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

Da fare

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 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 prospettive, 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

Lista delle figure

Lista delle tabelle

1 Introduzione

1.1 Contesto progetto

Nel mondo dei software ERP *(Enterprise Resource Planning)*, 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 CRM (*(Customer Relationship Management)*), il quale si occupa di tutte le modalità di gestione delle relazioni con il cliente.

1.2 L'idea di progetto

L'idea 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 rac-

comandazioni 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, 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 l'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, si voleva eseguire una sorta di preprocessing su di esse, andando a trasformarle in dei rating 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 sui 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**: le diverse tecniche di preprocessing utilizzate per trasformare i dati grezzi in valutazioni;
- **Capitolo 5**: le modalità con cui sono stati combinati i dati;
- **Capitolo 6**: una descrizione della libreria Cornac, dove sono implementati modelli e metriche per l'esecuzione di test;
- **Capitolo 7**: i risultati delle metriche rispetto i diversi algoritmi applicati al preprocessing dei dati;
- **Capitolo 8**: i risultati delle metriche rispetto i diversi algoritmi applicati al preprocessing dei dati nella loro versione combinata;
- **Capitolo 9**: i risultati delle metriche considerando il problema come un next basket recommendation;
- **Capitolo 10**: le conclusioni del lavoro svolto, andando a delineare problemi risolti, criticità e sviluppi per il futuro.

1.4 Convenzioni tipografiche

Il testo adotta le seguenti convenzioni tipografiche:

- ogni acronimo, abbreviazione, parola ambigua o tecnica viene spiegata e chiarificata alla fine del testo;
- ogni parola di glossario alla prima apparizione verrà etichetta come segue $parola^{[g]}$.

2
Analisi 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, ossia la parte descrittiva dove viene riportato l'acquirente, e una tabella per le posizioni della fattura, ossia la parte dove vengono riportati i materiali acquistati.
Abbiamo inoltre due tabelle che contengono rispettivamente l'anagrafica cliente e materiali, 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.
Quindi ricapitolando le principali tabelle disponibili sono le seguenti:

- **VBAK**: testata della fattura;
- **VBAP**: posizioni della fattura;
- **MARA**: anagrafica prodotto;
- **KNA1**: anagrafica materiali.

Andiamo ora a vedere 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 del'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**).

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 secondaria T179 vengono definiti i livelli di gerarchia e le diverse categorie. Vediamoli di seguito:

- **PRODH**: codice gerarchia prodotto;
- **STUFE**: livello gerarchia;
- **VTEXT**: descrizione testuale.

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

Nella tabella vengono mostrati i codici PRODH delle categorie di 1° livello, nella colonna *#tot* il numero di prodotti diversi per quella categoria, nella colonna *#sold* il numero di prodotti diversi acquistati almeno una volta appartententi a quella categoria ed infine il titolo della categoria. Come possiamo vedere le prime sei categorie con titolo in minuscolo hanno pochi prodotti catalogati in MARA e nessun prodotto venduto.

Chiaramento 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

Nella tabella 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 appartenti 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 rigurdanti 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 *dollari* sono state convertiti in *euro*;
- prodotti con misure di dimensione e volume, con unità di misura diversa rispettivamente dai *metri* e $metri^3$, 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

3.1 Introduzione

Uno dei campi più popolari al momento verso cui si rivolge una particolare attenzione è quello dei sistemi di raccomandazione, da ora in poi 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 cosidetti "attori" del problema, gli user e gli item, rispettivamente gli utenti del sistema e gli oggetti che si vuole consigliare. 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 intera in una scala da 1 a N, 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

		Items					
		1	*2*	...	*i*	...	*m*
Users	*1*	5	3		1	2	
	2		2				4
	:			5			
	u	3	4		2	1	
	:					4	
	n			3	2		

I rating sono organizzati in matrici, dette appunto user-item interaction matrix o semplicemente matrici dei rating (R), dove sulle righe abbiamo gli user mentre sulle colonne abbiamo gli item, 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 Task

L'obiettivo del sistema può essere quello di consigliare ad uno user una lista di N item, detta *TopN* che si ritiene possano interessargli, oppure dato un item si può trovare una lista di item che si considerino simili allo stesso in acccordo 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 il 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 sezione successive andiamo a spiegare più nel dettaglio il collaborative e conte-based filtering.

3.3.1 Collaborative filtering

Il collaborative filtering 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 UserKnn

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.

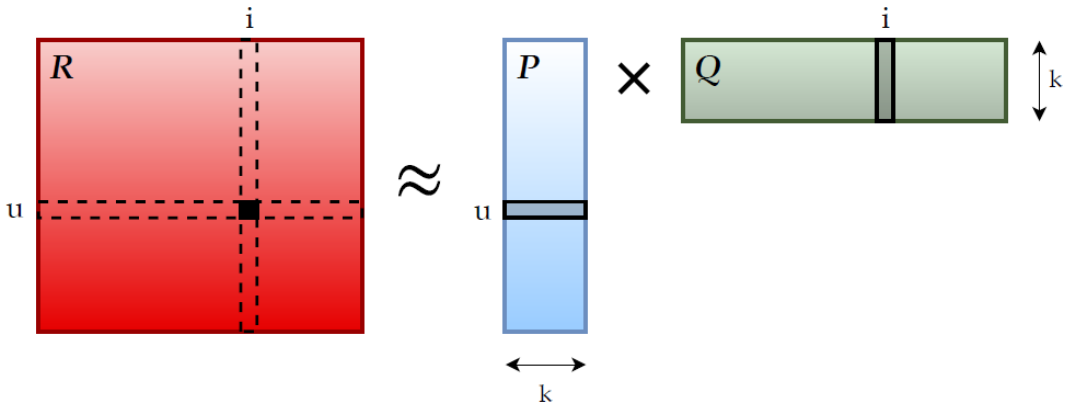
L'idea è quella di calcolare la similarità tra tutti gli user e fatta questa operazione è possibile calcolare il rating previsto per ogni item non valutato rispetto ad uno user. Per fare ciò andiamo a selezionare i k user con similarità più alta con il nostro user target e calcoliamo la media pesata 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 ItemKnn

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 andando a trovare per ciascuno di essi la lista di K item che ha valutato più simili ad esso, poi calcola la media pesata dei rating dei K item selezionati usando come peso la similarità.

