

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale

in Ingegneria Informatica

**Sistemi di Raccomandazione in ambito museale:  
progettazione e sviluppo di un modello  
Content-Based per le visite**

Relatore: Prof. Nicola Orio

Laureando: Fabio Paganin

Anno Accademico 2018/2019



*A mia madre, la donna più importante della mia vita...*

*A mio padre, che possa essere fiero di me...*



# Ringraziamenti

Ringrazio i miei genitori che mi hanno dato la possibilità di portare a termine gli studi, ringrazio moltissimo anche tutta la mia famiglia, senza la quale tutto questo non sarebbe stato possibile.

Ringrazio il dottor Daniel Zilio e il professor Nicola Orio che mi hanno seguito e aiutato per quasi sette mesi e che mi hanno dato l'opportunità di lavorare in un ambito estremamente attuale e d'avanguardia, quello dei Sistemi di Raccomandazione.

Ringrazio tutti i ragazzi della Residenza Universitaria Murialdo di Padova, amici fraterni, una seconda casa, una seconda famiglia.

Ringrazio tutti gli amici dell'Università, compagni di banco, di studio e di sofferenze.

*Ringrazio, inoltre, tutte le persone che ho amato e che hanno cambiato per sempre la mia vita.  
(Anche e soprattutto a loro è dedicato questo lavoro di tesi)*



# Indice

1	Introduzione 1	
1.1	Contesto generale: i Sistemi di Raccomandazione (Recommendation Systems) . . . . .	1
1.1.1	Obiettivi dei Sistemi di Raccomandazione. . . . .	3
1.1.2	Modelli di raccomandazione. . . . .	4
1.1.3	Esempi di Sistemi di Raccomandazione attuali . . . . .	9
1.1.4	Valutare un Sistema di Raccomandazione: perché è importante? . . . . .	12
1.2	Il problema della raccomandazione all'interno di un museo: Recommendation Systems museali. . . . .	13
2	Strumenti utilizzati 17	
2.1	Cosine distance . . . . .	17
2.2	Metriche per la valutazione . . . . .	19
2.3	Algoritmo di ordinamento Quicksort. . . . .	26
3	Introduzione al lavoro di tesi 29	
3.1	Dataset di riferimento . . . . .	29
3.2	TindArt . . . . .	33
4	Un modello di Sistema di Raccomandazione museale Content-Based: progettazione e sviluppo . . . . .	35
4.1	Studio preliminare: la funzione di similarità . . . . .	37
4.2	Algoritmo di raccomandazione: descrizione e valutazione . . . . .	60
5	Conclusioni e sviluppi futuri. . . . .	73
	Bibliografia. . . . .	75





# Capitolo 1

## Introduzione

Il problema dell'abbondanza di informazioni e contenuti reperibili (soprattutto all'interno delle piattaforme Web) ha portato, negli ultimi anni, ad una richiesta sempre più elevata di sistemi che permettessero un rapido accesso ad essi, abbattendo i tempi di ricerca.

La grande quantità di materiale disponibile, infatti, spesso spaventa un utente, il quale, soprattutto se piuttosto pigro, tende a deprimersi e smettere di cercare. D'altro canto un eccesso di risorse porta spesso a ricerche molto lunghe, al fine di trovare ciò che più ci aggrada. Esempi lampanti li possiamo trovare nella vita di tutti i giorni: grande richiesta e conseguente produzione di serie televisive, aumento spropositato delle fonti online per i più svariati temi ed argomenti, incremento spropositato della disponibilità di contenuti video e musicali, escalation dei social e delle informazioni in essi contenute, proliferare di articoli disponibili all'interno delle piattaforme di e-commerce, ... Prendere in considerazione tutto diventa, di fatto, impossibile.

Al fine di risolvere questo problema sono nati i Sistemi di Raccomandazione, cioè software, spesso integrati nelle applicazioni, che, sfruttando conoscenze pregresse sui gusti dell'utente o feedback espliciti da esso forniti, creano delle raccomandazioni personalizzate tentando di facilitare le sue scelte, aiutandolo così a reperire esclusivamente i contenuti che più sono conformi alle sue esigenze. Un esempio palese lo osserviamo in YouTube, il quale analizza i "Mi Piace", le visualizzazioni e i commenti di ogni singolo utente, al fine di fornirgli, sulla homepage principale, dei video mai visualizzati, scelti tra i milioni disponibili nel sito, ma che potrebbero piacergli.

I Sistemi di Raccomandazione si sono evoluti nel corso del tempo, fino ai giorni nostri, venendo utilizzati in numerosi ambiti, come, ad esempio, l'e-commerce, i Social, la pubblicità, le notizie, i film e le serie tv, la ricerca di contenuti online, la musica e, come suggerisce il titolo della tesi, in ambito museale.

Dopo una rapida panoramica sullo stato dell'arte dei Sistemi di Raccomandazione al giorno d'oggi e sulle applicazioni più diffuse, analizzeremo nel dettaglio il problema della raccomandazione all'interno di un museo, andando successivamente ad analizzare nello specifico il modello di Sistema di Raccomandazione progettato, oggetto di studio della tesi, focalizzandoci sui passi che hanno portato allo sviluppo e alla valutazione di un software che lo implementa.

L'obiettivo finale della tesi prefissato è stato, quindi, quello di progettare e sviluppare un modello di Sistema di Raccomandazione, con lo scopo di profilare un ipotetico utente di un museo (spesso con una vasta gamma di prodotti artistici) analizzandone i gusti, andando a consigliargli, successivamente, delle opere presenti all'interno della collezione che potrebbero piacergli, in modo da costituire un punto riferimento per chiunque volesse effettuare una visita, aiutandolo nella scelta delle opere d'arte e migliorando la sua esperienza.

### 1.1 Contesto generale: i Sistemi di Raccomandazione (Recommendation Systems)

I Sistemi di Raccomandazione sono strumenti software che utilizzano tecniche che sfruttano i feedback dell'utente, al fine di fornirgli i suggerimenti più pertinenti. Essi sono un'alternativa utile agli algoritmi di ricerca poiché aiutano gli utenti a scoprire elementi che altrimenti non avrebbero facilmente trovato.

I Recommendation Systems nascono da un'osservazione piuttosto semplice: le persone spesso si basano su raccomandazioni fornite da altri nel prendere decisioni quotidiane. Ad esempio, è comune fare affidamento su ciò che gli amici consigliano quando si vuole scegliere un libro da leggere o un canzone da ascoltare. Inoltre gli individui tendono a considerare molto le recensioni di film o ristoranti, al fine di effettuare una scelta.

Nel tentativo di imitare questo comportamento, i primi Recommendation Systems hanno applicato algoritmi per sfruttare raccomandazioni prodotte da una comunità di utenti e fornire tali raccomandazioni ad altri utenti in cerca di suggerimenti. Esse avvenivano consigliando articoli che gli utenti simili, o quelli con gusti simili, avevano gradito, il tutto basato sul principio che suggerimenti provenienti da utenti affini possono essere rilevanti e interessanti per l'utente attivo. Un'altra idea molto utilizzata suggerisce che gli interessi degli utenti possono essere modellati sulla base delle proprietà o degli attributi degli elementi che essi stessi hanno valutato o visualizzato in passato. Ad esempio un Recommendation System può considerare la navigazione verso una determinata pagina di un prodotto come un segno implicito di preferenza per gli articoli mostrati su quella pagina.

In modelli avanzati odierni, è inoltre possibile utilizzare dati contestuali come informazioni temporali, dati sulla posizione, conoscenze esterne o informazioni Social al fine di fornire consigli utili agli utenti. Si pensi ad esempio a Facebook, il quale sfrutta la posizione attuale per dare suggerimenti in merito alle attività presenti nella zona.

Il principio di base delle raccomandazioni è che esistono significative dipendenze tra l'utente e l'attività incentrata sull'elemento. Ad esempio, un utente interessato a un documentario storico ha maggiori probabilità di essere interessato a un altro documentario storico o a un programma educativo, piuttosto che a un film d'azione. In molti casi, varie categorie di articoli possono mostrare correlazioni significative, che possono essere sfruttate per formulare raccomandazioni più accurate. Negli ultimi anni, i Sistemi di Raccomandazione hanno dimostrato di essere un mezzo prezioso per affrontare il problema del sovraccarico di informazioni. Essi generano consigli utilizzando vari tipi di dati sugli utenti, sugli articoli disponibili e sulle transazioni precedenti, tutte archiviate in database personalizzati. Esempi classici di raccomandazione possono essere "quali elementi acquistare", "quale musica ascoltare", "quale serie tv seguire", ... L'utente può consultare questi consigli, accettarli o meno e può rilasciare anche un feedback. Al giorno d'oggi gli utenti sono in grado di fornire molto facilmente opinioni con un semplice clic del mouse. Una metodologia tipica per fornire feedback avviene sotto forma di valutazioni (ad esempio attraverso "Mi piace" o votazione da una a cinque stelline), in cui gli utenti votano, di fatto, un articolo, specificando la loro preferenza o meno verso di esso. Esistono tuttavia altre forme di feedback, le quali non sono altrettanto esplicite come, ad esempio, il semplice atto di un utente di acquistare un articolo o andare alla ricerca di un determinato prodotto, che può essere visto come una sorta di approvazione verso quell'articolo.

L'idea di base dei Sistemi di Raccomandazione è, quindi, quella di utilizzare queste varie fonti di dati per dedurre gli interessi dei clienti. Le azioni e feedback dell'utente possono inoltre essere archiviati ed essere riutilizzati per generare nuovi consigli nelle prossime interazioni con il sistema.

Poiché i consigli sono generalmente personalizzati, diversi utenti o gruppi di utenti beneficiano di suggerimenti diversi. Quante volte ci è capitato di accedere ad Amazon e notare, dalla homepage, che i nostri "Mi piace" o le nostre visualizzazioni/transazioni per certi determinati articoli permettono al sito di selezionare per noi contenuti simili, che verranno successivamente presentati a video in primo piano!

A volte vengono generate anche raccomandazioni non personalizzate. Esempi tipici possono essere "i libri più venduti", "gli album più ascoltati", "le opere più importanti all'interno di un museo". Essi prendono il nome di Hub e definiscono articoli molto noti, che devono assolutamente essere suggeriti.

L'entità a cui viene fornita la raccomandazione viene definita come **utente** (o cliente) e il prodotto consigliato viene anche definito **articolo** (o **item**). Ogni item può avere un valore positivo o negativo per l'utente (essere utile o meno) ed è dotato di attributi significativi (features). Pertanto, l'analisi delle raccomandazioni si basa spesso sulla precedente interazione (**transaction**) tra utenti e articoli, poiché, come si è capito, le conoscenze pregresse sono spesso buoni indicatori di scelte future. D'altro canto le raccomandazioni possono essere effettuate anche sulla base di requisiti specificati direttamente dall'utente piuttosto che sulla sua "storia passata".

Nei prossimi paragrafi andremo a delineare gli obiettivi principali di un Recommendation System, descriveremo con accuratezza le varie tecniche di raccomandazione che delineano i vari tipi di modelli e parleremo di alcune applicazioni esistenti.

### 1.1.1 Obiettivi dei Sistemi di Raccomandazione

Nella sezione precedente, abbiamo definito un Recommendation System come un software che fornisce agli utenti suggerimenti in termini di item che potrebbero piacergli. Ora desideriamo affinare questa definizione, prendendo in considerazione una serie di possibili obiettivi che un Sistema di Raccomandazione dovrebbe soddisfare.

L'**aumento delle vendite di prodotti**, al fine di aumentare i profitti, è l'obiettivo principale di un Recommendation System utilizzato da un qualsiasi commerciante. Raccomandando agli utenti articoli accuratamente selezionati in base ai loro gusti, i Sistemi di Raccomandazione portano gli item rilevanti all'attenzione dei clienti.

Sebbene l'obiettivo principale di un Sistema di Raccomandazione sia aumentare le entrate per il commerciante, ciò viene spesso raggiunto con metodi meno ovvi di quanto possa sembrare a prima vista, conseguendo prima obiettivi operativi e tecnici più comuni, come la **rilevanza**, ossia raccomandare gli articoli che sono rilevanti per l'utente, ben sapendo che quest'ultimo ha maggiori probabilità di considerare item che trova interessanti.

Altri obiettivi importanti vengono elencati di seguito:

- **Novità:** i Sistemi di Raccomandazione sono veramente utili quando l'elemento raccomandato è qualcosa che l'utente non ha mai visto. Ad esempio, i video popolari di uno YouTuber preferito raramente sarebbero nuovi per l'utente, il quale si stancherebbe facilmente del fatto che gli venissero continuamente proposti contenuti che ha già visualizzato in passato. Raccomandazione ripetuta di articoli popolari può anche portare a una riduzione della diversità delle vendite.
- **Serendipità:** significa che gli articoli raccomandati sono in qualche modo inaspettati, contrariamente a noiose e ovvie raccomandazioni, e quindi si crea nel cliente un certo tipo di stupore. La serendipità è diversa dalla novità in quanto le raccomandazioni sono davvero sorprendenti per l'utente. Ad esempio, se esce un nuovo album del proprio cantante preferito, allora la raccomandazione di quel CD rappresenta novità ma non è necessariamente serendipità. D'altra parte, quando viene raccomandato un album o una canzone di un cantante mai sentito prima (magari collegato in qualche modo al proprio cantante preferito), e lo stesso utente non avrebbe mai immaginato che quell'artista potesse attirarlo, allora si parla di serendipità. L'aumento della serendipità presenta spesso vantaggi commerciali e strategici a lungo termine, a causa della possibilità di scoprire aree di interesse completamente nuove da parte dell'utente, aumentando la diversità delle vendite. D'altro canto, si corre il rischio di raccomandare elementi irrilevanti.

- **Diversificazione delle raccomandazioni:** quando tutti gli articoli consigliati sono molto simili si aumenta il rischio che l'utente non gradisca nessuno di questi. D'altra parte, quando l'elenco consigliato contiene item di diversi tipi, è più probabile che l'utente possa gradire almeno uno di questi elementi. La diversità ha inoltre il vantaggio di garantire che l'utente non si annoi con la ripetuta raccomandazione di articoli simili.
- **Soddisfazione del cliente:** ad esempio, un utente di un museo che riceve raccomandazioni pertinenti sulle opere da visitare sarà più soddisfatto dell'esperienza e avrà maggiori probabilità di ritornare o di rilasciare recensioni positive assai utili per la pubblicità. Un metodo molto adatto per aumentare la soddisfazione del cliente è quello di fornirgli una spiegazione del motivo per cui gli è consigliato un determinato articolo.

Per comprendere la natura di questi obiettivi verranno discussi in seguito alcuni esempi famosi di Sistemi di Raccomandazione attuali.

## 1.1.2 Modelli di raccomandazione

Descriviamo ora le più importanti tecniche odierne di raccomandazione che delineano i vari tipi di modelli di Recommendation System.

I Sistemi di Raccomandazione lavorano solitamente con due tipi di dati: le interazioni utente-articolo (come votazioni o comportamento di acquisto) e le informazioni sugli attributi degli utenti e degli item (come profili testuali o parole chiave pertinenti, le **keywords**). I modelli di Recommendation System che utilizzano i primi tipi di dati sono indicati come **Collaborative Filtering**, mentre i metodi che utilizzano gli altri sono indicati come **Content-Based**. Esistono inoltre Sistemi di Raccomandazione basati sulla "conoscenza" (**Knowledge-Based**), in cui le raccomandazioni si basano su requisiti esplicitamente specificati dall'utente. Alcuni Recommendation Systems combinano questi diversi aspetti per creare **sistemi ibridi**. Essi possono sfruttare i punti di forza di vari tipi di Sistemi di Raccomandazione per creare tecniche in grado di offrire prestazioni migliori. All'interno di specifici ambiti la raccomandazione svolge un ruolo critico e potrebbe essere importante fornire suggerimenti sfruttando dati temporali, dati basati sulla posizione o dati provenienti da Social. Da quest'idea nascono i **Context-Based Recommendation Systems**. Di seguito verranno descritti più ampiamente i modelli sopra citati.

### Collaborative Filtering Models

Il modello Collaborative Filtering è uno dei più utilizzati. Esso sfrutta le valutazioni fornite dagli utenti agli item e la somiglianza tra essi (item-item, utente-utente) al fine di generare nuovi suggerimenti. Il metodo collaborativo, generalmente, utilizza le "matrici di rating", in cui ogni riga rappresenta un profilo utente, cioè un vettore continuamente aggiornato di votazioni del cliente agli item. Queste valutazioni possono essere booleane ("Mi piace", "Non mi piace") oppure valori che indicano il grado di preferenza.

Le matrici di rating hanno una caratteristica peculiare che, a volte, può diventare un problema: di solito sono molto sparse. Prendiamo in considerazione, come esempio, un'applicazione musicale in cui gli utenti indicano il loro gradimento o meno alle canzoni. Esistendo al mondo miliardi di canzoni, la maggioranza degli utenti avrà ascoltato solo una piccola parte di questo vasto universo, di

conseguenza, la maggior parte delle valutazioni risulterà mancante. I Recommendation Systems di tipo Collaborative Filtering si preoccupano di dare un valore a queste votazioni mancanti. Vedremo successivamente come questo compito risulti arduo in presenza di matrici molto sparse.

La maggior parte di questi modelli si concentra, come già detto, sull'utilizzo di correlazioni tra elementi o correlazioni tra utenti ai fini del processo di previsione. Esistono due tipi di metodi comunemente utilizzati:

- **Metodi Memory-Based**, più comunemente detti **Neighborhood-Based**. Di questa categoria fanno parte gli algoritmi che si basano sulla storia delle valutazioni inserite dagli utenti per predire votazioni mancanti. Si dividono in due sottocampi:
  - **User-Based Collaborative Filtering**. Si determinano gli utenti che sono simili tra loro secondo determinati criteri stabiliti dall'applicazione (creando insiemi di clienti). Successivamente, per ogni valutazione non data da un cliente ad un item, si calcola la media ponderata delle votazioni del gruppo di utenti simili a lui su quell'item, il tutto al fine di predire un possibile voto che l'utente stesso darebbe all'articolo. Pertanto, se due clienti hanno valutato canzoni in modo simile in passato, è possibile utilizzare le valutazioni di uno dei due su determinati pezzi al fine di prevedere le votazioni mancanti dell'altro su questi stessi.
  - **Item-Based Collaborative Filtering**. Al fine di prevedere la possibile votazione per un articolo da parte di un utente, per prima cosa, si determina un set di articoli simili all'item in questione. I voti dati dall'utente a questi ultimi vengono utilizzati per prevedere se esso apprezzerà o meno l'item stesso.
- **Metodi Model-Based**. In questa categoria rientrano gli algoritmi che sfruttano la storia del sistema, al fine di apprendere un modello che viene utilizzato per generare raccomandazioni.

I Sistemi di Raccomandazione che si basano su queste tecniche possono riscontrare dei problemi, in quanto potrebbe succedere, ad esempio, che tutti gli utenti "simili" al cliente non abbiano fornito alcuna valutazione per un determinato item. In questo caso la stima della possibile valutazione dell'articolo per quell'utente specifico diventa ardua. Ciò accade spesso in presenza di matrici di rating sparse, come già sottolineato.

## Content-Based Recommender Systems

Nei Sistemi di Raccomandazione Content-Based vengono usati gli attributi (**feature**) degli item, cioè un insieme di tutte le caratteristiche capaci di riassumerne il contenuto (da qui, appunto "Content-Based"), per formulare raccomandazioni.

Combinare con gli attributi degli item, in questi modelli, vengono utilizzate anche le valutazioni degli utenti. Ad esempio, si consideri una situazione in cui un utente ha valutato positivamente un'opera d'arte, ma non abbiamo accesso alle valutazioni di altri utenti. Escludendo l'utilizzo di un metodo Collaborative Filtering proprio per questo motivo, si può decidere di sfruttare la descrizione dell'opera in questione, espressa tramite parole chiave (**keyword**), le quali, se simili a quelle di altre opere artistiche, consentono la raccomandazione di queste ultime al cliente. Per fornire un esempio più concreto, ad un utente a cui piacciono le rappresentazioni di nudo potrebbe essere suggerita, come opera artistica, "La nascita di Venere" del Botticelli, oppure la "Venere di Milo", in quanto entrambi sono nudi artistici.

Per ciascun utente vengono visualizzati gli articoli che ha già acquistato o valutato, al fine di prevedere se gradirà o meno un item per il quale la sua valutazione o il suo comportamento di acquisto sono sconosciuti.

I Sistemi di Raccomandazione Content-Based presentano, anche per questo motivo, degli svantaggi che devono essere, in qualche modo, arginati:

- Questi modelli, in molti casi, forniscono **ovvie raccomandazioni**. Ad esempio, se un utente non ha mai valutato un articolo con un determinato set di parole chiave, tali item hanno poche probabilità di essere raccomandati. Questo fenomeno tende, quindi, a ridurre la diversità degli articoli consigliati.
- Sono suscettibili al problema della “partenza a freddo” (**Cold-Start problem**). Sebbene i metodi basati sul contenuto siano efficaci nel fornire raccomandazioni per nuovi articoli, come si può immaginare, non sono efficaci a fornirle per i nuovi utenti. Questo perché il modello, per l’utente, deve poter utilizzare le sue votazioni passate. Tali contenuti o conoscenze, infatti, potrebbero non essere sempre disponibili. Nel caso non si disponesse di queste, come ad esempio all’arrivo di un nuovo cliente, diventa più difficile applicare questo metodo. Pertanto sono stati progettati numerosi metodi specifici per risolvere il problema dell’avvio a freddo (la soluzione al problema Cold-Start utilizzata per gli scopi della tesi verrà esposta in seguito).

Sebbene quanto detto fornisca una visione convenzionale delle tecniche Content-Based, a volte vengono utilizzate versioni più elaborate di questo metodo. Ad esempio, gli utenti possono specificare parole chiave pertinenti nei propri profili e queste possono essere abbinare alle descrizioni degli articoli per formulare raccomandazioni. Tale approccio di esempio potrebbe essere utile negli scenari di avvio a freddo. Tuttavia, tali metodi sono spesso visti come una classe distinta di Sistemi di Raccomandazione, noti come sistemi basati sulla conoscenza (Knowledge-Based).

Come suggerisce il titolo della tesi, la tecnica utilizzata per lo sviluppo del modello di Sistema di Raccomandazione museale in questione è proprio quella Content-Based.

## Knowledge-Based Recommender Systems

I Sistemi di Raccomandazione Knowledge-Based lavorano cercando di suggerire item partendo dai dati a disposizione, dalle esigenze e dalle preferenze di ciascun utente, le quali vengono spesso raccolte in forma esplicita o dedotte. Il profilo utente può essere qualunque rappresentazione di conoscenza che deriva da questa inferenza.

I Sistemi di Raccomandazione Knowledge-Based sono particolarmente utili nel contesto di item che non vengono acquistati molto spesso, ad esempio immobili, automobili, richieste turistiche, servizi finanziari o beni di lusso. In tali casi, potrebbero non essere disponibili valutazioni sufficienti per il processo di raccomandazione, poiché gli articoli vengono acquistati raramente e con diversi tipi di opzioni dettagliate. Inoltre, la natura delle preferenze del consumatore può evolversi nel tempo quando si tratta di tali articoli. Ad esempio, il modello di un’auto di lusso, che poche persone possono permettersi, può evolversi in modo significativo nel corso di pochi anni, a seguito del quale le preferenze dell’utente possono mostrare un’evoluzione corrispondente.

Ad un determinato articolo possono essere associati attributi che corrispondono alle sue proprietà e un utente può essere interessato solo ad articoli con caratteristiche specifiche. Come esempio, sempre nel caso delle auto, esse possono avere diverse marche, modelli, colori, caratteristiche del

motore e gli interessi degli utenti possono essere regolati da una combinazione molto specifica di queste opzioni.

Il processo di raccomandazione viene eseguito sulla base di similarità tra i requisiti stabiliti dai clienti e le descrizioni degli articoli.

I Sistemi di Raccomandazione basati sulla conoscenza possono essere classificati in base al tipo di interfaccia (e conoscenza corrispondente) utilizzata per raggiungere gli obiettivi sopra menzionati:

- **Constraint-Based Recommendation System.** Gli utenti specificano requisiti o vincoli (ad esempio, limiti inferiori o superiori) sugli attributi dell'elemento. Ad esempio un utente che desidera acquistare un articolo potrebbe specificare il prezzo minimo e il prezzo massimo che è disposto a pagare nell'interfaccia resa a disposizione dal sistema (vedi **Fig.1**).
- **Case-Based Recommendation System.** L'utente specifica direttamente le caratteristiche specifiche che deve avere l'item in questione. Ad esempio, un utente in cerca di casa specificherà, nella finestra di dialogo fornita dal sistema, il numero di bagni che desidera avere, il prezzo che è disposto a pagare, ... (vedi **Fig.2**).

**EXAMPLE OF HYPOTHETICAL CONSTRAINT-BASED INTERFACE FOR HOME BUYING (constraint-example.com)**

[ ENTRY POINT ]

**I WOULD LIKE TO BUY A HOUSE SATISFYING THE FOLLOWING REQUIREMENTS:**

MIN. BR    MAX. BR    MIN. BATH    MAX. BATH

MIN. PRICE    MAX. PRICE    HOME STYLE    ZIP CODE

SUBMIT SEARCH

Fig.1: Ipotetico esempio di interfaccia utente per un Constraint-Based Recommendation System


Le metriche di similarità sono definite sugli attributi degli articoli, al fine di recuperare elementi simili.

Si noti che in entrambi i casi, il sistema offre l'opportunità all'utente di modificare i requisiti specificati, qualora non fosse soddisfatto dai risultati della raccomandazione suggerita.

Le interfacce grafiche sono particolarmente utili per esprimere feedback in tali sistemi, in cui gli utenti modificano iterativamente uno o più attributi di un item ad ogni iterazione. Attraverso una sequenza iterativa di richieste di modifica, è possibile arrivare ad un articolo desiderato.

A causa del loro uso di attributi sul contenuto, i sistemi basati sulla conoscenza ereditano alcuni degli stessi svantaggi dei sistemi Content-Based. Ad esempio le raccomandazioni nei sistemi basati sulla conoscenza a volte possono essere ovvie.

**EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION  
INTERFACE FOR HOME BUYING (critique-example.com)**

[ ENTRY POINT ] 

---

**I WOULD LIKE TO BUY A HOUSE SIMILAR TO ONE WITH THE FOLLOWING FEATURES:**

---

**I WOULD LIKE TO BUY AN HOUSE JUST LIKE THE ONE AT THE FOLLOWING ADDRESS:**

**Fig.2:** Ipotetico esempio di interfaccia utente per un Case-Based Recommendation System

## Demographic Recommender Systems

Il modello demografico classifica l'utente sulla base dei suoi attributi personali (età, altezza, peso, provenienza, ...) ed elabora il suggerimento in funzione della classe demografica a cui viene assegnato. Molti siti Web adottano soluzioni di personalizzazione semplici ed efficaci basate sui dati demografici. Ad esempio gli utenti vengono inviati a determinati siti Web in base alla lingua o al paese. Un primo Sistema di Raccomandazione, chiamato Grundy, raccomandava libri, sfruttando le caratteristiche dell'utente raccolte con l'uso di una form interattiva.

In molti casi, le informazioni demografiche possono essere combinate con un contesto aggiuntivo per guidare il processo di raccomandazione. Questo approccio è legato alla metodologia dei Sistemi di Raccomandazione Context-Based. Infatti, sebbene il modello demografico, di solito, non fornisca i migliori risultati su base autonoma, aumenta significativamente la potenza di altri Sistemi di Raccomandazione come componente di modelli ibridi.

I Sistemi di Raccomandazione demografici non sono soggetti al problema del "nuovo utente" in quanto forniscono suggerimenti semplicemente valutando le informazioni demografiche inserite.

Il limite principale consiste nella difficoltà di reperire le informazioni rilevanti al fine del rating, sia per la crescente sensibilizzazione sulla privacy che per il conseguente scetticismo degli utenti nel fornire informazioni personali in una form on-line.



## Context-Based Recommender Systems

I Sistemi di Raccomandazione basati sul contesto tengono conto di vari tipi di informazioni contestuali al fine di formulare raccomandazioni. Tali informazioni potrebbero includere tempo, posizione o dati Social dell'utente. Ad esempio, i tipi di abbigliamento consigliati da un sito di vestiti potrebbero dipendere sia dalla stagione che dalla posizione del cliente.

Il contesto dell'utente, quando cerca una raccomandazione, può quindi essere utilizzato per personalizzare meglio l'output del sistema. Ad esempio, in un contesto temporale, le raccomandazioni per le vacanze in inverno dovrebbero essere molto diverse da quelle fornite in estate e una raccomandazione del ristorante per un sabato sera con i propri amici dovrebbe essere diversa da quella suggerita per un pranzo di una giornata di lavoro con i colleghi.

È stato generalmente osservato che l'uso di queste informazioni contestuali può migliorare notevolmente l'efficacia del processo di raccomandazione, in quanto aumentano il grado di personalizzazione del suggerimento.

## Hybrid Recommender Systems

I metodi di raccomandazione descritti precedentemente presentano dei punti di debolezza, per i quali, a volte, è necessario combinarli al fine di minimizzare gli svantaggi derivanti da ciascun modello. La combinazione può avvenire in diversi modi, e il metodo risultante è detto ibrido.

Un sistema Hybrid, che combina due tecniche, tenta di utilizzare i vantaggi di una, al fine di correggere gli svantaggi dell'altra, o limitarli.

I modelli già menzionati, come visto, possono funzionare bene in diversi scenari. Ad esempio, i sistemi Collaborative-Filtering si basano sulle valutazioni della comunità, i metodi Content-Based sfruttano descrizioni testuali, i sistemi Knowledge-Based si focalizzano sulle interazioni con l'utente, stabilendo delle basi di conoscenza, e i sistemi demografici utilizzano gli attributi personali degli utenti per formulare raccomandazioni. Alcuni Sistemi di Raccomandazione, come i Knowledge-Based, sono più efficaci in situazioni di Cold-Start, in cui non è disponibile una quantità significativa di dati. Altri, come i metodi Collaborative-Filtering, sono più efficaci quando sono disponibili molti dati. Per fare un esempio, accoppiare un sistema collaborativo con uno basato sulla conoscenza, in questo caso, permetterebbe al primo di risolvere il problema dell'avvio a freddo.

In conclusione, i vari aspetti dei diversi tipi di sistemi possono essere combinati per ottenere migliori prestazioni, risolvendo le debolezze che tali sistemi avrebbero se fossero stand-alone. La potenza di più tipi di modelli viene così combinata, al fine di crearne uno migliore.

### 1.1.3 Esempi di Sistemi di Raccomandazione attuali

Il paragrafo precedente ci offre una panoramica sui modelli disponibili attualmente per la realizzazione di un Sistema di Raccomandazione efficiente.

I Recommendation Systems vengono utilizzati in centinaia di applicazioni, ognuna con regole particolari, con affinità e, a volte, punti di distacco dalla raccomandazione pura.

Le applicazioni che analizzeremo sfruttano spesso gli approcci ibridi, tentando di ridurre o di eliminare del tutto gli effetti negativi dei metodi puri.

Il primo fattore da considerare durante la progettazione di un Recommendation System è il dominio dell'applicazione, in quanto esso ha un effetto importante sull'approccio algoritmico che dovrebbe essere adottato.

Esistono molti tipi di domini:

- intrattenimento: ci si preoccupa di fornire consigli su film, musica, giochi;
- contenitori: notizie personalizzate, raccomandazioni di pagine Web, filtri e-mail;
- e-commerce: consigli da fornire ai consumatori sui prodotti da acquistare come, ad esempio, libri, magliette, gadget;
- servizi: raccomandazioni di esperti in un determinato campo, raccomandazioni di case in affitto;
- social: suggerimenti di possibili persone che si potrebbe conoscere e raccomandazioni personalizzate relative a contenuti particolari dei social media come tweet e stati di Facebook.

Lo sviluppatore di un Sistema di Raccomandazione per un determinato dominio dovrebbe comprendere gli aspetti specifici di tale dominio, i suoi requisiti, le sfide e le limitazioni dell'applicazione. Solo dopo aver analizzato questi fattori è possibile selezionare l'algoritmo di raccomandazione ottimale e progettare un'efficace interazione uomo-macchina. Ad esempio, nei Sistemi di Raccomandazione relativi ai social network (ad esempio Facebook, LinkedIn, Tweeter), bisogna considerare che gli utenti sono sovraccarichi di informazioni, attività e interazioni. Questi Recommendation Systems devono, quindi, mirare ad assistere l'utente nell'identificazione di contenuti pertinenti (Tweet, commenti, immagini, interazioni, ...).

