Andrea Pedretti

INTRODUZIONE AI SISTEMI DI RACCOMANDAZIONE

DATA SCIENCE PARMA 9/4/2019

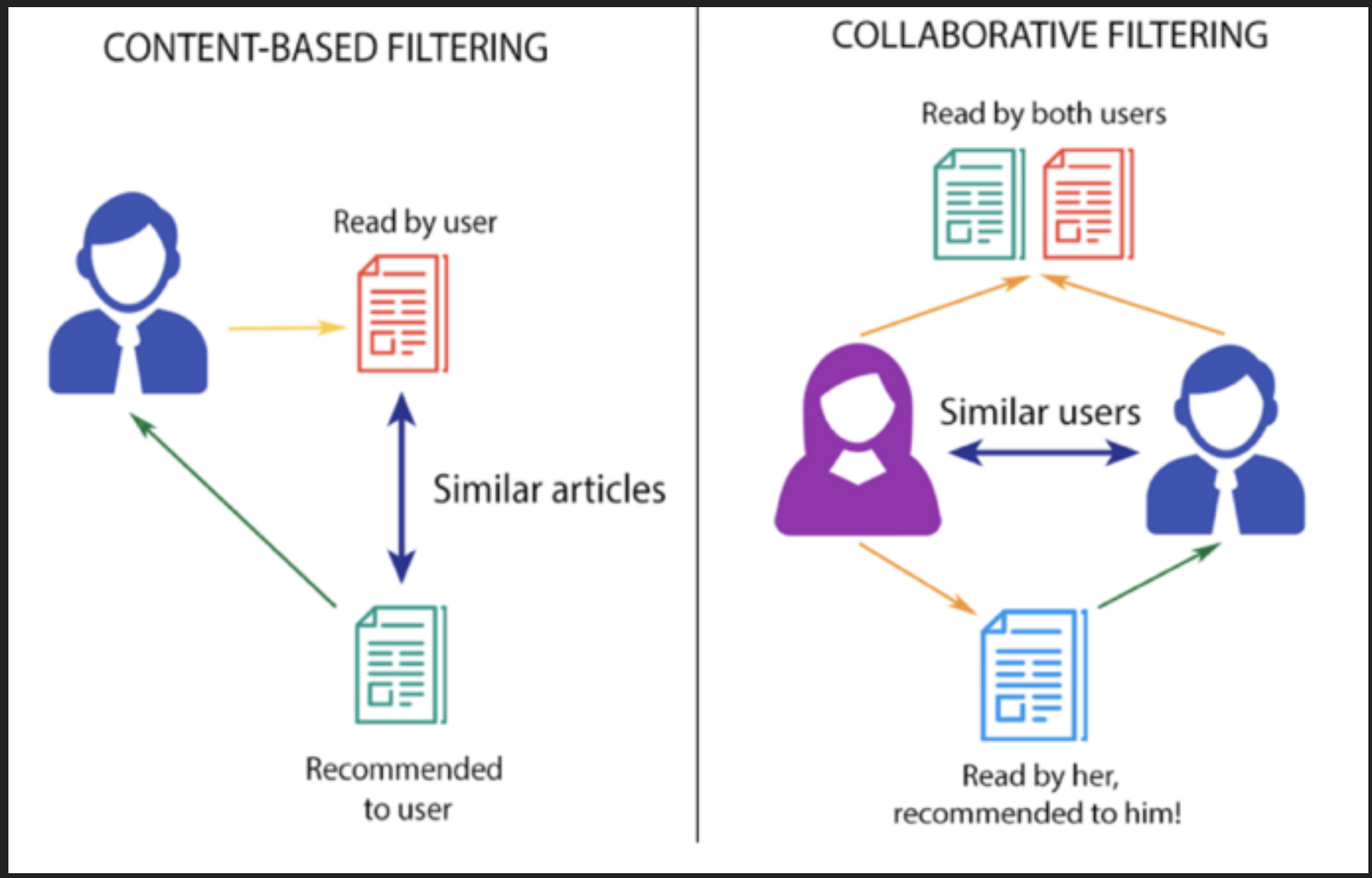
COS'È UN SISTEMA DI RACCOMANDAZIONE

- Un **sistema** di **raccomandazione** è un software di filtraggio dei contenuti (item) con l'obiettivo di fornire delle raccomandazioni personalizzate agli utenti (user)
- Utilizzi:
  - E-commerce
  - Piattaforme contenuti multimediali
  - Siti notizie
  - Piattaforme trading azioni
  - Social networks
  - ...

