based applicata al gruppo globale, ossia quello contenente tutte le triplete. Nel grafico composto 6.6 sottostante possiamo vedere sulle righe i tipi di dataset (macchine, ricambi, totale), mentre sulle colonne le *espressioni d'interesse*. Ciascun grafico poi mostra sulle ascisse la *scale* della matrice dei rating e sulle ordinate il risultato ottenuto da tale matrice rispetto la metrica  $NDCG@25$ .

La linea tratteggiata blu rappresenta il risultato del modello MostPop che viene usato come bound. In questo grafico composto per ogni valore della scala vengono riportati quattro puntini, per questa tecnica ogni matrice dei rating è stata tenuta sia in versione continua che in versione intera. A queste due matrici vengono applicati i modelli MF e UserKnn, da cui otteniamo quattro risultati, quelli di colore blu sono i risultati forniti dal modello MF, mentre quelli arancioni quelli del modello UserKnn.

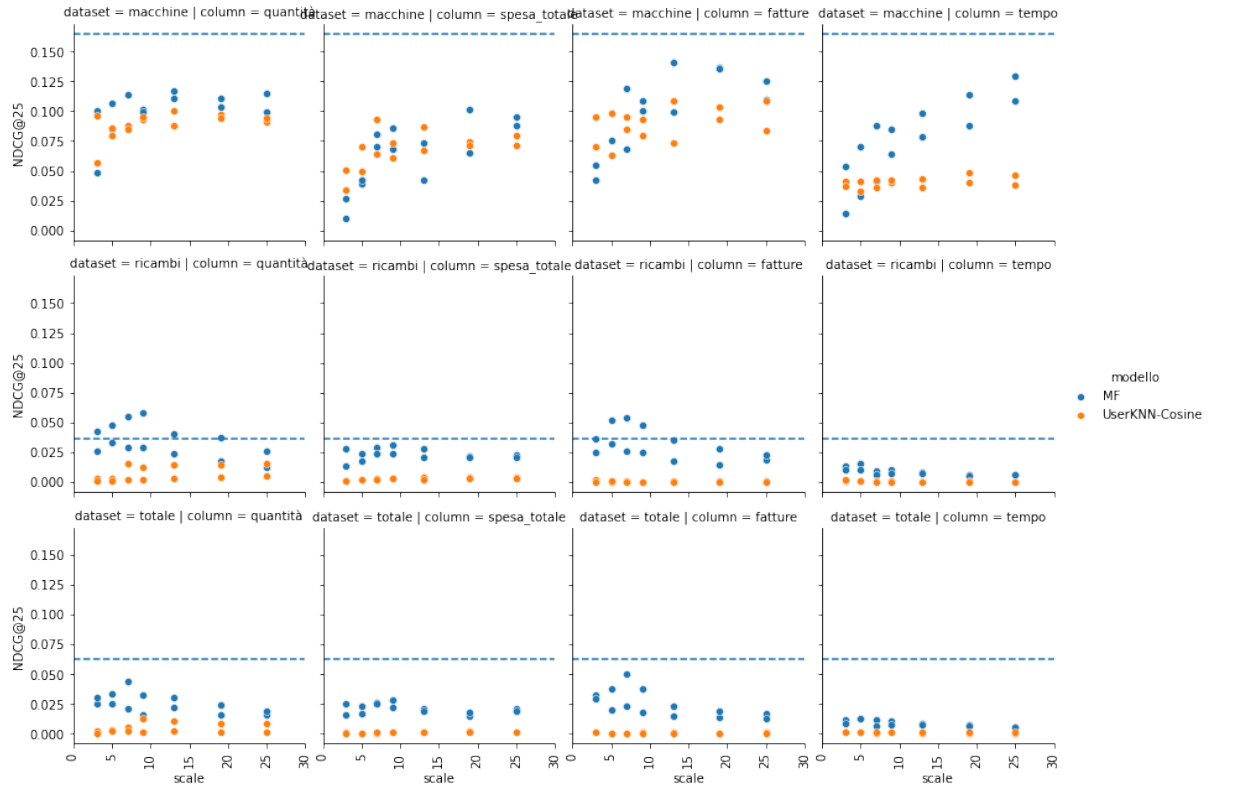


Figure 6.6: Risultati ordered-based gruppo globale.

Abbiamo dei risultati positivi nelle matrici di tipo dataset ricambi e espressione d'interesse quantità e numero fatture.

### 6.1.2.5 Tecnica ordered-based gruppo user-based

Di seguito possiamo vedere i risultati relativi la tecnica ordered-based su gruppi di triplette appartenenti allo stesso user. Il grafico composto 6.7 possiede la stessa struttura di quello (6.6) del paragrafo precedente.