Il Social Web viene anche sfruttato dai moderni motori di ricerca che si basano su tecniche di raccomandazione per affrontare le sfide del Web searching e implementare funzionalità di ricerca avanzate. In particolare, vari motori tentano di applicare una qualche forma di personalizzazione, generando risultati per una query dell'utente che non sono solo rilevanti ai fini della query, ma sono anche adattati alla cronologia delle ricerche e alle preferenze dell'utente dedotte dalla precedente attività sui Social.

Un'altra tendenza che sta interessando i motori di ricerca è l'aumento delle attività di searching sui dispositivi mobili. Sarebbe opportuno che, in questi casi, un Sistema di Raccomandazione sfruttasse il sistema di geolocalizzazione del dispositivo, al fine di fornire suggerimenti basati sulla posizione dell'utente.

Di seguito vengono presentati i Sistemi di Raccomandazione più famosi.

## **GroupLens**

GroupLens è stato un Sistema di Raccomandazione, costruito come prototipo di ricerca per il suggerimento delle notizie in Usenet (una rete mondiale formata da migliaia di server interconnessi, ognuno dei quali raccoglie gli articoli, le news, i messaggi, i post che le persone aventi accesso alla rete inviano ad un archivio ad accesso pubblico organizzato in gerarchie tematiche). Il sistema raccoglieva le valutazioni degli utenti di Usenet sugli articoli e le utilizzava per prevedere se questi item sarebbero potuti piacere o meno ad altri utenti. Come risulta ovvio, GroupLens utilizzava algoritmi **Collaborative-Filtering**.

Le idee generali sviluppate per questo sistema sono state estese anche al suggerimento di libri e film. I Sistemi di Raccomandazione corrispondenti sono stati indicati rispettivamente come BookLens e MovieLens.

### **Sistema di Raccomandazione di Amazon.com**

Amazon.com, oltre che un sito di e-commerce, è stato anche uno dei pionieri nei Sistemi di Raccomandazione. Durante i primi anni, fu uno dei pochi rivenditori ad avere la lungimiranza di rendersi conto dell'utilità di questa tecnologia.

Originariamente fondato come rivenditore di libri elettronici, la sua attività si è estesa praticamente a tutte le forme di prodotti.

Le raccomandazioni in Amazon.com sono fornite sulla base di valutazioni, comportamenti di acquisto e navigazione implicitamente forniti dai clienti. Esistono anche votazioni esplicite per gli item venduti e sono specificate dall'utente su una scala a 5 punti, con la valutazione più bassa di 1 stella e la valutazione più alta di 5 stelle.

Se si vuole acquistare qualcosa è obbligatorio creare un account Amazon ed effettuare l'accesso al sito tramite un meccanismo di autenticazione. Vengono così forniti consigli personalizzati agli utenti sulla pagina principale del sito, ogni volta che essi accedono al proprio account. In molti casi, vengono fornite anche spiegazioni per le raccomandazioni.

### **Sistema di Raccomandazione di film e serie di Netflix**

Netflix nasce come società di noleggio di DVD di film e di programmi televisivi per corrispondenza, che alla fine è stata estesa allo streaming.

L'attività principale odierna di Netflix è proprio quella di fornire servizi di streaming di film, serie e programmi televisivi su abbonamento. Viene offerta agli utenti la possibilità di valutare i film e gli spettacoli su una scala di 5 punti. Inoltre, Netflix memorizza anche le azioni dell'utente in termini di visualizzazioni. Queste valutazioni e azioni vengono quindi utilizzate per formulare raccomandazioni. Netflix fa inoltre un ottimo lavoro nel fornire spiegazioni per gli item consigliati. Presentare spiegazioni significative è importante per fornire all'utente una comprensione del perché potrebbero trovare interessante un determinato film o una determinata serie. Questo approccio migliora l'esperienza dell'utente.

### **Sistema di Personalizzazione di Google News**

Il Sistema di Personalizzazione di Google News è in grado di raccomandare notizie agli utenti in base alla cronologia dei clic. Questi ultimi sono associati a utenti specifici in base ai meccanismi di identificazione abilitati dagli account Gmail. L'atto di un utente che fa clic su una news può essere visto come una valutazione positiva per essa (feedback implicito). Non esiste, tuttavia, alcun meccanismo per mostrare il mancato gradimento delle news.

Algoritmi di raccomandazione **Collaborative-Filtering** vengono applicati ai dati raccolti, in modo da poter personalizzare la raccomandazione di articoli per classi specifiche di utenti.

## **Suggerimento di possibili amici in Facebook**

I siti di Social Network raccomandano spesso potenziali amici agli utenti, al fine di aumentare il numero di connessioni Social sul sito.

Questo tipo di raccomandazione ha obiettivi leggermente diversi di quella di un item generico. Mentre una raccomandazione di prodotto aumenta direttamente il profitto del commerciante facilitando le vendite, un aumento del numero di connessioni sociali migliora l'esperienza di un utente su un Social Network. Questo, a sua volta, ne incoraggia la crescita: più crescono i Social e più aumentano i loro ricavi derivanti dalla pubblicità. Pertanto, la raccomandazione di potenziali amici o collegamenti consente una migliore crescita e connettività della rete.

Tali forme di raccomandazioni si basano su relazioni strutturali anziché su dati di valutazione, pertanto la natura degli algoritmi sottostanti è completamente diversa.

### **1.1.4 Valutare un Sistema di Raccomandazione: perché è importante?**

Dato un algoritmo di raccomandazione, quanto bene si comporta? Come possiamo valutare la sua efficienza?

Una questione molto importante relativa al lato pratico di un Recommendation System è la necessità di valutarne la qualità.

La valutazione è richiesta nelle diverse fasi della realizzazione del sistema e per vari scopi.

In fase di progettazione è necessaria una valutazione per verificare la selezione delle tecniche di raccomandazione migliori nella realizzazione del modello scelto. Essa viene implementata off-line e le raccomandazioni calcolate dagli algoritmi devono essere confrontate con le valutazioni memorizzate dell'utente. Questo tipo di valutazione off-line viene generalmente condotto su dati di riferimento esistenti, se sono disponibili dati appropriati, o, in caso contrario, su dati raccolti.

Questi esperimenti possono misurare la qualità dell'algoritmo scelto nell'adempiere al suo compito di raccomandazione. Tuttavia, tale valutazione non può fornire alcuna comprensione circa la soddisfazione dell'utente sulla sua esperienza con il sistema. Gli algoritmi, infatti, potrebbero essere molto precisi nel risolvere il problema delle raccomandazioni di base, ovvero prevedere le valutazioni degli utenti, ma, per qualche altro motivo, il sistema potrebbe non essere apprezzato dai clienti, ad esempio perché le sue prestazioni non sono soddisfacenti per loro.

Pertanto, è richiesta anche una valutazione incentrata sull'utente finale. Durante la valutazione online, gli utenti reali interagiscono con il sistema senza essere consapevoli di ciò che accade in background. È quindi possibile analizzare le prestazioni del sistema dal punto di vista dell'utente e, distribuendo questionari di gradimento, capire cosa gli utenti possano riferire sulla propria esperienza con esso. Gli obiettivi dei Sistemi di Raccomandazione, infatti, vanno oltre l'accuratezza degli algoritmi. È proficuo fornire un'esperienza utile, divertente e personalizzata che porti alla fidelizzazione e alla soddisfazione dell'utente.

Ai fini della tesi il sistema è stato valutato solo off-line, su dati resi disponibili dall'applicazione TindArt, come vedremo in seguito. Uno sviluppo futuro potrebbe riguardare la valutazione di tale sistema in un contesto museale reale.

## 1.2 Il problema della raccomandazione all'interno di un museo: Recommendation Systems museali

I Sistemi di Raccomandazione possono essere inglobati in vari contesti, uno di questi è l'ambito museale.

Esistono, al giorno d'oggi, enormi musei in tutto il mondo, come, ad esempio, il Louvre a Parigi, il Prado a Madrid o l'Ermitage a San Pietroburgo, i quali contengono al loro interno svariate collezioni di opere d'arte provenienti da tutto il mondo, realizzate in diversi ambiti, con stili diversi e appartenenti a periodi, culture o correnti differenti. Di solito questi prodotti artistici possono essere distribuiti su più stanze, più piani e persino più edifici. Anche nei piccoli musei, a volte, il numero di mostre è abbastanza grande da rendere impossibile visualizzare tutte le collezioni in una volta sola, tenendo conto del fatto che, mentre la visita progredisce, l'interesse e l'attenzione del visitatore diminuiscono. Capita spesso, inoltre, di trovarsi a dover affrontare una mostra d'arte della quale si conosce davvero poco.

Una volta dentro ad un museo si è costretti, quindi, a prendere delle scelte. Sembra che domande come "dove andare?", "quali mostre vedere?" e "come trovare rapidamente opere interessanti per noi?" non siano facili da rispondere, in quanto i visitatori si trovano ad affrontare un sovraccarico di informazioni indotto dalla ricchezza del contenuto del museo. Essendo la visita al museo stesso un'attività di svago, un'esperienza educativa, in cui visitatori diversi hanno esigenze, obiettivi e interessi differenti e seguono i propri percorsi a proprio ritmo, è importante acquisire conoscenze sugli utenti e comprenderne, appunto, le esigenze, gli interessi e le preferenze, al fine di soddisfarli. La maggior parte delle mostre che essi vedono, infatti, potrebbero non essere di particolare interesse per loro. Ciò potrebbe fargli perdere opere d'arte o altri item che sono più legati alle loro preferenze personali. È dunque sicuramente importante visitare tutti i punti salienti del museo, i cosiddetti "Hub" (opere talmente famose che non possono essere saltate, ad esempio "La Gioconda" di Leonardo da Vinci al Louvre), ma anche espandere le conoscenze sulla collezione in base ai propri interessi specifici.

Queste sfide vengono affrontate dai Sistemi di Raccomandazione, i quali tengono conto, appunto, delle esigenze e delle preferenze degli utenti. I Recommendation Systems possono costituire una scelta vincente per un museo visto che la loro motivazione principale è quella di aiutare i visitatori ad affrontare questo "sovraccarico di informazioni", aiutandoli a trovare la propria strada nella collezione di opere d'arte, fornendo loro le informazioni giuste al momento giusto, aumentando la loro consapevolezza sui temi della storia dell'arte, consigliando loro le opere che potrebbero piacerli e, in sostanza, fornendo loro un'esperienza personalizzata e gradevole nella visita, magari stimolandoli e tentandoli a visitare il museo più spesso o a fornire recensioni positive, utili per altri potenziali visitatori. Una soluzione ottimale dovrebbe, da un lato, offrire agli utenti una selezione personalizzata di articoli in base alle loro preferenze e, dall'altro, fornire materiale approfondito su questi, che estenda le loro conoscenze.

Alcuni musei, come ad esempio il Louvre, offrono già software che consentono all'utente di personalizzare la propria visita, promuovendo così il patrimonio culturale anche nelle giovani generazioni. Ad esempio il Rijksmuseum di Amsterdam [3] fornisce vari servizi online attraverso il proprio sito Web, in cui i visitatori possono visualizzare in anteprima l'arredamento delle sale espositive e le introduzioni alle mostre. Dal 2005, il Rijksmuseum ha collaborato con ricercatori dell'Università Tecnologica di Eindhoven nel progetto CHIP (Cultural Heritage Information Personalization), parte del programma CATCH (Continuous Access to Cultural Heritage in the Netherlands), finanziato dalla Fondazione Olandese della Scienza, che promuove l'accesso ai beni culturali nei Paesi Bassi. È stata sviluppata un'applicazione Web che, sotto forma di un quiz

interattivo, aiuta gli utenti a trovare opere per loro potenzialmente interessanti nella collezione del Rijksmuseum. Questa applicazione presenta ai clienti, inizialmente, le opere d'arte del museo, in modo che essi le possano valutare. Sulla base di queste valutazioni, il sistema cerca altri item e argomenti correlati, sempre fra le opere del Rijksmuseum, che gli utenti potrebbero trovare interessanti. Per fare un esempio, se un utente attribuisce un punteggio elevato ai ritratti e scarso ai paesaggi, il sistema dedurrà che l'utente è più interessato all'argomento "Ritrattistica" che a quello "Paesaggistico". Il Recommendation System consiglia quindi "Ritrattistica" come Topic, insieme ad altri ritratti presenti all'interno della collezione del Rijksmuseum. Il sistema consente inoltre agli utenti di valutare le raccomandazioni fornite, offrendo il loro feedback personale sulla raccomandazione. Come si osserva dalla Fig.3, ogni utente può inoltre visualizzare il proprio profilo con le scelte già effettuate (può anche modificarle), e ottenere spiegazioni sul perché gli è stata consigliata una determinata opera.



Fig.3: Interfaccia di raccomandazione e profilazione utente del progetto CHIP

Va sottolineato come questa applicazione sfrutti le opere del museo stesso. Altre applicazioni preferiscono invece utilizzare opere correlate ad esse e non facenti parte della collezione, al fine di riuscire a profilare l'utente raccomandandogli opere all'interno del museo che potrebbero piacergli, ma senza "spoiler".

Alcuni musei hanno brevettato audioguide con, all'interno, un Sistema di Raccomandazione basato sull'analisi del comportamento del visitatore: traiettoria (misurazione della posizione accurata dell'utente, completa di tempo dedicato a ciascuna opera d'arte) e interazione con il dispositivo ("Mi piace" dati alle opere, ricerca di informazioni complementari) [4].

Altri musei, invece, si sono focalizzati nella creazione di applicazioni per smartphone che, profilando l'utente, non solo gli suggeriscono opere che potrebbero piacergli, ma utilizzano algoritmi tipici della ricerca operativa (ad esempio l'algoritmo che risolve il problema del commesso viaggiatore), al fine di consentirgli anche di visualizzare una mappa con il percorso più breve per poterle vedere tutte [5].

Notevoli sforzi della ricerca si sono incentrati anche sullo sviluppo di applicazioni di raccomandazione che sfruttano realtà virtuale e aumentata, già disponibili come aggiunta alle mostre in diversi musei, offrendo un'esperienza più interessante e coinvolgente per i visitatori [6]. Ai fini della creazione di questi Sistemi di Raccomandazione, dei modelli che vengono utilizzati e degli obiettivi da perseguire nella loro realizzazione e utilizzo, vale quanto già detto precedentemente.

Nel proseguo della relazione verrà analizzato nel dettaglio il modello di Sistema di Raccomandazione museale Content-Based progettato per la tesi, il quale ha lo scopo di profilare un ipotetico utente di un museo, consigliandogli delle opere che potrebbero piacergli. Verrà posta particolare attenzione sugli strumenti utilizzati, sui passi che hanno portato allo sviluppo e alla valutazione di un software che implementa il modello e sui risultati ottenuti.





## Capitolo 2

### Strumenti utilizzati

In questo capitolo vengono presentati tutti gli strumenti che sono stati utilizzati nello sviluppo del software che implementa il modello di Sistema di Raccomandazione museale Content-Based progettato per la tesi, ponendo particolare attenzione sul perché ha senso utilizzarli in questo contesto.

Saranno descritti nel dettaglio la distanza coseno (sezione 2.1), parte integrante della funzione di similarità fra opere d'arte del nostro Sistema di Raccomandazione, tutti gli strumenti utilizzati per la valutazione del software (sezione 2.2), e l'algoritmo di ordinamento Quicksort (sezione 2.3), impiegato per ordinare le opere in base al grado di similarità con un item prefissato, tutti argomenti e nozioni che potrebbero non essere ovvi per chiunque decida di affrontare lo studio trattato nella tesi.

### 2.1 Cosine distance

La cosine distance, detta anche cosine similarity, è un metodo per calcolare la similarità fra due vettori, semplicemente misurando il coseno fra di loro, di solito usata nell'analisi del testo.

Prima di esporre il procedimento per calcolarla e il perché ha senso utilizzarla in questo contesto, vale la pena di fare un passo indietro, riprendendo in mano la tipologia di modello presa in considerazione allo scopo di realizzare il nostro Sistema di Raccomandazione, ossia la metodologia Content-Based.

Come già detto, nei Sistemi di Raccomandazione basati sul contenuto gli item vengono descritti utilizzando degli attributi (features) che ne delineano, appunto, il contenuto. I Recommendation Systems Content-Based sono ampiamente utilizzati in questi scenari in cui è disponibile una quantità significativa di informazioni del genere. Come vedremo anche nel nostro caso, in molte occasioni questi attributi sono parole chiave che vengono estratte dalle descrizioni degli item, in altri casi, invece, sono già disponibili sotto forma di metadati strutturati. Oltre alle keywords possono essere utilizzati attributi relazionali come, ad esempio, autore, genere e corrente.

Le features vengono adoperate per creare profili di articoli (e anche di utenti) ai fini della raccomandazione. Come già detto, infatti, se un'utente ha votato positivamente un articolo con degli attributi, verosimilmente ha dato una valutazione positiva anche a queste features, morale per cui è più facile che gli vengano consigliati articoli che posseggono quelle stesse caratteristiche, o simili.

In questi casi, è necessario combinare gli attributi in un'unica rappresentazione strutturata.

Solitamente si applica il modello vettoriale in cui, dati  $n$  item, si assume che essi vengano rappresentati in uno spazio vettoriale composto da  $t$  dimensioni, dove  $t$  è il numero di termini (attributi relazionali e keywords totali). Di conseguenza l'articolo  $i$ -esimo viene rappresentato da un vettore di uni e zeri:

$$D_i = (d_{i1}, d_{i2}, \dots, d_{ij}, \dots, d_{it})$$

dove  $d_{ij}$  rappresenta la presenza o meno (1 o 0) del termine  $j$ -esimo per l'item  $i$ -esimo ( $D_i$ ).<sup>1</sup>

Una collezione di  $n$  articoli può essere rappresentata come una matrice dove ogni riga rappresenta un item e ogni colonna contiene i pesi che sono stati assegnati ad un termine per un particolare articolo:

	$t_1$	$t_2$	...	$t_j$	...	$t_t$
$D_1$	$d_{11}$	$d_{12}$	...	$d_{1j}$	...	$d_{1t}$
$D_2$	$d_{21}$	$d_{22}$	...	$d_{2j}$	...	$d_{2t}$
...	...	...	...	...	...	...
$D_i$	$d_{i1}$	$d_{i2}$	...	$d_{ij}$	...	$d_{it}$
...	...	...	...	...	...	...
$D_n$	$d_{n1}$	$d_{n2}$	...	$d_{nj}$	...	$d_{nt}$

Pertanto, ricapitolando, i sistemi basati sul contenuto operano in gran parte, ma non esclusivamente, nel dominio del testo (questa situazione si verifica anche nel nostro caso). Sebbene sia possibile utilizzare qualsiasi tipo di rappresentazione, l'approccio più comune è quello di convertire gli item in una rappresentazione spazio vettoriale in cui essi sono visti come vettori nei quali viene specificata la presenza o meno di ogni termine. Si effettua questa scelta perché esprime la rappresentazione più naturale per la descrizione degli articoli [1].

Data la rappresentazione vettoriale, se si volesse determinare quanto due item siano simili fra loro, bisognerebbe calcolare la distanza fra il vettore che rappresenta il primo articolo e il vettore che rappresenta il secondo. Preso un qualsiasi articolo, normalmente, per calcolare la distanza tra esso e qualsiasi altro item presente all'interno della collezione, si utilizza una **misura di similarità**: gli item per i quali corrisponde un valore maggiore di questa misura sono considerati più simili all'articolo in questione.

Diverse misure di similarità sono state proposte e prese in considerazione per il modello vettoriale, ma quella che è stata valutata come preferibile da adottare è quella chiamata, appunto, **cosine similarity** [11].

La misura della distanza coseno tra due item  $D_i$  e  $D_x$  viene mostrata di seguito:

$$\text{Cosine}(D_i, D_x) = \frac{\sum_j (d_{ij} \cdot d_{xj})}{\sqrt{\sum_j d_{ij}^2 \cdot \sum_j d_{xj}^2}}$$

dove  $j$  varia da 1 a  $t$ .

La misura cosine distance, come si può facilmente osservare, calcola il coseno dell'angolo fra i vettori che rappresentano i due item. Questi due vettori devono essere normalizzati, ossia entrambi devono avere lunghezza uguale. Quando questo si verifica, il coseno dell'angolo fra due vettori identici vale 1, mentre il coseno dell'angolo fra due vettori che non condividono nessun termine vale 0, esattamente come sottolinea la definizione stessa di coseno dell'angolo tra due vettori. Nel nostro caso, come vedremo, essendo le opere rappresentate da vettori di uni e zeri, dove 1 rappresenta la presenza di un determinato termine per un determinato item e 0 ne denota l'assenza, due prodotti artistici saranno più simili quanti più termini avranno in comune.

Si considerino, a titolo di esempio, due articoli:

---

<sup>1</sup> Si osserva, a titolo informativo, che  $d_{ij}$  normalmente rappresenta la frequenza del termine  $j$ -esimo nel documento  $i$ -esimo (tecnica tf-idf), tuttavia, come nel nostro caso, se questi pesi sono binari allora la cosine similarity coincide con la distanza di Jaccard. Molti schemi di pesatura sono stati proposti nel corso del tempo e si basano generalmente su variazioni dello schema tf-idf [11].

$$D_1 = (1, 1, 0), D_2 = (1, 0, 0)$$

i quali, come si può vedere, sono indicizzati da tre termini, dove i valori indicano i pesi associati a quei termini, e l'item:

$$D = (1, 0, 1)$$

indicizzato dagli stessi termini. Calcolando la cosine similarity fra i primi due e quest'ultimo risulta che

$$\text{Cosine}(D_1, D) = 0.5 \text{ e } \text{Cosine}(D_2, D) = 0.58.$$

L'item  $D_2$  risulta più simile a  $D$  rispetto a  $D_1$ .

Come sottolinea [11], essendo il modello vettoriale, di fatto, un modello empirico (cioè non vi sono per esso considerazioni di tipo metodologico), non ci sono ragioni teoriche che fanno preferire questa misura ad altre misure di similarità, ma, semplicemente, essa fornisce dei buoni risultati nella valutazione sperimentale. La cosine similarity viene spesso utilizzata nel dominio del testo a causa della sua capacità di adattarsi alle lunghezze variabili dei documenti (nel nostro caso alla variabilità del numero di termini posseduti da un item e, quindi, alla lunghezza del vettore).

Esistono altri tipi di misure di similarità, come, ad esempio, la correlazione di Pearson, la distanza euclidea e la distanza di Manhattan, ma ognuna di esse è più adatta a lavorare in altri ambiti, soprattutto quando i dati che si hanno a disposizione, vengono rappresentati in altri modi [1] [12] [13].

La cosine similarity è una misura molto utile nei Sistemi di Raccomandazione Content-Based per fare previsioni su item in cui la preferenza dell'utente è sconosciuta, come nel nostro caso. Ci si domanda come il nostro utente valuterebbe quegli item. Per ognuno di questi articoli si calcolano i  $k$  item più simili tra quelli già votati dall'utente usando la cosine distance. Viene determinato il valore medio del rating e lo si assegna come la valutazione prevista per l'articolo corrispondente. Questi item vengono poi ordinati in base al valore calcolato in questo passaggio e gli articoli principali (con rate più alto) vengono successivamente consigliati all'utente.

Come vedremo in seguito, agli scopi della tesi è stato progettato un modello di Sistema di Raccomandazione Content-Based che sfrutta tutte queste tecniche di rappresentazione dell'informazione sopracitate, con una funzione di similarità parametrica tra item (opere) ad hoc, la quale utilizza fortemente la distanza coseno.

## 2.2 Metriche per la valutazione

Abbiamo parlato di valutazione e del perché questa sia importante per conoscere l'efficienza e la qualità di un Sistema di Raccomandazione.

Ci accingiamo ora a parlare e a discutere delle metriche più importanti che vengono utilizzate per la valutazione di un Recommendation System, focalizzandoci su quelle impiegate per gli scopi della tesi.

Come già abbiamo visto, la valutazione si occupa di valutare per poi migliorare i sistemi, cioè essere in grado di capire meglio come effettivamente opera un Recommendation System e prendere decisioni utili, ad esempio per costruire nuovi sistemi. La valutazione riguarda l'efficacia di un Recommendation System, ovvero la misurazione di quanto bene un Sistema di Raccomandazione si comporta nel reperire item di interesse per una specifica utenza.

Ogni Sistema di Raccomandazione rappresenta, di fatto, un Sistema di Reperimento dell'Informazione e, come tale, ne condivide le caratteristiche principali. Per questo motivo, prima di parlare delle singole misure di valutazione, vale la pena di definire alcuni concetti importanti. Si definisce con il nome di **task**, l'obiettivo principale della nostra ricerca all'interno di una collezione di item (nel nostro caso potrebbe essere, ad esempio, "cercare le 30 opere più simili"). Per ogni task esistono più **topic**, i quali sono le reali esigenze informative degli utenti interessati dal task (nel caso delle 30 opere più simili, esempi di topic potrebbero essere "cercare le 30 opere più simili ad una determinata opera d'arte della collezione", con i topic che variano a seconda di quale opera viene presa in considerazione). È importante definire i **giudizi di rilevanza**, ossia un insieme dei giudizi che possono essere dati ad un singolo item restituito, in modo che esso sia giudicato rilevante o non rilevante per l'utenza (nel nostro caso i giudizi saranno semplicemente due: "rilevante" e "non rilevante"). Per ogni grado di rilevanza è possibile assegnare un valore numerico o peso (**relevance weight**). Quando abbiamo due gradi di rilevanza, e quindi due pesi, si parla di **rilevanza binaria**:

- Rilevante (R): 1;
- Non rilevante (NR): 0.

Esiste anche la possibilità di avere più gradi di rilevanza (multigraded relevance) e non semplicemente Relevant-Not relevant, e quindi più pesi [14]<sup>2</sup>.

Data una collezione di articoli e un topic, si definisce **run** il singolo esperimento che consente di recuperare gli item di interesse per quel topic (nel caso delle 30 opere più simili ad una determinata opera d'arte della collezione, una run rappresenta tutti quei 30 item, ordinati dal più simile al meno simile, ben inteso il fatto che si cambia run ogni volta che si modifica la funzione di similarità utilizzata). È pratica comune considerare gli item non recuperati durante una run come non rilevanti, tuttavia non è sempre così, infatti, come vedremo anche nel nostro caso, può essere semplicemente che su di essi non venga espresso alcun giudizio.

In diversi ambiti scientifici è molto usato il concetto di **ground truth**, parlando di quella funzione che, dato un task, una collezione di item e un topic per quel task, associa ad ogni coppia topic-articolo un grado di rilevanza, cioè parliamo di una funzione che, dato un topic e un item, dice se quest'ultimo è rilevante per quel topic. Una volta che è disponibile questa ground truth è possibile giudicare ogni singola run (assessment) rispetto al topic considerato (vedi Fig.4-5).

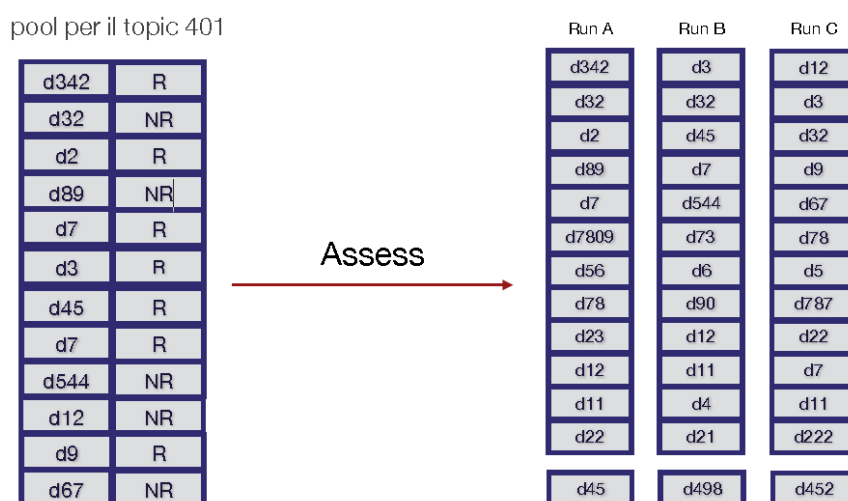


Fig.4: Assessment. A sinistra si vede il pool (ground truth), a destra le run per il topic considerato

<sup>2</sup> In questi casi esistono metriche per la valutazione dette misure a più gradi di rilevanza [14], diverse dalle misure binarie considerate per gli scopi della tesi.

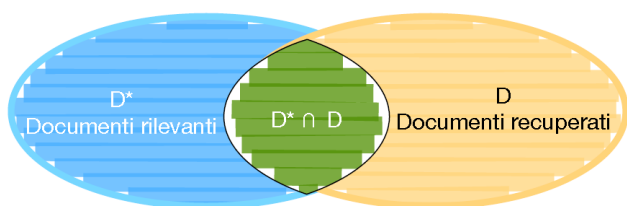
	Run A	Run B	Run C		Run A	Run B	Run C
1	R	R	NR	1	1	1	0
2	NR	NR	R	2	0	0	1
3	R	R	NR	3	1	1	0
4	NR	R	R	4	0	1	1
5	R	NR	NR	5	1	0	0
6	NR	NR	NR	6	0	0	0
7	NR	NR	NR	7	0	0	0
8	NR	NR	NR	8	0	0	0
9	NR	NR	NR	9	0	0	0
10	NR	NR	R	10	0	0	1
11	NR	NR	NR	11	0	0	0
12	NR	NR	NR	12	0	0	0
1000	R	NR	NR	1000	1	0	0

**Fig.5:** Assessment con pesi binari. A sinistra gli item rilevanti e non rilevanti per ogni run, a destra l'assegnazione dei corrispettivi valori numerici

Una volta definiti questi concetti vengono espone, di seguito, le misure con rilevanza binaria impiegate nella tesi, ai fini della valutazione.

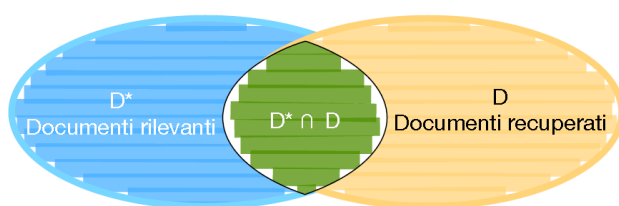
### Precisione e Richiamo

Data una run per un determinato topic, la Precisione (**Precision**) è definita come il numero di item rilevanti recuperati diviso il numero totale di item recuperati dalla stessa run (**Fig.6**), e il Richiamo (**Recall**) è definito come il numero di item rilevanti recuperati diviso il numero totale di item rilevanti esistenti nella collezione di articoli (**Fig.7**).



$$Prec = \frac{|D^* \cap D|}{|D|}$$

**Fig.6:** Precisione

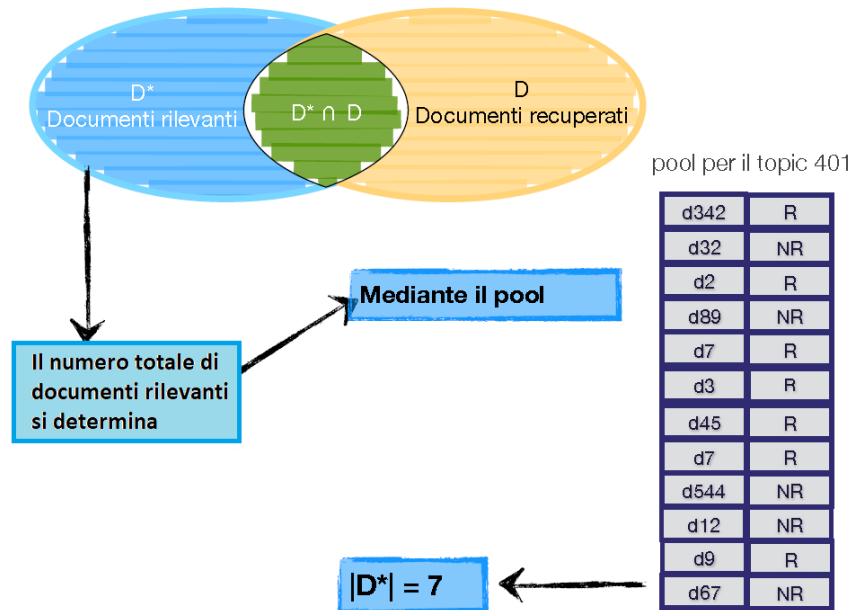


$$Rec = \frac{|D^* \cap D|}{|D^*|}$$

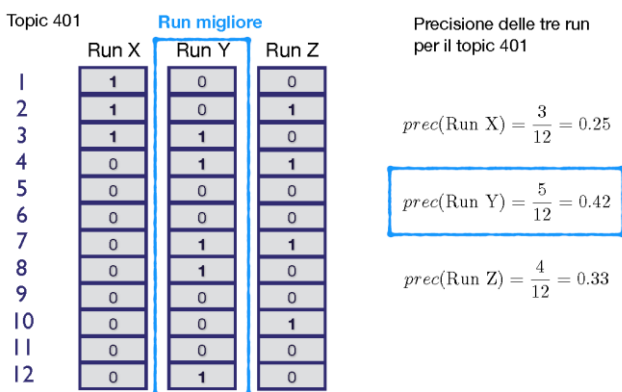
**Fig.7:** Richiamo

Precisione e Richiamo sono misure i cui valori sono sempre compresi tra 0 e 1. Più questi valori sono alti e migliore è la run. Convenzionalmente il valore 1 indica la migliore run possibile o run ideale e 0 la run peggiore. I valori di Recall e Precision possono essere indicati anche in percentuale, ad esempio Precision 0.6 indica Precisione del 60% e Recall 0.25 indica Richiamo del 25%.