3.3.1.3 Matrix Factorization (MF)

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, possiamo quindi creare 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, quello che vogliamo ottenere è una approsimazione della rating matrix attraverso la moltiplicazione di P e Q, ossia $R \approx P \cdot Q^T = \hat{R}$.



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 creando una funzione di loss apposita e andando ad allenare in modo alternato le matrici P e Q, carcando quindi di ridurre dopo ciascuna iterazione la differenza tra R e $\hat{R}$.
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)

Rispetto ai modelli precedenti che potevano funzionare sia con rating espliciti o impliciti, il VAECF 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. **Da completare**.

3.3.2 Content-based filtering

Il content-based filtering 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 (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 $TopN$ in ordine, dove un item non rilevante ha uno score minore rispetto a quello di uno rilevante, 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 correto 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 (normalized Discount Cumulative Gain) ci permette di calcolare una misura basata sulla posizione degli item rilevanti nella lista $TopN$, ossia quelli del test set.

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)} \qquad IDCG@k = \sum_{i=1}^{|REL_i|} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

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

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

Per prima cosa dobbiamo dire che la metrica lavora su una parte della lista $TopN$, ossia la parte contentente 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 denomicatore 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

4.1 Preprocessing matrici grezze

In questa sezione andremo a vedere diverse tecniche che sono state utilizzate per trasformare le matrici grezze 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 appartengonoù a quella categoria;

Vediamo ora i diversi metodi applicati ad un singolo gruppo.

4.1.3.1 Normalizzazione Min-Max

Una delle tecniche che viene proposta nella letteratura è quella della normalizzazione min-max, per applicarla andiamo a considerare un gruppo $G \subseteq V$ e applichiamo a ciascuna tripletta la funziona 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*.

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 trasformali 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 feature, se un'item apparteneva ad una di queste categorie andavamo ad impostare quella specifica cella a 1, altrimento 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 feature.

4.3.3 Nome prodotto

Il nome del prodotto poteva darci informazioni aggiuntive sulla similarità tra di essi, in quanto osservando l'anagrafica materiali i nomi sono assegnati in modo organizzato, quindi prodotti simili in cui cambiano solo alcune parti riportano lo stesso nome con le specifiche diverse.
Per ogni nome prodotto sono state eseguite le seguenti operazioni:

1. eliminazione di simboli e parti del nome non esplicativi, per esempio alcuni nomi avevano riportata anche la dimensione, già disponibili 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 quelle con almeno 10 occorrenze, ottenendo alla fine circa 600 parole. Ciascuna parola è stata assegnata ad una cella del vettore delle feature, se un item nel suo titolo contiene una di queste parole, imposta a 1 la cella corrispondente.

4.3.4 Dimensione, volume, peso

Per le informazioni riguardo lunghezza, larghezza, altezza, volume e peso si volevano creare per ciascuno di essi 20 intervalli ciascuno aventi circa lo stesso numero

di item. Si è assegnato al vettore delle feature una cella per ogni intervallo e se un'item aveva una delle misure all'interno dell'intervallo lo si impostava ad 1.

4.3.5 Profilo user

Una volta creata la matrice avente sulle righe gli item e sulle colonne le feature, 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 andremo ad operare sullo storico vendite andando a considerare le fatture come sessioni di d'acquisto applicando la *User Popularity-based CF*. Ragionando in termini di logica d'acquisto è possibile che i dealer acquistino sempre circa 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 MostPop, quindi si è pensato di muoversi nella direzione della popolarità con questa soluzione per vedere se i risultati migliorassero. L'obiettivo è quello di predire per ciascuno user gli item dell'utima 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 ordinato delle transazioni di uno user di cardinalità B_u definito come $\mathcal{B}_u = \{b_u^t | t \in 1, \ldots, B_u\}$. 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: $\pi_i^u = \frac{\mathcal{B}_u^i}{\mathcal{B}_u}$. Dato che la popolarità può variare nel tempo il paper introduce il concetto di recentezza, dicendo che per predirre l'ultima transazione potrebbe non essere efficace guardare nelle transazioni più

vecchie, quindi attraverso una finestra temporale sulle transazioni più recenti si va a calcolare la *recency aware user-wise popularity*, un modo di calcolare 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 $r \geqslant B_u$ 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, questo viene fatto andando a calcolare la similarità tra due user u e v con la similarità 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 α 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 consideremo solo user molto simili ad uno target.

5 Libreria cornac e esperimenti

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

5.1 Esperimenti sulle matrici grezze

5.1.1 Dataset

Le informazioni dello storico vendite sono state organizzate basandosi sulle due macrocategorie individuate nell'analisi: macchine e ricambi. Gli item considerati sono quelli acquistati almeno una volta mentre per i clienti quelli che hanno effettuato almeno un'acquisto. Abbiamo quindi tre tipi matrici grezze user-item contenenti solo item appartenenti alle macrocategoria delle macchine, solo dei ricambi ed infine una con tutti gli item detta totale. In generale le divisione rispetto i tipi di dataset sono le seguenti:

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

Per ognuno di questi tipi di matrice abbiamo le quattro versioni delle 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 posizione delle fatture di u in cui è presente i.

- **Spesa totale**: somma dei campi spesa totale (*NETWR*) di tutte le posizione 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 la data della fattura più vecchia, il delta temporale viene trasformato in giorni. Quindi più è alto il delta in giorni più recente sarà l'acquisto.

5.1.2 Libreria Cornac

La libreria cornac gestisce completamente gli esperimenti, dall'acquisizione dei dati fino alla verifica dei risultati. Nello specifico 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: implementazione dell'approccio del collaborative filtering memory-based, che di default prende in considerazione i 20 user più simili ad uno target.

