**Challenge A.A. 2013-2014**

**Ilario Maiolo (403253) e Anastasiya Zadorozhna (418276)**

La challenge consiste nell’implementazione di un sistema di raccomandazione che tratta i dati di Yelp (recensioni e consigli per ristoranti, shopping, vita notturna, divertimento, servizi e molto altro).

L’obiettivo finale del software è di suggerire dei punti di interesse (business) fornendo predizioni di rating per un business da parte di un certo utente.

Il software realizzato si basa sulle tecniche di collaborative filtering studiate durante il corso: User-based Nearest Neighbor e Item-based Nearest Neighbor. Il risultato migliore di predizione si ottiene dall’utilizzo combinato delle due tecniche citate. Inizialmente il sistema calcola le predizioni User-based e Item-based separatamente (in maniera indipendente). In seguito i due risultati sono combinati (con l’utilizzo della media ponderata).

**Le ipotesi di lavoro**

Si è deciso di non ricorrere all’utilizzo di una base di dati al fine di evitare di peggiorare le prestazioni (l’accesso alla base di dati è molto costoso in termini di tempo).

L’idea iniziale era quella di utilizzare una base di dati per contenere tutti i calcoli effettuati off-line, relativi alla similarità tra item (adjusted cosine similarity, approccio item-based). L’approccio descritto richiedeva tante connessioni con la base di dati e quindi un forte rallentamento.

La soluzione adottata attualmente è una mappa in memoria: quando si inizializzano le strutture dati, ad ogni utente viene associa una mappa che memorizza tutte le review da lui effettuate (il software lavora esclusivamente on-line).

Per aumentare le prestazioni del sistema, prima di creare le strutture dati si è deciso di eliminare il campo ‘text’ da ogni review del file review.json (questo campo ha dimensioni notevoli ma non viene utilizzato nel calcolo delle predizioni). È stata realizzata una classe apposita: GarbageManager, che ha questo compito.

Tutti i dati necessari al sistema sono ricavati dal dataset fornito (4 file json: business.json, checkin.json, review.json, user.json).

**Valutazioni interne**

Il dataset è suddiviso in training set (85%) e test set (15%) in maniera casuale a ogni iterazione del sistema.

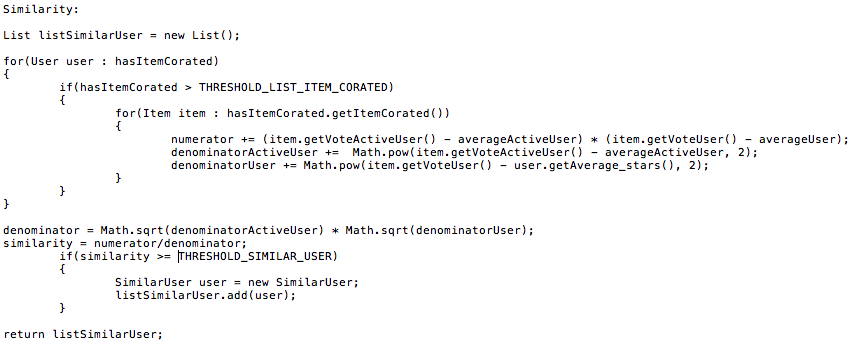
Il sistema è testato attraverso la metrica MAE (Mean Absolute Error) che misura quanto sono vicine le stime e le predizioni agli eventuali esiti.

**User-based**

In input è richiesto il rating database e l’ID dell’utente attuale o attivo (active user). In seguito sono identificati altri utenti (peer user/nearest neighbor) che hanno avuto in passato delle preferenze simili a quelle dell’utente attivo.