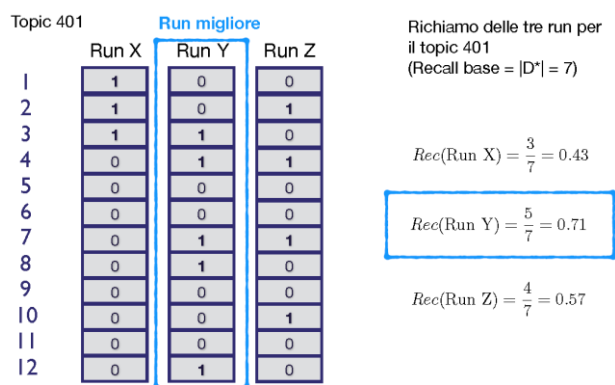
Nelle **Fig.8-9-10** vengono presentati degli esempi di calcolo di Precision e Recall per le run definite nella **Fig.5**, evidenziando, sia in termini di Precisione che di Richiamo, la run migliore.



**Fig.8:** Determinazione del numero totale di item rilevanti all'interno della collezione mediante la ground truth (pool).  $|D^*|$  è anche chiamato la **Recall Base** per un topic



**Fig.9:** Esempio di calcolo della Precisione



**Fig.10:** Esempio di calcolo del Richiamo

## Precisione a diversi livelli di cut-off

Precisione e Recall non sono, tuttavia, misure "ranked", in quanto sono calcolate su insiemi di item e l'ordinamento (il rank) non è tenuto in considerazione.

In diversi task di ricerca, come del resto anche il nostro, l'ordinamento degli item è molto importante. Quindi una run che ha più articoli rilevanti in posizioni alte dovrebbe essere considerata migliore di una run con lo stesso numero di articoli rilevanti, ma in posizioni basse, pensiamo, ad esempio, ad un task di ricerca sul Web.

Per questo motivo si introduce la Precisione a diversi livelli di cut-off, la quale sta ad indicare una normale Precisione, ma calcolata sui primi k item:

$$prec@k = p@k = p[k].$$

In  $p@k$  chiamiamo k "livello di cut-off" o valore di taglio.

Nelle **Fig.11-12** vengono presentati degli esempi di calcolo di  $p@1$  e  $p@10$  per le run definite nella **Fig.5**. Come possiamo osservare, la run migliore cambia in entrambi i casi.

Topic 401

	Run X	Run Y	Run Z
1	1	0	0
2	1	0	1
3	1	1	0
4	0	1	1
5	0	0	0
6	0	0	0
7	0	1	1
8	0	1	0
9	0	0	0
10	0	0	1
11	0	0	0
12	0	1	0

Si considera solo il primo documento

$$\text{prec}@1(X) = 1/1 = 1$$

$$\text{prec}@1(Y) = 0/1 = 0$$

$$\text{prec}@1(Z) = 0/1 = 0$$

Fig.11: Esempio di calcolo della p@1

Topic 401

	Run X	Run Y	Run Z
1	1	0	0
2	1	0	1
3	1	1	0
4	0	1	1
5	0	0	0
6	0	0	0
7	0	1	1
8	0	1	0
9	0	0	0
10	0	0	1
11	0	0	0
12	0	1	0

Si considerano i primi 10 documenti

$$\text{prec}@10(X) = 3/10 = 0.33$$

$$\text{prec}@10(Y) = 4/10 = 0.40$$

$$\text{prec}@10(Z) = 4/10 = 0.40$$

Fig.12: Esempio di calcolo della p@10

Esiste un caso particolare di  $\text{prec}@k$ , ovvero quando  $k$ , il cut-off, è uguale alla Recall Base. In questi casi si parla di Precision At Recall Base, o semplicemente **Precision At Recall** ( $p[\text{RB}]$  o  $p@r$ ) ed è una misura molto usata nell'ambito della valutazione (Fig.13).

Topic 401

	Run X	Run Y	Run Z
1	1	0	0
2	1	0	1
3	1	1	0
4	0	1	1
5	0	0	0
6	0	0	0
7	0	1	1
8	0	1	0
9	0	0	0
10	0	0	1
11	0	0	0
12	0	1	0

Per il topic  $t=401$ ,  $\text{RB}_{401} = 7$ , quindi si considerano i primi 7 documenti

$$\text{prec}@RB_t(X) = 3/7 = 0.43$$

$$\text{prec}@RB_t(Y) = 3/7 = 0.43$$

$$\text{prec}@RB_t(Z) = 3/7 = 0.43$$

Fig.13: Esempio di calcolo della p[RB]

## Curva Richiamo-Precisione (R-P Curve)

Dopo aver definito Recall e Precision, l'obiettivo è quello di introdurre una misura che unisca questi due e ci permetta di avere una visione d'insieme dell'efficacia di un Recommendation System. Vogliamo calcolare la Precision in funzione di  $r$  punti di Recall, in modo da poter disegnare una curva con i punti di Richiamo sull'asse delle ascisse e la Precisione corrispondente sulle ordinate. Ciò che si fa è, in sostanza, data una run, calcolare Richiamo e Precisione ad ogni livello di cut-off e disegnare la R-P Curve corrispondente. Le Fig.14-15 mostrano come disegnare questa curva per le run definite nella Fig.5.

Come si può intuire dalla Fig.16, utilizzando la Curva Richiamo-Precisione, il confronto tra due run sullo stesso topic non è sempre possibile. Inoltre, sempre usando questa misura, il calcolo delle prestazioni di una stessa run su più topic (media tra i valori) è impossibile. La motivazione principale sta nel fatto che i punti di Richiamo, sui quali viene calcolata la Precisione, non sempre coincidono. Per avere dei punti fissi di Richiamo queste R-P Curves devono essere interpolate in quei punti. La Curva Interpolata permette di confrontare più run sullo stesso topic e la stessa run su più topic. Ai fini di calcolare quella che si chiama **Interpolated R-P Curve**, il metodo standard che viene utilizzato è un po' contro intuitivo [14], vediamo perché.

Documenti rilevanti del pool per il topic 401 (Recall base =  $|D^*| = 7$ )

r	Run Y	Rec(r)	Prec(r)
1	NR	0	0
2	NR	0	0
3	R	0.14	0.33
4	R	0.29	0.50
5	NR	0.29	0.40
6	NR	0.29	0.33
7	R	0.43	0.43
8	R	0.57	0.50
9	NR	0.57	0.44
10	NR	0.57	0.40
11	NR	0.57	0.36
12	R	0.71	0.42

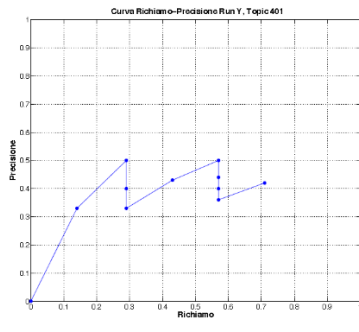


Fig.14: R-P Curve Run Y

Documenti rilevanti del pool per il topic 401 (Recall base =  $|D^*| = 7$ )

r	Run Z	Rec(r)	Prec(r)
1	NR	0	0
2	R	0.14	0.5
3	NR	0.14	0.33
4	R	0.29	0.50
5	NR	0.29	0.40
6	NR	0.29	0.33
7	R	0.43	0.43
8	NR	0.43	0.38
9	NR	0.43	0.33
10	R	0.57	0.40
11	NR	0.57	0.36
12	NR	0.57	0.33

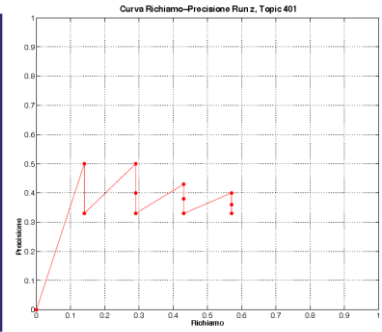


Fig.15: R-P Curve Run Z

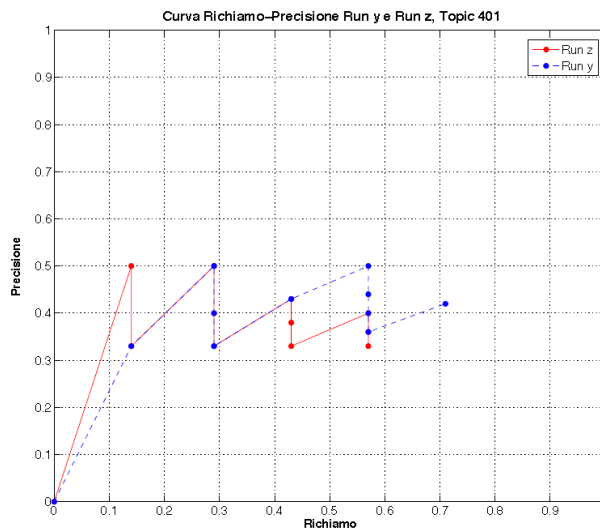


Fig.16: R-P Curve di Run Y e Z a confronto

Il primo passo è quello di definire 11 punti fissi di Richiamo, con l'obiettivo di fissare un dominio per tutte le run:

$$rp \in \{0, 0.1, 0.2, 0.3, 0.4, \dots, 1\}.$$

Successivamente, per ogni run e per ognuno di questi punti, viene calcolata la Precisione interpolata nella seguente maniera:

$$IP_{rp} = \max\{\text{Prec}(r) \mid 1 \leq r \leq N, \text{Rec}(r) \geq rp\}$$

dove N è il numero di item restituiti dalla run e  $\text{Prec}(r)$  e  $\text{Rec}(r)$  sono la Precisione e il Richiamo calcolati sui primi r item. Le Fig.16-17-18 mostrano come disegnare questa curva per le run definite nella Fig.5.

Perché viene effettuato questo calcolo e non si utilizza, piuttosto, una semplice interpolazione lineare? La spiegazione di tutto questo sta nel fatto che, osservando la Curva Richiamo-Precisione Interpolata risultante, chiunque sarebbe disposto a richiamare qualche altro item se questo volesse dire aumentare la percentuale dell'insieme di articoli richiamati rilevanti (vale a dire se la Precisione di un insieme di item recuperati più grande è più alta). Come si può vedere dalla Fig.19, adesso confrontare due run diverse diventa possibile.



Documenti rilevanti del pool per il topic 401 (Recall base =  $|D^*| = 7$ )

r	Run Y	Rec(r)	Prec(r)	$IP_{rp} = \max_{(1 \leq r \leq N \mid Rec(r) \geq \tau_p)} Prec(r)$
1	NR	0	0	$IP_0 = \max_{(1 \leq r \leq N \mid Rec(r) \geq 0)} Prec(r) = 0.5$
2	NR	0	0	
3	R	0.14	0.33	$IP_{0.1} = \max_{(1 \leq r \leq N \mid Rec(r) \geq 0.1)} Prec(r) = 0.5$
4	R	0.29	0.50	
5	NR	0.29	0.40	
6	NR	0.29	0.33	...
7	R	0.43	0.43	$IP_{0.7} = \max_{(1 \leq r \leq N \mid Rec(r) \geq 0.7)} Prec(r) = 0.42$
8	R	0.57	0.50	
9	NR	0.57	0.44	
10	NR	0.57	0.40	...
11	NR	0.57	0.36	
12	R	0.71	0.42	

Fig.16: Interpolated R-P Curve run Y

Documenti rilevanti del pool per il topic 401 (Recall base =  $|D^*| = 7$ )

r	Run Z	Rec(r)	Prec(r)	rp	$IP_{rp}$
1	0	0	0	0	0.50
2	1	0.14	0.5	0.1	0.50
3	0	0.14	0.33	0.2	0.50
4	1	0.29	0.50	0.3	0.50
5	0	0.29	0.40	0.4	0.50
6	0	0.29	0.33	0.5	0.50
7	1	0.43	0.43	0.6	0.42
8	0	0.43	0.38	0.7	0.42
9	0	0.43	0.33	0.8	0
10	1	0.57	0.40	0.9	0
11	0	0.57	0.36	1	0
12	0	0.57	0.33		

Fig.17: Interpolated R-P Curve run Z

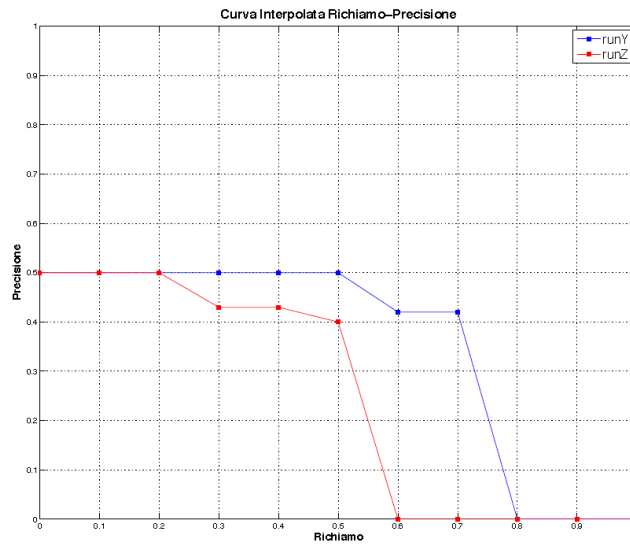


Fig.18: Interpolated R-P Curve di run Y e Z a confronto

Inoltre, sempre dalla Fig.18, si può notare come la run Y sia migliore della run Z, in quanto, a parità di Richiamo, fornisce prestazioni migliori in termini di Precisione. Inoltre, all'aumentare del Recall, la curva della run Y scende più lentamente di quella della run Z.

La Curva Richiamo-Precisione Interpolata permette di definire, partendo da essa, un valore numerico unico per ogni run, detto **11-Point Average Precision (11-pt-AP)**.

$$11pt - AP = \frac{\sum_{rp \in \{0,0.1,\dots,1\}} IP_{rp}}{11}$$

Esso non è altro che la media tra tutti e 11 i punti di Precisione interpolata.

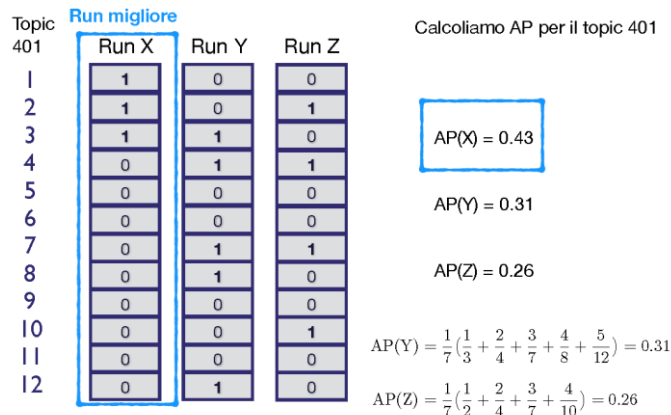
## Average Precision

Infine consideriamo una delle misure binarie più utilizzate nell'ambito della valutazione, l'Average Precision. Come abbiamo già detto, Precisione e Richiamo sono metriche basate sull'intero insieme di item restituiti dal sistema, ma non tengono conto dell'ordine degli articoli restituiti. Per i sistemi che restituiscono una sequenza ordinata di item, è preferibile che questo ordine venga considerato. L'Average Precision è una metrica top heavy, ossia dà più peso (o valore) alla run che mette più item rilevanti in posizioni alte di rank. L'AP è un valore compreso tra 0 e 1.

Dato un topic, la Recall Base **RB** relativa, e una run  $r_t$  di lunghezza  $N$  (con  $N$  numero naturale), l'Average Precision si calcola nella seguente maniera:

$$AP = \frac{1}{RB} \sum_{k=1}^N r_t[k] \cdot p@k$$

dove  $r_t[k]$  vale 1 se l'item in posizione  $k$  della run è rilevante, 0 viceversa. Nella **Fig.19** viene presentato il calcolo della AP per le run definite nella **Fig.5**.



**Fig.19:** Esempio di calcolo della Average Precision

Per spiegare perché l'Average Precision è top heavy, supponiamo che, in seguito ad un miglioramento del sistema, un item rilevante restituito venga spostato dalla posizione 2 alla posizione 1 per una determinata run. Prima di questo miglioramento l'item contribuiva ad AP di 1/2, cioè 0.5, mentre dopo il miglioramento contribuisce di 1. Viceversa, supponiamo che il miglioramento di cui sopra produca lo spostamento di un documento rilevante dalla posizione 100 alla posizione 99. Questo produrrà un piccolo impatto su AP, visto che il contributo del documento passa da 1/100 (0.01) a 1/99 (0.0101).

L'Average Precision ci dice come valutare una run per uno specifico topic, ma come possiamo valutare una run su tutti i topic della collezione? Si calcola l'AP per ogni topic e successivamente si fa la media aritmetica di questi valori, ottenendo così la **MAP (Mean Average Precision)**. Data una run  $R$  e un insieme di topic  $T$  la MAP si calcola nella seguente maniera:

$$MAP(R) = \frac{\sum_{t \in T} AP(r_t)}{|T|}$$

## 2.3 Algoritmo di ordinamento Quicksort

Presentiamo ora l'algoritmo di ordinamento Quicksort, utilizzato agli scopi della tesi, come vedremo, per ordinare le opere in base al grado di similarità con un item prefissato (dalla meno simile alla più simile).

Quicksort è un algoritmo di ordinamento ricorsivo e dicotomico (appartiene alla classe degli algoritmi divide-et-impera).

Dato un array di elementi, ad ogni stadio della ricorsione si effettua un ordinamento parziale di una sequenza di questi elementi da ordinare. Si prende a caso un elemento della sequenza che assume

il ruolo di perno (pivot) per quello stadio. Il pivot, ora, si trova nella sua posizione definitiva. Successivamente la procedura denominata “partition” consente di portare a sinistra dell’array i valori minori del perno e a destra quelli maggiori. Si procede ricorsivamente sui due sottovettori così creati (il vettore degli elementi maggiori del perno e il vettore degli elementi minori), arrestandosi quando, nella scomposizione, si creano vettori di un solo elemento.

Lo pseudocodice per il Quicksort è il seguente:

**Procedure Quicksort**(A, b, e)

```
Input A vettore di elementi
begin
  if b < e then
    begin
      l ← Partition(A, b, e)
      Quicksort(A, b, l - 1)
      Quicksort(A, l + 1, e)
    end
  end
end
```

**Function Partition**(A, b, e)

```
begin
  pivot ← A[e]
  i ← (b-1)
  for j ← b to (e-1) do
    begin
      if A[j] <= pivot then
        begin
          i ← i + 1
          Scambia(A[i], A[j])
        end
      end
    end
  Scambia(A[i+1], A[e])
  return (i+1)
end
```

con b indice del primo elemento dell’array ed e indice dell’ultimo elemento dell’array. Partition restituisce il punto di spezzamento dell’array nei due sottovettori.

L’algoritmo Quicksort è un algoritmo in place, in quanto non crea una copia del vettore in input per operare l’ordinamento. Pertanto un algoritmo in place risparmia memoria rispetto ad un algoritmo non in place.

Si noti che la complessità in termini di memoria, cioè l’occupazione di memoria richiesta dall’algoritmo, è  $O(n)$ , dove n è la lunghezza dell’array da ordinare.

L’algoritmo nel caso peggiore ha un costo quadratico in termini di complessità temporale. Tuttavia il caso peggiore si verifica solo quando lo sbilanciamento è totale, cioè quando l’algoritmo di partizionamento restituisce una partizione di lunghezza n-1 e una di lunghezza 0. In questo caso il tempo di esecuzione è  $\Theta(n^2)$ . Il punto cruciale in QuickSort è scegliere il miglior perno! Se vogliamo evitare che la scelta del partizionamento ci conduca ad un tempo quadratico, è sufficiente scegliere come pivot l’elemento mediano della sequenza. L’elemento centrale è, ovviamente, il migliore, in quanto dividerebbe la lista in due sotto-liste uguali. Tuttavia trovare l’elemento intermedio da un

elenco non ordinato è difficile e richiede tempo, ecco perché è stato preso come perno l'ultimo elemento.

L'algoritmo Quicksort, come vedremo, è parte integrante dell'algoritmo di raccomandazione. È stato scelto fra i tanti perché, oltre ad essere un algoritmo intuitivo e malgrado questo tempo di esecuzione lento nel caso peggiore (che nell'esecuzione del nostro software si verifica molto raramente), risulta essere spesso la migliore scelta pratica di ordinamento, perché in media è notevolmente efficiente visto che il suo tempo medio di esecuzione è  $\Theta(n \log_2(n))$ . Come già detto, offre il vantaggio di ordinare in loco e funziona bene anche in ambienti con memoria virtuale [15].

## Capitolo 3

### Introduzione al lavoro di tesi

Il software che implementa il modello di Sistema di Raccomandazione museale Content-Based progettato per la tesi sfrutta i dati provenienti da TindArt [10], un'applicazione per Android sviluppata dal dottor Daniel Zilio, dottorando del Dipartimento di Beni Culturali dell'Università di Padova. Questa app, scaricabile in versione Beta da Google Play<sup>3</sup>, raccoglie le preferenze degli utenti relativamente ad una sequenza di opere d'arte appartenenti ad un dataset di riferimento, permettendoci così di avere dei dati sull'utente al fine di poterlo profilare, capendone i gusti, aggirando così il problema del "Cold-Start". TindArt semplicemente visualizza sullo schermo dello smartphone delle immagini di opere d'arte, a cui l'utente esprime la propria preferenza cliccando "LIKE" oppure "NOPE". TindArt verrà descritta nella sezione 3.2 del capitolo, ponendo attenzione sui risultati che restituisce, i quali sono stati successivamente utilizzati come input dal software sviluppato per la tesi, al fine di suggerire articoli potenzialmente interessanti per l'utente che effettua una visita.

L'insieme di immagini visualizzate dall'app è stato raccolto durante uno studio iniziale che ha coinvolto un gruppo di studenti. Inizieremo descrivendo questa collezione (o dataset) di riferimento (sezione 3.1), contenente tutte le opere che sono presenti in TindArt e che sono state successivamente votate, tramite l'app, da degli utenti, andando a descrivere che caratteristiche ha, come è stata raccolta e tutte le informazioni che abbiamo su di essa.

### 3.1 Dataset di riferimento

Descriviamo ora la collezione di riferimento contenente tutte le opere che possono essere votate su TindArt e che, di conseguenza, sono state sfruttate dal nostro software con lo scopo di fornire raccomandazioni.

Il dataset è stato fornito dal dottor Daniel Zilio sotto forma di due file xml. Nel primo di questi, come si può vedere dalla **Fig.20**, ogni opera viene descritta tramite dei tag e possiede, nell'ordine, un id univoco, il titolo, la data di creazione, le dimensioni, l'autore, il tipo dell'opera (tag <met\_classification>), la tecnica con cui è stata creata (ad esempio "Olio su tela") e una sua descrizione. Ad ogni opera è associato un set di keyword (parole chiave che la descrivono) presenti nel secondo file xml e collegate ad essa mediante l'id (vedi **Fig.21**).

Le opere di questa collezione sono state raccolte mediante uno studio condotto su 61 studenti frequentanti il corso di "Storia e tutela dei Beni artistici e musicali" dell'Università di Padova. Ognuno di loro ha avuto il compito di scegliere cinque opere d'arte che gli piacessero dall'Heilbrunn Timeline of Art History, l'archivio digitale online del Metropolitan Museum di New York (i metadati dei file xml provengono proprio dal sito del museo<sup>4</sup>). Una volta scelte le opere d'arte, i partecipanti hanno dovuto scaricare l'immagine digitale corrispondente a ciascuna opera e copiarne i metadati, caricandoli successivamente su una piattaforma web dedicata.

---

<sup>3</sup> [https://play.google.com/store/apps/details?id=tindart\\_evo.meeple.tindart](https://play.google.com/store/apps/details?id=tindart_evo.meeple.tindart)

<sup>4</sup> <https://www.metmuseum.org/toah/>

```

<row>
  <id>2</id>
  <titolo>Sergeant of the Colonial Regiment </titolo>
  <date>1906-7 </date>
  <dimension>35 1/2 x 28 in. (90.2 x 71.1 cm)</dimension>
  <autore>Albert Marquet</autore>
  <met_classification>Paintings</met_classification>
  <medium>Oil on canvas</medium>
  <description> Albert Marquet has long been associated with Fauvist painters who,
  for a brief period between 1904 and 1908, immersed themselves in richly saturated hues,
  the last engagement with color before Cubism. Like fellow painters Vlaminck and Derain,
  Marquet often painted seaside ports and rivers meandering through cities, though his
  palette was softer and more subdued. Marquet's "Sargeant of the Colonial Regiment",
  of which another version exists in the Musée des Beaux-Arts in Bordeaux, is one of his few portraits.
  Yet his rather opaque characterization is less about personality than military decorum.
  The figure's gold epaulettes, stripes, and shiny brass buttons bespeak his role as assistant
  to the quartermaster-corporal, decorative details that define and relieve the dark blue mass
  of his military jacket. Marquet's choice of subject coincides with the fervent French
  colonization of Indochina, the West Indies, and northern Africa in the early years of the twentieth century,
  following France's crushing defeat in the Franco-Prussian War. </description>
</row>

```

Fig.20: Esempio di opera descritta tramite tag xml

```

<row>
  <id>2</id>
  <keyword>A.D. 1900-present</keyword>
</row>

<row>
  <id>2</id>
  <keyword>Albert Marquet</keyword>
</row>

<row>
  <id>2</id>
  <keyword>Canvas</keyword>
</row>

<row>
  <id>2</id>
  <keyword>Europe</keyword>
</row>

<row>
  <id>2</id>
  <keyword>France</keyword>
</row>

<row>
  <id>2</id>
  <keyword>Oil paintings</keyword>
</row>

<row>
  <id>2</id>
  <keyword>Paintings</keyword>
</row>

```

Fig.21: Alcune keyword relative all'opera con id 2

Va detto, per correttezza, che alcuni studenti hanno sbagliato a scrivere l'URL del sito Web e hanno scelto le opere d'arte dal sito ufficiale del Metropolitan Museum<sup>5</sup> anziché dall'archivio digitale Heilbrunn Timeline of Art History. Tuttavia si è deciso ugualmente di tenere tutte le opere e i metadati di queste ultime sono stati adattati manualmente.

Su questi dati è stato fatto uno studio, effettuato dal dottor Daniel Zilio e dal professor Nicola Orio, docente universitario del Dipartimento di Beni Culturali, in collaborazione con Camilla Toniolo, dottoressa in Progettazione e gestione del turismo culturale presso l'Università di Padova. Ciò che si è andato ad indagare è stata la coerenza degli studenti nella scelta delle opere.

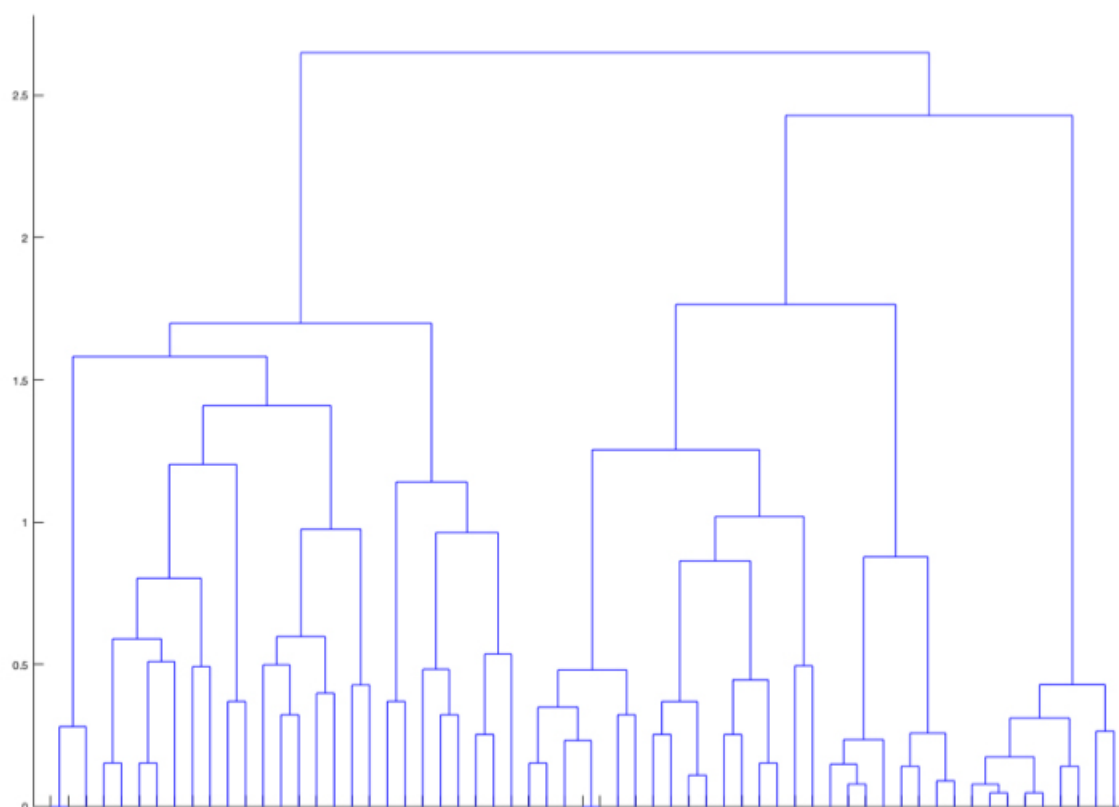
<sup>5</sup> <https://www.metmuseum.org/art/collection/>

Lo studio è stato svolto associando a tutti gli item un genere artistico e una corrente artistica. Tutto ciò ha permesso di aggiungere ai metadati di ogni opera d'arte due campi addizionali, usando come riferimento le informazioni provenienti da WikiArt<sup>6</sup>, combinate con quelle già possedute.

Partendo da un insieme di 35 correnti e 25 generi artistici, alcune correnti simili sono state poi unite, arrivando ad ottenerne 21. Lo stesso procedimento è stato fatto con i generi, ottenendo 11 classi differenti.

Lo scopo di tutto ciò è stato quello di scoprire la coerenza degli studenti nella scelta delle opere, cioè se i partecipanti sceglievano costantemente opere d'arte da un gruppo limitato di correnti e generi artistici oppure no. Come si evince da [10], la maggior parte dei partecipanti ha scelto cinque opere d'arte appartenenti a quattro diversi movimenti artistici. Per quanto riguarda il genere, pure in questo caso la maggior parte degli studenti ha scelto opere appartenenti a più di tre generi diversi. Successivamente è stata effettuata una cluster analysis degli studenti in base alla corrente e al genere. L'idea di aver scelto correnti o generi in comune "avvicina" tra loro gli studenti dal punto di vista delle loro preferenze.