- ItemKnn: 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: implementazione del matrix factorization, metodo model-based del collaborative filtering, di default considera 10 user più simili ad uno target;

- VAECF: modello per la versione implicita, usato per avere un risultato di base.

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 dividerli esternamente con le rispettive percentuali: 70% al training set, 15% per validation e test set. 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 fare di training.

5.1.3 Esperimenti sulle singole matrici grezze

Come riportato nel capitolo relativo il preprocessing delle matrici grezze, non è stata definita una 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 una delle distribuzioni disponibili, si è proceduto a testarle entrambe. Dalle combinazioni di: scala, distribuzione dei rating, espressione di interesse e dataset, si sono ottenute circa 1500 matrici dei rating. Per eseguire i test si è proceduti come segue:

1. Ottengo un risultato di base con il modello MostPop;
2. Poi per ogni tecnica testo ciascuna combinazione di matrice dei rating, scala e funzione di distribuzione se usata, si eseguono le seguenti operazioni: Poi per ciascuna tecnica si procede a testare tutte le matrici dei rating annesse ad essa, per ciascuna di esse si eseguono le seguenti operazioni:
   (a) *Alleno* i modelli (MF, $UserKnn$) con valori di deafult 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 si procede ad un tuning dei parametri sul modello con quella matrice dei rating;
   (c) Una volta concluso il tuning, si sceglie il modello migliore in accordo al validation set;
   (d) Si va a confrontare la valutazione finale basata sul test set, con quella di base ottenutta con il $VAECF$.
   (e) Tra tutte quelle che superano quest'ultimo passaggio scegliamo la migliore, una per ogni dataset.

Il confronto dei risultati non è puramente matematico ma va a valutare il trand generale del metodo, se buona parte dei risultati sono migliori del bound si procede con il tuning, mentre se solo alcuni valori sono leggermente meglio li si lascia perdere.

5.1.4 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 descritti nel capitolo precedente. Per gli esperimenti si sono usati lo stesso training, validation e test set usati per gli esperimenti precedenti. La fase preliminare di esperimenti mirava a capire se questo metodo potesse fornire risultati migliori rispetto quelli di base dati da $MostPop$ e $VAECF$.

5.2 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, infatti 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 hanno fatto si di essere poco influenti nella media finale, mentre la gerarchia prodotto e i nomi hanno fatto si di raccomandare i prodotti più popolari. Concludendo non si è voluto sviluppare ulteriormente l'approccio preferendo provare altro.

5.3 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. Si sono andati a selezionare 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 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 avevo acquistato ricambi. L'nsieme 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 transizioni sono state a formare il training set.

Si avevano i seguenti hyper-parametri:

- 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 andato a testare ogni combinazione di hyper-parametri sul validation set, selezionati quelli migliori si è calcolata la valutazione finale del test set.

6 Risultati sperimentali

In questo capitolo andremo a vedere i risultati sperimentali per ogni metodo.

6.1 Risultati preprocessing matrici grezze

Nelle sezione successive andremo a vedere i risultati delle valutazioni di ciascun modello rispetto la metrica AUC e la $NDCG$ con $k \in \{5, 10, 25, 100\}$, in particolare la metrica che riteniamo sia più significativa è la seconda in quanto si basa sul ranking degli item nella lista $TopN$.
Per i confronti abbiamo scelto di usare $NDCG$ con $k = 25$. Nei seguenti grafici

6.1.1 Risultati base

In questa sezione andremo a vedere i risultati del modello MostPop e del VAECF, che verranno utilizzati come bound per valutare quelli succcessivi.

6.1.1.1 MostPop

Come detto il modello MostPop restituisce per ogni user la stessa lista di item più popolari. Andiamo a vedere i risultati per dataset sul validation set:

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

Useremo questi risultati come bound per quelli degli altri modelli (MF, UserKnn), nel caso in cui questi siano migliori si procederà al tuning dei parametri e al calcolo finale dei risultati sul test set.
Quelli riportati qui di seguito sono i risultati del MostPop sul test set.

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

6.1.1.2 VAECF

Il modello VAECF 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 $\{sigmoid, tanh, elu, relu, relu6\}$.

dataset	n_{int}	n_{hid}	act_f
macchine	5	30	relu
ricambi	5	30	sigmoid
totale	5	30	sigmoid

Il tuning dei parametri ha portato a considerare tutte le possibili combinazioni dei parametri sopra citati, andiamo a vedere per ogni dataset quali parametri hanno restituito sul validation set i risultati migliori.

Andiamo a vedere i risultati per dataset sul validation set dopo il tuning dei parametri:

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

Dato che i risultati della tabella precedente sono quelli del validation set su un modello con parametri ottimizzati, non possiamo direttamente usarli per il confronto con
Vediamo i risultati del modello validato sul test set:

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

6.1.2 Risultati MF e UserKnn

In questa sezione andremo a vedere per ogni tecnica di preprocessing adottata sulle matrici grezze i risultati ottenuti. La valutazione si è divisa in due fasi:

1. preliminare: confrontiamo i risultati dei modelli (MF, UserKnn) con parametri di default sulle matrici dei rating, con il risultato bound dato dal MostPop;

2. avanzata: dopo aver effettuato il tuning sui parametri dei modelli con le matrici rimanenti, si confrontano con il bound dato dal VAECF.

Cominceremo prima con la tecnica basata su min-max, poi con quella ordered-based per concludere infine con quella basata sui prodotti totali. Spesso si farà riferimenti ai risultati dei modelli, intendendo il risultato dato dei modelli basato sulle diverse matrici dei rating.