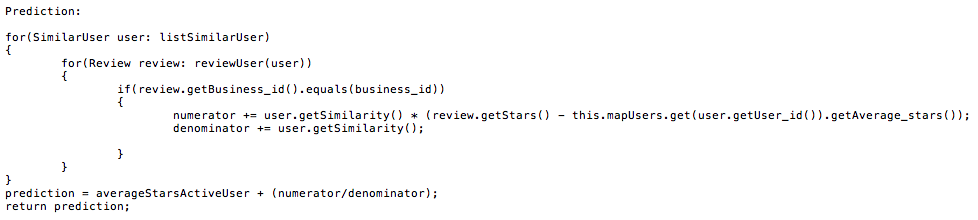
La similarità è calcolata con l’utilizzo della formula di Pearson (**Pearson Correlation Coefficient**):

N.B. i valori sono compresi tra +1 (strong positive) e -1 (strong negative).



Per fare una predizione bisogna effettuare una scelta sui neighbor da considerare per il calcolo (considerando solo quelli che superano una certa soglia di similarità) e il grado con il quale valutare le loro opinioni (le soluzioni applicate sono descritte in seguito, nel paragrafo che tratta i miglioramenti per l’approccio user-based).

Le predizioni sono calcolate utilizzando la seguente formula:



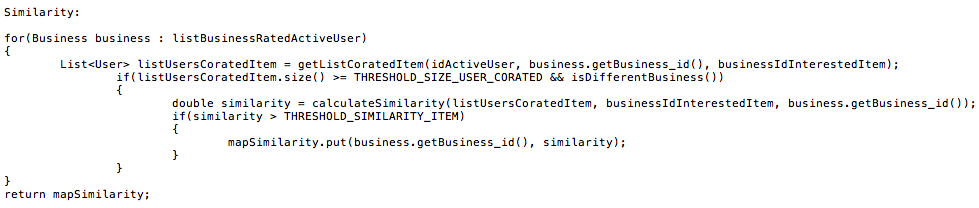
**Miglioramenti dell’approccio user-based**

* Si ordina la lista degli utenti simili in base al valore della similarità e si applica, poi, un ulteriore incremento al valore della similarità degli utenti nelle prime posizioni (utenti più simili). Nel nostro caso, tale incremento, viene applicato solamente ai primi utenti della lista ordinata, in particolare ad un numero di utenti pari ad un decimo della lunghezza della lista (listSimilarUser.size()/10).
* Il sistema considera anche il caso in cui due utenti abbiano pochi item (business) in comune per i quali le loro opinioni possano essere simili solo per caso e quello in cui vi siano molti item sui quali i loro pareri concordano. Si utilizza un fattore che riduce linearmente il peso quando il numero di co-rated item è basso (*significance weighting*).
* Il neighborhood è selezionato considerando un valore minimo specifico di soglia per la similarità fra utenti. Se la similarità tra due utenti è troppo bassa, questo utente non viene considerato nel calcolo della predizione.

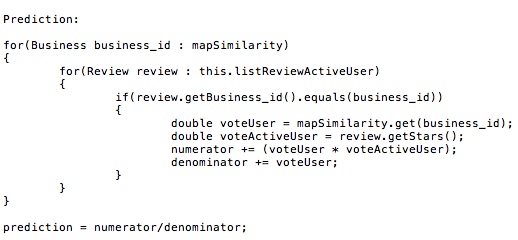
**Item-based**

L’idea base dell’approccio item-based è quella di calcolare le similarità tra item invece della similarità tra utenti.

Per calcolare la similarità si utilizza la **adjusted cosine measure**.



Le predizioni sono calcolate utilizzando la seguente formula:



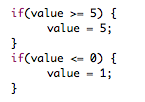
**Miglioramenti item-based**

* Si applica l’approccio content-based. Sono stati costruiti dei range in base al numero di categorie in comune tra i due item (si fanno i confronti sul campo ‘categories’ nel file business.json). Secondo il numero ottenuto, il valore della similarità tra questi item viene incrementato.

**Item-based/User-based combinato**

La predizione combina il valore ottenuto con l’approccio user-based e quello ottenuto con l’approccio item-based. Si utilizza la media ponderata che si basa sul "peso" degli elementi (ciascuno dei due valori è moltiplicato per il relativo peso, si sommano i prodotti ottenuti e infine si divide per la somma dei pesi).

Alla fine i valori delle predizioni sono arrotondati e normalizzati:



­