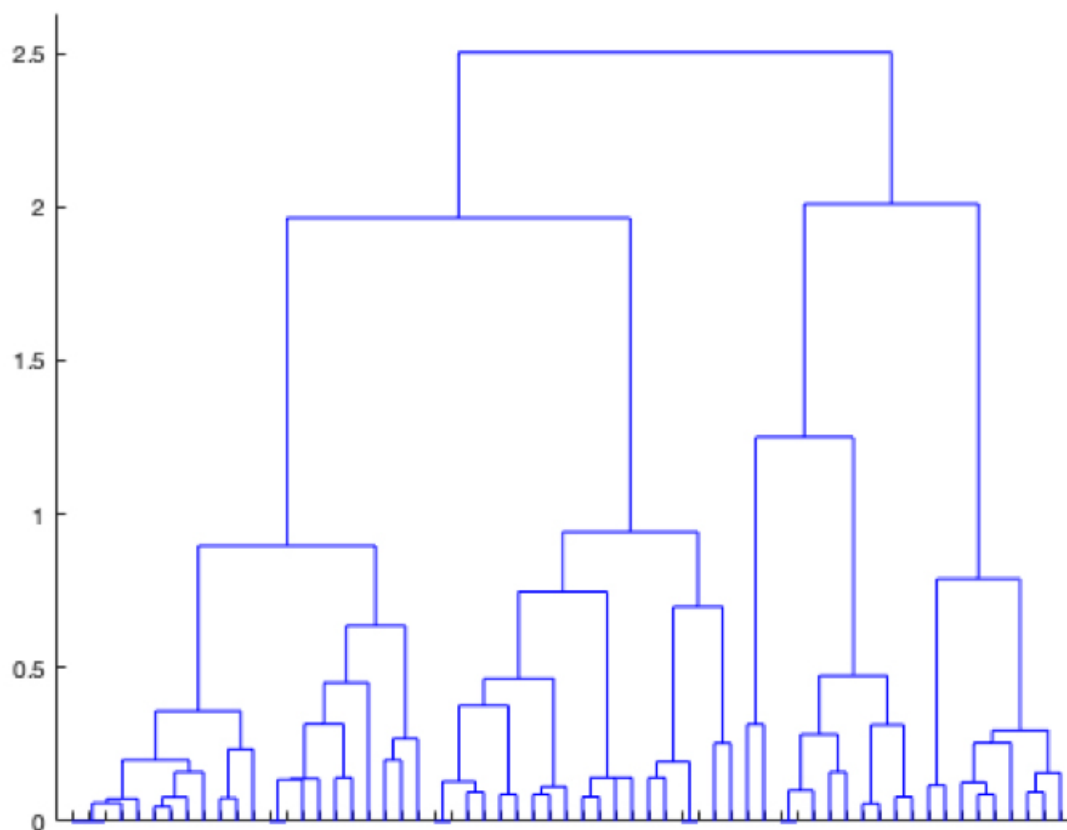
Come si può vedere dal dendrogramma presentato in **Fig.22**, se si raggruppano gli studenti per corrente artistica, ci sono due grossi cluster principali. Scendendo a partire dall'alto se ne possono individuare anche tre.



**Fig.22:** Cluster analysis di studenti in base alla somiglianza delle correnti artistiche scelte

Se si osserva il dendrogramma presentato in **Fig.23**, se si raggruppano gli studenti per genere artistico, anche qui ci sono due grossi cluster principali. Scendendo a partire dall'alto se ne possono individuare anche quattro.

<sup>6</sup> [www.wikiart.com](http://www.wikiart.com)



**Fig.23:** Cluster analysis di studenti in base alla somiglianza dei generi artistici scelti

Tutto ciò dimostra che esistono trend comuni fra gli studenti, per i quali le scelte effettuate da uno studente possono essere ereditate da un altro che fa parte dello stesso cluster. La morale di tutto ciò suggerisce che i partecipanti possono essere profilati in base alle loro scelte.

In conclusione, gli esperimenti svolti dimostrano come gli studenti abbiano effettuato la scelta delle opere in maniera coerente e non puramente a caso, arrivando addirittura ad avere la possibilità di definire una sorta di profilo di uno studente rispetto ad un altro<sup>7</sup>.

Tutto ciò dimostra la validità della collezione raccolta. Se, viceversa, le opere fossero state scelte casualmente, il rischio di errore sarebbe stato elevato (ad esempio il rischio di avere nel dataset tutte le opere dello stesso artista, diminuendo la diversificazione degli item) e sarebbe stato più difficile ottenere dei cluster di studenti.

La collezione di riferimento presenta tutt'ora 352 opere d'arte, le quali vengono visualizzate in TindArt con lo scopo di essere votate da dei potenziali utenti.

---

<sup>7</sup> Date queste evidenze, si è reso possibile effettuare uno studio sulla funzione di similarità creata allo scopo della tesi. Questo studio sfrutta la classificazione delle opere per genere e corrente artistica e verrà presentato più avanti nella relazione.



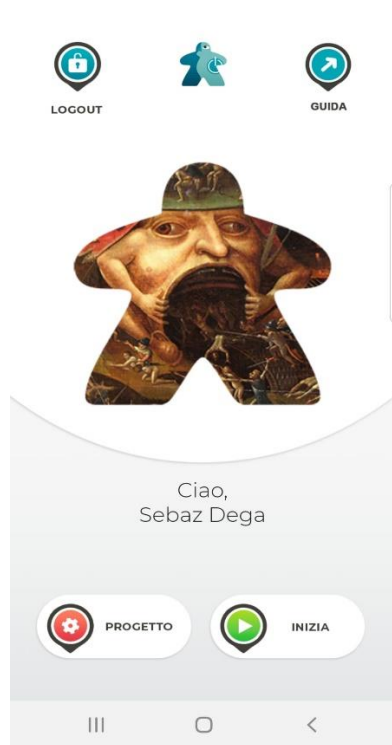


Fig.24: Schermata Home di TindArt

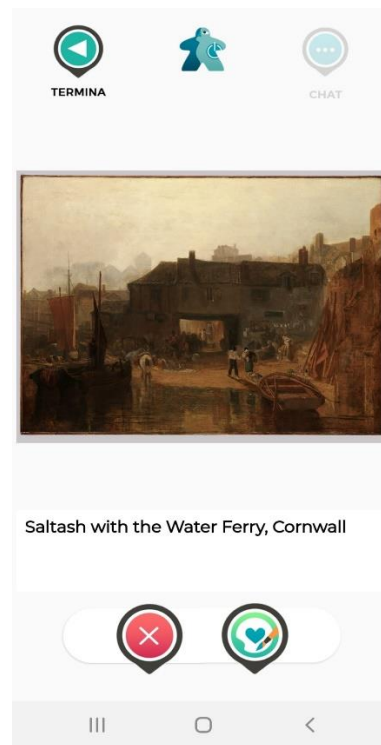


Fig.25: Schermata principale di TindArt

## 3.2 TindArt

Proviamo ad immaginare la situazione in cui un utente in fila ad un museo non sappia come ammazzare il tempo. Tindart potrebbe essere la soluzione ai suoi problemi!

Come già detto, TindArt è un'app sviluppata per dispositivi Android, chiamata così in quanto imita l'applicazione per appuntamenti Tinder nel modo di indicare le preferenze per un item (nel nostro caso le opere d'arte), in modo tale da fornire uno strumento intuitivo all'utente.

Una volta scaricata l'applicazione, l'utente crea un account e quindi viene creata una sessione. Nella schermata principale (vedi Fig.24) sono presenti cinque pulsanti:

- Logout: viene utilizzato per disconnettersi dalla sessione in corso;
- Guida: rimanda ad un tutorial dell'applicazione;
- Progetto: mostra informazioni sul progetto;
- Inizia: inizia la valutazione delle opere;
- PERSONAL INFORMATION: è il logo al centro, attualmente utilizzato solo per uno studio preliminare.

Come si può vedere dalla Fig.25 l'applicazione mostra sequenzialmente, in ordine casuale, le opere d'arte provenienti dal dataset di riferimento descritto nella Sezione 2.1. L'utente può valutare queste opere nel seguente modo:

- o usando il pulsante verde per LIKE o il pulsante rosso per NOPE;
- oppure facendo "swipe", cioè scorrendo il dito da sinistra a destra per LIKE, da destra a sinistra per NOPE.

Come si può vedere, esiste anche un pulsante in alto a destra chiamato Chat, attualmente inutilizzato.

Una sessione può essere interrotta in qualsiasi momento dall'utente con il pulsante in alto a sinistra chiamato Termina, ed è possibile riavviarla in un secondo momento. Una volta che si clicca sul pulsante Termina, l'app prova a fare una semplice raccomandazione, in base alla corrente e al genere delle opere votate. Se si termina una sessione senza aver votato neanche un'opera, questa raccomandazione non viene effettuata, viceversa viene visualizzata sullo schermo la GIF di "Pulp Fiction" (**Fig.26**), la quale sta a significare che l'applicazione non sa cosa suggerire perché non è stata effettuata alcuna scelta.



**Fig.26:** GIF dal film "Pulp Fiction": l'applicazione non sa cosa suggerire

Successivamente alla creazione dell'app e alla sua distribuzione su Google Play, è stato chiesto ad un secondo gruppo di studenti universitari, frequentanti il corso di "Progettazione e gestione del turismo culturale" dell'Università di Padova, di utilizzarla. In un mese 54 studenti l'hanno scaricata e testata, fornendo un totale di 9882 scelte.

Da questo insieme di interazioni, supponendo che alcune di esse siano state casuali, sono state rimosse le scelte per le quali è stato impiegato un tempo inferiore a 0.5 secondi (immaginando che fossero state fatte a caso) e superiore a 20 secondi (immaginando che l'utente, effettuando la scelta, probabilmente era distratto da altre attività).

Le informazioni ottenute dall'analisi dell'esperienza dell'utente mostrano che TindArt potrebbe essere uno strumento molto utile per raccogliere informazioni sugli utenti in uno scenario museale pre-visita.

Grazie al dottor Daniel Zilio è stato possibile disporre delle valutazioni di questi 54 utenti sulle opere del dataset di riferimento. Anche queste ultime sono state fornite sotto forma di file xml<sup>8</sup>, permettendoci così di avere dei dati sull'utente al fine di poterlo profilare, capendone i gusti, e aggirando così il problema della "partenza a freddo". Immaginiamo una situazione reale: se non avessimo a disposizione dati sui gusti degli utenti che entrano al museo, come potremmo profilarli fornendo loro raccomandazioni efficaci di opere che potrebbero gradire?

---

<sup>8</sup> All'interno di questo file sono presenti dei record per ogni votazione (data dall'utente all'opera). Se l'attributo descritto dal tag <vote> ha valore 1 significa che l'opera ha ricevuto un LIKE dall'utente, se ha valore 0 significa che ha ricevuto un NOPE, altrimenti semplicemente il record non è presente.

## Capitolo 4

### Un modello di Sistema di Raccomandazione museale Content-Based: progettazione e sviluppo

Il modello di Sistema di Raccomandazione museale progettato e sviluppato agli scopi di questa tesi ha il compito di profilare un ipotetico utente di un museo, analizzandone i gusti, andando a consigliargli, successivamente, delle opere presenti all'interno della collezione che potrebbero piacergli.

Come già è stato anticipato, l'approccio che si è stimato essere il migliore ai fini del dataset utilizzato, presentato nel Paragrafo 3.1, è quello Content-Based. La collezione di opere d'arte, infatti, come visto, viene descritta utilizzando parole chiave e attributi relazionali per ogni item. Queste features relative al contenuto vengono utilizzate dal nostro sistema, combinate con le valutazioni degli utenti alle suddette opere, con lo scopo di formulare raccomandazioni.

Come per ogni Sistema di Raccomandazione Content-Based, anche in questo caso si è dovuto affrontare il problema Cold-Start. La domanda che ci si è posti è stata la seguente: "come si può profilare un nuovo utente capendone i gusti e consigliargli, successivamente, opere d'arte che potrebbero piacergli?". Come anticipato nel Paragrafo 3.2, chi svolge il compito di raccogliere dati sui nuovi clienti è TindArt, un'applicazione che propone agli utenti le opere del dataset di riferimento, chiedendogli di votarle con un LIKE, oppure un NOPE. Anche questi dati sono stati successivamente utilizzati come input dal software sviluppato per la tesi, al fine di suggerire articoli potenzialmente interessanti per l'utente che effettua una visita. Un esempio di questi dati restituiti dall'app TindArt viene presentato in **Tab.1**. Come si può vedere, ogni utente può aver dato un LIKE all'opera (valore 1), un NOPE (valore 0), oppure può anche essere che non l'abbia votata (magari perché ha terminato la sessione spontaneamente prima di aver valutato tutti gli item).

UTENTI	COLLEZIONE DI OPERE D'ARTE										
	TRAINING SET							TEST SET			
	Opera1	Opera2	Opera3	Opera4	Opera5	.	.	.	OpTest1	OpTest2	...
Utente1	1	0	0	0	1	.	.	.	1	1	...
Utente2	0	1		1	1	.	.	.		1	...
.	.	.	.	.	.	.	.	.	.	.	...
.	.	.	.	.	.	.	.	.	.	.	...
.	.	.	.	.	.	.	.	.	.	.	...
Utente54	0	1	1	1	1	.	.	.	0	0	...

**Tab.1:** Esempio di dati restituiti dall'applicazione TindArt

Una volta acquisite le preferenze, il nostro sistema procede alla raccomandazione. L'algoritmo di raccomandazione verrà accuratamente descritto nella sezione 4.2 del capitolo, nella quale verranno esposti anche i risultati della valutazione off-line del sistema. Al fine di effettuare tale valutazione, il software sviluppato partiziona i dati in due insiemi, come si osserva sempre dalla **Tab.1**:

- **Training Set:** detto anche insieme di addestramento, è l'insieme dei dati utilizzati per addestrare il nostro sistema alla raccomandazione. Come Training Set si è deciso di prendere l'80% delle opere del dataset di riferimento e le corrispettive preferenze espresse dagli utenti per esse.

- **Test Set:** è l'insieme dei dati utilizzati per valutare le prestazioni del sistema. Come Test Set si è deciso di prendere il restante 20% delle opere del dataset di riferimento e le corrispondenti preferenze espresse dagli utenti per esse.

Il sistema, come vedremo, sfrutta i dati del Training Set come conosciuti, al fine di stimare una possibile valutazione per le opere del Test Set, ovvero vengono utilizzate le votazioni date da ogni utente ad ogni opera del Training Set, immaginando che siano la nostra conoscenza pregressa sull'utente stesso, facendo a finta di non conoscere le votazioni date dagli stessi utenti alle opere del Test Set, che vengono stimate dal nostro sistema. Come già osservato, infatti, la valutazione off-line prevede che le raccomandazioni calcolate dagli algoritmi debbano essere successivamente confrontate con le vere valutazioni memorizzate dell'utente alle opere del Test Set. Questi esperimenti possono misurare la qualità dell'algoritmo progettato nell'adempiere al suo compito di raccomandazione.

Riassumendo, l'obiettivo principale della nostra ricerca all'interno della collezione di item (**task**) è quello di raccomandare ad un utente, data una certa conoscenza pregressa su di esso, le opere di un museo che potrebbero piacergli.

La collezione sperimentale, creata ai fini della valutazione e costruita prima di iniziare gli esperimenti, è composta dalla tripla seguente:

- **Dataset di riferimento:** corpus di documenti formato da 352 opere d'arte e presentato nel Paragrafo 3.1;
- **Topics:** le reali esigenze informative degli utenti interessati al task, ovvero, dato un profilo utente (le sue preferenze sulle opere d'arte relative all'80% scelto come Training Set), ci si domanda quali siano le opere del Test Set che potrebbero piacergli, al fine di raccomandargliele;
- **Giudizi di rilevanza:** i giudizi che possono essere dati ad un singolo item del Test Set raccomandato, in modo che esso sia giudicato rilevante o non rilevante per l'utente. Essi sono semplicemente due: "Rilevante" e "Non rilevante". I giudizi di rilevanza indicano quindi se una raccomandazione effettuata è giusta o è sbagliata. Come si intuisce, la nostra **ground truth** è, quindi, il Test Set, in quanto, dato un topic e un item raccomandato, mi dice se quest'ultimo è veramente rilevante per l'utente (se effettivamente l'aveva votato positivamente).

Ad ogni grado di rilevanza è associato un **relevance weight**, precisamente 1 per "Rilevante" e 0 per "Non rilevante", la cosiddetta **rilevanza binaria**.

Al fine di compiere i passi dell'algoritmo di raccomandazione, data un'opera del Test Set, per predire il voto che l'utente darebbe a quell'opera e decidere se raccomandargliela o meno, vengono prese le opere del Training Set simili a questa e viene effettuata una media pesata tra i voti che l'utente ha dato a queste ultime. Per capire quali siano effettivamente le "opere più simili" è stato fatto uno studio preliminare, delineato nella sezione 4.1 del capitolo, in cui è stata creata una funzione di similarità parametrica tra le opere ad hoc, utilizzata successivamente dal nostro algoritmo. Come osserveremo, così come il profilo utente è definito dai voti che ha dato agli item, le opere, essendo descritte da features, sono rappresentate da vettori di uni e zeri, dove 1 rappresenta la presenza di un determinato termine per un determinato item e 0 ne denota l'assenza. Due prodotti artistici saranno più simili quanti più termini avranno in comune. Chiaramente, se poi l'utente ha dato una valutazione positiva ad un'opera del Training Set, l'ha data implicitamente anche alle sue features, morale per cui è più facile che gli vengano consigliati articoli del Test Set che posseggono quelle stesse caratteristiche, o simili.

## 4.1 Studio preliminare: la funzione di similarità

Il nostro Sistema di Raccomandazione, al fine di adempiere ai suoi scopi, necessita di una funzione di similarità fra le opere d'arte del dataset di riferimento, ossia di un metodo che calcoli quanto due opere siano simili tra loro.

Come abbiamo già visto nel Paragrafo 3.1, ogni item del dataset viene descritto da dei metadati, i quali sono, nell'ordine, un id univoco, il titolo, la data di creazione, le dimensioni, l'autore, il tipo dell'opera, la tecnica con cui è stata creata e una descrizione testuale, oltre ad un set di keyword che la descrivono.

Inizialmente l'idea è stata quella di creare una funzione di similarità che contasse il numero dei metadati in comune fra due opere (chiaramente gli item con più elementi in comune sarebbero stati considerati come più simili). Tuttavia il problema di una funzione del genere nasce dal fatto che non esiste la possibilità di creare uno spazio dimensionale per le opere e non si riuscirebbe, quindi, a capire se effettivamente un'opera con più metadati in comune con un item in questione sarebbe più o meno simile ad esso di una che ne condivide di meno. Potrebbe essere ad esempio che, date due opere, una di queste condivida 12 dei suoi 20 termini<sup>9</sup> in comune con l'item in questione, mentre l'altra ne condivida solo 10 ma sia effettivamente descritta solamente da questi 10 termini. Quale delle due è più simile all'item in questione? In effetti 12 è maggiore di 10, ma 10 su 10 è sicuramente più di 12 su 20, ovvero la seconda opera condivide tutti i suoi termini con l'item in questione, mentre la prima solo una parte.

Il Sistema di Raccomandazione sviluppato agli scopi della tesi è di tipo Content-Based, morale per cui si è deciso di utilizzare la **cosine distance** (già descritta interamente nel Paragrafo 2.1) come funzione di similarità, in quanto, non solo ci permette di rappresentare le opere all'interno di uno spazio vettoriale al fine di poterle confrontare tra loro, ma, inoltre, come si è visto sempre nel Paragrafo 2.1, risulta essere il miglior metodo per calcolare la similarità tra due item in un sistema basato sul contenuto.

A questo punto, proprio come prevede il modello vettoriale, essendo che le opere sono descritte da features, si è deciso di rappresentare ciascuna di esse tramite un vettore di zeri e uni, dove 1 rappresenta la presenza di un determinato termine per una determinata opera e 0 ne denota l'assenza. Per fare un esempio concreto, supponiamo di avere due opere d'arte per le quali vogliamo calcolare quanto siano simili fra loro. Seguendo il modello vettoriale esse vengono rappresentate dal sistema in uno spazio vettoriale composto da  $t$  dimensioni, dove  $t$  è la cardinalità dell'unione fra i termini della prima opera e i termini della seconda (i doppietti vengono giustamente eliminati). Di conseguenza le due opere vengono rappresentate come segue:

$$\begin{aligned}O_1 &= (m_{11}, m_{12}, \dots, m_{1j}, \dots, m_{1t}) \\O_2 &= (m_{21}, m_{22}, \dots, m_{2j}, \dots, m_{2t})\end{aligned}$$

dove  $m_{ij}$  rappresenta, appunto, la presenza o meno (1 o 0) del termine  $j$ -esimo per l'opera  $i$ -esima ( $O_i$ ).

Se viene calcolata la distanza coseno tra  $O_1$  e  $O_2$  ( $\text{Cosine}(O_1, O_2)$ ), questa funzione ritorna un numero tra 0 e 1. Più il risultato è alto e più le due opere sono simili. Per come è fatta la cosine similarity, due prodotti artistici si assomigliano di più quanti più termini hanno in comune.

---

<sup>9</sup> Con "termini" si intende l'insieme dei metadati e delle keyword che descrivono l'opera.

Come si può intuire, si è deciso di dare lo stesso peso a tutti i termini, cioè o 1 o 0, perché non c'è ragione di preferire un termine rispetto ad un altro dando a quel termine peso maggiore. In questo modo si prende in considerazione, ad esempio, il termine dell'autore, ma non gli si dà quel peso determinante che fa sì che tutte le opere dello stesso autore siano considerate molto più simili tra loro dalla funzione di similarità, in quanto opere che condividono lo stesso numero di keyword e non l'autore con un item, sono sì meno simili ad esso rispetto ad opere che condividono anche l'autore oltre alle parole chiave, ma non molto ai fini dell'applicazione.

Tuttavia, a questo punto, sorge un problema. La funzione coseno appare perfettamente adatta ai nostri scopi, ma siamo veramente sicuri che, al fine di calcolarla, ci servano tutti i termini che descrivono l'opera? Proviamo a ragionare.

Come è stato osservato da alcuni esperimenti iniziali (effettuati con la funzione che calcolava il numero esatto dei metadati in comune), se due opere condividono lo stesso **autore** e/o la stessa **data di creazione**, allora, solitamente, hanno anche molte più keywords in comune, ergo sono molto più simili tra loro. Questo ci suggerisce come non si possa fare a meno di trascurare questi due metadati nel calcolo della funzione di similarità tra due opere.

Nonostante l'importanza che riveste la data, tuttavia, essa ha riservato non pochi problemi, inizialmente, nel tentativo di estrapolarla. Alla fine si è deciso di trascurarla. La data di creazione si presenta, infatti, in troppe forme diverse fra le varie opere, tra le quali alcuni esempi vengono presentati di seguito:

- 1850;
- 1936-37;
- XVIII secolo;
- Meiji period;
- First half of the 14th century;
- 1720-1750;
- 1520s;
- 1860-6;
- 1<sup>st</sup>-2<sup>nd</sup> century A.D.;
- Ca. 1580-85;
- Mid 1630s;
- Spring/Summer 1999;
- ...

Le forme in cui si presenta, quindi, sono decisamente troppe e troppo variabili per cui valga veramente la pena di tentare di estrapolare il metadato data. Inoltre si è osservato come informazioni relative al periodo siano già contenute nell'insieme delle keywords della stragrande maggioranza delle opere. Per questi motivi si è deciso di trascurare la data nel calcolo della funzione di similarità.

Per quanto riguarda il **tipo** dell'opera e la **tecnica** con cui è stata creata, queste potrebbero essere caratteristiche che non è desiderabile trascurare. Pensiamo al caso in cui, ad esempio, venissero a visitare il museo dei clienti più interessati a certi tipi di produzione delle opere piuttosto che ad altri. A questo punto tipo e tecnica diventerebbero due metadati discriminanti. Quindi, a parità di termini in comune, conoscere la tecnica con cui è stata creata un'opera e il tipo dell'opera può permetterci di cambiare il valore restituito dalla distanza coseno fra due articoli della collezione.

Nonostante l'importanza di questi due metadati, tuttavia, valgono le stesse considerazioni fatte per la data, ossia esistono informazioni relative al tipo dell'opera e alla tecnica con cui è stata creata che

sono già contenute nell'insieme delle keywords della stragrande maggioranza delle opere, ragion per cui anche questi metadati diventano un'informazione ridondante, che può essere trascurata nel calcolo della funzione di similarità.

Come si può intuire, quindi, le **keywords** descrivono molto bene le opere d'arte, visto anche il fatto che, come introdotto nel Paragrafo 3.1, vengono da un database prefissato<sup>10</sup>, cioè sono uniformate, ovvero si attinge sempre dallo stesso pool di parole chiave per descrivere le opere d'arte. Esse rappresentano, probabilmente, l'informazione più importante che abbiamo per il calcolo della funzione di similarità.

L'**id** e le **dimensioni** dell'opera sono completamente trascurabili per il calcolo della cosine distance, in quanto sono sempre diversi per ogni opera. Inoltre le dimensioni, come la data, vengono espresse sempre in forme differenti, cioè non esiste un pattern comune per rappresentare le proporzioni di un'opera.

Il **titolo** dell'opera è, molto probabilmente, un'informazione importante per calcolare la distanza tra due item. Supponiamo, ad esempio, di avere due dipinti intitolati "Madonna con bambino", o magari due opere che, nel titolo, contengono il termine "Madonna". Chiaramente queste opere sono più simili tra loro rispetto che ad altre.

Come si può intuire, il titolo è formato da più parole, molte delle quali sono stopword<sup>11</sup>. Ai fini del calcolo della funzione cosine distance è stata precedentemente effettuata l'eliminazione delle stopword<sup>12</sup> e dei doppioni dal titolo. Infatti, solo i termini che non sono stopword vengono effettivamente usati per la creazione dello spazio vettoriale, in quanto sono gli unici che portano un vero contenuto informativo e sono veramente utili ai fini della similarità tra due item. Lo stesso procedimento è stato adoperato per la descrizione dell'opera.

A questo punto ci si è domandati se valesse effettivamente la pena utilizzare anche i termini della **descrizione** nel calcolo della distanza coseno, insieme all'autore, al titolo e alle keywords. In effetti le parole chiave, da sole, già descrivono bene le nostre opere. Inoltre risulta ovvio come l'inserimento della descrizione abbassi il valore della cosine similarity, visto il fatto che è molto "rumorosa". Infatti, a differenza delle keywords che sono uniformate, la descrizione testuale può essere fatta utilizzando qualsiasi insieme di parole (si possono anche descrivere due opere identiche utilizzando termini completamente differenti). Volendo calcolare la cosine distance tra due opere, ogni termine (non stopword) presente nella descrizione di una e assente nella descrizione dell'altra, alza il valore del denominatore della funzione, a discapito del numeratore. Tuttavia il fatto di avere valori molto bassi di distanza coseno non è determinante in quanto, come già detto, al Sistema di Raccomandazione interessa recuperare le opere più simili ad un determinato articolo della collezione. Quindi, se inserendo la descrizione tutti i valori di similarità fra le opere e questo item si abbassano, al sistema interessano comunque i valori più alti fra questi. Eliminare a priori la descrizione dell'opera dal calcolo della funzione di similarità sarebbe un danno, in quanto non sempre le keywords vengono rese disponibili da un museo. Molto spesso, ciò che un museo possiede delle sue opere d'arte sono solamente delle descrizioni testuali.

In conclusione, a quanto pare, sembrerebbe che i termini del titolo, i termini della descrizione, l'autore e le keywords siano le informazioni più importanti per il calcolo della cosine similarity.

Nonostante tutto, però, queste considerazioni fatte fino a questo momento, per quanto lecite, erano comunque delle supposizioni che dovevano essere valutate. Più precisamente, questa similarità fra le opere è basata sulla similarità dei descrittori, non su delle considerazioni di esperti

---

<sup>10</sup> Il database del Metropolitan Museum.

<sup>11</sup> Parole portatrici di scarso contenuto informativo, se usate indipendentemente da altre parole. Ad esempio articoli, pronomi personali, preposizioni, congiunzioni.

<sup>12</sup> Per raggiungere tale scopo è stata usata una stoplist di parole inglesi ottenuta dall'unione di 6 stoplist reperite online.

storici dell'arte. Si è supposto che, mettendo insieme informazioni su titolo, descrizione, autore e keywords, più termini due opere condividono in comune e più simili queste sono tra loro, ma non sapevamo se questa fosse una buona "guess". Bisognava valutarlo.

Inoltre, ci si è chiesti se servisse davvero tutta questa informazione. Di fatto, permanevano ancora dei dubbi su quanto fosse importante l'utilizzo o meno della descrizione piuttosto che delle keywords.

In sostanza, siamo davvero sicuri che informazioni relative al titolo, all'autore, alle keywords e alla descrizione, se messe insieme, siano tutte ugualmente utili a fornire le raccomandazioni migliori per il nostro sistema? Cosa succederebbe se avessimo solo una parte di questi dati? Ad esempio, come detto prima, cosa succederebbe se il museo ci fornisse solo la descrizione delle opere?

L'obiettivo principale di ogni Sistema di Raccomandazione è quello di fornire i suggerimenti migliori per ogni utente. Come abbiamo precedentemente osservato, il nostro Recommendation System utilizza la funzione di similarità con lo scopo di recuperare le opere più simili ad un determinato item. Di fatto si tratta di un sottoproblema di Reperimento dell'Informazione all'interno del dataset delle opere, dove il task di ricerca è, appunto, "cercare le opere più simili". L'idea consiste nel fatto che, quanto meglio questo obiettivo viene raggiunto, tanto migliore sarà la raccomandazione finale. Del resto, se voglio stimare il voto che l'utente darebbe ad un item per decidere se raccomandarglielo o meno, restituendo le opere effettivamente più simili (al fine di calcolare questo voto tramite una media pesata tra le valutazioni che l'utente ha dato a queste ultime) avrò maggiori probabilità che il voto stimato sia corretto.

Si è deciso, quindi, di modificare la **funzione di similarità** nel seguente modo:

$$\text{Dist}(O_1, O_2) = \alpha \cdot A(O_1, O_2) + \beta \cdot B(O_1, O_2) + \gamma \cdot C(O_1, O_2) + \delta \cdot D(O_1, O_2)$$

$$\alpha + \beta + \gamma + \delta = 1, \quad 0 \leq \alpha, \beta, \gamma, \delta \leq 1$$

- Modulo  $A(O_1, O_2)$ : rappresenta la cosine distance tra le due opere calcolata utilizzando esclusivamente i termini presenti nei titoli delle due.
- Modulo  $B(O_1, O_2)$ : rappresenta la cosine distance tra le due opere calcolata utilizzando il termine "autore" di ciascuna delle due (se l'autore è lo stesso B vale 1, altrimenti vale 0).
- Modulo  $C(O_1, O_2)$ : rappresenta la cosine distance tra le due opere utilizzando esclusivamente le keywords delle due.
- Modulo  $D(O_1, O_2)$ : rappresenta la cosine distance tra le due opere utilizzando esclusivamente i termini presenti nelle descrizioni delle due.

Anche questa funzione di similarità, così come la cosine similarity, ritorna un valore tra 0 e 1. Supponiamo infatti che tutti i termini siano uguali, e quindi  $A = B = C = D = 1$ . Allora  $\text{Dist}(O_1, O_2) = 1$ . Chiaramente, anche qui, più il risultato è alto e più le due opere sono ritenute simili.

Grazie a questa nuova funzione di similarità, che come vedremo è stata quella utilizzata nel nostro sistema, abbiamo delle "manopole sperimentali", le variabili  $\alpha, \beta, \gamma, \delta$ , le quali sono utili a bilanciare l'effetto del titolo, dell'autore, delle keywords e della descrizione nel calcolo della similarità tra due opere. Grazie ad esse è possibile fare degli esperimenti nel calcolo delle opere più simili ad un determinato item, per esempio considerare solo la descrizione nel calcolo della similarità ( $\alpha = \beta = \gamma = 0, \delta = 1$ ), oppure sperimentare cosa succederebbe utilizzando solo le keywords ( $\alpha = \beta = \delta = 0, \gamma = 1$ ), al fine di vedere se funzionano meglio della descrizione, o addirittura vedere cosa succederebbe se bilanciassimo l'effetto delle keywords e quello della descrizione, combinandoli nel calcolo della similarità ( $\alpha = \beta = 0, \gamma = \delta = 0.5$ ). Un altro ragionamento che questa funzione permetterebbe di fare sarebbe quello di vedere in che misura titolo e autore sarebbero utili. E se, ad esempio,



impostassimo  $\beta = 0$ , la nostra funzione di similarità sarebbe in grado di recuperare ugualmente le opere dello stesso autore tra le più simili, ben sapendo che, in effetti, produzioni artistiche dello stesso creatore, di solito, sono sempre piuttosto affini tra loro?