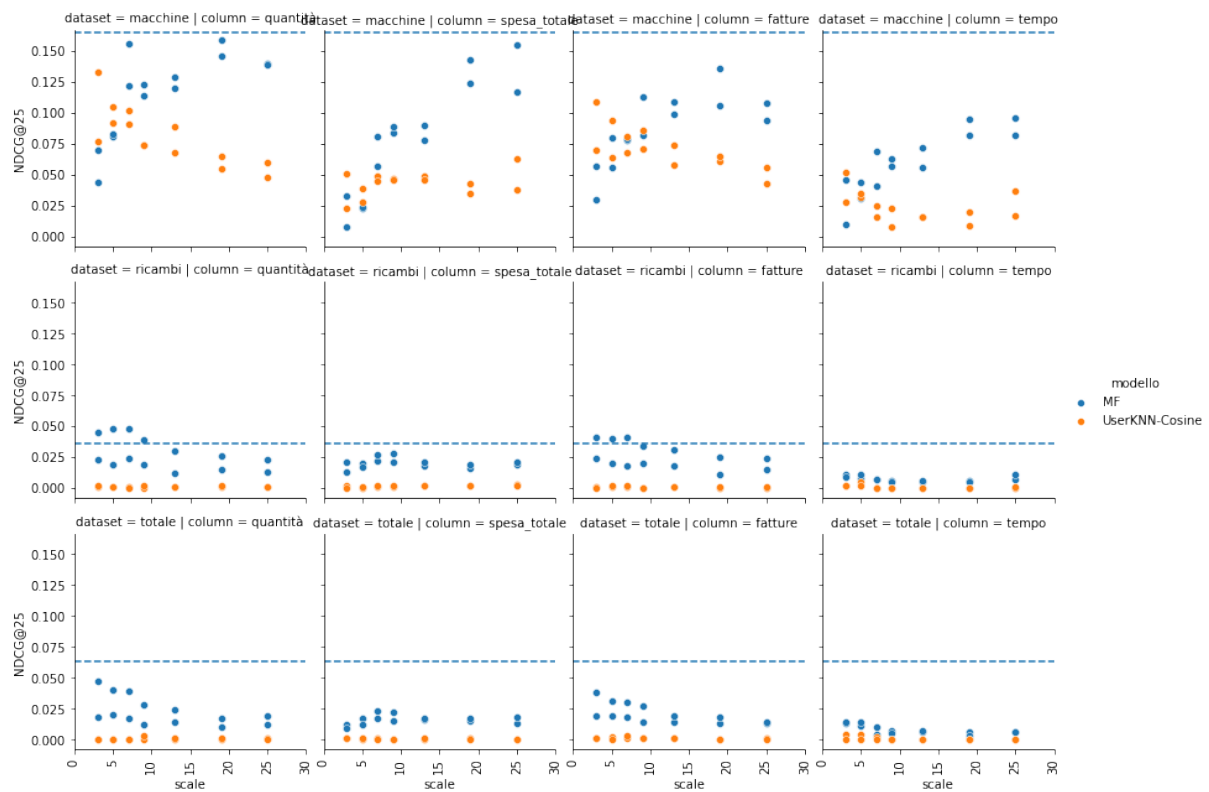


Figure 6.7: Risultati ordered-based gruppo user-based.

Abbiamo qualche risultato positivo sempre per il tipo dataset ricambi con espressione d'interesse quantità e numero fatture, possiamo dire che siano leggermente peggiori dei precedenti.

### 6.1.2.6 Tecnica ordered-based gruppo user-category-based

Vediamo la tecnica di preprocessing ordered-based applicata al gruppo user-category-based, dove le triplette vengono divise per user e per categoria. Le macchine hanno a disposizione solo la divisione in categorie secondo la gerarchia prodotto, mentre i ricambi possono essere divisi secondo le categorie di 2° e 3° livello della gerarchia prodotto e dal gruppo merceologico. Per il tipo dataset totale si proverà ad usare prima solo le categorie della gerarchia prodotto, poi le categorie della gerarchia prodotto solo per le macchine mentre per i ricambi le categorie del gruppo merceologico.

#### 6.1.2.6.1 Macchine

Come possiamo vedere dal grafico composto 6.8, sulle righe abbiamo i livelli della gerarchia prodotto (1°, 2°, 3° livello), mentre sulle colonne abbiamo le espressioni d'interesse. Non sono presenti risultati che superino la soglia critica.

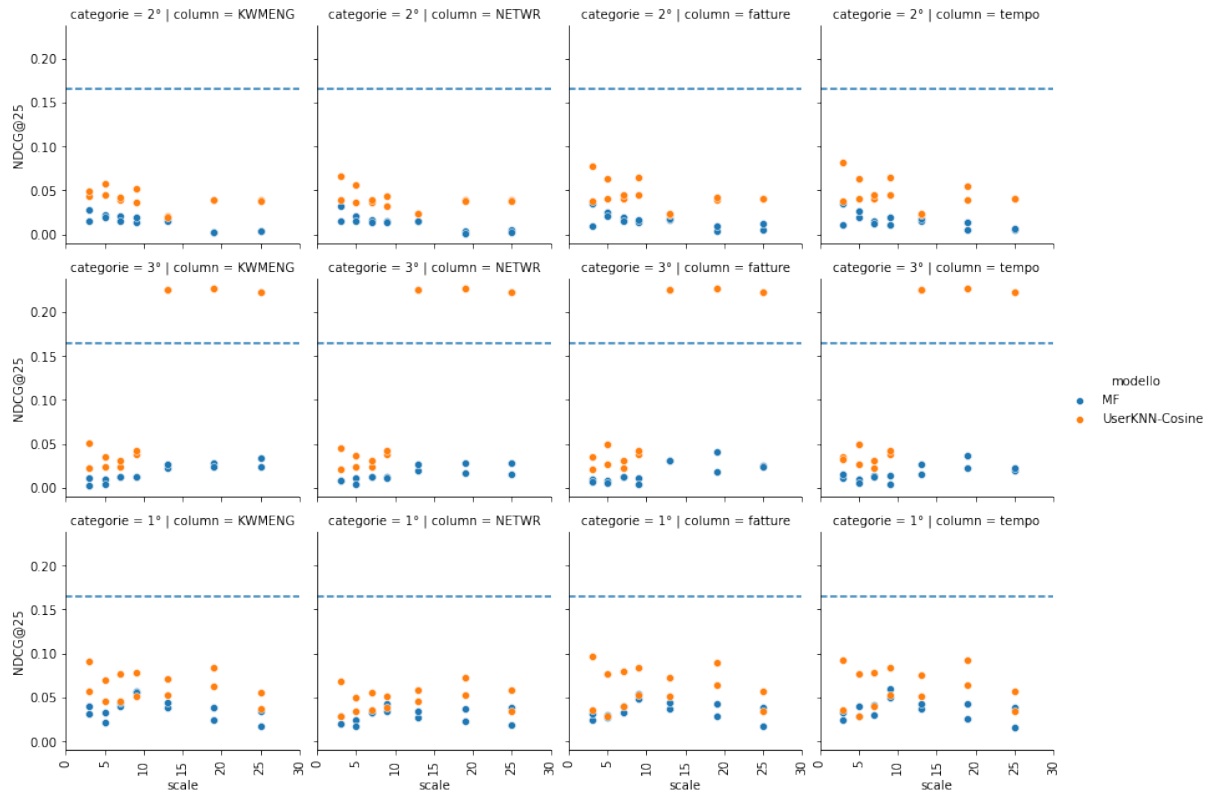


Figure 6.8: Risultati ordered-based gruppo user-category based dataset macchine.