PERCHÉ UN SISTEMA DI RACCOMANDAZIONE

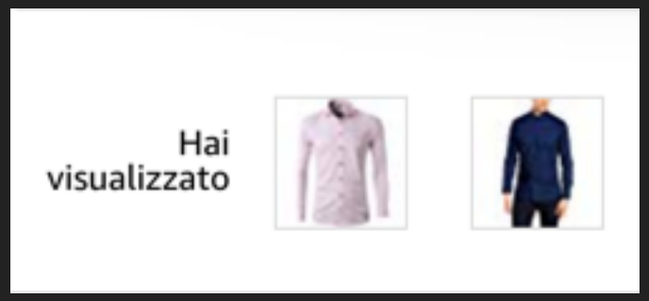
- Aumentare le vendite grazie ad offerte personalizzate in base alle scelte dell'utente
- Aumentare il tempo speso sulla piattaforma
- Gestire problema information overload delle piattaforme online
- Riuscire a sfruttare la coda lunga
- Migliorare la user experience
- Aumentare retention clientela

METODI UTILIZZATI



METODI UTILIZZATI

- Content based filtering
  - raccomandazioni basate sulla similarità degli item (parole-chiave e attributi)
  - raccomandazioni basate sulla profilazione degli utenti collegati ad item con caratteristiche simili