Grazie a questa nuova funzione di similarità parametrizzata è stato possibile rispondere a tutte queste domande, ponendosi come obiettivo quello di trovare il valore migliore per ciascuna delle variabili  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ , al fine di effettuare la restituzione delle opere più simili nel migliore dei modi possibili, con lo scopo di ottenere, successivamente, le raccomandazioni migliori per un ipotetico utente.

Per raggiungere questo scopo, come vedremo di seguito, sono stati fatti degli esperimenti, in ciascuno dei quali, per ciascuna opera d'arte, sono state calcolate le 30 più simili ad essa secondo i risultati forniti dalla funzione di similarità, settando queste manopole nei diversi modi. In sostanza, per ogni item è stata calcolata la similarità con tutti gli altri articoli della collezione, ordinandoli, successivamente, dal valore più alto determinato dalla funzione al valore più basso, mediante algoritmo Quicksort (Paragrafo 2.3), restituendo, come detto, la lista dei primi 30<sup>13</sup>.

Tuttavia, per ogni opera, di questi 30 item restituiti come "più simili", quali sono davvero pertinenti e quali no? Avremmo avuto bisogno di un esperto storico dell'arte che ci dicesse quali fossero le opere veramente più simili e quali no fra quelle restituite per ogni item, al fine di comprendere, per ogni combinazione di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ , quanto bene funzionasse il nostro sistema. Parlando in termini di Richiamo e Precisione, per ogni run (si ha una run diversa ogni volta che vengono modificate  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ ) non avevamo a disposizione l'insieme degli elementi rilevanti. Ciononostante esistono delle potenziali evidenze che ci hanno permesso di definire 4 diversi insiemi di item rilevanti, per ciascuno dei quali è stato fatto un esperimento diverso:

- Il primo insieme degli elementi rilevanti è formato dalle opere del dataset che condividono lo **stesso autore** con l'item in questione, quello per il quale calcoliamo i 30 articoli più simili. L'evidenza sta nel fatto che, come già osservato precedentemente, ci si aspetta che opere dello stesso autore siano posizionate in alto nella lista restituita, ovvero abbiano valori alti della funzione di similarità (opere dello stesso autore di solito sono più simili tra loro).
- Avendo a disposizione i dati descritti nel Paragrafo 3.1 e sapendo che sono stati presi in maniera coerente, un'altra evidenza sta nel fatto che ci si aspetta che opere della stessa corrente, dello stesso genere o scelte dallo stesso studente, siano in posizioni alte della rank list (del resto, opere della stessa corrente o dello stesso genere di solito sono più simili tra loro). Quindi è stato possibile definire tre ulteriori insiemi diversi di opere rilevanti:
  - i. l'insieme delle opere del dataset che condividono la stessa corrente con l'item in questione;
  - ii. l'insieme delle opere del dataset che condividono lo stesso genere con l'item in questione;
  - iii. l'insieme delle opere del dataset che, come l'item in questione, sono state scelte dallo stesso studente.

Quando si effettuano esperimenti di questo tipo, valutando il "motore di ricerca" su queste caratteristiche generali, se si osserva che mediamente gli item definiti rilevanti vengono messi in fondo o non vengono restituiti nella rank list (valori di Precisione, Richiamo e Average Precision bassi), allora si capisce che il motore va migliorato. Nel nostro caso, visto che per ogni opera si ottiene una diversa rank list a seconda delle combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ , per migliorare il reperimento

---

<sup>13</sup> Valore settato sperimentalmente (30 sono circa il 10% delle opere).

delle 30 opere per ogni item è sufficiente modificare queste quattro manopole, fintanto che non si ottiene una combinazione di queste che mi dà i valori migliori di Richiamo e Precisione.

In conclusione, questo studio è servito a stabilire con che valori si è deciso di settare  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$  della nostra funzione di similarità, al fine di utilizzarla nell'algoritmo principale, con lo scopo finale di ottenere raccomandazioni migliori.

Di seguito vengono presentati i quattro esperimenti sopracitati.

## 1) Insieme degli elementi rilevanti: opere dello stesso autore

Al fine di compiere questi esperimenti, come già accennato, è stata implementata la funzione di similarità sopracitata mediante software, con lo scopo di trovare il valore migliore per ciascuna delle variabili  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ . Tutte le opere del dataset sono state acquisite e, per ognuna di esse, è stata calcolata la similarità con tutti gli altri articoli della collezione mediante questa funzione, restituendo i 30 più simili, sempre per ognuna di esse, settando le manopole in diversi modi. Date queste opere restituite per ogni item, come primo esperimento, si è deciso di utilizzare come insieme degli elementi rilevanti le opere del dataset che condividono lo **stesso autore** con l'item in questione. Come già detto, ci si aspetta che opere dello stesso autore siano posizionate in alto nella lista restituita, ovvero abbiano valori alti della funzione di similarità, o che comunque vengano recuperate, visto che opere dello stesso autore, di solito, sono più simili tra loro.

Come prima azione,  $\beta$  è stata settata a 0 per ogni prova. Chiaramente, se  $\beta$  fosse impostata ad un valore maggiore di 0, essendo essa la variabile che bilancia l'effetto dell'autore nella funzione, tutte le opere con lo stesso autore sarebbero automaticamente spinte più in alto nella rank list, come conferma una delle prove effettuate, ergo sarebbe come barare. Ciò che si voleva veramente osservare era se, impostando  $\beta = 0$ , la nostra funzione di similarità sarebbe stata in grado di recuperare ugualmente le opere dello stesso autore tra le più simili, ben sapendo che produzioni artistiche dello stesso creatore, di solito, sono sempre piuttosto affini tra loro.

Per ognuno degli item del dataset (tranne per le opere in cui l'autore compare solo una volta), è stata effettuata la valutazione dell'insieme dei 30 articoli restituiti come potenzialmente "più simili". Sono state calcolate le seguenti misure di valutazione (vedi Paragrafo 2.2):

- **Richiamo**: si vuole vedere quante opere dello stesso autore sono state richiamate sul totale di opere dello stesso autore presenti nel dataset.
- **Precisione a diversi livelli di cut-off ( $p@1$ ,  $p@3$ ,  $p@5$ ,  $p@10$ ,  $p@15$ ,  $p@20$ ,  $p@30$ ,  $p@r$ )**: si vuole vedere quante opere dello stesso autore sono state richiamate su una parte, o sul totale, delle opere restituite. Tutto ciò è stato calcolato per vedere come si comporterebbe il sistema se venissero restituite le prime  $k$  opere (ci si aspetta che la Precisione tenda a scendere all'aumentare di  $k$ , visto che ogni run, per essere considerata "buona", dovrebbe avere i documenti rilevanti in posizioni alte di rank).
- **Average Precision**: si vuole vedere quale, tra tutte le run (combinazione di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ ) mette più documenti rilevanti in posizioni alte di rank.
- **Curva Richiamo-Precisione Interpolata e 11-pt-AP**: si vuole vedere l'andamento della Precisione all'aumentare del Richiamo. Come si è visto nel Paragrafo 2.2, più alta è la curva e migliore è la run.

Il desiderio, quindi, era quello di scoprire, per ogni combinazione di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ , quanto bene funzionasse il nostro sistema.

Come ripetuto più volte, uno degli obiettivi principali era quello di capire il rapporto keywords-descrizione. Quindi, i parametri sono stati settati nei seguenti modi:

- $\alpha = \beta = \gamma = \delta = 1/4$
- $\alpha = \beta = 0$

$\gamma$	$\delta$
0	1
1	0
0.5	0.5

- $\beta = 0$

$\alpha$	$\gamma$	$\delta$
1/3	1/3	1/3
1/3	2/3	0
1/3	0	2/3

In questo modo è stato possibile considerare solo la descrizione nel calcolo della similarità ( $\alpha = \beta = \gamma = 0, \delta = 1$ ), oppure sperimentare cosa succederebbe utilizzando solo le keywords ( $\alpha = \beta = \delta = 0, \gamma = 1$ ), al fine di vedere se funzionano meglio della descrizione nel reperimento delle 30 opere per ogni item, o addirittura vedere cosa succederebbe se bilanciassimo l'effetto delle keywords e quello della descrizione. Inoltre sono state fatte delle prove in cui si è tentato di inserire anche il titolo ( $\alpha = 1/3$ ).

Alla fine di ogni prova (ricordiamo che le prove, o run, variano al variare di  $\alpha, \beta, \gamma$  e  $\delta$ ) è stato calcolato il valor medio dei risultati di ciascuna metrica di valutazione calcolata per ciascuno degli insiemi delle opere restituite. I risultati numerici sono stati riportati in **Tab.2** e **Tab.3** (graficamente nelle **Fig.27** e **Fig.28**).

Come si può osservare dai risultati ottenuti, se  $\alpha = \beta = \gamma = \delta = 1/4$ , viene dato un peso eccessivo all'autore nella funzione di similarità, per il quale, se le opere hanno lo stesso autore in comune, il risultato della funzione aumenta di 0.25 e queste vengono spinte molto più in alto nella rank list rispetto alle opere che non condividono lo stesso autore (+0 sul valore della funzione di similarità). Quindi, ovviamente, tutti i risultati di Richiamo e Precisione hanno valori altissimi, così come l'Average Precision.

Tuttavia, come sottolineato precedentemente, a noi interessano solo le combinazioni con  $\beta = 0$ . Notiamo come i valori ottenuti siano dei buoni risultati, soprattutto in termini di Richiamo (considerando il fatto che la valutazione viene fatta in base ad una evidenza). Per quanto riguarda la Precisione, essa, come previsto, tende a scendere gradualmente all'aumentare di  $k$ , dimostrando come ci sia una certa tendenza a piazzare gli item rilevanti in posizioni alte di rank.

Ciò che si osserva inizialmente è che l'uso o meno del titolo nella funzione di similarità non cambia di molto i valori di Richiamo e Precisione, anzi, sembrerebbe addirittura svantaggioso in termini di Precisione e AP. Questo risultato sottolinea il fatto che non sempre lo stesso autore utilizza gli stessi termini per intitolare un'opera.

Osservando anche le Interpolated R-P Curves, si può notare come l'uso delle keywords funzioni nettamente meglio dell'uso della descrizione, esattamente come era stato previsto, visto che due opere uguali possono essere descritte con termini completamente diversi, mentre le keywords sono uniformate.

Nonostante ciò, il risultato più sorprendente è che keywords e description permettono di ottenere, se combinati, i risultati migliori in tutti i casi.

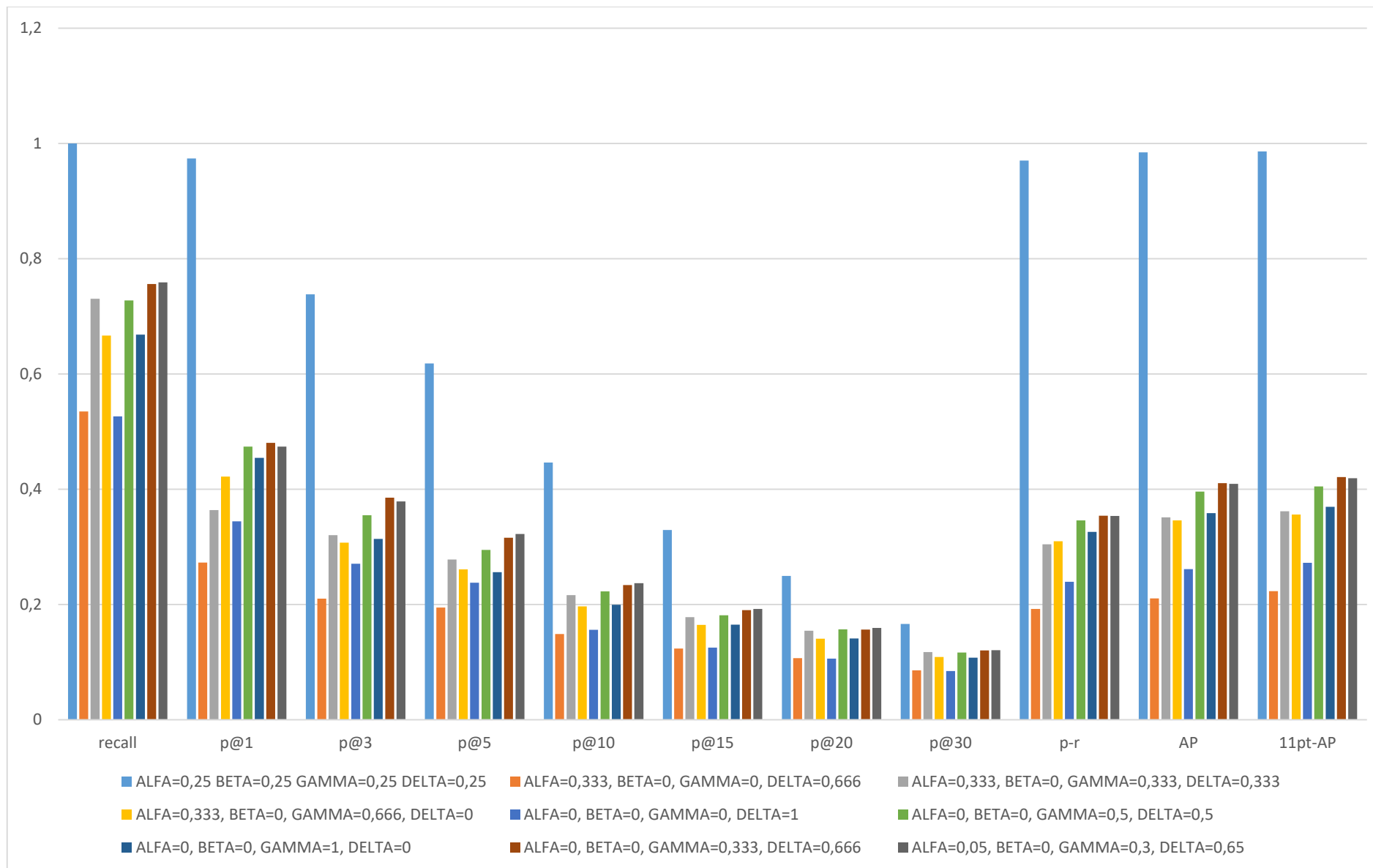
Questi risultati mostrano molto chiaramente la natura Content-Based del sistema: keywords e description sono relative al contenuto e danno i risultati migliori, il titolo non necessariamente è relativo al contenuto dell'opera.

Tentando di cavalcare la cresta dell'onda, sono state fatte due ulteriori prove, una senza l'utilizzo del titolo e l'altra con, sempre combinando l'utilizzo delle keywords e della descrizione, ma, questa volta, tentando di proporzionare i valori di  $\alpha$ ,  $\gamma$  e  $\delta$  rispetto alla quantità di termini contenuti, rispettivamente, nel titolo, tra le keywords e nella descrizione:

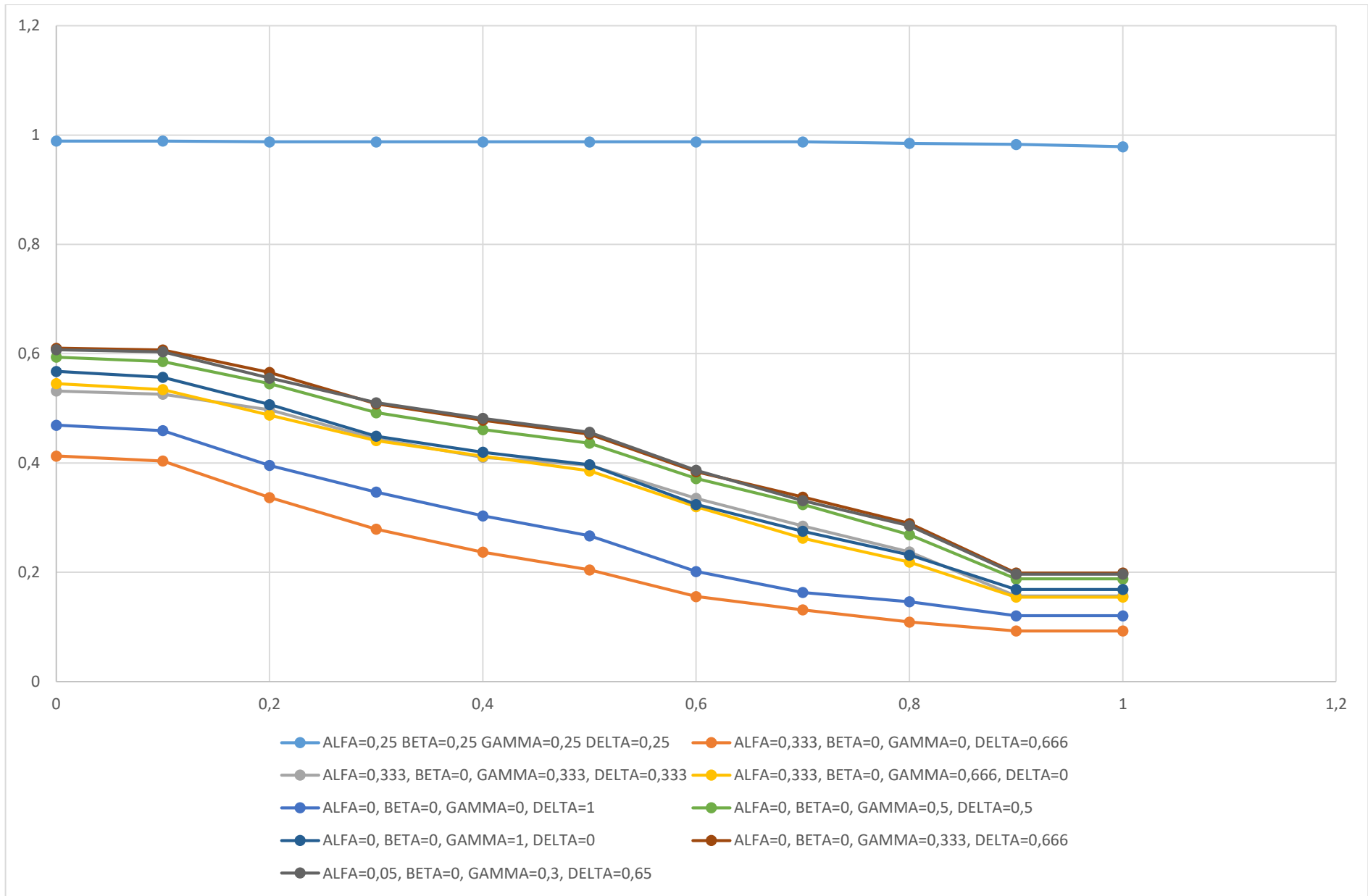
- 1)  $\alpha = 0$ ,  $\beta = 0$ ,  $\gamma = 1/3$ ,  $\delta = 2/3$
- 2)  $\alpha = 0.05$ ,  $\beta = 0$ ,  $\gamma = 0.3$  e  $\delta = 0.65$

cioè dando molto più peso alla descrizione che contiene molti più termini, poi alle keywords che sono circa la metà dei termini della descrizione, ed infine al titolo, il quale è formato da un numero esiguo di termini rispetto alla descrizione testuale e alle parole chiave.

Esattamente come per le altre prove, l'inserimento o meno del titolo non cambia di molto i risultati. Tuttavia, tentando di proporzionare i valori di  $\gamma$  e  $\delta$  rispetto alla quantità di termini contenuti tra le keywords e nella descrizione, si ottengono risultati leggermente migliori che dando a ciascuno dei due lo stesso peso. Come vedremo negli altri esperimenti, questa caratteristica non viene del tutto mantenuta.



**Fig.27:** Risultati della valutazione dell'esperimento 1 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$



**Fig.28:** Curve Richiamo-Precisione Interpolate dell'esperimento 1 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$

	Recall	p@1	p@3	p@5	p@10	p@15	p@20	p@30	p-r	AP	11pt-AP
$\alpha=0,25 \beta=0,25 \gamma=0,25 \delta=0,25$	1	0,974026	0,738095238	0,618181818	0,446104	0,329004329	0,249350649	0,166233766	0,970421047	0,984679625	0,986405841
$\alpha=0,333, \beta=0, \gamma=0, \delta=0,666$	0,535052051	0,272727	0,20995671	0,194805195	0,148701	0,123376623	0,106818182	0,085714286	0,192326324	0,210479178	0,223169993
$\alpha=0,333, \beta=0, \gamma=0,333, \delta=0,333$	0,730638013	0,363636	0,32034632	0,277922078	0,216234	0,177922078	0,154545455	0,117316017	0,304362503	0,351035928	0,36153428
$\alpha=0,333, \beta=0, \gamma=0,666, \delta=0$	0,666643476	0,422078	0,307359307	0,261038961	0,196753	0,164502165	0,140584416	0,108874459	0,30975572	0,346045687	0,356076834
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,526213667	0,344156	0,270562771	0,237662338	0,155844	0,125108225	0,106168831	0,084415584	0,239388786	0,261213526	0,272080341
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,727661822	0,474026	0,354978355	0,294805195	0,222727	0,181385281	0,156818182	0,116450216	0,345938982	0,395655588	0,404939962
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,668068439	0,454545	0,313852814	0,255844156	0,199351	0,164935065	0,140909091	0,107792208	0,325827149	0,358645603	0,369545364
$\alpha=0, \beta=0, \gamma=0,333, \delta=0,666$	0,755952381	0,480519	0,385281385	0,315584416	0,233766	0,19004329	0,156493506	0,12012987	0,35403525	0,410589409	0,420970413
$\alpha=0,05, \beta=0, \gamma=0,3, \delta=0,65$	0,758766234	0,474026	0,378787879	0,322077922	0,237013	0,192207792	0,159415584	0,120779221	0,353566275	0,409155986	0,419162131

**Tab.2:** Risultati numerici della valutazione dell'esperimento 1 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$

	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$\alpha=0,25 \beta=0,25 \gamma=0,25 \delta=0,25$	0,989177489	0,989177	0,987554113	0,987554113	0,987554	0,987554113	0,987554113	0,987554113	0,98495671	0,983054572	0,978773314
$\alpha=0,333, \beta=0, \gamma=0, \delta=0,666$	0,412738454	0,403663	0,336796729	0,27898442	0,237021	0,204381701	0,155606668	0,13128608	0,10909696	0,092647209	0,092647209
$\alpha=0,333, \beta=0, \gamma=0,333, \delta=0,333$	0,531703685	0,525805	0,497485843	0,444482038	0,410412	0,396125076	0,335284141	0,284637207	0,237410326	0,156765548	0,156765548
$\alpha=0,333, \beta=0, \gamma=0,666, \delta=0$	0,545336913	0,534352	0,487585407	0,441023074	0,412385	0,385592616	0,320079179	0,262392073	0,218694324	0,154702391	0,154702391
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,469104185	0,459196	0,395583462	0,346889245	0,303247	0,266931295	0,201595688	0,163155457	0,146036356	0,120572548	0,120572548
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,593628326	0,585674	0,545087361	0,492103407	0,461165	0,436399753	0,371646367	0,324089754	0,268774415	0,187885949	0,187885949
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,567673835	0,556854	0,506987367	0,449063728	0,419694	0,396865221	0,324043933	0,275052257	0,231605493	0,168579858	0,168579858
$\alpha=0, \beta=0, \gamma=0,333, \delta=0,666$	0,610094379	0,606848	0,56565701	0,508205049	0,478243	0,452962275	0,384203424	0,337702881	0,289222378	0,198768052	0,198768052
$\alpha=0,05, \beta=0, \gamma=0,3, \delta=0,65$	0,607201745	0,603522	0,555320927	0,510452703	0,481692	0,456170288	0,386710573	0,331111615	0,285254943	0,196673501	0,196673501

**Tab.3:** Curve Richiamo-Precisione Interpolate dell'esperimento 1 per diverse combinazioni  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ : risultati numerici

## 2) Insiemi degli elementi rilevanti: opere della stessa corrente, dello stesso genere o scelte dallo stesso studente

Successivamente al primo, sono stati realizzati altri tre esperimenti che sfruttano i dati e le evidenze descritte nel Paragrafo 3.1. Ci si aspetta che opere della stessa corrente, dello stesso genere o scelte dallo stesso studente<sup>14</sup>, siano in posizioni alte della rank list (opere della stessa corrente o dello stesso genere di solito sono più simili tra loro) o che comunque vengano recuperate. Quindi è stato possibile definire tre ulteriori insiemi di opere rilevanti, per ciascuno dei quali è stato fatto un esperimento diverso:

- l'insieme delle opere del dataset che condividono la **stessa corrente** con l'item in questione;
- l'insieme delle opere del dataset che condividono lo **stesso genere** con l'item in questione;
- l'insieme delle opere del dataset che, come l'item in questione, sono state scelte dallo **stesso studente**.

Il software, per questi esperimenti, è stato aggiornato, aggiungendo ad ogni opera del dataset le informazioni sulla corrente di cui fa parte, sul genere e sugli studenti che l'hanno scelta.

Come nel primo esperimento, anche qui, per ogni opera del dataset sono state calcolate le 30 più simili, settando  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$  in diversi modi, con lo scopo di trovarne i valori migliori ai fini della valutazione, in termini di Richiamo e Precisione.

In questi casi, rispetto a prima,  $\beta$  può chiaramente essere utilizzato. In questo modo è stato possibile capire anche quanto conta l'autore ai fini della similarità fra due opere.

Per ognuno degli item del dataset, in ciascuno dei tre esperimenti, è stata effettuata la valutazione dell'insieme dei 30 articoli restituiti al variare di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ . Anche in questo caso sono state calcolate le seguenti misure:

- **Richiamo**: opere dello stesso genere/corrente/studente che sono state richiamate sul totale delle opere dello stesso genere/corrente/studente presenti nel dataset.
- **Precisione a diversi livelli di cut-off ( $p@1$ ,  $p@3$ ,  $p@5$ ,  $p@10$ ,  $p@15$ ,  $p@20$ ,  $p@30$ ,  $p@r$ )**: opere dello stesso genere/corrente/studente che sono state richiamate su una parte, o sul totale, delle opere restituite. Tutto ciò è stato calcolato per vedere, per ogni esperimento, come si comporterebbe il sistema se venissero restituite le prime  $k$  opere (ci si aspetta che la Precisione tenda a scendere all'aumentare di  $k$ , visto che ogni run, per essere considerata "buona", dovrebbe avere gli item rilevanti in posizioni alte di rank).
- **Average Precision**: per ogni esperimento, quale, tra tutte le run (combinazione di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ ) mette più documenti rilevanti in posizioni alte di rank.
- **Curva Richiamo-Precisione Interpolata e 11-pt-AP**: per ogni esperimento, l'andamento della Precisione all'aumentare del Richiamo. Come si è visto nel Paragrafo 2.2, più alta è la curva e migliore è la run.

L'obiettivo di questi ulteriori esperimenti era sempre quello di scoprire, per ogni combinazione di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ , quanto bene funzionasse il nostro sistema.

All'inizio sono stati settate le manopole tutte allo stesso valore ( $\alpha = \beta = \gamma = \delta = 1/4$ ), con lo scopo di avere un punto di partenza per osservare come si comporta il sistema bilanciando i valori delle

---

<sup>14</sup> Ricordiamo che, come descritto nel Paragrafo 3.1, ogni studente ha scelto 5 opere secondo i suoi gusti. Queste sono le opere che fanno parte del dataset di riferimento. Ognuna di esse è stata classificata per corrente e genere.



manopole. Successivamente si è provato a togliere  $\alpha$  ( $\alpha = 0, \beta = \gamma = \delta = 1/3$ ) al fine di vedere se, anche in questo esperimento, l'apporto dato dal titolo sarebbe stato insignificante.

Come prima si è tentato di vedere cosa succederebbe considerando solo le keywords ( $\alpha = \beta = \delta = 0, \gamma = 1$ ) o solo la descrizione ( $\alpha = \beta = \gamma = 0, \delta = 1$ ), tentando anche di bilanciare questi valori.

Infine si è provato sperimentalmente ad aumentare o diminuire ciascun valore, cercando di individuare la combinazione migliore di  $\alpha, \beta, \gamma$  e  $\delta$ .

Dall'esperimento precedente si è osservato che le keyword e la descrizione, se combinati, funzionano bene agli scopi del reperimento. Inoltre, come si può immaginare, ogni autore, generalmente, fa parte della stessa corrente e ha realizzato opere solitamente dello stesso genere. Sfruttando queste informazioni, sicuramente utili, e combinandole, si è provato a dare un valore alto all'autore ( $\beta$ ), basso al titolo ( $\alpha$ ) e medio alle keyword ( $\gamma$ ) e alla descrizione ( $\delta$ ).

Ulteriormente, anche in questo caso, sono state fatte ulteriori prove, sfruttando i risultati dell'esperimento precedente, cioè si è tentato di proporzionare i valori di  $\beta, \gamma$  e  $\delta$  rispetto alla quantità di termini contenuti, rispettivamente, nell'autore, tra le keyword e nella descrizione, con lo scopo di vedere se, anche in questo caso, i risultati miglioravano sensibilmente.

Alla fine di ogni prova è stato calcolato il valor medio dei risultati di ciascuna metrica di valutazione calcolata per ciascuno degli insiemi delle opere restituite. I risultati sono stati riportati:

- in **Tab.4** e **Tab.5** (graficamente nelle **Fig.29** e **Fig.30**) per l'esperimento con le opere della stessa **corrente** come insieme degli elementi rilevanti;
- in **Tab.6** e **Tab.7** (graficamente nelle **Fig.31** e **Fig.32**) per l'esperimento con le opere dello stesso **genere** come insieme degli elementi rilevanti;
- in **Tab.8** e **Tab.9** (graficamente nelle **Fig.33** e **Fig.34**) per l'esperimento con le opere scelte dallo stesso **studente** come insieme degli elementi rilevanti;

Come si è osservato dai risultati ottenuti, valgono praticamente le stesse considerazioni per tutti e tre gli esperimenti.

Notiamo immediatamente come i valori ottenuti in termini di Recall e Precision siano discretamente buoni sia per la corrente che per il genere (considerando il fatto che la valutazione viene fatta in base a delle semplici evidenze). Per quanto riguarda lo studente, c'è un errore di fondo che ci ha portato ad avere valori di Richiamo e Precisione bassi: noi sappiamo quali sono le 5 opere che ha scelto, quindi sappiamo che gli piacciono, ma non sappiamo se le altre gli piacciono o no.

Osservando la Precisione, essa, come previsto, tende a scendere gradualmente all'aumentare di  $k$ , dimostrando come ci sia una certa tendenza a piazzare gli item rilevanti in posizioni alte di rank. Osservando, in particolar modo, la  $p@1$  per ciascuno dei tre esperimenti, essa è molto alta, questo significa che, in molti casi, il primo elemento suggerito è corretto.

Se osserviamo le prime due prove effettuate, con  $\alpha = \beta = \gamma = \delta = 1/4$  la prima e con  $\alpha = 0, \beta = \gamma = \delta = 1/3$  la seconda, otteniamo dei risultati molto simili per ogni misura. Questo ci ha portato ad affermare definitivamente, anche in base ai risultati ottenuti nel primo esperimento, la scarsa utilità del titolo nella funzione di similarità. Da qui in poi sono state fatte esclusivamente prove con  $\alpha = 0$  (fatta eccezione per una).