### 6.1.2.6.2 Ricambi

Come possiamo vedere dal grafico composto 6.9, sulle righe abbiamo le categorie di 1° e 2° livello della gerarchia prodotto e la divisione del gruppo merceologico (*MATKL*), mentre sulle colonne abbiamo le espressioni d'interesse. Abbiamo risultati che superano la soglia critica dividendo le triplette secondo le categorie di 2° e 3° livello con le espressioni quantità, numero fatture e recentezza.

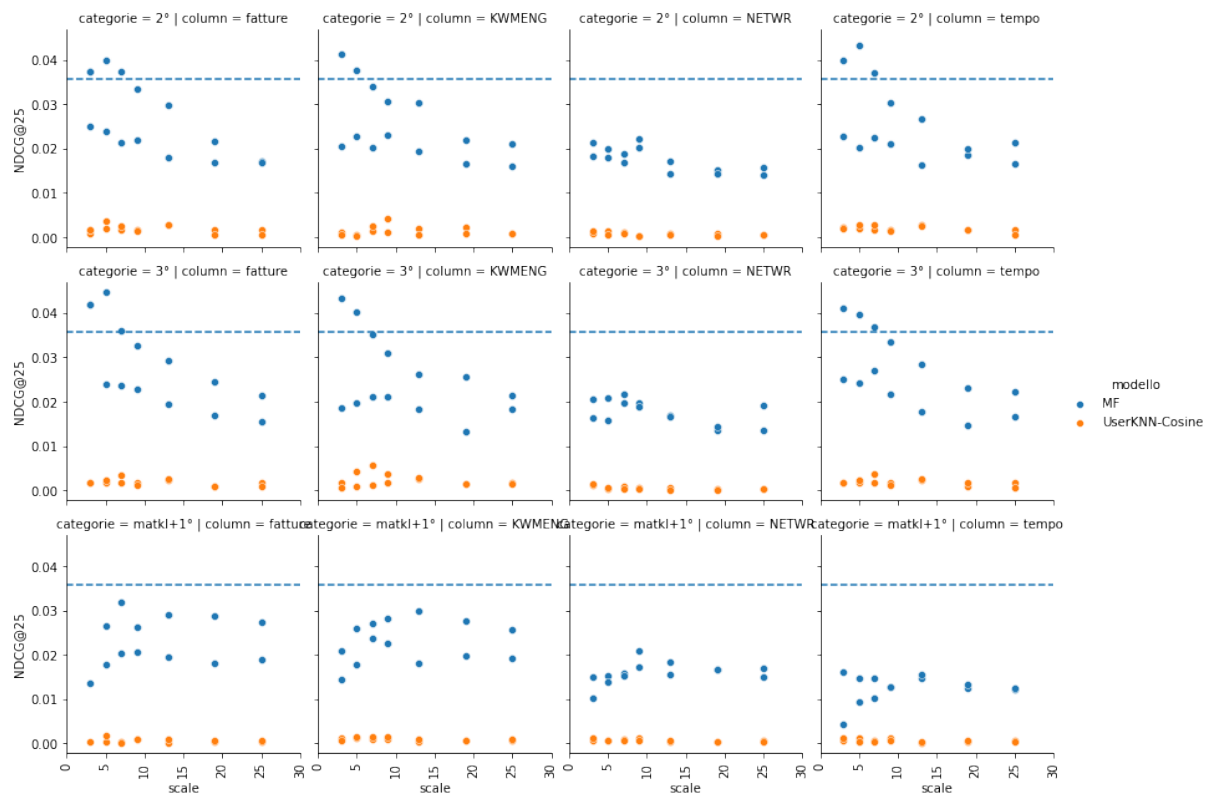


Figure 6.9: Risultati ordered-based gruppo user-category based dataset ricambi.

### 6.1.2.6.3 Totale

Come possiamo vedere dal grafico composto 6.10, sulle righe abbiamo il 1°, 2° e 3° livello della gerarchia prodotto e successivamente le precedenti categorie applicate solo alle macchine, mentre i ricambi si sono divisi secondo il gruppo merceologico. Sulle colonne abbiamo le espressioni d'interesse. Non abbiamo alcun risultato sopra la soglia.