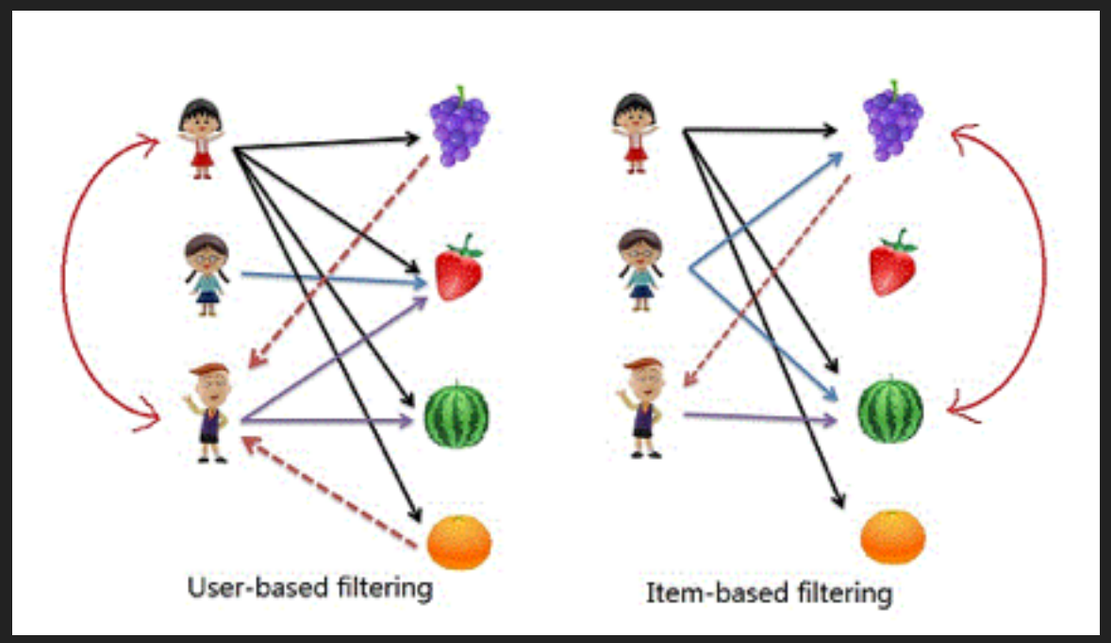


CONTENT BASED FILTERING

- Vantaggi
  - capacità di raccomandare i nuovi oggetti (cold start problem mitigato)
  - efficace con disponibilità di attributi testuali degli item
- Limiti
  - raccomandazioni poco accurate per nuovi utenti (possibile mitigazione durante fase onboarding)
  - raccomandazioni ‘banali’ e sovraspecializzate, sono molto legate alle proprie scelte precedenti, incapaci di offrire contenuti diversi ma potenzialmente interessanti

METODI UTILIZZATI

- Collaborative filtering: analizzano la similarità fra utenti sulla base delle valutazioni
  - user-user
  - item-item (utile con utenti > item)



COLLABORATIVE BASED FILTERING

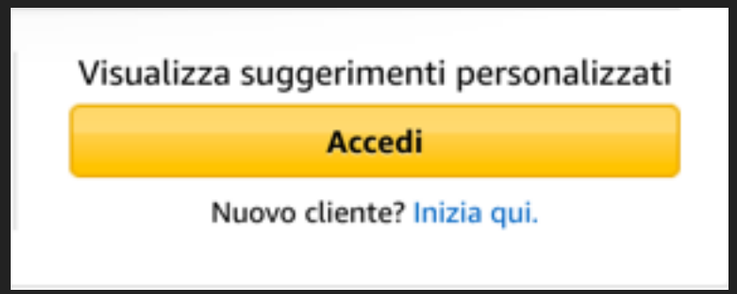
- Vantaggi
  - individuano relazioni tra utenti
  - capacità di effettuare raccomandazioni meno prevedibili
  - applicabili a qualsiasi dominio funzionale
- Limiti
  - cold start problem: è necessario accumulare un grande quantitativo di dati per poter effettuare una previsione
  - banana problem: ci item correlati con quasi tutto, analogamente è complessa la gestione di item molto rari (ad esempio automobile)

COLD START PROBLEM

- Scarsa capacità di previsione (mancano informazioni sulle interazioni) durante:
  - bootstrap sistema raccomandazione
  - nuovo item
  - nuovo utente
- Possibili soluzioni:
  - reperire informazioni sulle preferenze degli utenti durante la fase di on-boarding con domande mirate ed eventuale richiesta di informazioni anagrafiche per profilare l'utente
  - proporre all'utente gli item più popolari (visti, venduti)
  - clusterizzare i dati in base alla similarità dei loro attributi (metodi ibridi)

METODI UTILIZZATI

- Hybrid methods



CONTENT BASED FILTERING – FUNZIONAMENTO (1)

- Obiettivo: calcolare la similarità fra ogni item ed effettuare un rank degli item più simili all'item preso in esame. Gli item verranno raccomandati seguendo quell'ordine.
- Come descrivere gli item? Vettori di TF-IDF! Ogni elemento è rappresentato dalla funzione di peso TF(t)*IDF(t) per una determinato parola (t) del dizionario (bag-of-words) di tutti gli item.
- TF(t) = frequenza del termine t nel documento i-esimo / numero totale dei termini nel documento i-esimo
- IDF(t) = ln(numero totale documenti / numero documenti contenenti il termine t)

	AI	Bigdata	Banana	Result	Weight
item1	2,334	1,98	0,02	3,56	2,05
item2	1,46	2,74	0	4,07	2,15
item3	1,19	2,50	0	3,88	1,50
item4	0	1,02	3,72	1,88	2,33

PRE PROCESSING DEL TESTO

- rimuovere stop word e tag
- rimuovere caratteri accentati e speciali
- lemmatization e stemming
- espandere le contrazioni

CONTENT BASED FILTERING – FUNZIONAMENTO (2)