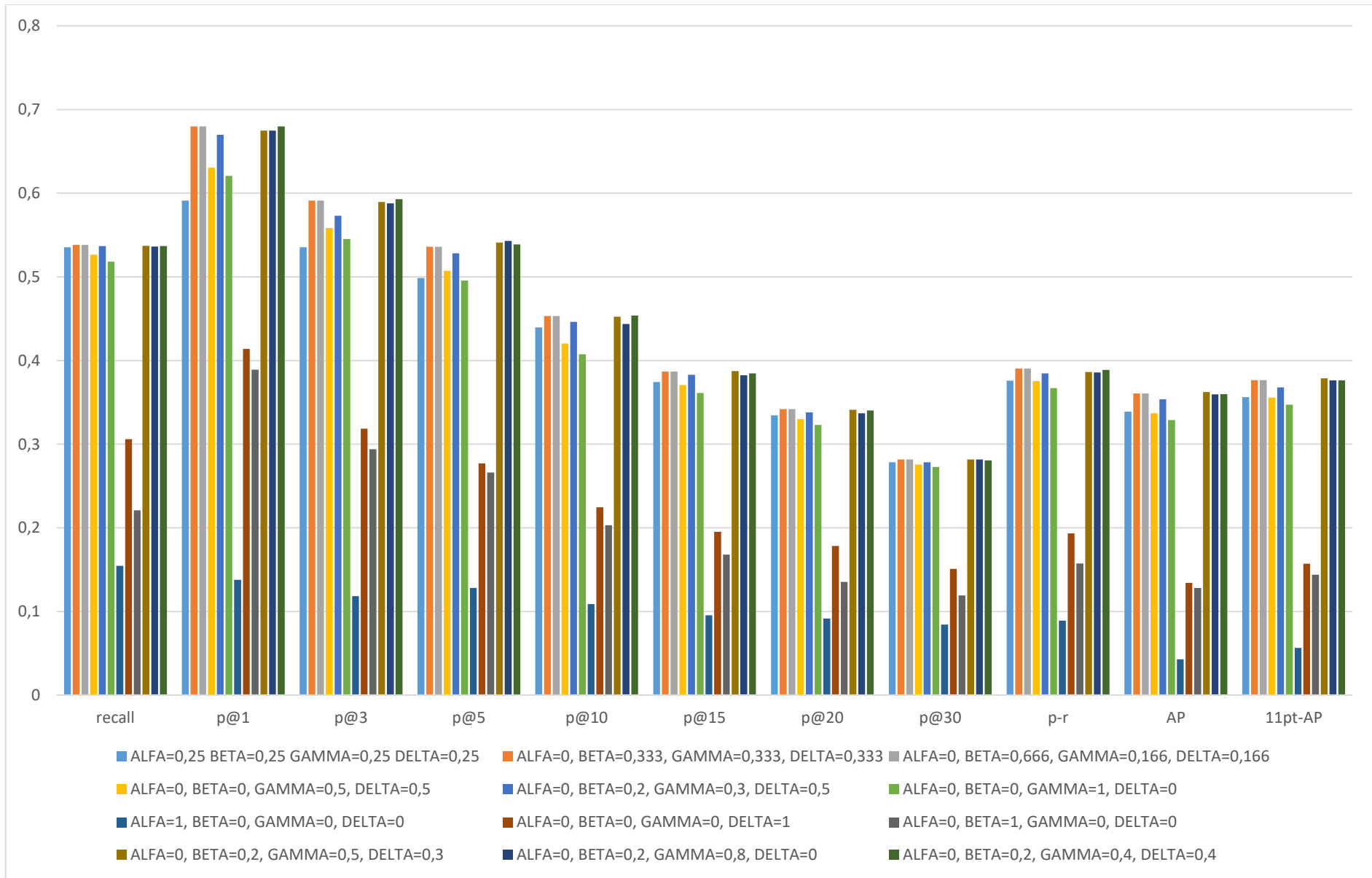
Nella terza prova si è tentato di dare un valore alto all'autore ( $\beta = 2/3$ ), medio alle keyword ( $\gamma = 1/6$ ) e alla descrizione ( $\delta = 1/6$ ), per i motivi sopracitati. Ciò che si ottiene è che, in ciascuno dei tre esperimenti, i risultati di tutte le misure di valutazione sono uguali a quelli della seconda prova ( $\alpha = 0, \beta = \gamma = \delta = 1/3$ ). Come si spiega questo fatto? Nel calcolo della funzione di similarità, se le opere hanno lo stesso autore in comune, il risultato aumenta di 0.333 nella seconda prova e 0.666 nella terza. Queste opere, nella seconda prova, vengono spinte molto più in alto nella rank list rispetto a

quelle che non condividono lo stesso autore (+0 sul valore della funzione di similarità), fino ad arrivare ai primi posti. Questo accade anche nella terza prova, dove queste opere vengono spinte ancora più in alto nella rank list, ma erano già ai primi posti! Vengono spinte più in alto coloro che erano già le più alte, quindi la rank list non cambia.

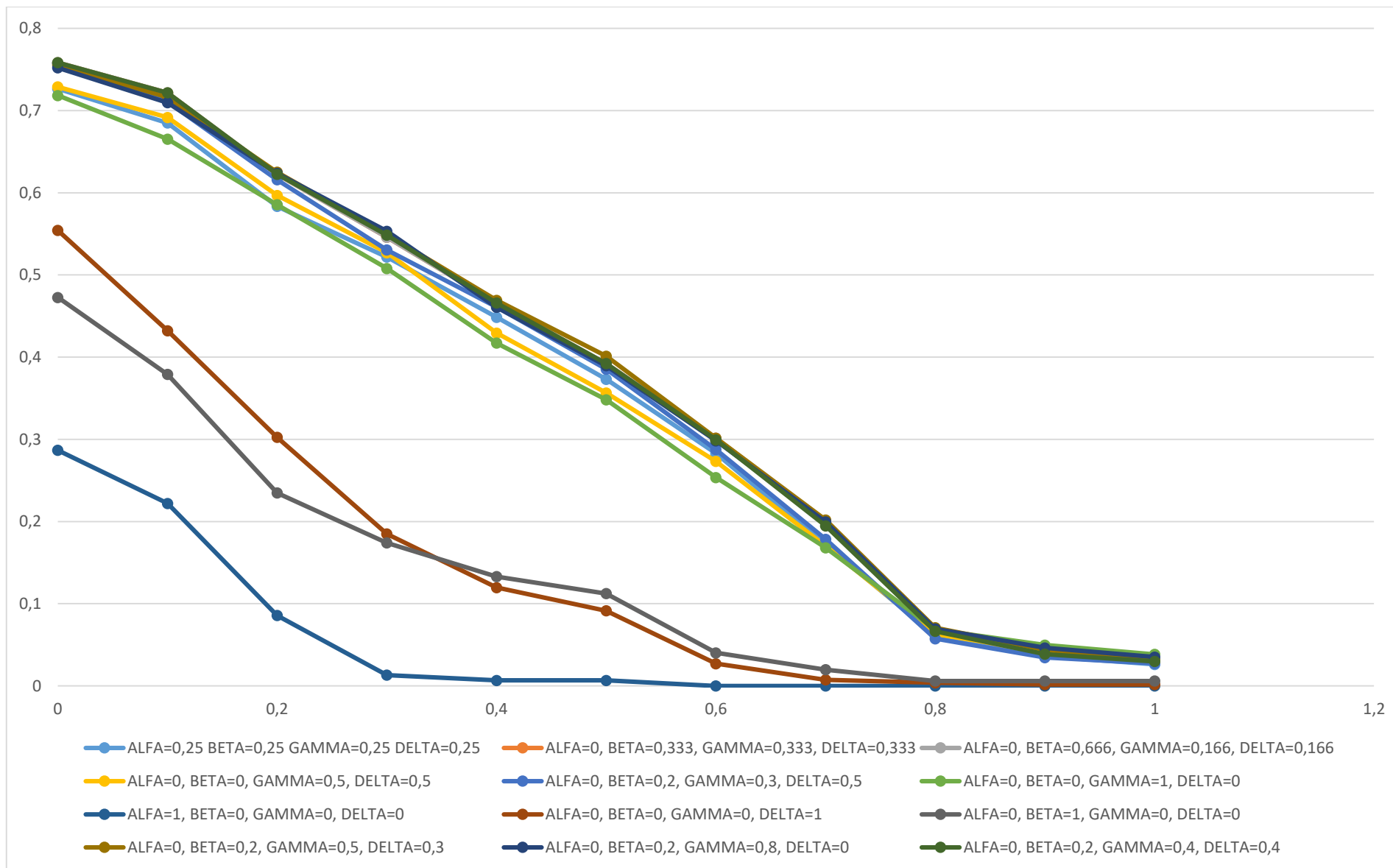
Nella quarta prova si è tentato di eliminare l'autore dal calcolo della funzione di similarità, bilanciando l'effetto delle keywords e della descrizione ( $\beta = 0, \gamma = \delta = 0.5$ ). In questo caso tutti i valori delle misure si abbassano nell'esperimento della corrente (rimangono piuttosto simili negli esperimenti del genere e dello studente). L'autore, nel suo piccolo, sembrerebbe essere utile per il calcolo della funzione di similarità.

Tentando di bilanciare i valori di  $\beta, \gamma$  e  $\delta$  rispetto alla quantità di termini contenuti, rispettivamente, nell'autore, tra le keywords e nella descrizione (quinta prova), si nota che, a differenza del primo esperimento, questa volta i risultati delle misure non migliorano. Provando a fare ulteriori prove nell'intorno di questi valori di  $\beta, \gamma$  e  $\delta$  (decima, undicesima e dodicesima prova), anche in questo caso, i risultati non cambiano di molto.

Osservando anche le Interpolated R-P Curves, si può notare come l'uso esclusivo delle keywords funzioni nettamente meglio dell'uso esclusivo della descrizione, dell'autore e del titolo, esattamente come era stato previsto. Addirittura, se vengono considerate esclusivamente le keywords nel calcolo della funzione di similarità, si ottengono valori delle misure di valutazione che sono molto simili a quelli più alti calcolati nelle altre prove.



**Fig.29:** Risultati della valutazione dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
Insieme degli elementi rilevanti: opere della stessa corrente



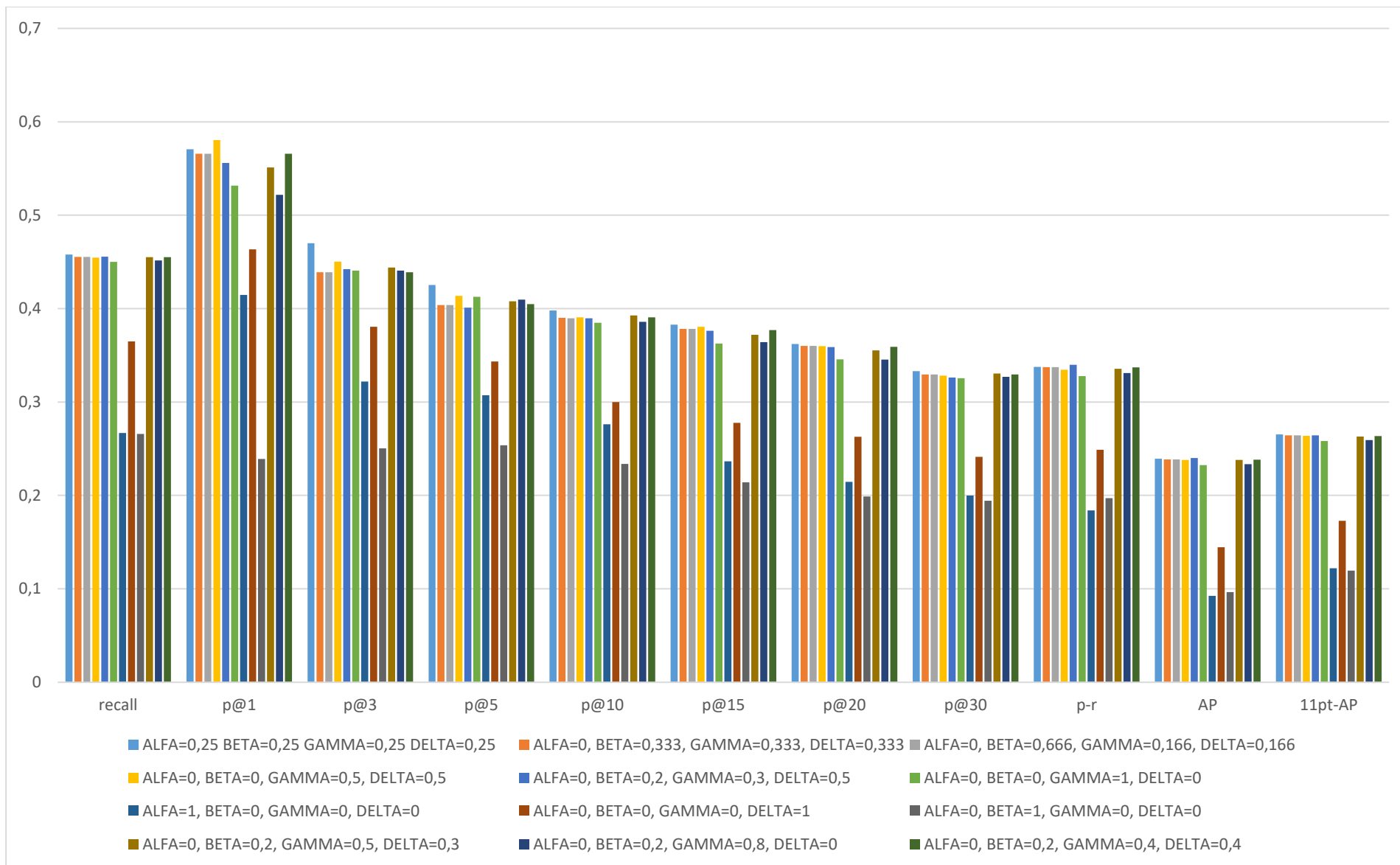
**Fig.30:** Curve Richiamo-Precisione Interpolate dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
 Insieme degli elementi rilevanti: opere della stessa **corrente**

	Recall	p@1	p@3	p@5	p@10	p@15	p@20	p@30	p-r	AP	11pt-AP
$\alpha=0,25, \beta=0,25 \gamma=0,25 \delta=0,25$	0,535396956	0,591133	0,535303777	0,498522167	0,439409	0,374384236	0,334482759	0,278325123	0,375947755	0,338734701	0,356038417
$\alpha=0, \beta=0,333, \gamma=0,333, \delta=0,333$	0,538223278	0,679803	0,591133005	0,535960591	0,453202	0,386863711	0,341871921	0,281609195	0,390364115	0,360519198	0,376401055
$\alpha=0, \beta=0,666, \gamma=0,166, \delta=0,166$	0,538223278	0,679803	0,591133005	0,535960591	0,453202	0,386863711	0,341871921	0,281609195	0,390364115	0,360519198	0,376401055
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,526513996	0,630542	0,558292282	0,507389163	0,420197	0,370771757	0,330049261	0,275533662	0,375481208	0,336931562	0,355593655
$\alpha=0, \beta=0,2, \gamma=0,3, \delta=0,5$	0,536861105	0,669951	0,573070608	0,528078818	0,446305	0,382922824	0,337931034	0,278325123	0,384667813	0,353582392	0,367897973
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,518196953	0,62069	0,545155993	0,495566502	0,407389	0,361247947	0,322906404	0,2727422	0,367116272	0,328797442	0,34734867
$\alpha=1, \beta=0, \gamma=0, \delta=0$	0,154482693	0,137931	0,118226601	0,128078818	0,108867	0,095566502	0,091625616	0,084400657	0,088989561	0,042645042	0,056469229
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,306038392	0,413793	0,318555008	0,276847291	0,224631	0,195402299	0,178325123	0,15090312	0,19321194	0,134240143	0,156978091
$\alpha=0, \beta=1, \gamma=0, \delta=0$	0,220789676	0,389163	0,293924466	0,266009852	0,202956	0,167816092	0,135221675	0,119047619	0,157252652	0,128050298	0,14399512
$\alpha=0, \beta=0,2, \gamma=0,5, \delta=0,3$	0,537026611	0,674877	0,589490969	0,5408867	0,452217	0,387520525	0,341133005	0,281609195	0,386388693	0,362419021	0,378664908
$\alpha=0, \beta=0,2, \gamma=0,8, \delta=0$	0,536219277	0,674877	0,587848933	0,542857143	0,443842	0,38226601	0,336945813	0,281773399	0,385683986	0,35950286	0,37615704
$\alpha=0, \beta=0,2, \gamma=0,4, \delta=0,4$	0,536786496	0,679803	0,592775041	0,538916256	0,453695	0,38456486	0,340394089	0,28045977	0,388770949	0,359736503	0,376248392

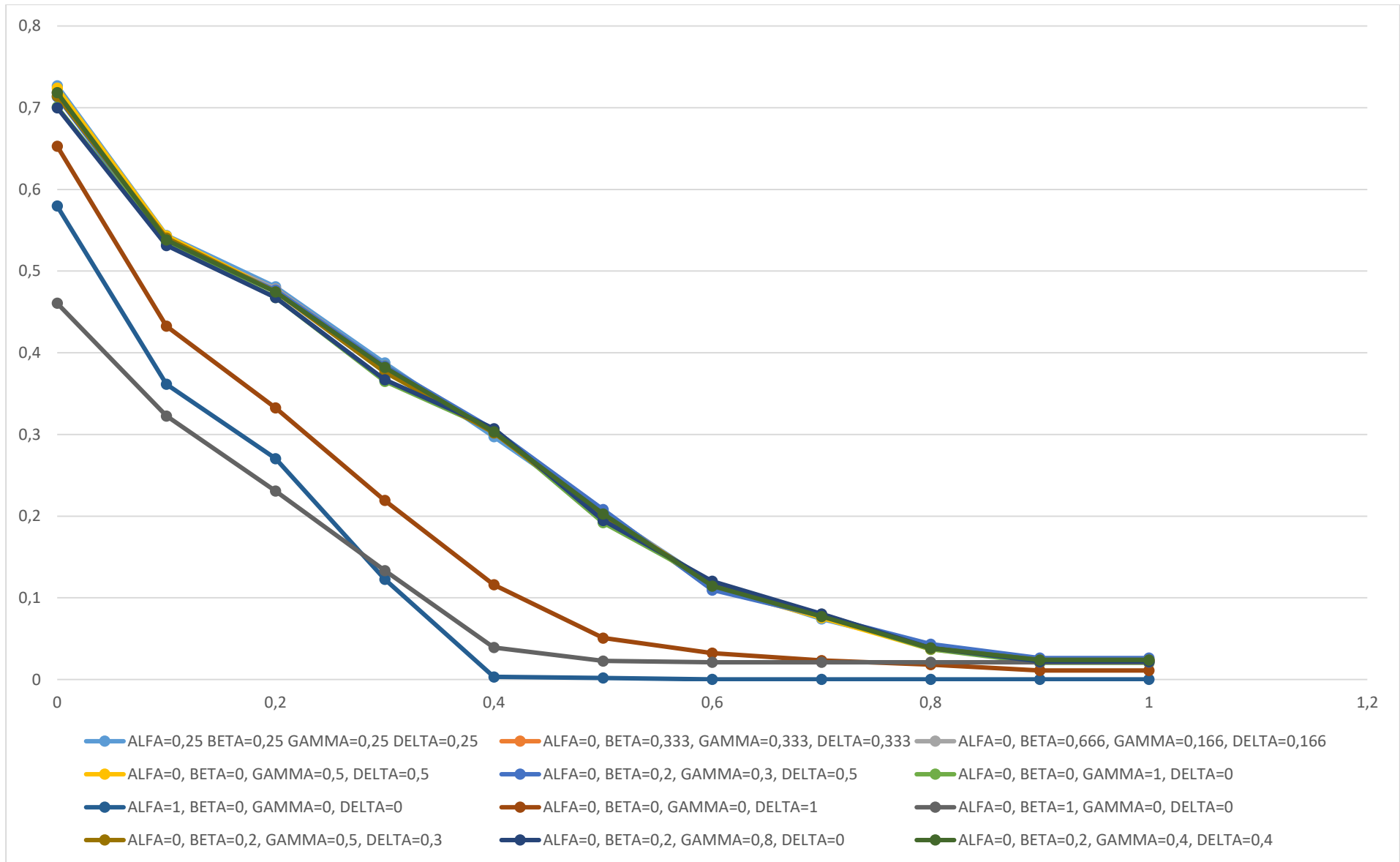
**Tab.4:** Risultati numerici della valutazione dell'esperimento 2 per diverse combinazioni di  $\alpha, \beta, \gamma$  e  $\delta$ .  
Insieme degli elementi rilevanti: opere della stessa **corrente**

	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$\alpha=0,25, \beta=0,25 \gamma=0,25 \delta=0,25$	0,726527261	0,685015	0,583526975	0,52183813	0,448569	0,373111835	0,282913083	0,171615686	0,061027116	0,036037985	0,026240184
$\alpha=0, \beta=0,333, \gamma=0,333, \delta=0,333$	0,758217818	0,721536	0,622277277	0,545943818	0,465719	0,392262862	0,301409547	0,199078875	0,066146868	0,038387213	0,029432601
$\alpha=0, \beta=0,666, \gamma=0,166, \delta=0,166$	0,758217818	0,721536	0,622277277	0,545943818	0,465719	0,392262862	0,301409547	0,199078875	0,066146868	0,038387213	0,029432601
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,728930272	0,691734	0,59682058	0,527096997	0,429385	0,356536311	0,27327695	0,170176516	0,062896789	0,041815324	0,032860711
$\alpha=0, \beta=0,2, \gamma=0,3, \delta=0,5$	0,757711975	0,710858	0,615868188	0,530408762	0,46102	0,385536196	0,287635981	0,178472439	0,057485379	0,034481247	0,027399067
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,718376034	0,665375	0,585454421	0,507992864	0,417249	0,348268699	0,253564203	0,168172993	0,068344179	0,049691297	0,038346876
$\alpha=1, \beta=0, \gamma=0, \delta=0$	0,286680468	0,222029	0,085774385	0,013213423	0,006732	0,006732348	0	0	0	0	0
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,554206024	0,432256	0,30243851	0,184968974	0,119724	0,091490299	0,027096191	0,007494722	0,003800141	0,001642036	0,001642036
$\alpha=0, \beta=1, \gamma=0, \delta=0$	0,472662924	0,378976	0,234816946	0,174047462	0,133005	0,112315271	0,040229885	0,019704433	0,006062903	0,006062903	0,006062903
$\alpha=0, \beta=0,2, \gamma=0,5, \delta=0,3$	0,75531206	0,716133	0,625137015	0,549284199	0,46926	0,401222486	0,301521944	0,201986423	0,070758308	0,044012046	0,030685952
$\alpha=0, \beta=0,2, \gamma=0,8, \delta=0$	0,75213952	0,709711	0,623598265	0,553193633	0,460826	0,389686114	0,29831869	0,199517654	0,069333757	0,046312286	0,035089877
$\alpha=0, \beta=0,2, \gamma=0,4, \delta=0,4$	0,758391796	0,72171	0,622615459	0,54857694	0,465922	0,392292242	0,299412617	0,194859532	0,066475276	0,03871562	0,029761008

**Tab.5:** Curve Richiamo-Precisione Interpolate dell'esperimento 2 per diverse combinazioni  $\alpha, \beta, \gamma$  e  $\delta$ : risultati numerici.  
Insieme degli elementi rilevanti: opere della stessa **corrente**



**Fig.31:** Risultati della valutazione dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
Insieme degli elementi rilevanti: opere dello stesso genere



**Fig.32:** Curve Richiamo-Precisione Interpolate dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
 Insieme degli elementi rilevanti: opere dello stesso **genere**

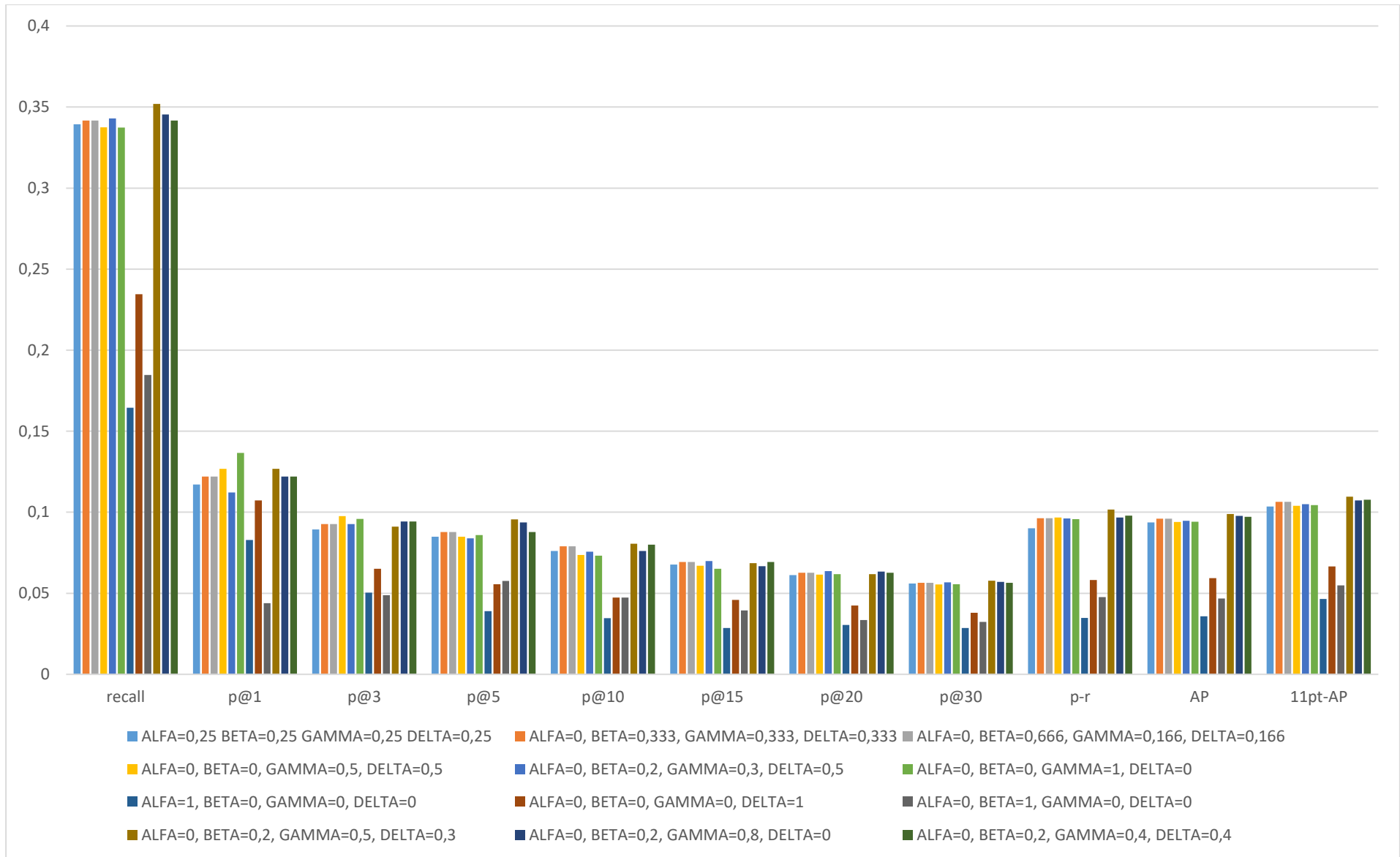
	Recall	p@1	p@3	p@5	p@10	p@15	p@20	p@30	p-r	AP	11pt-AP
$\alpha=0,25, \beta=0,25 \gamma=0,25 \delta=0,25$	0,45800549	0,570732	0,469918699	0,425365854	0,398049	0,382764228	0,362195122	0,33300813	0,337713547	0,239246873	0,265274544
$\alpha=0, \beta=0,333, \gamma=0,333, \delta=0,333$	0,45538983	0,565854	0,43902439	0,403902439	0,390244	0,378211382	0,36	0,329593496	0,337439596	0,238600506	0,264212908
$\alpha=0, \beta=0,666, \gamma=0,166, \delta=0,166$	0,45538983	0,565854	0,43902439	0,403902439	0,389756	0,378211382	0,36	0,329593496	0,337439596	0,238597735	0,264212908
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,45469829	0,580488	0,450406504	0,413658537	0,390732	0,380487805	0,359756098	0,328292683	0,334539119	0,238075315	0,263877616
$\alpha=0, \beta=0,2, \gamma=0,3, \delta=0,5$	0,45552564	0,556098	0,442276423	0,40097561	0,389756	0,376260163	0,358780488	0,326341463	0,339788389	0,239929954	0,264370856
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,450138669	0,531707	0,440650407	0,412682927	0,384878	0,362601626	0,345609756	0,325365854	0,327660754	0,232376025	0,258318319
$\alpha=1, \beta=0, \gamma=0, \delta=0$	0,266747792	0,414634	0,32195122	0,307317073	0,276098	0,236422764	0,214634146	0,199837398	0,183982728	0,092377705	0,121922702
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,364823496	0,463415	0,380487805	0,343414634	0,3	0,277723577	0,262682927	0,241300813	0,248939394	0,144543351	0,172794587
$\alpha=0, \beta=1, \gamma=0, \delta=0$	0,265911115	0,239024	0,250406504	0,253658537	0,233659	0,21398374	0,198780488	0,194308943	0,19708175	0,096397043	0,119578978
$\alpha=0, \beta=0,2, \gamma=0,5, \delta=0,3$	0,45509837	0,55122	0,443902439	0,407804878	0,392683	0,37203252	0,355365854	0,330406504	0,335538574	0,237994018	0,262973419
$\alpha=0, \beta=0,2, \gamma=0,8, \delta=0$	0,45161515	0,521951	0,440650407	0,409756098	0,385854	0,364227642	0,345365854	0,32699187	0,331147274	0,233449758	0,259295039
$\alpha=0, \beta=0,2, \gamma=0,4, \delta=0,4$	0,455173028	0,565854	0,43902439	0,404878049	0,390732	0,376910569	0,35902439	0,329430894	0,337222794	0,238231802	0,26346314

**Tab.6:** Risultati numerici della valutazione dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
Insieme degli elementi rilevanti: opere dello stesso genere