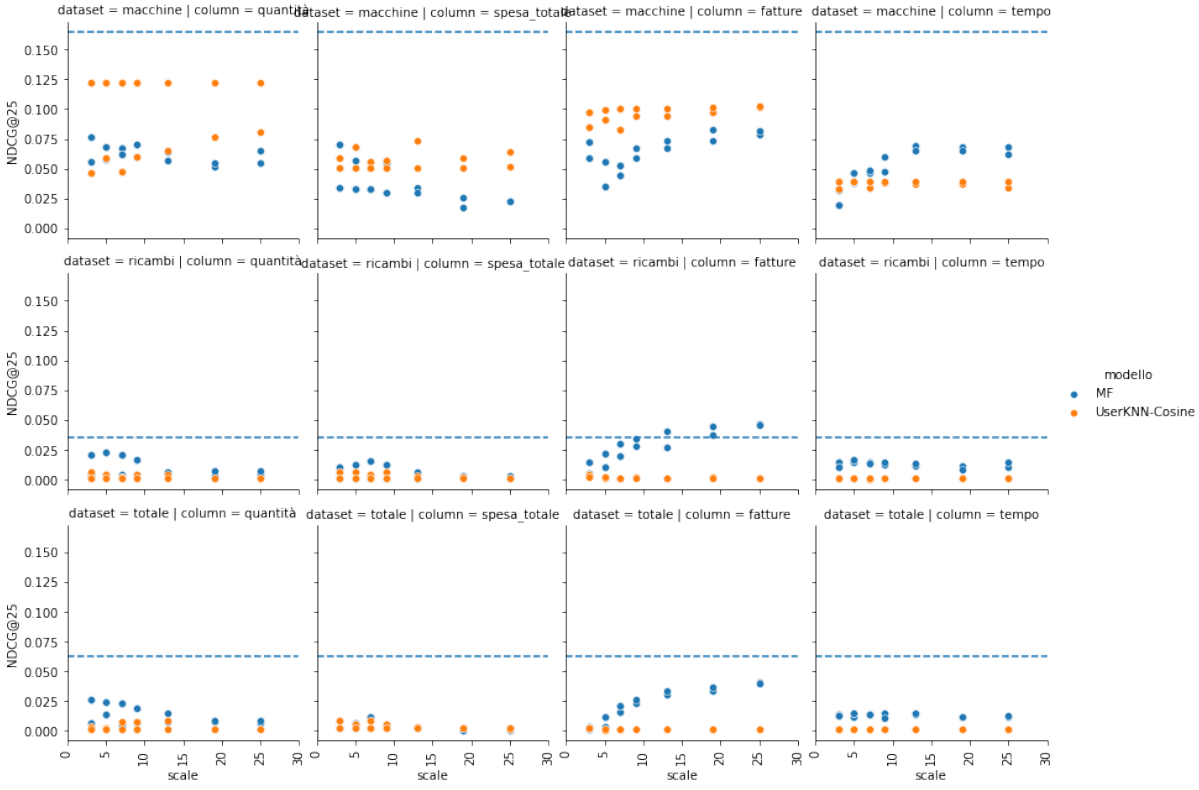
6.1.2.1 Normalizzazione min-max

Come spiegato nel capitolo dedicato alle tecniche ora andremo ad osservare i risultati rispetto diversi gruppi di triplette.

6.1.2.1.1 Gruppo globale

In questa sezione vedremo la normalizzazione min-max applicata al gruppo globale, ossia quello contenente tutte le triplette. Nella tabella sottostante possiamo vedere sulle righe i dataset (macchine, ricambi, totale), mentre sulle righe possiamo vedere 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 $NDCG@25$.

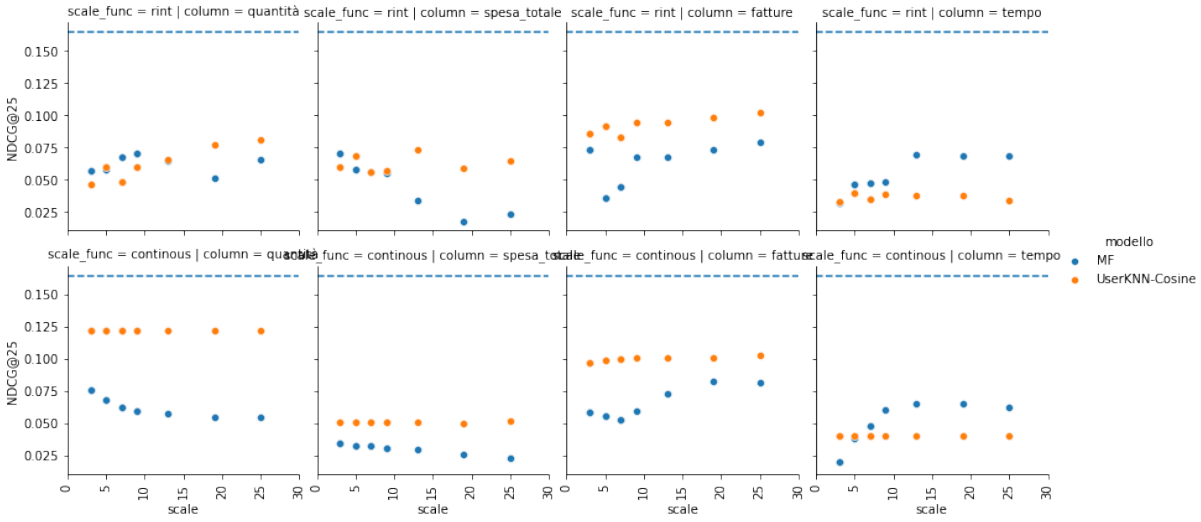
La linea tratteggiata blu rappresenta il risultato del modello MostPop che viene usato come bound. In questo grafico composto per ogni valore della scala vengono riportati quattro puntini, per questa tecnica ogni matrice dei rating è stata tenuta sia in versione continua che in versione intera, a queste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



Diamo un commento ai risultati per ogni tipo di dataset:

6.1.2.1.1.1 Macchine

Per il seguente dataset non presenta alcun risultato che superi il livello di sbarramento, possiamo dire però che le espressioni d'interesse quantità e numero di fatture risultano essere migliori rispetto le restanti. Notiamo anche che la scala sembra avere effetto sui risultati anche se questi possiamo dire che si trovino circa nello stesso intervallo di valori della metrica. Il modello UserKnn sembra funzionare meglio rispetto MF per tutte le espressioni d'interesse tranne che per la recentezza. Per ogni valore della scala si possono vedere due pallini per ciascun colore, questo perché c'è una versione della matrice continua e una intera. Vediamo di seguito come si comportano:



La prima riga utilizza le matrici intere mentre la seconda quelle continue, come possiamo vedere i risultati della versione continua sono più lineari rispetto gli altri. Nonostante ciò possisamo dire che i risultati siano tendenzialmente simili, a parte che per la prima colonna (quantità) dove il modello UserKnn migliora di molto con la versione della matrice continua. In generale questo varà per tutti i grafici che andremo a vedere dove viene applicato min-max.