- Come calcolare la similarità fra item (vettori tf-idf)?
  - cosine similarity

$$cosine(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

  - distanza euclidea

$$d(i,j)=\sqrt{\sum_{k=1}^{n}(x_{ik}-y_{jk})^2}$$

  - coefficiente correlazione di Pearson

$$Sim(u_i, u_k) = \frac{\sum_j (r_{ij} - r_i)(r_{kj} - r_k)}{\sqrt{\sum_j (r_{ij} - r_i)^2 \sum_j (r_{kj} - r_k)^2}}$$

  - distanza Jaccard: applicabile solo quando i vettori (insiemi) contengono valori binari

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

CONTENT BASED FILTERING – FUNZIONAMENTO (3)

- Gli attributi degli item possono essere anche rappresentati in modo binario, in questo caso gli item saranno descritti da vettori di elementi booleani, i quali potrebbero ad esempio rappresentare alti valori tf-idf (1) e bassi valori tf-idf (0)

	adventure	horror	comedy	Spielberg	Di Caprio
item1	0	0	1	0	1
item2	1	0	1	0	0
item3	1	0	0	1	0
item4	0	1	0	0	10

NETFLIX PRIZE (OTTOBRE 2006 – 18 SETTEMBRE 2009)

- Training dataset
  - 100.480.507 rating dati da 480.189 user a 17.770 film
  - Probe dataset 1.408.395 rating
  - t-uple formate da <user, film, data voto, voto>
- Qualifying dataset
  - Quiz dataset: 1.408.342 rating. Dato ai partecipanti e usato per calcolare gli score dei leaderboard
  - Test dataset: 1.408.789 rating. Usato per valutare concorrenti
  - t-uple formate da <film, user, data voto>

IMPLICIT EXPLICIT INFORMATION

- Dati espliciti
  - rating dati dagli utenti, acquisti.
- Dati impliciti
  - click
  - item visti
  - tempo di consultazione
  - la tendenza nel recensire un certo tipo di item può indicare una preferenza per quel tipo di item (ad es recensioni ristoranti giapponesi)

UTILITY MATRIX

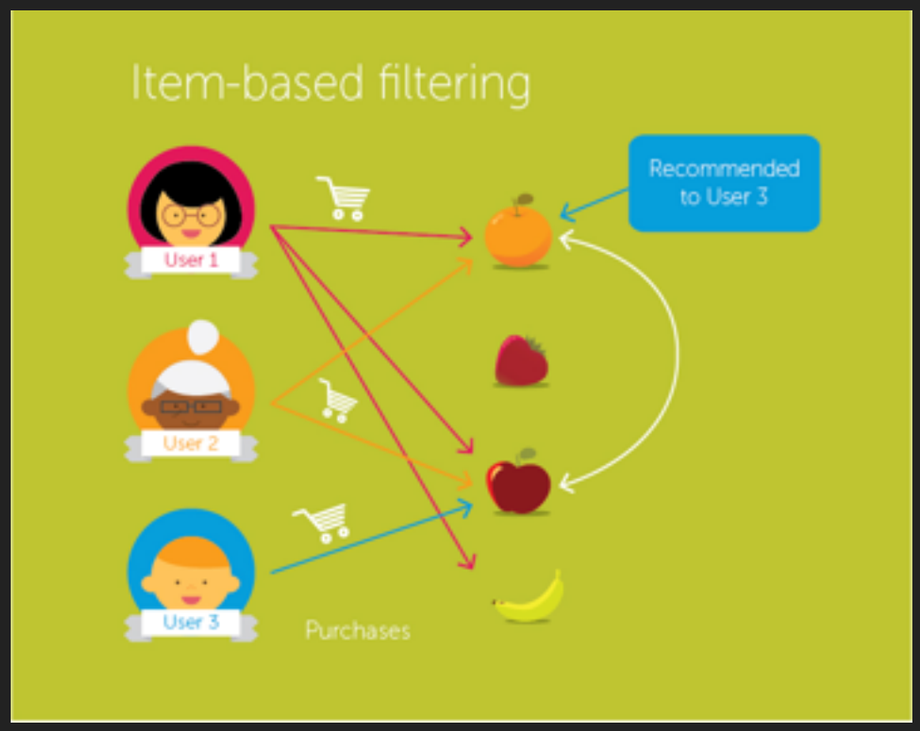
User/ Item	Tu la conosci Claudia?	The Matrix	The Departed
Andrea	3	3	
Luca		2	5
Michele		4	
Sergey		1	