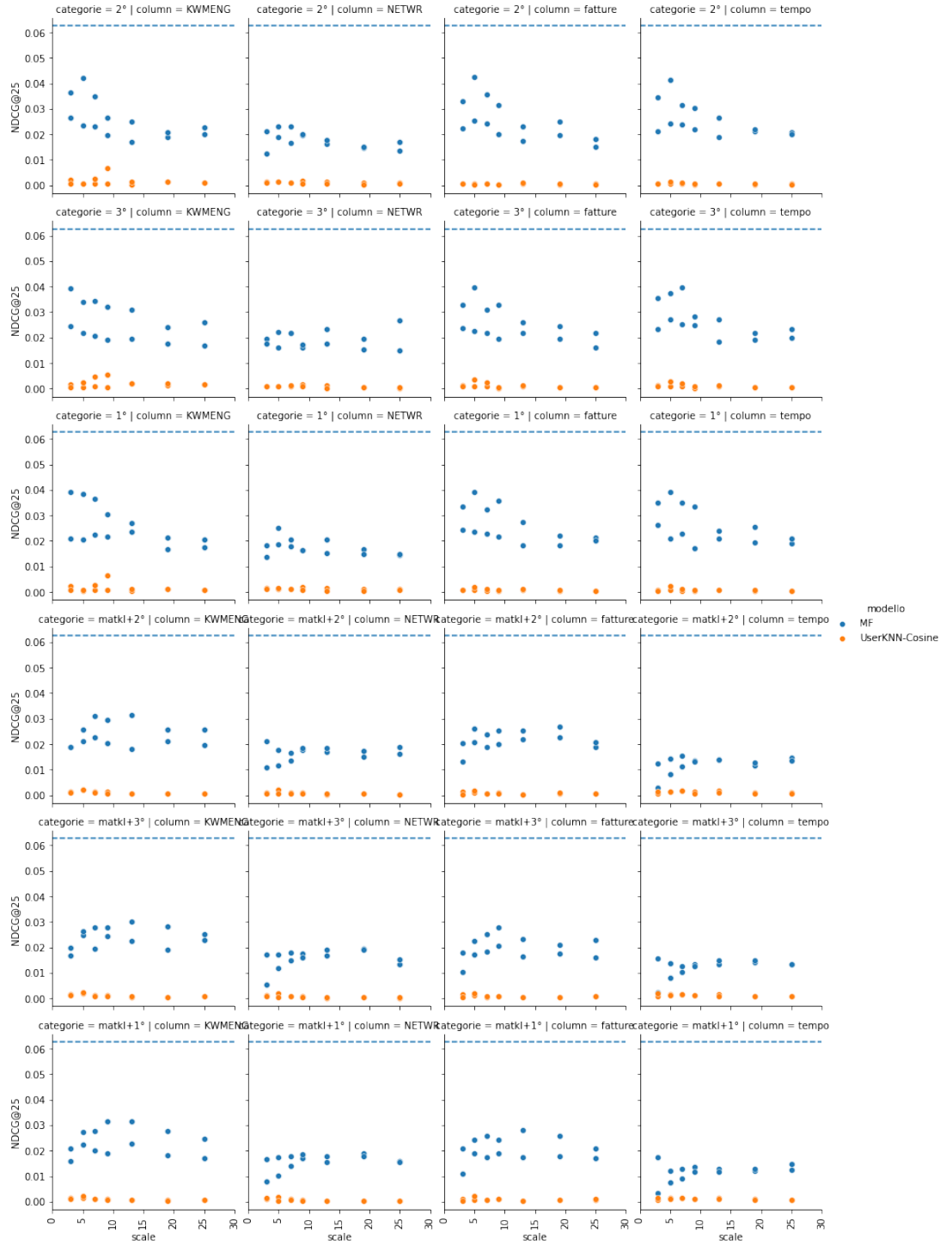


Figure 6.10: Risultati ordered-based gruppo user-category based dataset totale.

### 6.1.2.7 Tecnica product-based

In questa sezione vedremo la tecnica di preprocessing product-based, dove si assegna ad ogni item lo stesso rating. Nel grafico composto 6.11 possiamo vedere sulle righe i tipi di dataset (macchine, ricambi, totale), mentre sulle colonne le *espressioni d'interesse*. Ciascun grafico poi mostra sulle ascisse la *scale* della matrice dei rating e sulle ordinate il risultato ottenuto da tale matrice rispetto la metrica  $NDCG@25$ . La linea tratteggiata blu rappresenta il risultato del modello MostPop che viene usato come bound. In questo grafico composto per ogni valore della scala vengono riportati quattro puntini, per questa tecnica ogni matrice dei rating è stata tenuta sia in versione continua che in versione intera. A queste due matrici vengono applicati i modelli MF e UserKnn, da cui otteniamo quattro risultati, quelli di colore blu sono i risultati forniti dal modello MF, mentre quelli arancioni quelli del modello UserKnn. Possiamo notare come questo approccio fornisca risultati positivi per ciascun tipo di dataset e per quasi tutte le espressioni d'interesse.

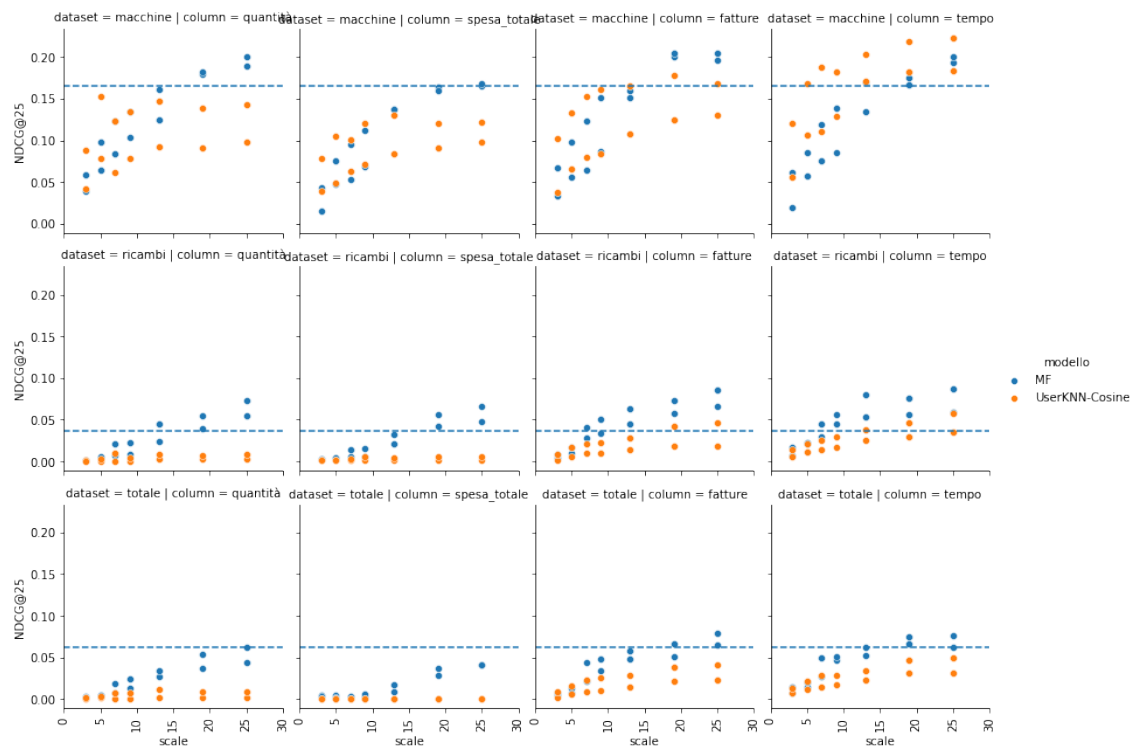


Figure 6.11: Risultati tecnica product-based.

### 6.1.3 Fase test avanzata

Per sintetizzare i risultati precedenti usiamo la tabella 6.6 dove sulle righe sono presenti i diversi metodi annessi ai gruppi e sulle colonne il tipo dataset, dov'è presente un 1 sappiamo che c'è stato almeno un risultato sopra la soglia.

<i>tecniche</i>	<i>macchine</i>	<i>ricambi</i>	<i>totale</i>
min-max globale		1	
min-max user-based		1	
min-max user-category-based		1	
ordered globale		1	
ordered user-based		1	
ordered user-category-based	1	1	
product-based	1	1	1

Table 6.6: Riassunto tecniche che hanno superato la fase preliminare per dataset.