6.1.2.1.1.2 Ricambi

Per questo tipo di dataset abbiamo un'espressione d'interesse che ha superato la soglia di sbarramento, quella del numero di fatture, potrebbe essere che questa abbia una distribuzione dei valori di partenza più piatta rispetto le controparti

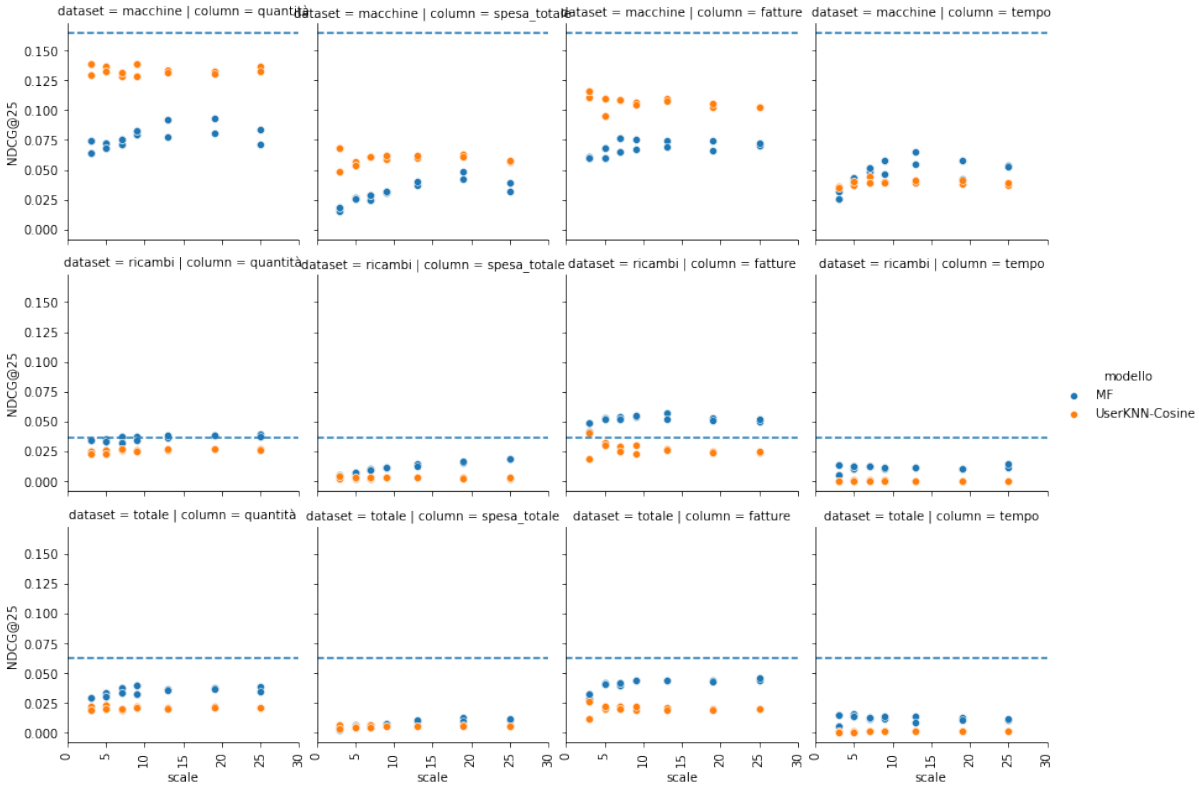
(quantità, spesa totale), in quanto ciascun prodotto ha una quantità e un costo diverso, mentre il numero di fatture è più legato alla frequenza d'acquisto. Per quanto riguarda la scala sembra esserci una tendenza per l'espressione numero di fatture a migliorare all'aumentare del valore massimo di scala. Le affermazioni riportate precedentemente per quanto riguarda matrici intere e continue valgono anche qui.

6.1.2.1.1.3 Totale

In questo tipo di dataset non abbiamo valutazioni che superino la soglia, la tendenza dei dati richiama molto quella del dataset tipo ricambi, non a caso il totale è formato per buona parte dalle triplette dei ricambi.

6.1.2.1.2 Gruppi user-based

In questa sezione vedremo la normalizzazione min-max applicata al gruppo user-based, ossia quello dove ciascun gruppo contiene solo le triplette di un singolo user. Il grafico composto di seguito possiede la stessa disposizione di quello del paragrafo precedente.



Notiamo come ci siano un leggero miglioramento dei risultati rispetto il gruppo globale, vediamo singolarmente ciascun tipo di dataset.

6.1.2.1.2.1 Macchine

Possiamo notare che i risultati siano migliorati rispetto i precedenti, ma non hanno ancora superato la soglia per passare alla fase avanzata, il modello UserKnn fuziona molto bene rispetto MF, come in precedenza le espressioni d'interesse quantità e numero di fatture funzionano meglio delle altre due. Anche qui si nota una tendenza della scala ad avere risultati migliori con valori intorno a 25.

6.1.2.1.2.2 Ricambi

Con questo gruppo possiamo notare come l'espressione numero di fatture rispetto il tipo dataset ricambi sia molto migliorata e ora quasi tutte i risultati con modello MF sono sopra la soglia critica. Inoltre anche la quantità si trova in alcuni risultati intorno al valore soglia. In generale comunque i risultati sembrano in generale migliori del gruppo globale.