COLLABORATIVE FILTERING – MEMORY BASED (1)

- Approccio Memory Based: si basa sulla storia delle valutazioni dell'utente per effettuare raccomandazioni
  - user to user (più usato): cerca gli utenti più simili basandosi sui rating. La similarità viene calcolata solitamente con cosine similarity o coefficiente di Pearson. Vengono poi raccomandati gli item con il rating medio più alto fra gli utenti più simili
    - grossi limiti di performance nei sistemi sparsi (quasi tutti), non è molto scalabile
    - bias verso oggetti troppo comuni (banana problem): filtri ad hoc, separazione tra oggetti molto e poco comuni, etc
    - Esempio di funzionamento: un utente naviga per la prima volta su You Tube e guarda un video musicale degli Oasis, non verranno per forza mostrati gli item simili al video degli Oasis (per esempio Rock, anni 90, Gallagher, etc). Ma l'utente verrà associato ad un cluster di persone che hanno visto quel video: verranno quindi proposti i video maggiormente visti dalle persone di quel cluster. Quando aumenterà il numero di informazioni, i cluster di appartenenza saranno più specifici e corretti.

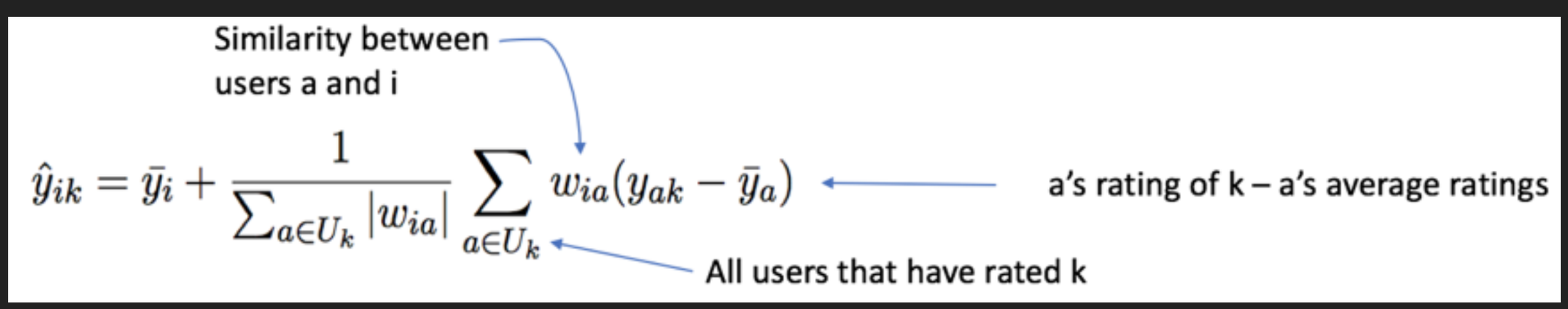
COLLABORATIVE FILTERING – MEMORY BASED (2)

- item to item (inventato da Amazon): analizza le associazioni/correlazioni tra gli item, basandosi sui ratings degli utenti (interazioni passate)
  - due item vengono considerati simili se hanno ricevuto ratings simili dallo stesso user
  - Esempio: per capire se un item deve essere raccomandato ad un utente, viene calcolata la media pesata dei rating degli altri item, quelli più simili verranno consigliati all'utente
- E' più robusto dell'approccio basato sugli user poiché le valutazioni sugli item sono più stabili



COLLABORATIVE FILTERING – MEMORY BASED (3)