Se dovessimo stilare una classifica su questi risultati preliminari, potremmo affermare i seguenti punti:

- la **tecnica product-based** funziona molto bene su tutti i dataset e finora ha restituito i risultati migliori;
- la **normalizzazione min-max** funziona solo sul tipo dataset ricambi, tra i tre gruppi quelli più efficaci sono quello user-based e user-category-based, sembra funzionare meglio su questo tipo di dataset rispetto la tecnica ordered-based;
- la **tecnica ordered-based** funziona sul tipo dataset macchine con il gruppo user-category-based ma risulta peggiore della tecnica product-based, mentre sul tipo dataset ricambi i gruppi che funzionano meglio sono quello globale e user-category-based.

Dopo aver visto i risultati delle tecniche nella fase preliminare, si passa alla fase avanzata selezionando tra tutte le matrici dei rating quelle che hanno ottenuto un valore della metrica superiore alla soglia critica data dal MostPop.

Per ciascuna combinazione di matrice dei rating e modello, che avesse restituito un risultato superiore a quello soglia, si è proceduto ad un tuning dei parametri.

Una volta trovato quello migliore, in accordo al validation set, si è calcolato il risultato finale della metrica sul test set. Nella fase avanzata il nuovo bound era quello del VAECF, vediamo ora i risultati del test set con relativa soglia.

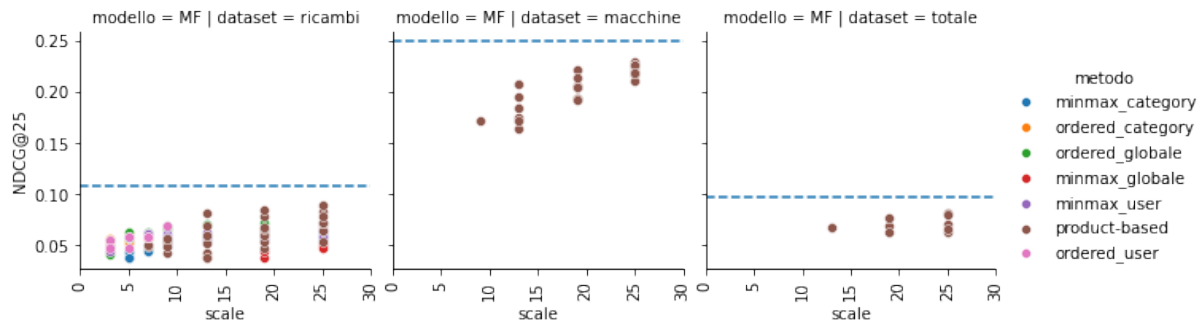


Figure 6.12: Risultati MF ottimizzato su test set con matrici selezionate.

Nel grafico 6.12 sono riportati i risultati ottenuti dai modelli MF ottimizzati sul test set. Nelle colonne troviamo rispettivamente il tipo dataset ricambi, macchine, totale con i loro relativi nuovi bound. Possiamo vedere in tutti e tre i grafici che i risultati non superano i valori soglia.

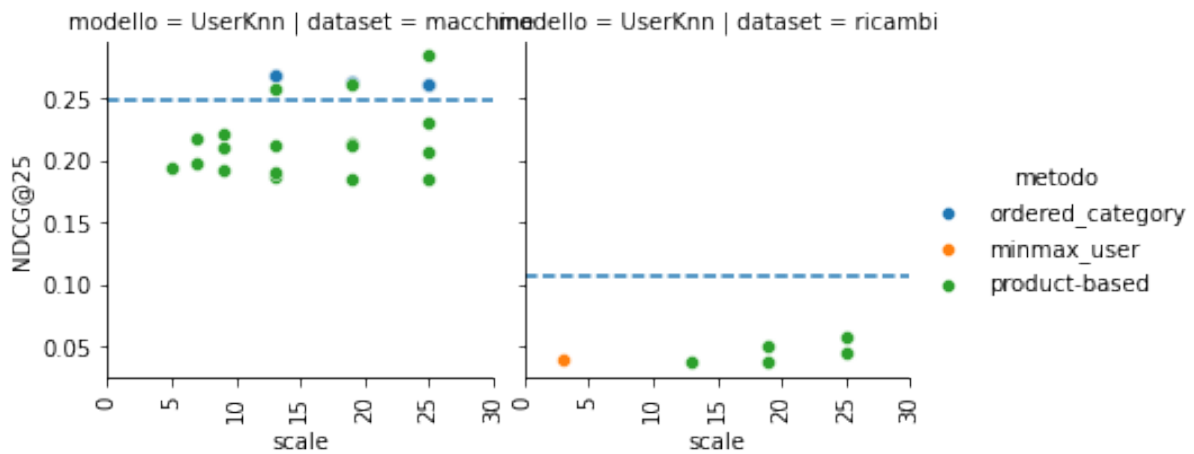


Figure 6.13: Risultati UserKnn ottimizzato su test set con matrici selezionate.

Nel grafico 6.13 vengono mostrati i risultati ottenuti dai modelli UserKnn ottimizzati sul test set. Le colonne riportano rispettivamente il tipo dataset macchine e ricambi, possiamo vedere come il metodo product-based primeggi anche se ci

sono due risultati della tecnica ordered-based, basata su categorie, oltre la soglia. Per il tipo dataset ricambi non c'è alcun risultato oltre la soglia.

Per quanto riguarda la distribuzione dei rating nelle matrici, quelle che hanno risultati superiori alla soglia hanno per circa il 66% una distribuzione gaussian-like. Inoltre per quanto riguarda i valori della scala, i risultati migliori si sono ottenuti con il valore massimo di questa (25).

Vediamo ora i risultati finali sul test set per ogni tipo di dataset nella tabella 6.14.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.7288	0.1737	0.2223	0.2844	0.3203
ricambi	0.4930	0.0801	0.0919	0.1075	0.1401
totale	0.4343	0.0988	0.0874	0.0964	0.1398

Figure 6.14: Tabella risultati finali sul test set.

Riportiamo per ogni tipo di dataset tutte le informazioni relative alla matrice dei rating e alla tecnica con cui è stata generata, oltre che il modello utilizzato.