6.1.2.1.2.3 Totale

Non abbiamo alcun risultato ancora sopra la soglia, anche se i risultati sembrano un po' meglio dei precedenti.

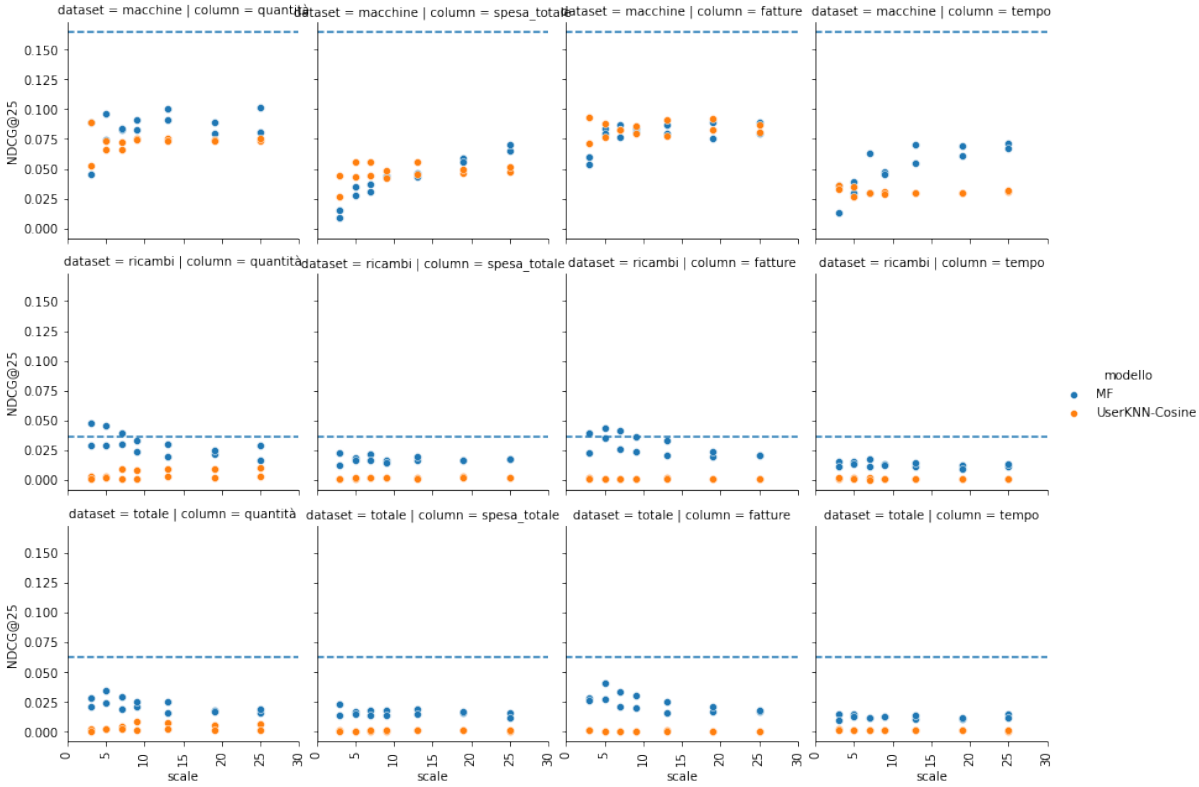
6.1.2.1.3 Gruppo user-category-based Ora andremo a vedere la normalizzazione min-max applicata però al gruppo user-category-based, dove le triplette vengono divise per user e per categoria. Le categorie utilizzate sono quelle della gerarchia prodotto e del gruppo merceologico. Vediamo ora per ogni tipo di dataset i risultati sperimentali.

6.1.2.2 Tecnica ordered-based

6.1.2.2.1 Gruppo globale

In questa sezione vedremo la tacnica di preprocessing ordered-based applicata al gruppo globale, ossia quello contenente tutte le triplette. Nella tabella sottostante possiamo vedere sulle righe i dataset (macchine, ricambi, totale), mentre sulle righe possiamo vedere le *espressioni d'intresse*. 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. Inoltre i puntini di colore blu rappresentano i risultati forniti dal modello MF, mentre quelli arancioni dallo UserKnn.



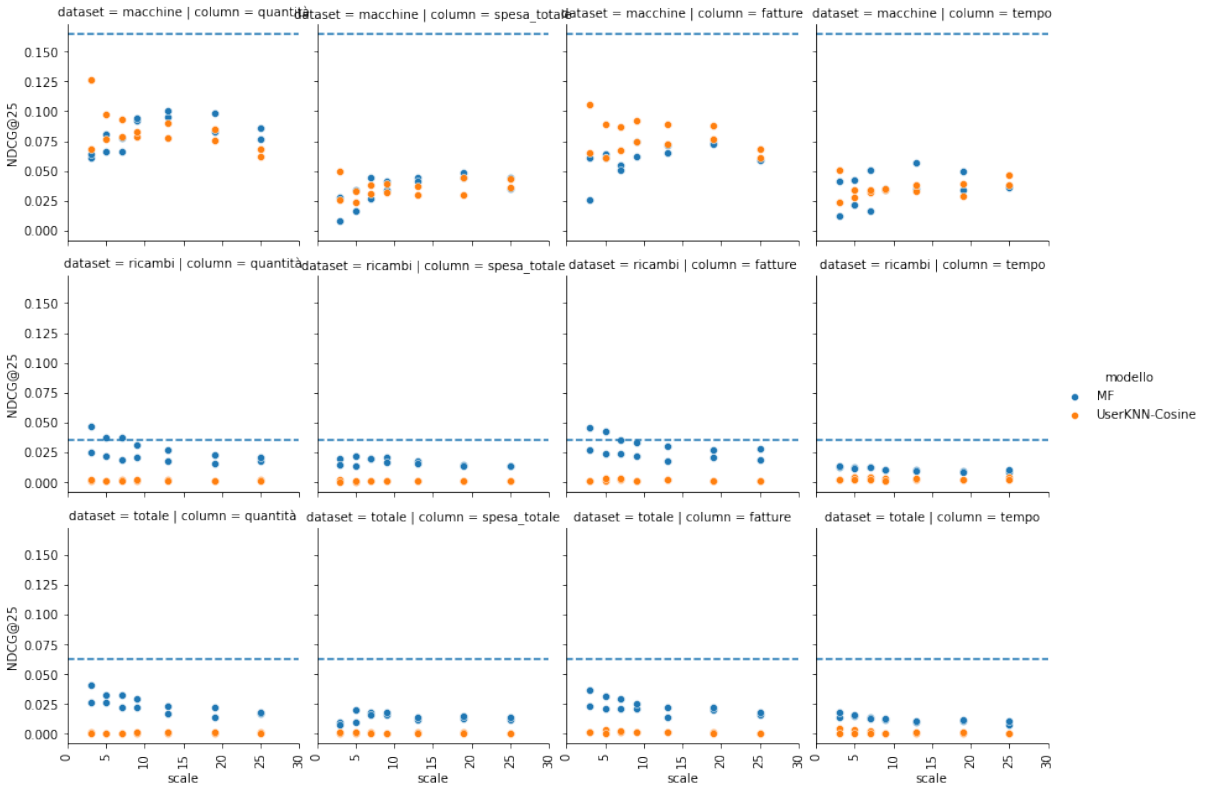
Nella tabella di grafici sopra riportati possiamo vedere sulle righe i dataset (macchine, ricambi, totale), mentre sulle colonne possiamo vedere le espressioni d'interesse (quantità,)