- Algoritmo Neirest Neighbour
- la similarità fra user può essere calcolata con cosine similarity



- Necessario correggere sottraendo la media dei rating di ogni utente, per evitare il bias dovuto ai differenti metri di giudizio di ogni utente. Analogamente alla media dei rating, ci possono essere altri fattori che vanno corretti

COLLABORATIVE FILTERING – MODEL BASED

- Approccio Model Based:
  - i dati storici vengono usati per far apprendere un algoritmo di machine learning (reti bayesiane, reti neurali, matrix factorization, PCA, SVD, ..)
  - Matrix factorization (Simon) Funk SVD: il vantaggio rispetto all'algoritmo standard Nearest Neighbor è che anche se due utenti non hanno valutato gli stessi film, è ancora possibile trovare una similarità tra di loro se condividono gusti simili: i fattori latenti

MATRIX FACTORIZATION

	Fiat Panda	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea	4	4	4	4	4
Luca	4	4	4	4	4
Michele	4	4	4	4	4
Sergey	4	4	4	4	4

	Fiat Panda	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea	4	5	2	3	1
Luca	5	1	1	1	2
Michele	5	1	3	2	3
Sergey	2	3	1	4	2

	Fiat Panda	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea	3	1	1	3	1
Luca	1	2	4	1	3
Michele	3	1	1	3	1
Sergey	4	3	5	4	4

MATRIX FACTORIZATION

	Lancia Y	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea	3	1	1	3	1
Luca					
Michele	3	1	1	3	1
Sergey					

	Fiat Panda	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea	3			3	
Luca	1			1	
Michele	3			3	
Sergey	4			4	

	Fiat Panda	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea					
Luca	1	2	4	1	3
Michele	3	1	1	3	1
Sergey	4	3	5	4	4

	Fiat Panda	BMW X6	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea		1	1		1
Luca		2	4		3
Michele		1	1		1
Sergey		3	5		4

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Andrea	3	1	1	3	1
Luca	1	2	4	1	?
Michele	3	1	1	3	1
Sergey	4	3	5	4	4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Sportiva	3	1	1	3	1
SUV	1	2	4	1	3

	Sportiva	SUV
Andrea	1	0
Luca	0	1
Michele	1	0
Sergey	1	1

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

Creazione Matrici rettangolari NxK e KxN il cui prodotto è la matrice NxM

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
Sportiva	1,2	3,1	0,3	2,5	0,2
SUV	2,4	1,5	4,4	0,4	1,1

	Sportiva	SUV
Andrea	0,2	0,5
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,44	1,37	2,26	0,7	0,59
1,32	1,53	1,85	0,91	0,5
2,76	3,37	3,73	2,07	1,02
1,68	1,99	2,32	1,2	0,63

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,2	3,1	0,3	2,5	0,2
H2	2,4	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,2	0,5
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,44	1,37	2,26	0,7	0,59
1,32	1,53	1,85	0,91	0,5
2,76	3,37	3,73	2,07	1,02
1,68	1,99	2,32	1,2	0,63

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

Come scegliere le feature H1 e H2?
Aumentare il numero di fattori latenti permettere di migliorare la personalizzazione delle raccomandazioni, tuttavia un numero troppo alto porterà a problemi di overfitting nel modello

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,2	3,1	0,3	2,5	0,2
H2	2,4	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,2	0,5
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,44	1,37	2,26	0,7	0,59
1,32	1,53	1,85	0,91	0,5
2,76	3,37	3,73	2,07	1,02
1,68	1,99	2,32	1,2	0,63

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,2	3,1	0,3	2,5	0,2
H2	2,4	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,2	0,5
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,44	1,37	2,26	0,7	0,59
1,32	1,53	1,85	0,91	0,5
2,76	3,37	3,73	2,07	1,02
1,68	1,99	2,32	1,2	0,63

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

A questo punto si può comparare la matrice calcolata con la matrice target

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,2	3,1	0,3	2,5	0,2
H2	2,4	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,2	0,5
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,44				