<i>dataset</i>	<i>modello</i>	tecnica	espressione interesse	scala	distribuzione
macchine	UserKnn	product-based	recentezza	25	gaussian-like
ricambi	VAECF	implicito		1	
totale	VAECF	implicito		1	

Figure 6.15: Informazioni sul modello e la tecnica dei risultati della tabella 6.14

## 6.2 Risultati matrici grezze combinate

Vediamo i risultati delle tecniche basate sulla combinazione delle matrici, si è utilizzato il validation set per il training e il test set per i risultati finali. Entrambe le tecniche restituiscono risultati simili tra loro e di poco al di sotto di quelli ottenuti con le matrici singole. Nelle sezioni successive vedremo due grafici aventi sulle righe i tipi di dataset e sulle colonne le espressioni d'interesse. La linea tratteggiata è il bound dato dal VAECF sul test set.

### 6.2.1 Combinazione liste $TopN$

Questa tecnica, nonostante nell'incrocio dataset macchine - tecnica product-based sembri funzionare, in realtà è stato fatto un test misurando il coefficiente di similarità delle liste  $TopN$  prodotte da ciascun modello e si è visto che è quasi sempre molto alto. Quindi la combinazione delle liste non produce una lista così diversa da quelle di partenza. Vediamo i risultati nel grafico composto 6.16.

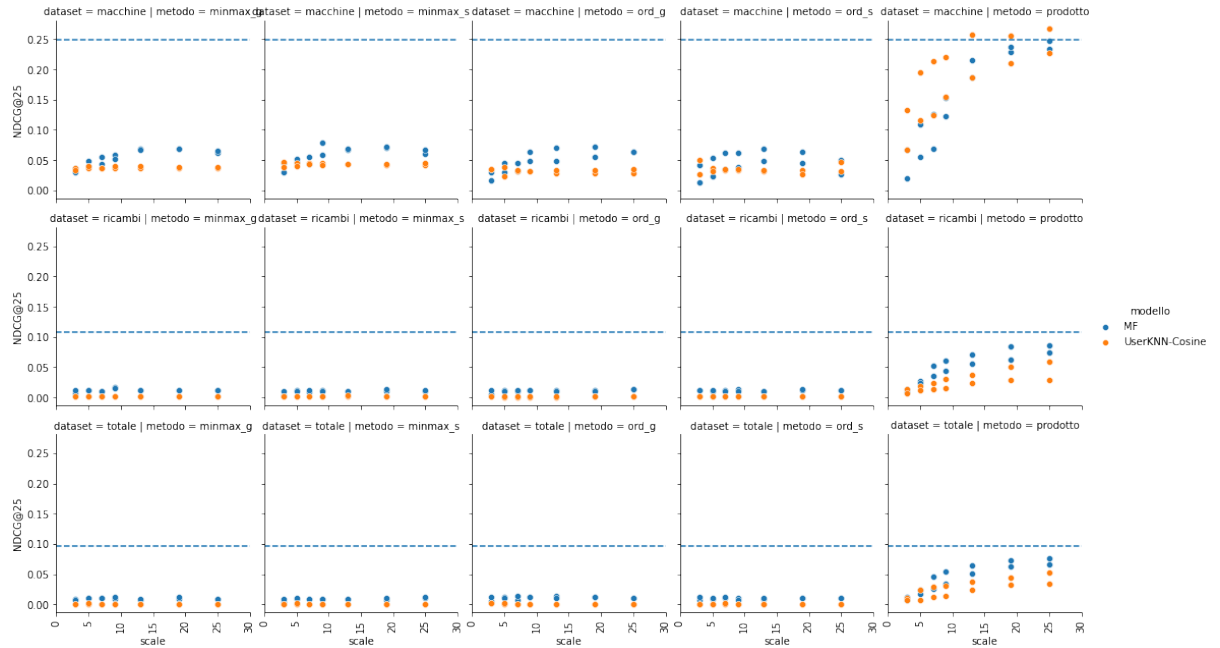


Figure 6.16: Grafico composto metodo basato su combinazione liste  $TopN$ .

Nella tabella 6.7 vediamo i risultati migliori per ciascun dataset sul test set, tutti forniti dalla tecnica product-based.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.5658	0.1516	0.2049	0.2683	0.3041
ricambi	0.7873	0.0756	0.0759	0.0865	0.1427
totale	0.7782	0.0577	0.0648	0.0767	0.1273

Table 6.7: Tabella risultati finali combinazione liste  $TopN$ .

## 6.2.2 Media matrici dei rating

Il motivo per cui la versione basata sulle liste *TopN* non funziona è che le liste di partenza non sono così diverse in quanto molte di esse si basano sull'ordinamento o su una normalizzazione su scala simile, la matrice media ottenuta sarà quindi molto simile a quelle di partenza.