6.1.2.2.2 Gruppi user-based

Di seguito possiamo vedere i risultati relativi la tecnica ordered-based su gruppi di triplette appartenenti allo stesso user.

I grafici sono organizzati come segue: sulle righe i dataset (macchine, ricambi, totale), mentre sulle righe possiamo vedere le *espressioni d'intresse*. 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. Inoltre i puntini di colore blu rappresentano i risultati forniti dal modello MF,

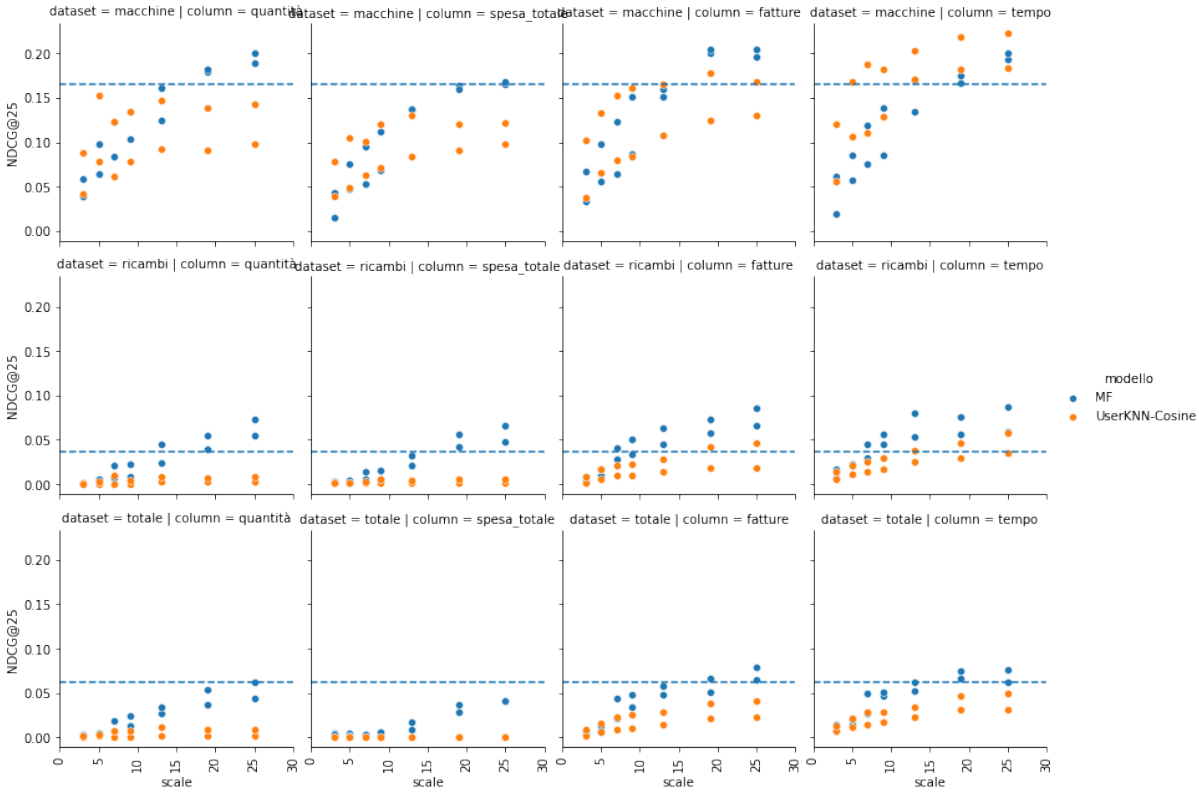
mentre quelli arancioni dallo UserKnn.



Possiamo vedere come il metodo non funzioni se non per alcuni risultati nell'incrocio (ricambi,quantità) che superano di poco il livello bound.