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,4	3,1	0,3	2,5	0,2
H2	2,5	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,3	0,6
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,4	3,1	0,3	2,5	0,2
H2	2,5	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,3	0,6
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,92				

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,4	3,1	0,3	2,5	0,2
H2	2,5	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,3	0,6
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,92	1,83			

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,4	3,1	0,3	2,5	0,2
H2	2,5	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,3	0,6
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

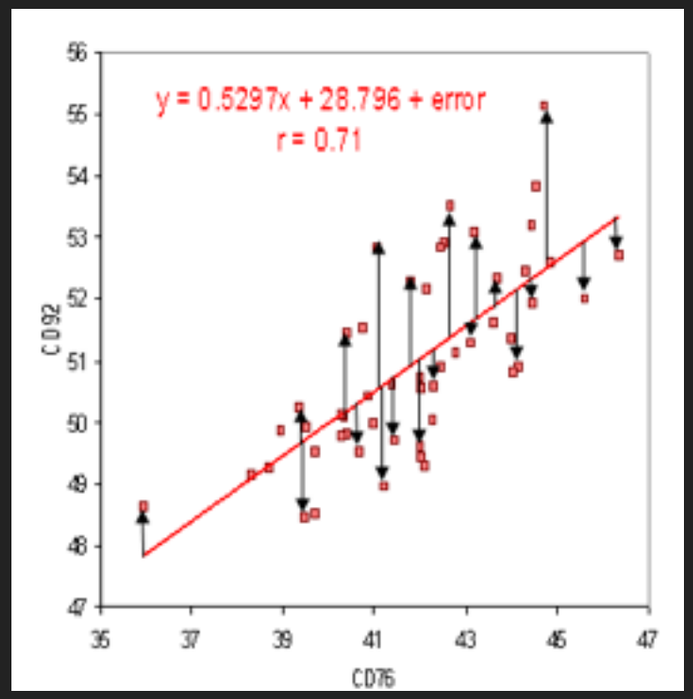
1,92	1,83			

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

MATRIX FACTORIZATION – ERROR FUNCTION

- Quando fermarsi? Di quanto aumentare/diminuire gli iperparametri? Quanto la matrice che abbiamo calcolato è lontana dalla matrice target?
- Root Mean Square Error (RMSE): è la deviazione standard dei residui.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$



MATRIX FACTORIZATION – ERROR FUNCTION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,4	3,1	0,3	2,5	0,2
H2	2,5	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,3	0,6
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,92	1,83			

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

Errore = $(1,92 - 3)^2 + (1,83 - 1)^2 + \ldots$

MATRIX FACTORIZATION – GRADIENT DESCENT

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	1,4	3,1	0,3	2,5	0,2
H2	2,5	1,5	4,4	0,4	1,1

	H1	H2
Andrea	0,3	0,6
Luca	0,3	0,4
Michele	0,7	0,8
Sergey	0,4	0,5

1,92	1,83			

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

L’obiettivo è minimizzare la funzione di errore, si prende derivata dell’errore per capire

quanto aumentare o diminuire i valori degli iperparametri

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	3	1	1	3	1
H2	1	2	4	1	3

	H1	H2
Andrea	1	0
Luca	0	1
Michele	1	0
Sergey	1	1

	1	1	3	
1		4	1	3
3	1	1		
	3	5		4

MATRIX FACTORIZATION

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	3	1	1	3	1
H2	1	2	4	1	3

	H1	H2
Andrea	1	0
Luca	0	1
Michele	1	0
Sergey	1	1

	1	1	3	
1		4	1	3
3	1	1		
	3	5		4

	Fiat Panda	BMW X5	Maserati Quattroporte	Citroen C3	Maserati Stelvio
H1	3	1	1	3	1
H2	1	2	4	1	3

	H1	H2
Andrea	1	0
Luca	0	1
Michele	1	0
Sergey	1	1

3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

The End