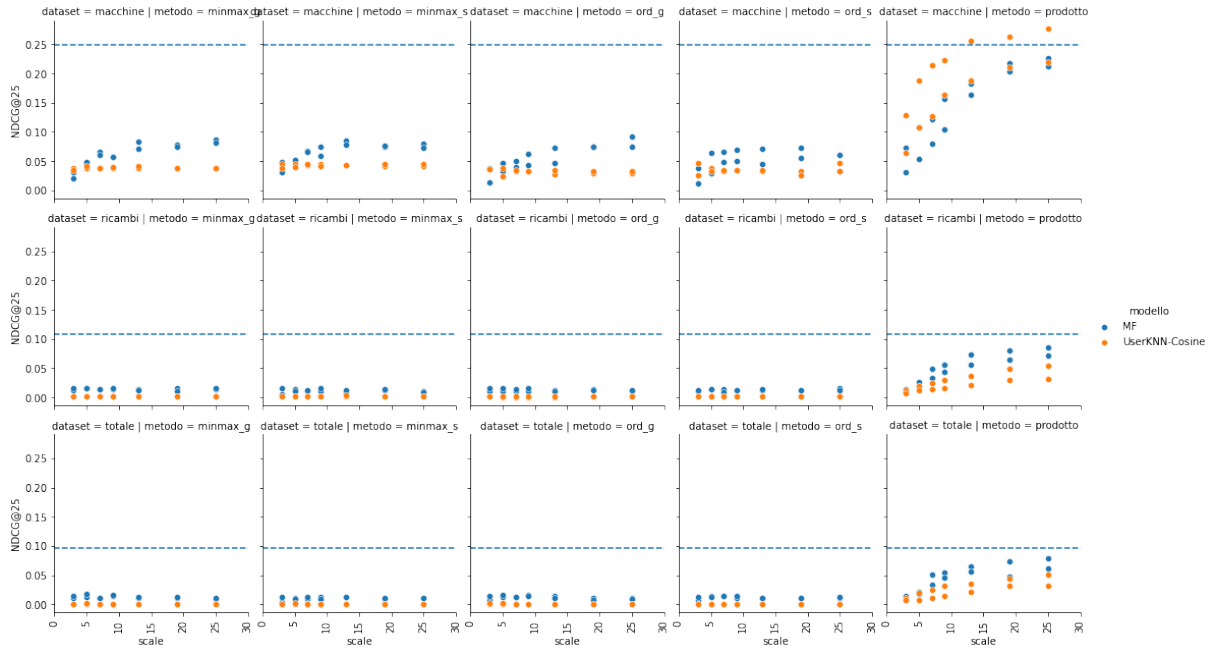


Figure 6.17: Grafico composto dei risultati metodo basato sulla media matrici grezze.

Nella tabella 6.8 vediamo i risultati migliori per ciascun dataset sul test set.

<i>dataset</i>	<i>AUC</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.7288	0.1708	0.2149	0.2775	0.3133
ricambi	0.8861	0.0757	0.0766	0.0860	0.1416
totale	0.8810	0.0605	0.0649	0.0792	0.1256

Table 6.8: Tabella risultati finali del metodo basato su liste *TopN* sul test set.

### 6.3 Risultati con approccio next-basket

In questa sezione vedremo i risultati dell'approccio next-basket. Per prima cosa riportiamo quali sono stati i parametri selezionati con il tuning per ciascun dataset, riportati in tabella 6.9.

<i>dataset</i>	$\alpha$	$q$	$r$
macchine	0.5	100	$\infty$
ricambi	0.75	50	$\infty$
totale	0	100	$\infty$

Possiamo notare che alla fine il tuning ha portato ad avere una finestra di recentezza  $r = \infty$ , quindi stiamo usando la popolarità *popularity user-wise*.

Table 6.9: Tabella parametri selezionati.

Possiamo inoltre vedere che la località  $q$  è comunque alta mentre, per quanto riguarda *alpha*, osserviamo che nel dataset macchine si calcola la probabilità composta al 50%. Nei ricambi si dà più importanza a quella dello user in esame, ed infine nel totale si considera solo la probabilità composta dello user esterno.

Vediamo i risultati, in tabella 6.10, dei modelli ottimizzati sul validation set.

<i>dataset</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.5832	0.6278	0.6506	0.6627
ricambi	0.1728	0.1892	0.2381	0.3317
totale	0.2196	0.2295	0.2834	0.3653

Table 6.10: Tabella risultati approccio next-based sul validation set.

E ora i corrispondenti risultati, in tabella 6.11, con il test set

<i>dataset</i>	<i>NDCG@5</i>	<i>NDCG@10</i>	<i>NDCG@25</i>	<i>NDCG@100</i>
macchine	0.6049	0.6476	0.6726	0.6741
ricambi	0.2261	0.2403	0.2915	0.3811
totale	0.1955	0.2045	0.2595	0.3537

Table 6.11: Tabella risultati approccio next-based sul test set.

Ricordiamo che questi risultati non sono confrontabili con quelli delle sezioni precedenti, ma sono comunque molti interessanti.



## 6.4 Confronto critico

Il metodo di normalizzazione min-max non altera l'iniziale distribuzione dei valori, questi dopo essere stati scalati saranno per la maggior parte concentrati all'inizio dell'intervallo di rating. Questo fa sì che molti user abbiano solo item considerati pessimi, questo porta a similarità molto alta tra tutti gli user e non permette di distinguere gli item interessanti per esso. Il metodo in generale non funziona e non se ne raccomanda l'applicazione.

Il metodo ordered-based rispetto al precedente risolve molti problemi legati alla distribuzione dei rating, infatti ora la cardinalità di ciascun valore nella scala dei rating è circa uguale. La media dei rating degli user è collocata nella parte centrale della scala, mentre quella degli item copre tutta la scala, anche questo metodo non sembra funzionare proprio in virtù della variabilità dei rating per ogni item, anche se i risultati sono leggermente migliori dei precedenti.

Il metodo product-based assegnando ad un item lo stesso rating per tutti gli user ha fatto sì che la media dei rating degli user non sia più centrata, in compenso però nella similarità è aiutata in quanto i rating di uno stesso item sono tutti uguali. Il suo funzionamento è molto legato alla popolarità rispetto le espressioni d'interesse e questo sembra aver dato risultati superiori rispetto ai precedenti, sicuramente la popolarità è uno degli ingredienti fondamentali per il raggiungimento degli obiettivi prefissati. In generale anche i risultati ottenuti con MostPop e VAECF sono bassi, possiamo quindi dire osservando i dati che il preprocessing sulle matrici grezze non funziona e di riflesso neanche quello combinato che si basa su di esse. Lavorare sullo storico vendite con i precedenti metodi potrebbe non essere la strada giusta, con il metodo next-basket invece si sono ottenuti risultati promettenti anche se allenato su dataset più ristretti e per questo più incline ad ottenere risultati positivi. Il dataset tipo macchina è relativamente piccolo già di suo, quindi risulta ancora più facile ottenere buoni risultati, mentre quello totale funziona comunque peggio rispetto a quello di tipo ricambi, dall'unione di macchine e ricambi non sembra si possa ottenere quindi risultati positivi. Ciò che si può concludere da questi test è che la popolarità giochi un ruolo importante nelle raccomandazioni e che l'approccio da seguire sia quello next-basket magari combinato con le espressioni d'interesse.



# 7

## Conclusioni

### 7.1 Problemi critici risolti

Dopo aver visto i risultati siamo in grado di dare una valutazione generale agli stessi. Per quanto riguarda l'approccio del preprocessing alle matrici grezze siamo in grado di dire che le tecniche group-based, ossia la normalizzazione min-max e la tecnica ordered-based, non hanno restituito alcun risultato positivo. Si è però notato che confrontare le triplette per categorie in base agli item può portare ad un miglioramento dei risultati. La tecnica product-based che gioca fortemente sulla popolarità degli item dal punto di vista delle espressioni d'interesse, ha ritornato buoni risultati soprattutto rispetto la matrice grezza della recentezza.