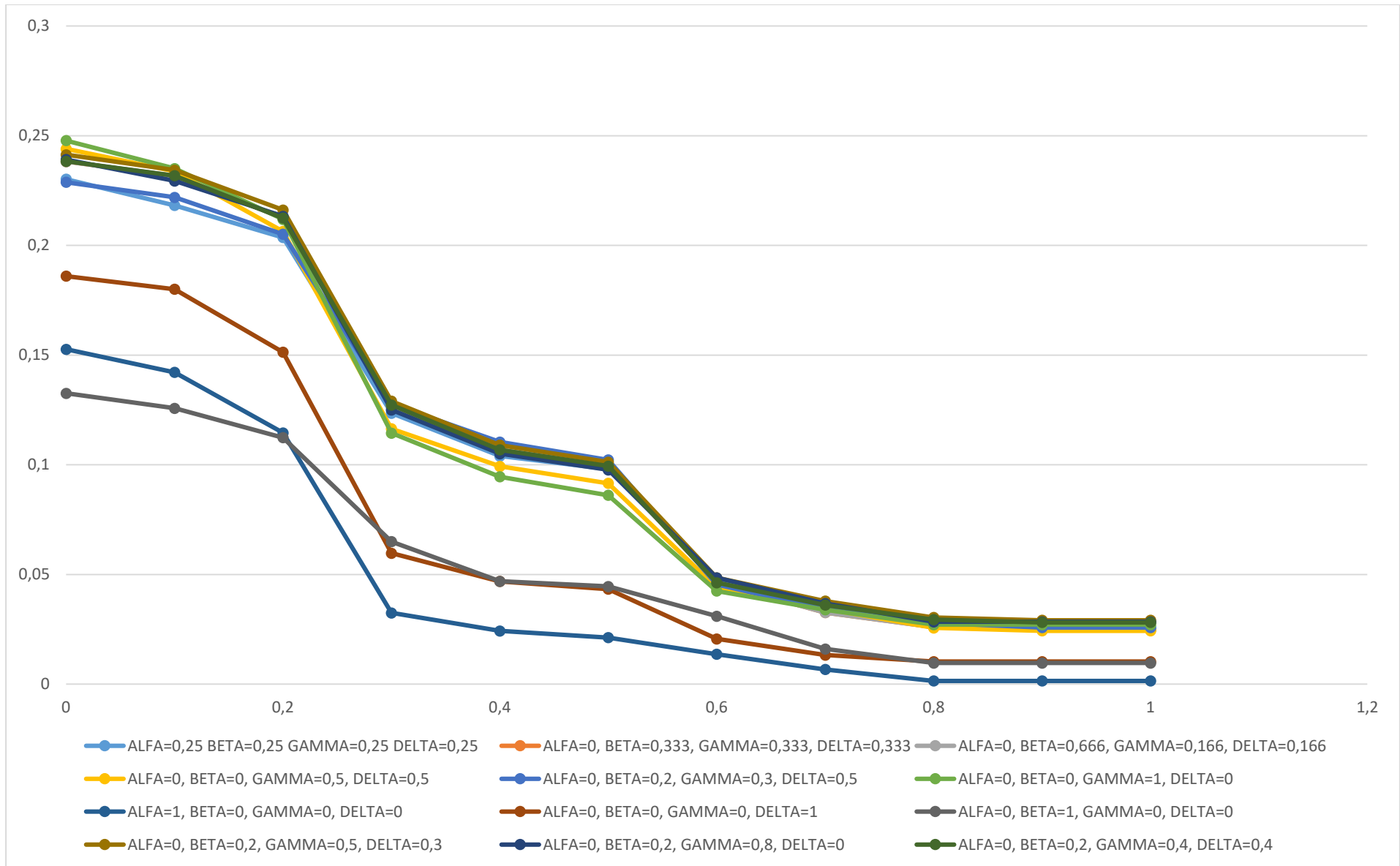
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$\alpha=0,25, \beta=0,25 \gamma=0,25 \delta=0,25$	0,726990882	0,543751	0,480785682	0,387658602	0,297257	0,205030885	0,116909678	0,074192112	0,040907695	0,022268054	0,022268054
$\alpha=0, \beta=0,333, \gamma=0,333, \delta=0,333$	0,718659884	0,538228	0,478054104	0,383447701	0,304101	0,205550959	0,114459116	0,077248703	0,038726657	0,023932632	0,023932632
$\alpha=0, \beta=0,666, \gamma=0,166, \delta=0,166$	0,718659884	0,538228	0,478054104	0,383447701	0,304101	0,205550959	0,114459116	0,077248703	0,038726657	0,023932632	0,023932632
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,724201148	0,543272	0,475238574	0,384087247	0,303419	0,202149329	0,112898537	0,075674581	0,037100641	0,022306616	0,022306616
$\alpha=0, \beta=0,2, \gamma=0,3, \delta=0,5$	0,713472114	0,535927	0,476824975	0,384011536	0,30595	0,20808785	0,109686356	0,078032147	0,043403722	0,026341463	0,026341463
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,701069029	0,534362	0,467738356	0,365190257	0,305173	0,192065375	0,118823461	0,078835891	0,037179264	0,02053278	0,02053278
$\alpha=1, \beta=0, \gamma=0, \delta=0$	0,57972009	0,361558	0,270499541	0,122812032	0,003225	0,001979772	0,000271003	0,000271003	0,000271003	0,000271003	0,000271003
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,653092872	0,432602	0,332571922	0,219337755	0,116101	0,050843268	0,032343473	0,023297183	0,018365439	0,01109281	0,01109281
$\alpha=0, \beta=1, \gamma=0, \delta=0$	0,460770726	0,322735	0,23078203	0,133398522	0,039228	0,022764228	0,021138211	0,021138211	0,021138211	0,021138211	0,021138211
$\alpha=0, \beta=0,2, \gamma=0,5, \delta=0,3$	0,7143535	0,5406	0,474603282	0,376350911	0,302018	0,200688278	0,118151422	0,07992038	0,038272887	0,023874361	0,023874361
$\alpha=0, \beta=0,2, \gamma=0,8, \delta=0$	0,699997649	0,531383	0,467695077	0,367193461	0,307263	0,194825005	0,120410546	0,080408149	0,038751522	0,022158796	0,022158796
$\alpha=0, \beta=0,2, \gamma=0,4, \delta=0,4$	0,718655988	0,538208	0,47457647	0,382238921	0,303277	0,202838311	0,114459116	0,077248703	0,038726657	0,023932632	0,023932632

**Tab.7:** Curve Richiamo-Precisione Interpolate dell'esperimento 2 per diverse combinazioni  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ : risultati numerici.  
Insieme degli elementi rilevanti: opere dello stesso genere





**Fig.33:** Risultati della valutazione dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
Insieme degli elementi rilevanti: opere scelte dallo stesso **studente**



**Fig.34:** Curve Richiamo-Precisione Interpolate dell'esperimento 2 per diverse combinazioni di  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ .  
 Insieme degli elementi rilevanti: opere scelte dallo stesso **studente**

	Recall	p@1	p@3	p@5	p@10	p@15	p@20	p@30	p-r	AP	11pt-AP
$\alpha=0,25, \beta=0,25, \gamma=0,25, \delta=0,25$	0,339318988	0,117073	0,089430894	0,084878049	0,076098	0,067642276	0,061219512	0,055934959	0,090134311	0,093646389	0,103456676
$\alpha=0, \beta=0,333, \gamma=0,333, \delta=0,333$	0,341723169	0,121951	0,092682927	0,087804878	0,079024	0,069268293	0,062682927	0,056422764	0,096236624	0,096034937	0,106360953
$\alpha=0, \beta=0,666, \gamma=0,166, \delta=0,166$	0,341723169	0,121951	0,092682927	0,087804878	0,079024	0,069268293	0,062682927	0,056422764	0,096236624	0,096034937	0,106360953
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,337539872	0,126829	0,097560976	0,084878049	0,073659	0,06699187	0,061463415	0,055447154	0,096749584	0,093941803	0,103997895
$\alpha=0, \beta=0,2, \gamma=0,3, \delta=0,5$	0,342915044	0,112195	0,092682927	0,083902439	0,07561	0,069918699	0,063658537	0,056747967	0,096176704	0,094749321	0,105014336
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,33740445	0,136585	0,095934959	0,085853659	0,073171	0,06504065	0,061707317	0,055609756	0,095693808	0,094134244	0,104322982
$\alpha=1, \beta=0, \gamma=0, \delta=0$	0,164419516	0,082927	0,050406504	0,03902439	0,034634	0,028617886	0,030487805	0,028617886	0,034768916	0,035842181	0,046538821
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,234457587	0,107317	0,06504065	0,055609756	0,047317	0,045853659	0,042439024	0,037886179	0,05823306	0,059279743	0,066544593
$\alpha=0, \beta=1, \gamma=0, \delta=0$	0,184619767	0,043902	0,048780488	0,057560976	0,047317	0,039349593	0,033414634	0,032357724	0,047592711	0,046830649	0,054816772
$\alpha=0, \beta=0,2, \gamma=0,5, \delta=0,3$	0,351959328	0,126829	0,091056911	0,095609756	0,080488	0,068617886	0,061707317	0,057723577	0,101608284	0,098956926	0,109619603
$\alpha=0, \beta=0,2, \gamma=0,8, \delta=0$	0,34550004	0,121951	0,094308943	0,093658537	0,076098	0,066666667	0,063414634	0,057073171	0,096753543	0,097705047	0,107269463
$\alpha=0, \beta=0,2, \gamma=0,4, \delta=0,4$	0,341723169	0,121951	0,094308943	0,087804878	0,08	0,069268293	0,062682927	0,056422764	0,09786264	0,097228437	0,107642646

**Tab.8:** Risultati numerici della valutazione dell'esperimento 2 per diverse combinazioni di  $\alpha, \beta, \gamma$  e  $\delta$ .  
Insieme degli elementi rilevanti: opere scelte dallo stesso **studente**

	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$\alpha=0,25, \beta=0,25, \gamma=0,25, \delta=0,25$	0,230174507	0,218244	0,203607301	0,123533835	0,103979	0,098021921	0,048126102	0,035608651	0,026389087	0,025169575	0,025169575
$\alpha=0, \beta=0,333, \gamma=0,333, \delta=0,333$	0,238255361	0,231693	0,212504275	0,127123003	0,106745	0,099332967	0,046216848	0,032647954	0,02601809	0,024717277	0,024717277
$\alpha=0, \beta=0,666, \gamma=0,166, \delta=0,166$	0,238255361	0,231693	0,212504275	0,127123003	0,106745	0,099332967	0,046216848	0,032647954	0,02601809	0,024717277	0,024717277
$\alpha=0, \beta=0, \gamma=0,5, \delta=0,5$	0,244021564	0,23378	0,206380352	0,116394355	0,099264	0,091530437	0,044007496	0,034341042	0,025619834	0,024319021	0,024319021
$\alpha=0, \beta=0,2, \gamma=0,3, \delta=0,5$	0,228820263	0,221925	0,205109164	0,127723821	0,110401	0,102212433	0,045385773	0,034058447	0,027956552	0,025782779	0,025782779
$\alpha=0, \beta=0, \gamma=1, \delta=0$	0,247805362	0,235047	0,211869319	0,114446309	0,094524	0,086115665	0,042477847	0,033900416	0,027122263	0,027122263	0,027122263
$\alpha=1, \beta=0, \gamma=0, \delta=0$	0,152663308	0,142131	0,114530162	0,032468335	0,024267	0,021188525	0,013658137	0,00667427	0,001448833	0,001448833	0,001448833
$\alpha=0, \beta=0, \gamma=0, \delta=1$	0,186025958	0,180044	0,15131799	0,059703706	0,046831	0,043302164	0,020638072	0,013274703	0,010284239	0,010284239	0,010284239
$\alpha=0, \beta=1, \gamma=0, \delta=0$	0,132563055	0,125762	0,112372828	0,064950249	0,046865	0,044533883	0,030992468	0,016040322	0,00963501	0,00963501	0,00963501
$\alpha=0, \beta=0,2, \gamma=0,5, \delta=0,3$	0,241340095	0,23431	0,216132589	0,128986758	0,108971	0,101193332	0,048468005	0,037841175	0,030391468	0,029090655	0,029090655
$\alpha=0, \beta=0,2, \gamma=0,8, \delta=0$	0,239146739	0,229407	0,21329206	0,124989752	0,105048	0,097689167	0,048474519	0,036741366	0,028391655	0,028391655	0,028391655
$\alpha=0, \beta=0,2, \gamma=0,4, \delta=0,4$	0,238331334	0,231769	0,21249314	0,127269987	0,106821	0,099409113	0,046216848	0,036062588	0,029432724	0,028131911	0,028131911

**Tab.9:** Curve Richiamo-Precisione Interpolate dell'esperimento 2 per diverse combinazioni  $\alpha, \beta, \gamma$  e  $\delta$ : risultati numerici.  
Insieme degli elementi rilevanti: opere scelte dallo stesso **studente**

Complessivamente, dai risultati ottenuti nei quattro esperimenti precedenti emergono dei fatti:

- il titolo è di scarsa utilità nel calcolo della funzione di similarità;
- l'autore, le keywords e la descrizione permettono di ottenere, se combinati, i risultati migliori nel reperimento;
- le keywords possono anche essere utilizzate da sole nel calcolo della funzione di similarità, in quanto, non solo sono puramente relative al contenuto dell'opera (ricordiamo la natura Content-Based del nostro sistema), ma sono uniformate e descrivono molto accuratamente le opere del dataset, a differenza della descrizione, permettendo di ottenere valori delle misure di valutazione molto vicini a quelli migliori di ogni prova.

Da questi fatti sembrerebbe che, mettendo insieme le informazioni su descrizione, autore e keywords, più termini due opere condividono in comune e più simili queste sono tra loro. Queste circostanze ci hanno portato, nella fase successiva della tesi (progettazione e sviluppo dell'algoritmo di raccomandazione), ad usare la funzione di similarità

$$\text{Dist}(O_1, O_2) = \alpha \cdot A(O_1, O_2) + \beta \cdot B(O_1, O_2) + \gamma \cdot C(O_1, O_2) + \delta \cdot D(O_1, O_2)$$

già descritta precedentemente, utilizzando i seguenti valori per le manopole:

$$\alpha = 0, \beta = \gamma = \delta = 1/3$$

siccome hanno dato i risultati migliori, in termini di Recall e Precision, nei vari esperimenti, e piccole variazioni di essi non portano cambiamenti significativi nelle prestazioni della funzione.

## 4.2 Algoritmo di raccomandazione: descrizione e valutazione

Descriviamo ora l'algoritmo di predizione/raccomandazione progettato e sviluppato durante la tesi, il quale ha come obiettivo quello di raccomandare opere d'arte in un potenziale contesto museale, cioè a degli utenti di un museo.

Quando un utente entra in un museo, se avesse a disposizione un'applicazione come TindArt, potrebbe ammassare il tempo in coda giocandoci, votando le opere presenti su questa app.

L'algoritmo sfrutta i dati provenienti da un'applicazione stile TindArt come conoscenza pregressa sull'utente. Queste informazioni sono utili all'algoritmo per capire i gusti dei clienti, cosa piace ad un utente e cosa non piace. I dati vengono poi analizzati con lo scopo di predire il voto che l'utente darebbe a ciascuna opera della collezione del museo in questione, il tutto per decidere se raccomandargliela o meno. Per raggiungere questo scopo l'algoritmo prende ciascuna opera del potenziale museo e calcola la probabilità che essa sia gradita o meno dall'utente, il tutto tramite la **funzione di score** (o di predizione) mostrata di seguito

$$\text{Score}(a; i) = \sum_{j \in V(i)} \left( r_{aj} \cdot \frac{1}{p(j)} \right)$$

L'algoritmo, inizialmente, al fine di sfruttare la conoscenza pregressa resa disponibile dall'utente "a" mediante l'app, raccoglie le opere del dataset dell'applicazione più simili all'opera "i" del museo (V(i)). Per svolgere questo compito, l'algoritmo principale utilizza la funzione di similarità descritta

nel paragrafo 4.1. Queste opere affini vengono restituite, ordinate mediante algoritmo Quicksort (Paragrafo 2.3), dalla più simile alla meno simile.

Data  $j \in V(i)$ ,  $r_{aj}$  è la votazione che l'utente "a" ha dato all'opera "j"<sup>15</sup>, mentre  $p(j)$  è la posizione che j occupa all'interno della lista ordinata delle opere più simili ad i.

Successivamente l'algoritmo calcola, tramite  $\text{Score}(a; i)$ , la media pesata delle votazioni che l'utente ha dato a queste opere più simili, con peso l'inverso della posizione all'interno della lista ordinata. In questo modo, giustamente, le votazioni date alle opere del dataset dell'app presenti in  $V(i)$  e in posizioni alte di rank, hanno più peso rispetto a quelle contenute in  $V(i)$  e in posizioni più basse di rank.

Per fare un esempio pratico, supponiamo che l'algoritmo raccolga, all'interno del dataset dell'applicazione, le seguenti opere affini ad "i", ordinate dalla più simile alla meno simile, con le corrispettive votazioni da parte dell'utente

OPERA	VOTO
1	LIKE
2	LIKE
3	NOPE
4	NOT VOTED
5	LIKE
6	NOPE
...	...

e supponiamo che il voto numerico corrispondente a LIKE sia 1, -1 per NOPE e 0 per NOT VOTED. Allora la funzione di score, come predizione di voto che l'utente darebbe all'opera "i" della collezione del museo, restituisce il seguente risultato:

$$\text{Score}(a; i) = 1 + \frac{1}{2} - \frac{1}{3} + \frac{1}{5} - \frac{1}{6} + \dots$$

Se il risultato della funzione di score sta sopra una certa soglia, allora l'opera "i" del museo viene raccomandata al cliente, altrimenti no. Questa soglia viene impostata in base alle necessità stabilite dal museo, a seconda di quante opere si vuole raccomandare e degli obiettivi che si vuole ottenere dalla raccomandazione.

Infine le opere raccomandate vengono restituite all'utente ordinate, mediante algoritmo Quicksort, da quella con valore della funzione di score più alto a quella con valore più basso

Come già anticipato nell'introduzione al capitolo, per testare l'algoritmo è stato realizzato un software che lo implementa, tramite il quale è stato possibile verificare l'efficacia della raccomandazione, anche utilizzando la funzione di similarità progettata (descritta nel Paragrafo 4.1). Inizialmente il software acquisisce i dati delle opere d'arte del dataset di riferimento (presentato nel Paragrafo 3.1) e le preferenze degli utenti relative a ciascuna di esse. Per ogni opera si tiene memoria di tutti gli utenti che l'hanno votata positivamente (LIKE) e di coloro che, invece, le hanno dato un voto negativo (NOPE). Chi non fa parte di queste due liste significa che non ha espresso alcuna valutazione per l'opera. Come già è stato anticipato, tutti questi dati provengono dall'applicazione TindArt, descritta nel Paragrafo 3.2.

Una volta acquisite le preferenze degli utenti, il nostro software procede alla raccomandazione.

<sup>15</sup> Nel nostro caso 1 = LIKE, -1 = NOPE, 0 = NON VOTATA.

Al fine di effettuare la valutazione, come già anticipato nell'introduzione al capitolo, il software sviluppato partiziona i dati in due insiemi, **Training Set** (80% delle opere del dataset di riferimento e le corrispettive preferenze espresse dagli utenti per esse) e **Test Set** (il restante 20 % delle opere del dataset di riferimento e le corrispettive preferenze espresse dagli utenti per esse).

Per ogni utente il sistema sfrutta i dati relativi alle sue preferenze all'interno del Training Set come conoscenza pregressa, al fine di predire una possibile votazione che l'utente stesso darebbe alle opere del Test Set, delle quali si fa finta di non conoscere le preferenze.

La situazione viene schematizzata nella **Fig.35**.



**Fig.35:** Esempio grafico di predizione del voto di un utente generico ad un'opera del Test Set

Il software implementa sia la funzione di score che la funzione di similarità descritta nel Paragrafo 4.1.

Per ognuna delle opere del Test Set, inizialmente, viene utilizzata questa funzione di similarità, al fine di restituire le 30 opere<sup>16</sup> più simili, tutte e 30 ordinate dalla più affine alla meno affine tramite ordinamento Quicksort. Successivamente il software utilizza la funzione di score per determinare il voto stimato. Se questo voto è maggiore di una certa soglia allora l'opera del Test Set viene raccomandata all'utente, altrimenti no (anche le opere raccomandate vengono restituite all'utente ordinate, mediante ordinamento Quicksort, da quella con valore della funzione di score più alto a quella con valore più basso). Se un'opera viene consigliata ad un cliente e questo l'ha votata positivamente la raccomandazione è corretta, se l'ha votata negativamente la raccomandazione è sbagliata, se non l'ha votata non può essere espresso alcun giudizio. Come già osservato, infatti, la valutazione off-line prevede che le raccomandazioni calcolate dagli algoritmi debbano essere successivamente confrontate con le vere valutazioni memorizzate dell'utente alle opere del Test

<sup>16</sup> Valore settato sperimentalmente nello studio della funzione di similarità descritto nel Paragrafo 4.1 (30 sono circa il 10% delle opere).

Set, il tutto per misurare la qualità dell'algorithmo progettato nell'adempire al suo compito di raccomandazione.

Si è deciso, inoltre, di effettuare una **cross-validation**, ovvero si è deciso di suddividere il dataset di riferimento in 5 gruppi di egual numerosità (20% del dataset), utilizzando come Test Set iterativamente un gruppo alla volta (il Training Set viene rappresentato dal restante 80%). Ciò al fine di verificare la bontà dell'algorithmo di predizione utilizzato. Quindi, in sostanza sono state effettuate 5 prove, in ognuna delle quali è stata eseguita la valutazione utilizzando un Test Set differente.

#### PRIMA PROVA

Dataset di riferimento	
20% Test Set	80% Training set

#### SECONDA PROVA

Dataset di riferimento	
	20% Test Set

#### TERZA PROVA

Dataset di riferimento	
	20% Test Set

#### QUARTA PROVA

Dataset di riferimento	
	20% Test Set

#### QUINTA PROVA

Dataset di riferimento	
	20% Test Set

Come abbiamo detto, dato un utente e un'opera del Test Set, viene calcolata la funzione di score e, se il valore restituito è maggiore di una certa **soglia**, allora l'opera viene raccomandata. Ma, ai fini della tesi, a che valore si è deciso di impostare questa soglia? Si è scelto di lasciarsi guidare dai dati a disposizione. Essendo che la funzione di score viene calcolata in base ai voti dell'utente per le 30 opere più simili, e che il voto numerico corrispondente a LIKE è 1, -1 per NOPE e 0 per NON VOTATA, essa restituisce dei valori che vanno da 3.994 nel caso migliore (tutte le opere più simili sono state votate positivamente dall'utente) a -3.994 (tutte le opere più simili sono state votate negativamente dall'utente)<sup>17</sup>. A questo punto è stata formulata un'ipotesi: utilizzando lo 0 come soglia sul valore restituito dalla funzione di predizione, tutte le opere del Test Set per le quali la votazione stimata è maggiore di 0 vengono raccomandate, le altre no. In effetti questa soglia viene presa esattamente a metà e consente di non raccomandare né le opere per cui la votazione predetta è 0, né le opere con valore negativo della funzione di score. Lo 0 lo si ottiene solo in un caso, cioè quando l'utente non ha votato neanche una delle opere più simili, quindi non esiste la possibilità di affermare che l'opera del Test Set per la quale viene calcolata la funzione di predizione possa realmente piacere

<sup>17</sup> Se ci si pensa, nel caso in cui un utente abbia votato positivamente tutte le 30 opere più simili allora

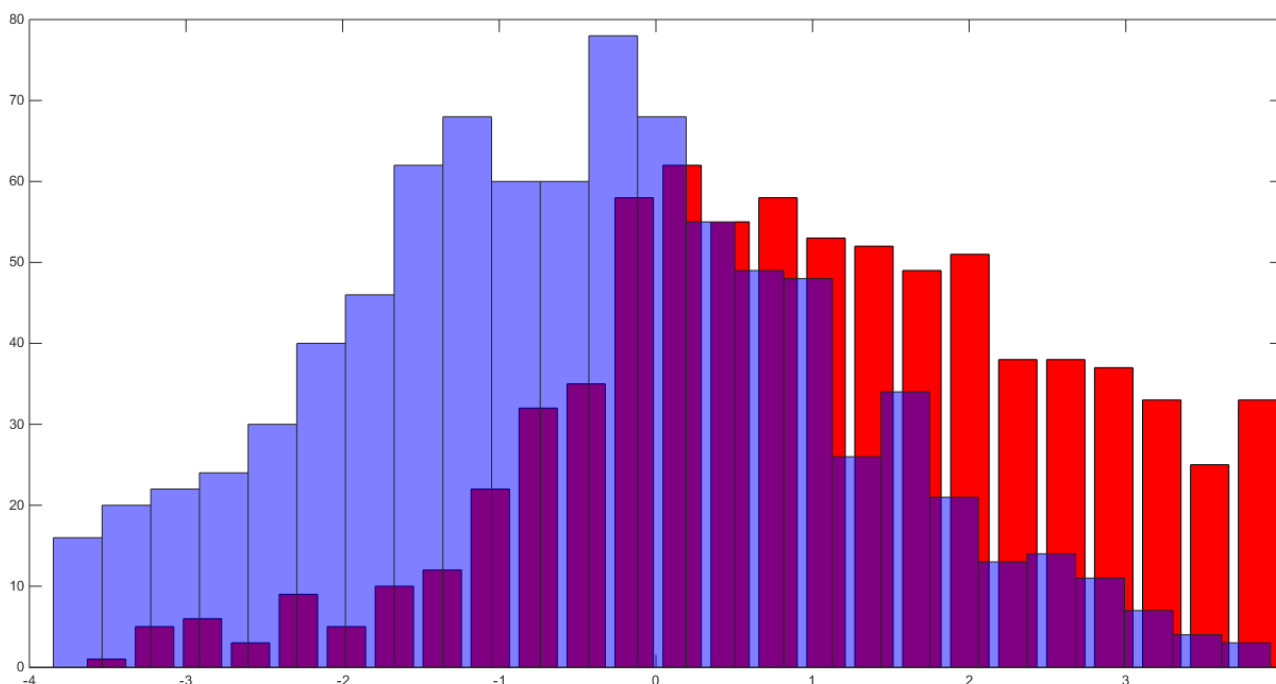
$Score(a; i) = 1 + 1/2 + 1/3 + \dots + 1/30 = 3.994.$

Nel caso in cui un utente abbia votato negativamente tutte le 30 opere più simili allora

$Score(a; i) = -1 - 1/2 - 1/3 - \dots - 1/30 = -3.994.$

all'utente in questione. Quest'ipotesi è stata successivamente confermata mediante degli esperimenti effettuati sul dataset di riferimento.

Nel primo di questi esperimenti il software è stato eseguito utilizzando come Test Set il primo 20% di opere del dataset, al fine di calcolare, per ogni utente, tutti i risultati della funzione di score su questo 20% di opere (come ricordiamo, i risultati possibili vanno da 3.994 a -3.994). Gli esiti sono stati riportati nell'istogramma in **Fig.36**, nel quale l'asse delle ascisse indica tutti i valori possibili della funzione di predizione. Ogni barra rossa indica il numero dei risultati della funzione di score che stanno in quell'intervallo e per cui la vera votazione memorizzata dell'utente è LIKE. Viceversa ogni barra blu indica il numero dei risultati della funzione di score che stanno in quell'intervallo e per cui la vera votazione memorizzata dell'utente è NOPE. Chiaramente, in questo istogramma sono compresi i risultati di tutti gli utenti.



**Fig.36:** Valori della funzione di score calcolati sul il primo 20% di opere del dataset come Test Set

Come si può notare dall'istogramma, la "campana dei LIKE" è, giustamente, più spostata verso destra, mentre la "campana dei NOPE" si concentra di più nella parte negativa del grafico, esattamente come ci si aspettava.

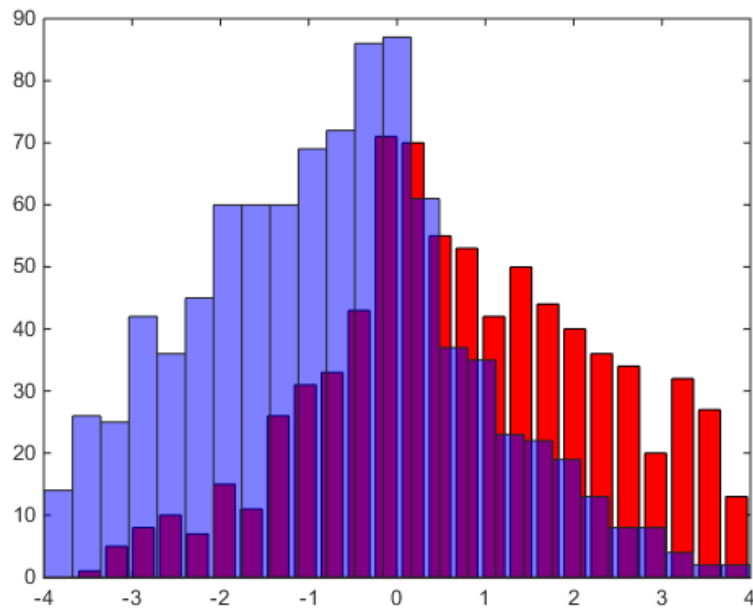
Come obiettivo per la tesi si è deciso di individuare una soglia che riuscisse a minimizzare i falsi positivi e i falsi negativi<sup>18</sup>. Grazie ad un foglio di calcolo è stato possibile effettuare un conteggio e il punto di spezzamento di questo primo esperimento è stato individuato in 0,333333333 (osservando l'istogramma, questo è circa il punto in cui si incrociano le due campane), che rappresenta la soglia per cui vengono minimizzati i falsi negativi e i falsi positivi.

Successivamente sono stati condotti altri quattro esperimenti analoghi, effettuati utilizzando come Test Set, rispettivamente, il secondo 20% delle opere del dataset di riferimento, il terzo 20%, il quarto e il quinto, esattamente come per la valutazione. I risultati per ciascuno di questi esperimenti sono riportati nelle **Fig.37-38-39-40**, congiuntamente alle soglie individuate per ciascuno di essi. Al fine di individuare una possibile soglia che andasse bene ai fini della tesi è stata calcolata la media dei cinque valori ottenuti in questi esperimenti:

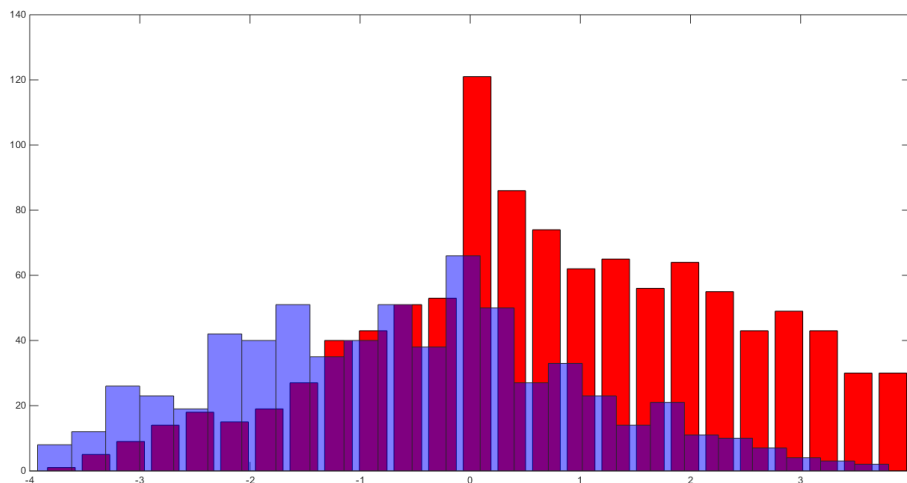
<sup>18</sup> Falsi positivi: opere che vengono raccomandate, ma per le quali la vera votazione dell'utente è NOPE.

Falsi negativi: opere che non vengono raccomandate, ma per le quali la vera votazione dell'utente è LIKE.

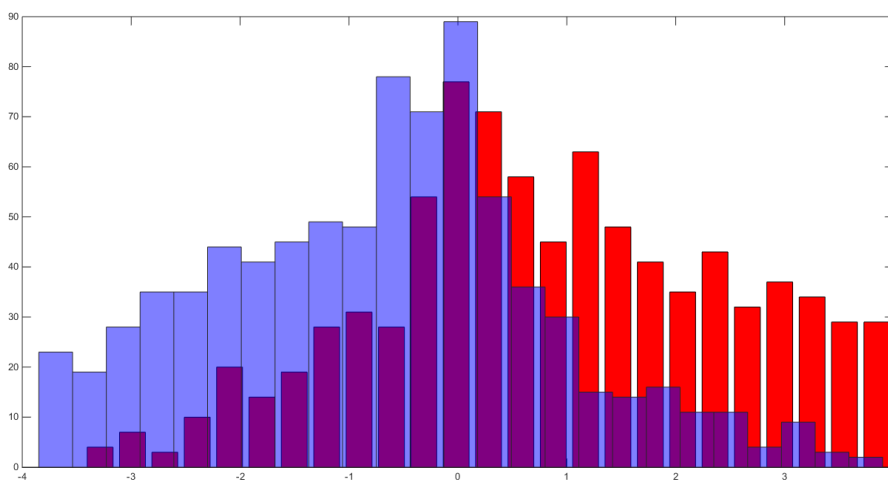




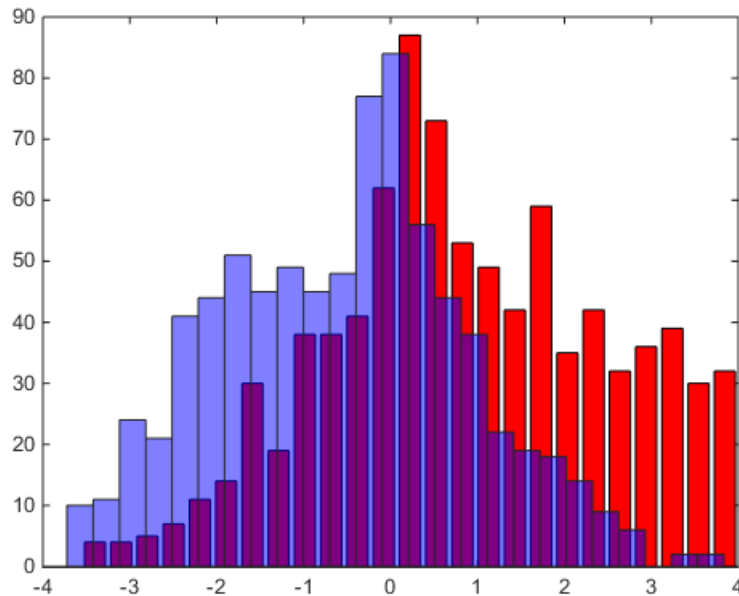
**Fig.37:** Valori della funzione di score calcolati sul il secondo 20% di opere del dataset come Test Set.  
Soglia individuata = 0,03333333



**Fig.38:** Valori della funzione di score calcolati sul il terzo 20% di opere del dataset come Test Set.  
Soglia individuata = -0,008



**Fig.39:** Valori della funzione di score calcolati sul il quarto 20% di opere del dataset come Test Set.  
Soglia individuata = -0,223



**Fig.40:** Valori della funzione di score calcolati sul il quinto 20% di opere del dataset come Test Set.  
Soglia individuata = 0,0465

$$\text{Threshold} = \frac{(0,333333333 + 0,033333333 - 0,008 - 0,223 + 0,0465)}{5} = 0,036433333$$

Come si può osservare, questa soglia calcolata usando i dati reali conferma l'ipotesi che era stata fatta, in quanto 0,036433333 è un punto di spezzamento molto vicino a zero, ma leggermente superiore, in modo tale da non raccomandare né le opere per cui la votazione predetta è 0, né le opere con valore negativo della funzione di score.