6.1.2.2.3 Gruppo user-category-based

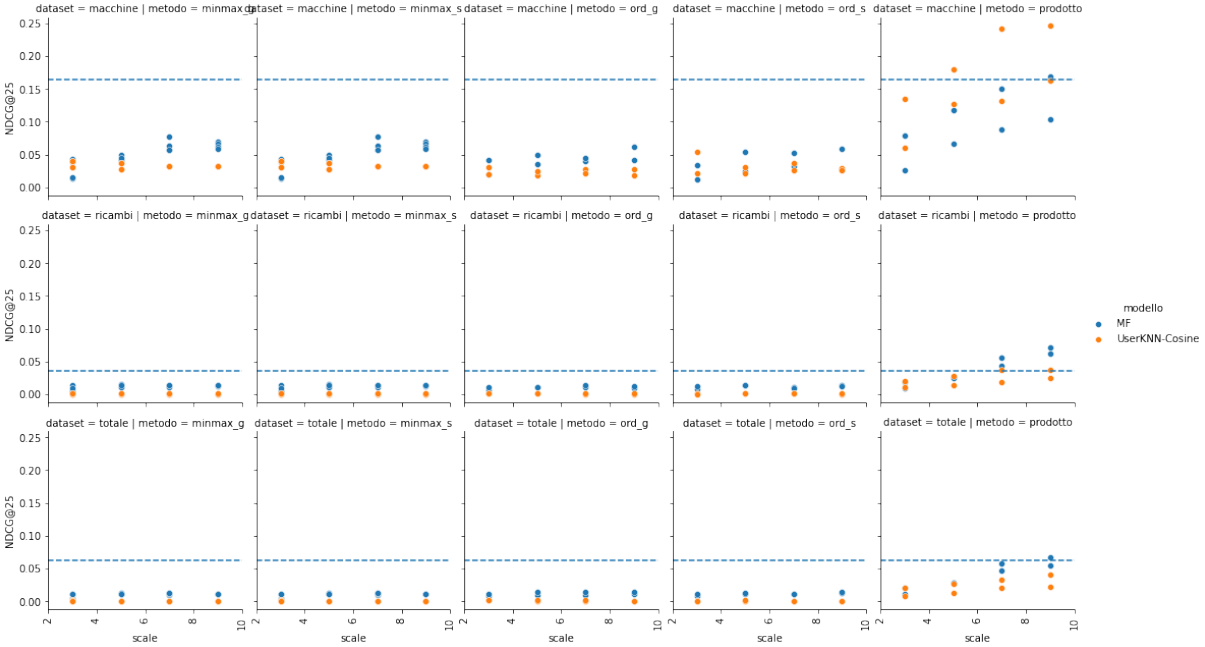
6.1.2.3 Tecnica product-based



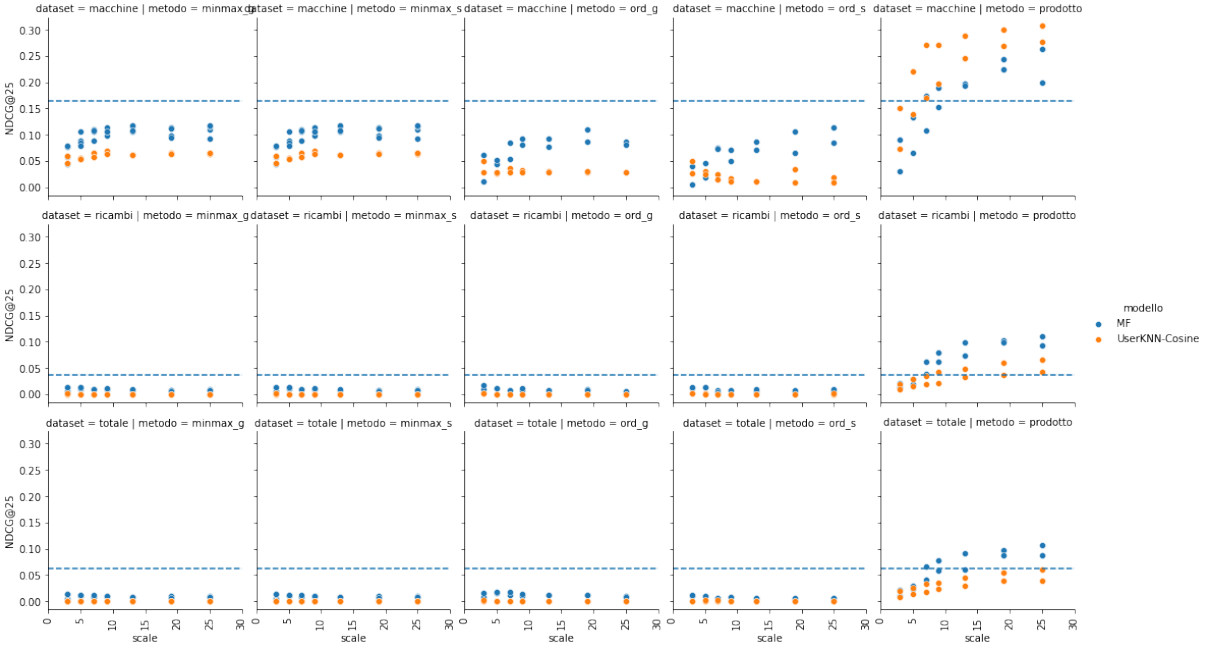
6.2 Risultati matrici grezze combinate

Vediamo di seguito i risultati delle versioni combinate.

6.2.1 Combinazione liste *TopN*



6.2.2 Media matrici dei rating



6.3 Esperimenti con approccio next-basket

In questa sezione vedremo i risultati dell'approccio next-based. Per prima cosa andiamo a vedere quali sono stati i parametri selezionati con il tuning per ciascun dataset.

dataset	α	q	r
macchine	0.5	100	∞
ricambi	0.75	50	∞
totale	0	100	∞

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

Possiamo inoltre vedere che la località q è comunque alta, mentre per quanto riguarda *alpha* abbiamo che le macchine calcolano la probabilità composta al 50%, nei ricambi si da più importanza a quella dello user in esame, ed infine nel totale si considera solo la probabilità composta dello user esterno.
Vediamo i risultati sperimentali con i modelli ottimizzati sul validation set:

dataset	$NDCG$@5	$NDCG$@10	$NDCG$@25	$NDCG$@100
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

E ora i corrispondenti risultati con il test set:

dataset	$NDCG$@5	$NDCG$@10	$NDCG$@25	$NDCG$@100
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

Ricordiamo che questi risultati non sono confrontabili con quelli delle sezioni precendenti, però i risultati sembrano molti interessanti.

Glossario

Lista degli acronimi