L'aver generato matrici dei rating su scale diverse è stata un'idea positiva in quanto ha influenzato i risultati finali, questi sono stati più alti dove la scala era maggiore. I risultati in generale non sono comunque soddisfacenti in termini quantitativi e sicuramente possono essere migliorati, inoltre il tipo dataset totale, unione di macchine e ricambi insieme, non ha restituito risultati migliori delle controparti separate, quindi si ritiene opportuno considerarli come dataset separati. Per quanto riguarda l'approccio combinato non si sono ottenuti i risultati sperati in quanto le matrici di partenza e le liste *TopN* prodotte sono troppo simili tra loro. L'approccio next-basket ha restituito buoni risultati ma va ricordato che il

bacino di utenti si è ridotto da circa 320 a 250 user, in ogni caso i risultati per quanto riguarda il dataset tipo macchine sono molto buoni, mentre per i ricambi siamo su valori inferiori, anche se c'è da dire che rispettivamente i due dataset lavorano su circa 500 e 7500 item. In conclusione i risultati ottenuti sul dataset macchine sono incoraggianti, non si può dire lo stesso per quello dei ricambi, ma l'idea è quella di trattarli separatamente.

## 7.2 Problemi aperti

Credo possa esistere una relazione tra i risultati positivi del metodo product-based, con espressione d'interesse recentezza, e quelli dati dall'approccio next-basket, in quanto entrambi sono basati sulla popolarità e sulla recentezza. Sarebbe interessante indagare in questo senso per capire eventualmente se si possano combinare i due approcci. Sarebbe possibile trovare nei dati altre espressioni d'interesse o possiamo usarle come feature per l'approccio content-based. Non abbiamo esplicitato il confronto tra item in quanto l'approccio content-based non aveva abbastanza informazioni e il modello ItemKnn, sia per risultati scarsi che per tempi di valutazione è stato accantonato. Non sono stati affrontati problemi quali cold start per nuovi user e item.

## 7.3 Futuri sviluppi

Sicuramente uno degli sviluppi possibili è quello di applicare la divisione secondo categorie all'approccio product-based, andando così a confrontare per popolarità item più "simili". Inoltre anche gli approcci combinati hanno usato modelli su cui non è stato effettuato il tuning dei parametri, questo potrebbe migliorare i risultati anche se le matrici rimangono comunque molto simili tra loro, invece per ovviare a ciò potremmo combinare matrici create con i diversi metodi sviluppati.

# Glossario

- **Matrici Grezze User-item:** matrici aventi sulle righe gli user e sulle colonne gli item, nell'incrocio troviamo misurazioni quantitative, quali la quantità per esempio. Sono dette grezze in quanto devono essere raffinate per ottenere successivamente dei rating;
- **Espressioni D'interesse:** aspetti che cercano di misurare l'interesse di uno user verso un'item, nello specifico di questo lavoro sono la quantità acquistata, la spesa totale e il numero di fatture di quell'item, la recentezza dell'acquisto;
- **Dealer:** sono aziende che acquistano prodotti da un'altra azienda e poi li rivendono a clienti terzi;
- **Preprocessing:** serie di operazioni atte a manipolare i dati per migliorare le performance dei modelli a cui vengono applicati;
- **Matrice Dei Rating:** una matrice use-item avente valori in una scala ben definita;
- **Backpropagation:** algoritmo per l'allenamento delle reti neurali;
- **TopN:** lista di lunghezza N contenente tutti gli item ordinati secondo gli interessi di uno user;



# Lista degli acronimi

<b>ERP</b> .....	Enterprise Resource Planning
<b>CRM</b> .....	Customer Relationship Management
<b>BTB</b> .....	Business To Business
<b>RS</b> .....	Sistema di Raccomandazione
<b>MF</b> .....	Matrix Factorization
<b>CF</b> .....	Collaborative Filtering
<b>AE</b> .....	Auto-Encoder
<b>VAE</b> .....	Variational Auto-Encoder
<b>VAECF</b> .....	Variational Auto-Encoder for Collaborative Filtering
<b>AUC</b> .....	Area Under the Curve
<b>NDCG</b> .....	Normalized Discount Cumulative Gain
<b>UP-CF</b> .....	User Popularity-based Collaborative Filtering
<b>UserKnn</b> .....	User k-nearest neighbour
<b>ItemKnn</b> .....	Item k-nearest neighbour





# Riferimenti bibliografici

- [1] Bank, D., Koenigstein, N., and Giryas, R. (2021). Autoencoders.
- [2] Breese, J. S., Heckerman, D., and Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. Inglese. *Microsoft Research*.
- [3] Dhanjal, C., Gaudel, R., and Clemencon, S. (2015). AUC optimisation and collaborative filtering.
- [4] Faggioli, G., Polato, M., and Aioli, F. (2020). Recency aware collaborative filtering for next basket recommendation. Inglese. *Association for Computing Machinery*.
- [5] Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.
- [6] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Inglese. *IEEE Computer Society*.
- [7] Liang, D., Hoffman, M. D., Krishnan, R. G., and Jebara, T. (2018). Variational autoencoders for collaborative filtering. Inglese. *International World Wide Web Conferences Steering Committee*.
- [8] Lops, P., de Gemmis, M., and Semeraro, G. (2011). *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105.
- [9] Salah, A., Truong, Q.-T., and Lauw, H. W. (2020). Cornac: A comparative framework for multimodal recommender systems. *Journal of Machine Learning Research*, 21(95):1–5.
- [10] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Inglese. *Association for Computing Machinery*.

- [11] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. *Springer-Verlag Berlin Heidelberg*.
- [12] Singh, P., Dutta Pramanik, P., Dey, A., and Choudhury, P. (2021). Recommender systems: An overview, research trends, and future directions. *International Journal of Business and Systems Research*, 15:14–52.
- [13] Wang, Y., Wang, L., Li, Y., He, D., Liu, T.-Y., and Chen, W. (2013). A theoretical analysis of ndcg type ranking measures.