Alla fine si è deciso di utilizzare questa soglia, la quale, come vedremo, consente di ottenere dei buoni risultati in termini di Richiamo e Precisione. Tuttavia, a volte, in casi come questo, minimizzare anche i falsi negativi potrebbe non essere lo scopo di un museo. Potrebbe essere, infatti, che per un museo vada bene minimizzare esclusivamente i falsi positivi, in quel caso la soglia sarebbe più alta. In effetti ad un utente, solitamente, dà più fastidio che gli venga consigliata un'opera che non gradisce, piuttosto che non gli venga suggerita un'opera che gli piace. Infatti, della prima l'utente si accorge, della seconda no.

## Valutazione

Dopo aver descritto l'algoritmo predittivo e il software realizzato per testarlo vengono esposti di seguito i risultati della valutazione del nostro motore di raccomandazione.

Il software è stato eseguito 5 volte, usando, di volta in volta, un Test Set differente, come già era stato accennato. In ognuna di queste cinque prove, per ogni utente è stata effettuata la valutazione dell'insieme delle opere raccomandate. Confrontando queste raccomandazioni calcolate dall'algoritmo con le vere valutazioni memorizzate dell'utente alle opere del Test Set, sono state determinate le seguenti misure di valutazione:

- **Richiamo:** si vuole vedere a quante delle opere raccomandate dal software l'utente ha messo realmente un LIKE, sul totale di opere del Test Set a cui l'utente ha messo LIKE (infatti, come già detto nell'introduzione al capitolo, questi ultimi sono gli item rilevanti).

- **Precision e Precisione a diversi livelli di cut-off (p@1, p@2, p@3, p@4, p@5):** si vuole vedere a quante delle opere raccomandate dal software l'utente ha messo realmente un LIKE, su una parte, o sul totale, delle opere restituite. Nel calcolo della Precisione si pone un piccolo problema. Come abbiamo già detto, gli item rilevanti sono le opere del Test Set a cui l'utente ha messo LIKE. Le opere a cui l'utente ha messo NOPE sono, quindi, chiaramente non rilevanti. Cosa succede alle opere non votate? Semplicemente, non gli viene associato alcun giudizio di rilevanza, ergo non vengono considerate nel calcolo della Precisione. Esempio:

OPERA RACCOMANDATA	VOTO DELL'UTENTE	
1	LIKE	p@1 = 1
2	LIKE	p@2 = 1
3	NOT VOTED	
4	LIKE	p@3 = 1
5	NOPE	p@4 = 3/4
6	NOPE	p@5 = 3/5
7	LIKE	p@6 = 4/6
8	NOT VOTED	
		Precision = 4/6

Tutto ciò è stato calcolato per vedere quanto bene si comporterebbe il sistema se venissero raccomandate solo le prime k opere. Come sappiamo, infatti, l'algoritmo restituisce le opere raccomandate ordinandole da quella con valore della funzione di score più alto a quella con valore più basso. In questo modo, l'opera con valore della funzione di predizione più alto è considerata come quella che, potenzialmente, più di tutte le altre potrebbe piacere all'utente.

Per scopi di studio, oltre a p@1 e p@2, per ogni insieme di opere raccomandate ad un utente sono state calcolate anche p@3, p@4 e p@5, anche se va detto che non sempre viene restituito un insieme di opere raccomandate di cardinalità superiore o pari a 5, anzi, secondo un controllo effettuato sui risultati, l'insieme più piccolo di opere restituite ha cardinalità 2. In questi rari casi p@3, p@4 e p@5 vengono calcolate supponendo che gli elementi non esistenti siano, in realtà, esistenti e non rilevanti. Esempio:

OPERA RACCOMANDATA	VOTO DELL'UTENTE	
1	NOPE	p@1 = 0
2	LIKE	p@2 = 1/2
		p@3 = 1/3
		p@4 = 1/4
		p@5 = 1/5

- **Average Precision:** si vuole vedere quanto bene l'algoritmo riesce a mettere più item rilevanti in posizioni alte di rank.
- **Curva Richiamo-Precisione Interpolata e 11-pt-AP:** si vuole vedere l'andamento della Precisione all'aumentare del Richiamo.

Alla fine di ogni prova è stato calcolato il valor medio dei risultati di ciascuna delle metriche di valutazione calcolate sugli insiemi delle opere raccomandate. I risultati numerici per ciascuna delle 5 prove sono stati riportati in **Tab.10** e **Tab.11** (graficamente nelle **Fig.41** e **Fig.42**). Nella prima prova (**K = 0**) è stato utilizzato come Test Set il primo 20% delle opere del dataset di riferimento, nella seconda (**K = 1**) è stato utilizzato come Test Set il secondo 20%, nella terza (**K = 2**) il terzo 20%, nella quarta (**K = 3**) il quarto e nella quinta (**K = 4**) il quinto.

Il desiderio era quello di scoprire quanto bene funzionasse il nostro sistema, verificando l'efficacia della raccomandazione. A questo scopo, al fine di verificare l'ANDAMENTO MEDIO del nostro motore di raccomandazione, è stato calcolato il valor medio dei risultati di ciascuna metrica di valutazione calcolata per ciascuno dei Test Set utilizzati (per ciascuna delle prove effettuate). I risultati numerici sono stati riportati sempre in **Tab.10** e **Tab.11** (graficamente nelle **Fig.43** e **Fig.44**).

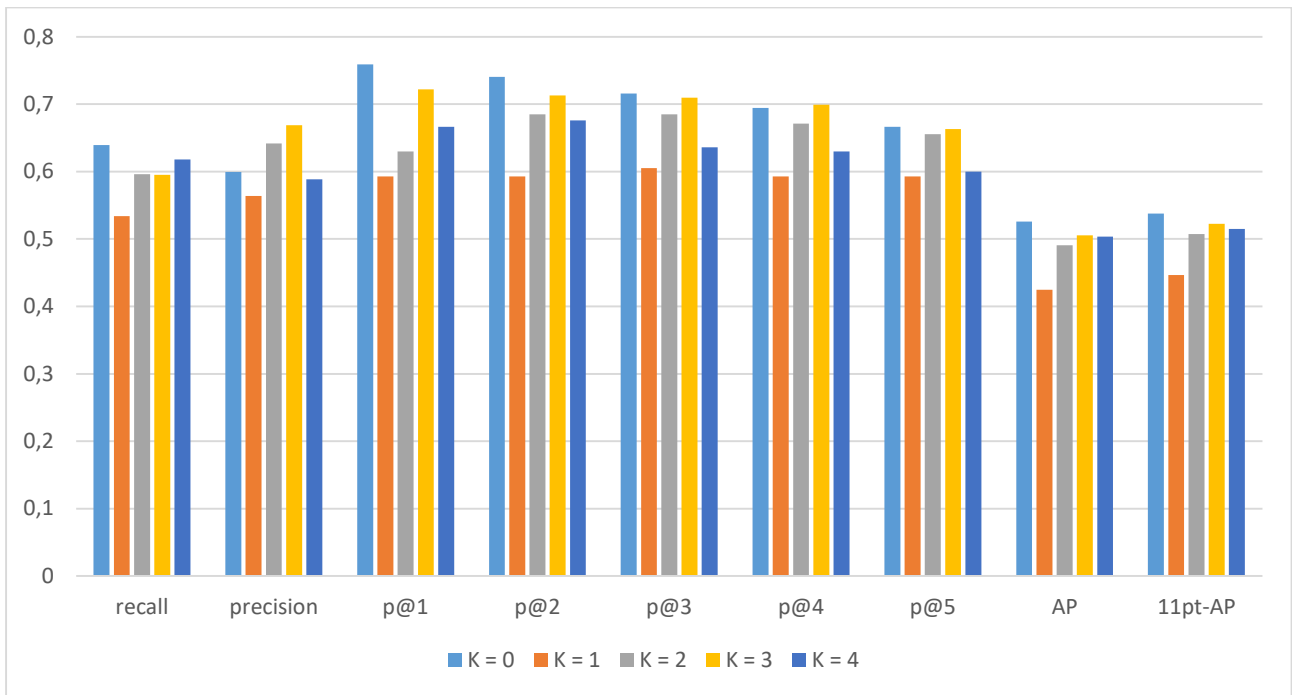
Analizzando il comportamento generale del sistema, che è quello che interessa maggiormente, si può notare immediatamente come le Precision a diversi livelli di cut-off calcolate siano piuttosto alte. La  $p@1$ , come vediamo, ha un valore superiore a 0.67. Questo significa che, nel 67% dei casi circa, la prima opera raccomandata è corretta, ossia due volte su tre è un'opera realmente gradita dall'utente. La  $p@2$  è molto vicina alla  $p@1$ , e così anche la  $p@3$ . Di solito la Precisione tenderebbe a scendere aumentando  $k$ , in quanto vengono aggiunti elementi di possibile "rumore". In questo caso, essendo essa quasi uguale sino al terzo elemento raccomandato, significa che possono essere raccomandate anche le prime due-tre opere e, due volte su tre, queste raccomandazioni saranno corrette.

La Precisione, in media, comincia a scendere a partire dal quarto elemento raccomandato, tuttavia questo calo è dovuto anche al fatto che, nelle prove effettuate, qualche volta ci sono meno di 5 opere raccomandate per un utente, come già è stato detto.

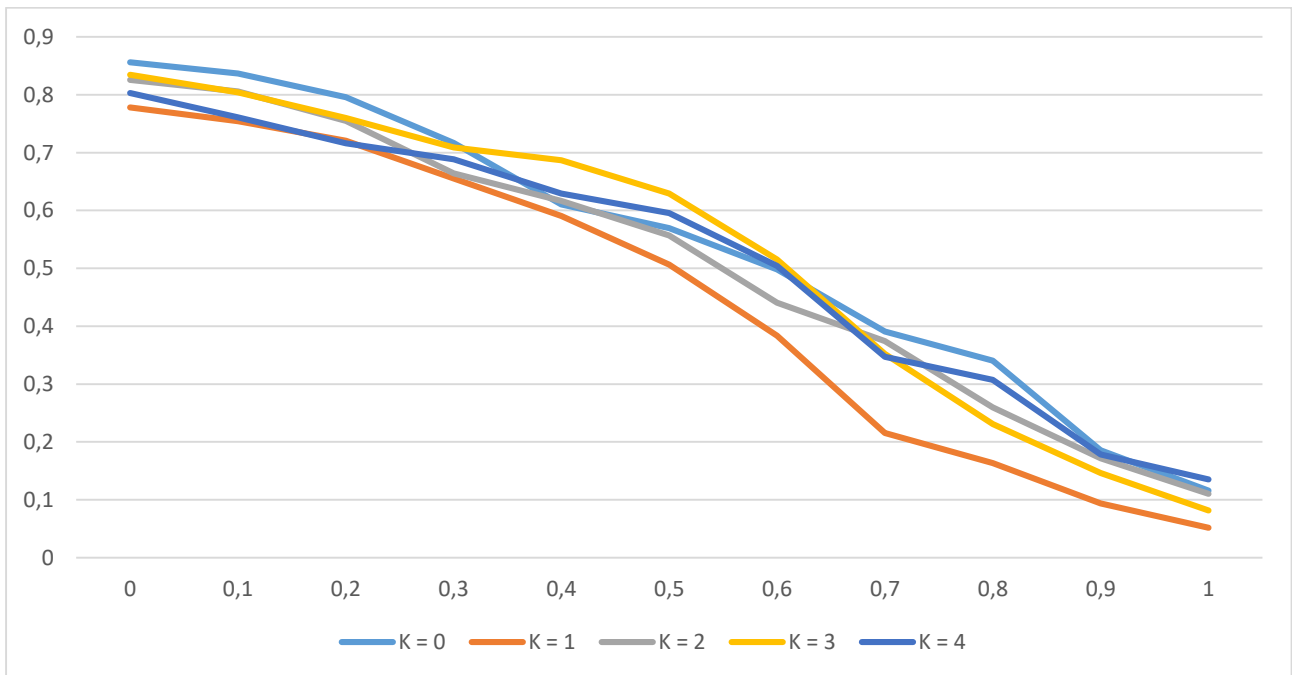
Il sistema fornisce dei buoni risultati anche in termini di Average Precision, segno che l'algoritmo riesce piuttosto bene a metter più item rilevanti in posizioni alte di rank.

Anche il Richiamo e la Precisione totali sono discretamente buoni. Il nostro sistema riesce a richiamare in media il 60% delle opere rilevanti per l'utente. Di tutte le opere richiamate, sempre in media, tre su cinque sono gradite all'utenza.

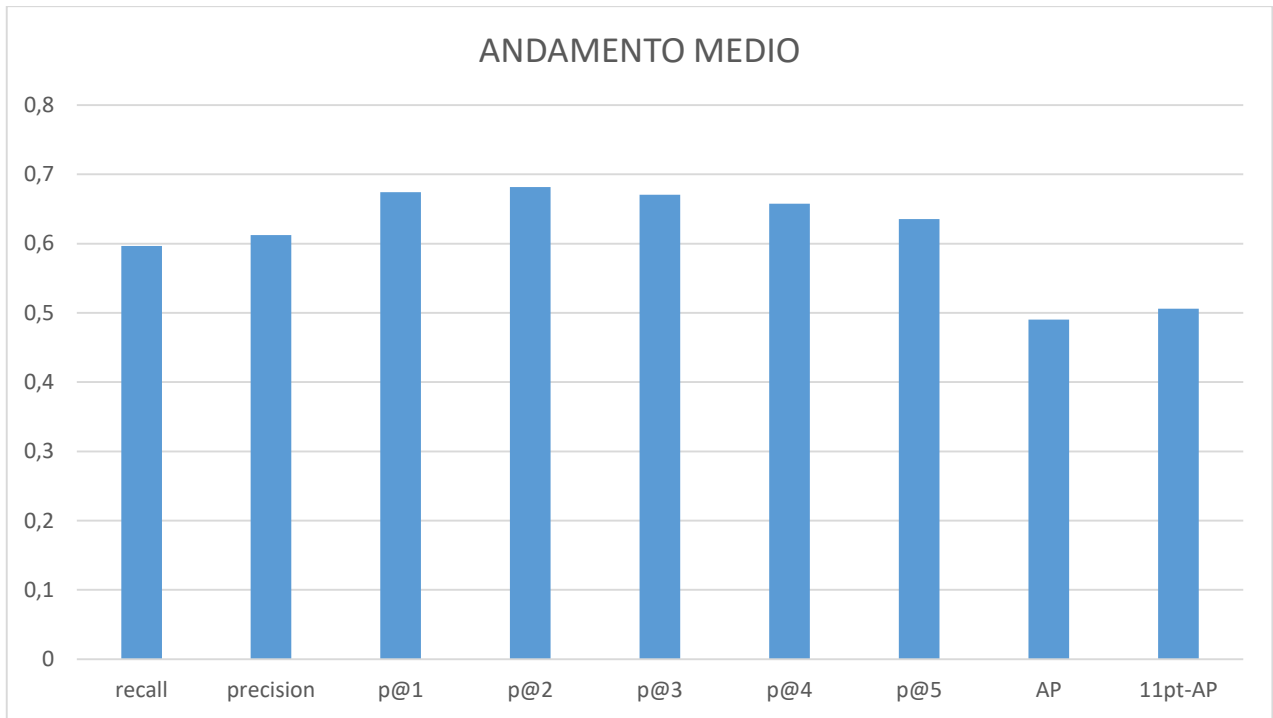
Ad essere precisi va detto che questi valori ottenuti sono una sottostima dei valori possibili, in quanto, nel nostro caso particolare, ci sono opere raccomandate che l'utente non ha votato, le quali, giustamente, non sono state prese in considerazione ai termini del calcolo di questi valori. Considerando anche queste opere, tutti valori di Richiamo e Precisione potrebbero essere anche molto più alti.



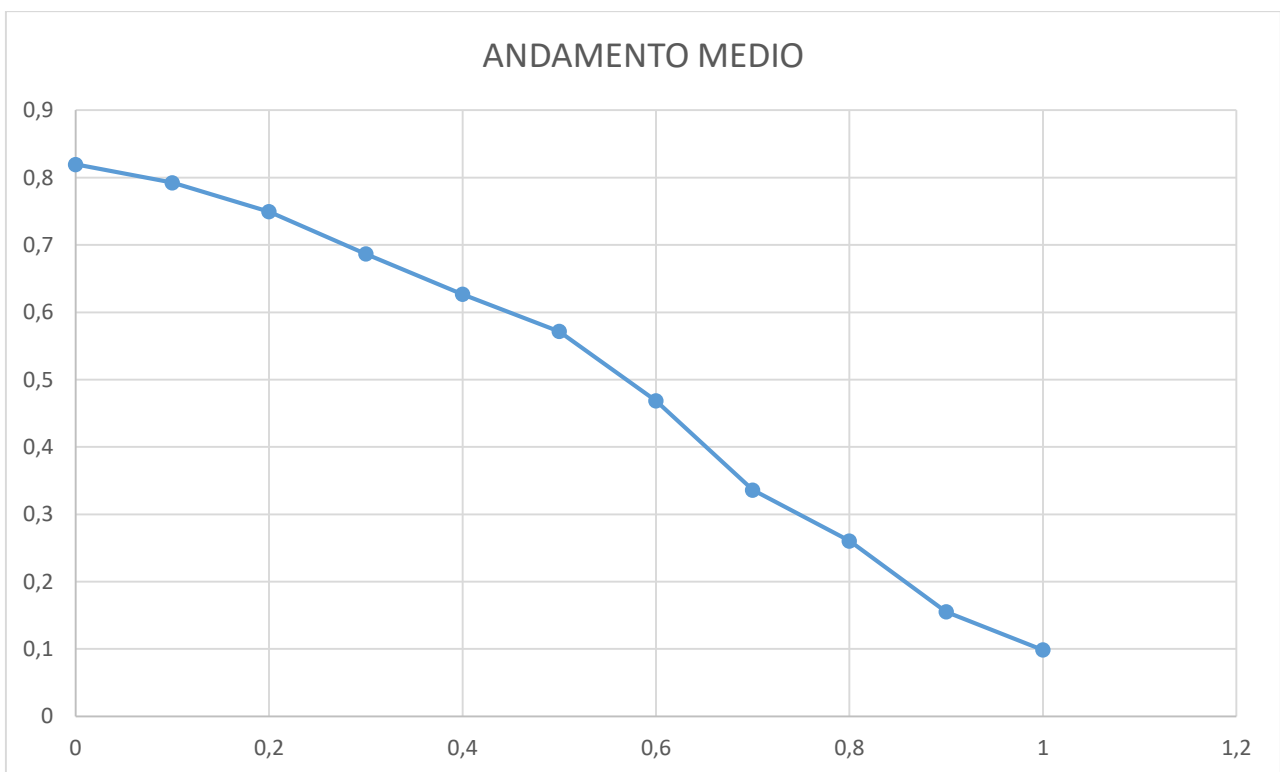
**Fig.41:** Risultati della valutazione della raccomandazione per ciascuna delle 5 prove



**Fig.42:** Curve Richiamo-Precisione Interpolate per ciascuna delle 5 prove



**Fig.43:** Risultati generali della valutazione del sistema



**Fig.44:** Curva Richiamo-Precisione Interpolata che rappresenta l'ANDAMENTO MEDIO del sistema in termini di Richiamo e Precisione

	Recall	Precision	p@1	p@2	p@3	p@4	p@5	AP	11pt-AP
<b>K = 0</b>	0,639136316	0,599564596	0,759259259	0,740740741	0,716049383	0,694444444	0,666666667	0,525808572	0,537883838
<b>K = 1</b>	0,53396273	0,563948345	0,592592593	0,592592593	0,604938272	0,592592593	0,592592593	0,424655398	0,446684428
<b>K = 2</b>	0,59604755	0,641633532	0,62962963	0,685185185	0,685185185	0,671296296	0,655555556	0,490793378	0,507390338
<b>K = 3</b>	0,595295916	0,668733004	0,722222222	0,712962963	0,709876543	0,699074074	0,662962963	0,505475887	0,522656808
<b>K = 4</b>	0,617931655	0,588636425	0,666666667	0,675925926	0,635802469	0,62962963	0,6	0,503719057	0,515060584

<b>ANDAMENTO MEDIO</b>	0,596474833	0,61250318	0,674074074	0,681481481	0,67037037	0,657407407	0,635555556	0,490090458	0,505935199
------------------------	-------------	------------	-------------	-------------	------------	-------------	-------------	-------------	-------------

**Tab.10:** Risultati numerici della valutazione della raccomandazione per ciascuna delle 5 prove e ANDAMENTO MEDIO del sistema

	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
<b>K = 0</b>	0,85627572	0,836738005	0,796087452	0,716867346	0,610247135	0,569512407	0,498414769	0,390734385	0,340683315	0,185631866	0,115529823
<b>K = 1</b>	0,778182003	0,754482709	0,720387089	0,655432112	0,590826964	0,506214283	0,383683685	0,215498088	0,163250667	0,093931158	0,051639948
<b>K = 2</b>	0,825907771	0,805772556	0,75467846	0,664338182	0,616762965	0,55706431	0,440420169	0,374433047	0,259936538	0,17171036	0,11026936
<b>K = 3</b>	0,834455976	0,804209063	0,759743438	0,708774708	0,686649484	0,629183319	0,515439933	0,352129918	0,231047662	0,14621365	0,081377736
<b>K = 4</b>	0,802885106	0,761078816	0,715896916	0,688268774	0,629340821	0,595648598	0,504363118	0,34728256	0,307434166	0,178562481	0,134905073
<b>ANDAMENTO MEDIO</b>	0,819541315	0,79245623	0,749358671	0,686736225	0,626765474	0,571524583	0,468464335	0,3360156	0,26047047	0,155209903	0,098744388

**Tab.11:** Curve Richiamo-Precisione Interpolate per ciascuna delle 5 prove e ANDAMENTO MEDIO del sistema: risultati numerici





## Capitolo 5

### Conclusioni e sviluppi futuri

I Sistemi di Raccomandazione sono entrati ormai a far parte della vita quotidiana. Li possiamo apprezzare ovunque, da Amazon a Facebook, da Spotify a YouTube. Come abbiamo visto si adattano alla perfezione a qualunque ambito, arrivando persino a quello museale, suggerendo ai clienti quali opere potrebbero essere di loro gradimento all'interno di un museo, in modo da costituire un punto di riferimento per chiunque volesse effettuare una visita, aiutandolo nella scelta delle opere d'arte e migliorando la sua esperienza.

Durante la tesi è stato progettato e sviluppato un modello di Sistema di Raccomandazione museale di tipo Content-Based, con lo scopo di profilare un ipotetico utente di un museo, analizzandone i gusti, andando a consigliargli, successivamente, delle opere presenti all'interno della collezione che potrebbero piacergli.

Al fine di raggiungere questo obiettivo, il software che implementa questo modello sfrutta i dati provenienti da TindArt [10], un'applicazione per Android sviluppata dal dottor Daniel Zilio, dottorando del Dipartimento di Beni Culturali dell'Università di Padova. Questa app raccoglie le preferenze degli utenti ("LIKE" oppure "NOPE") relativamente ad una sequenza di opere d'arte appartenenti ad un dataset di riferimento, raccolto in seguito ad uno studio iniziale che ha coinvolto un gruppo di studenti, permettendoci di poter profilare un utente, capendone i gusti, aggirando così il problema del "Cold-Start".

Le opere d'arte del dataset di riferimento, come abbiamo visto, vengono descritte utilizzando parole chiave e attributi relazionali (features relative al contenuto) e vengono utilizzate dal nostro sistema, combinate con le valutazioni degli utenti relative ad esse, con lo scopo di formulare raccomandazioni.

Il nostro Recommendation System, al fine di adempiere ai suoi scopi, usa la funzione di similarità parametrica progettata durante la tesi, la quale, sfruttando la cosine distance, permette di confrontare due opere d'arte basandosi sulla similarità tra i loro metadati (titolo, descrizione, autore e keywords), nell'assunzione che più termini due opere condividono in comune e più simili queste sono tra loro.

Durante la progettazione della funzione ci si è domandati quanto fosse importante l'utilizzo o meno della descrizione piuttosto che delle keywords e se servissero davvero anche informazioni sul titolo e sull'autore delle opere al fine di calcolarne la similarità. I parametri della funzione sono utili, appunto, per bilanciare l'effetto di questi metadati sul risultato finale. Sono stati effettuati degli studi al fine di capire come settare questi parametri, il tutto con lo scopo di ottenere risultati migliori in termini di raccomandazione. Banalmente, se un utente ha votato positivamente un'opera con degli attributi, verosimilmente ha dato una valutazione positiva anche a queste caratteristiche, morale per cui è più facile che gli vengano consigliate opere che posseggono quelle stesse features, cioè le opere simili. Tuttavia si è visto da questi studi che non tutti questi attributi sono utili. Queste analisi condotte sulla funzione e i buoni risultati finali della valutazione del sistema hanno portato ad affermare che i termini del titolo sono di scarsa utilità, mentre l'autore, le keywords e la descrizione permettono di ottenere, se combinati, i risultati migliori nel calcolo della funzione di similarità. Le keywords, inoltre, possono anche essere utilizzate da sole per verificare la similarità fra due opere, in quanto sono uniformate e puramente relative al contenuto dell'opera (ricordiamo la natura Content-Based del nostro sistema), quindi descrivono molto accuratamente le opere del dataset, a differenza della descrizione (può accadere infatti che due opere d'arte identiche vengano descritte usando parole completamente diverse).

L'algoritmo di raccomandazione progettato, il quale sfrutta questa funzione di similarità (con i parametri settati in modo da utilizzare i termini relativi all'autore, alle keywords e alla descrizione nel calcolo della similarità fra due opere), applica una funzione di predizione che, sfruttando la conoscenza pregressa sull'utente, tenta di stimare se un'opera potrebbe o meno piacergli. Questa funzione restituisce un valore numerico. Se questo è maggiore di una certa soglia l'opera viene raccomandata, altrimenti no.

È stato compiuto uno studio anche su questa soglia, al fine di individuare per essa il valore che minimizzasse i falsi positivi e i falsi negativi.

Per testare l'algoritmo, come precedentemente detto, è stato realizzato un software che lo implementa, grazie al quale è stato possibile valutare l'efficacia della raccomandazione.

Dai i risultati finali appare chiaro come il sistema progettato consenta di raccomandare delle opere ad un utente che, nella maggior parte dei casi, sarebbero realmente gradite. Inoltre, se lo scopo del sistema fosse quello di raccomandare solo le prime 3 opere ritornate dall'algoritmo, statisticamente due volte su tre queste raccomandazioni sarebbero corrette.

Uno sviluppo futuro possibile potrebbe essere quello di effettuare una valutazione dell'algoritmo incentrata su utenti finali in un contesto museale reale, la cosiddetta valutazione online.

Tutto ciò ci permetterebbe di analizzare le prestazioni del sistema dal punto di vista dell'utente reale e, distribuendo questionari di gradimento, si potrebbe capire cosa gli utenti possano riferire sulla propria esperienza con esso. In questo modo si potrebbe andare oltre l'accuratezza dell'algoritmo, migliorando il sistema in modo da fornire un'esperienza utile, divertente e personalizzata che porti alla soddisfazione dell'utente finale.

Una volta effettuato questo passo fondamentale, le tecniche di predizione di questo sistema potrebbero essere integrate nell'applicazione TindArt, al fine di renderla un'app museale per la raccomandazione di opere all'interno di uno specifico museo.

Il mondo frenetico in cui viviamo ci lascia poco tempo da dedicare alle cose che più amiamo. Frequentare un museo potrebbe essere un'attività di svago educativa per la quale non sempre si ha a disposizione il tempo necessario. A volte, inoltre, i musei sono talmente grandi che è impossibile visualizzare tutte le collezioni in una volta sola, tenendo conto del fatto che, mentre la visita progredisce, l'interesse e l'attenzione del visitatore diminuiscono. Capita spesso, inoltre, di trovarsi a dover affrontare una mostra d'arte della quale si conosce davvero poco o una collezione che non sia di particolare interesse. Ciò potrebbe farci perdere opere d'arte che sono più legate alle nostre preferenze personali.

I Sistemi di Raccomandazione museale possono migliorare notevolmente la qualità della visita. Tenendo conto delle esigenze in termini di tempo e delle preferenze degli utenti, i Recommendation Systems possono costituire una scelta vincente per un museo, aiutando i visitatori a trovare la propria strada nella collezione di opere d'arte, fornendo loro le informazioni giuste al momento giusto, aumentando la loro consapevolezza sui temi della storia dell'arte, consigliando loro le opere che potrebbero piacergli e, in sostanza, fornendo loro un'esperienza personalizzata e gradevole nella visita.

## Bibliografia

- [1] C.C. Aggarwal, Recommender Systems: The Textbook, Springer International Publishing Switzerland 2016.
- [2] Francesco Ricci, Lior Rokach, Bracha Shapira Editors: Recommender Systems Handbook, Second Edition Springer Science+Business Media New York 2011, 2015.
- [3] Lora Aroyo, Rogier Brussee, Lloyd Rutledge, Peter Gorgels, Natalia Stash, and Yiwen Wang. Personalized Museum Experience: The Rijksmuseum Use Case. In Museums and the Web 2007, San Francisco, USA, April 11-14, 2007.
- [4] Ivan Keller and Emmanuel Viennet. Recommender systems for museums: Evaluation on a real dataset. In Fifth International Conference on Advances in Information Mining and Management, July 2015.
- [5] Daniel Le Berre, Pierre Marquis, and Stephanie Roussel. Planning personalised museum visits. ICAPS 2013 - Proceedings of the 23rd International Conference on Automated Planning and Scheduling, 01 2013.
- [6] A. Tav\_car, C. Antonya, and E.V. Butila. Recommender system for virtual assistant supported museum tours. Informatica (Slovenia), 40(3):279-284, 2016.
- [7] Tsvi Kuik, Einat Minkov, and Keren Kahanov. Graph-based recommendation in the museum. In Proceedings of the First International Workshop on Decision Making and Recommender Systems (DMRS2014), Bolzano, Italy, September 18-19, 2014, pages 46-48, 2014.
- [8] Pierpaolo Di Bitonto, Maria Laterza, Teresa Roselli, and Veronica Rossano. A recommendation system to promote local cultural heritage. Journal of e-learning and Knowledge Society, 7(3), 2011.
- [9] Y.-C. Hsieh and P.-S. You. An efficient encoding scheme for a new multiple-type museum visitor routing problem with must-see and select-see exhibition rooms. International Journal of Computational Intelligence Systems, 10(1):677-689, 2017.
- [10] Daniel Zilio, Nicola Orio, and Camilla Toniolo. TindArt, an Experiment on User Profiling for Museum Applications.
- [11] Maristella Agosti. Modello vettoriale. Reperimento dell'informazione (IR) – aa 2016-2017, Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova.
- [12] Ajay Agarwal, Minakshi Chauhan. Similarity Measures used in Recommender Systems: A Study. International Journal of Engineering Technology Science and Research, Volume 4 Issue 6, June 2017.
- [13] C. Aggarwal. Data mining: the textbook. Springer, New York, 2015.

[14] Maristella Agosti, Gianmaria Silvello. Information Retrieval Evaluation. Valutazione sperimentale: Campagne internazionali e misure di efficacia. Reperimento dell'informazione (IR) – aa 2016-2017, Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova.

[15] Thomas H. Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein. Introduction to algorithms. Third edition. The MIT Press, Cambridge, Massachusetts, 2009.