Cuprins

[1 Introducere 3](#_Toc12014213)

[2 Sisteme de recomandări 4](#_Toc12014214)

[2.1 Content-based filtering 4](#_Toc12014215)

[2.1.1 Avantaje 4](#_Toc12014216)

[2.1.2 Dezavantaje 5](#_Toc12014217)

[2.1.3 Concluzie 5](#_Toc12014218)

[2.2 Collaborative filtering 5](#_Toc12014219)

[2.2.1 Avantaje 6](#_Toc12014220)

[2.2.2 Dezavantaje 6](#_Toc12014221)

[2.2.3 Concluzie 6](#_Toc12014222)

[2.3 Concluzie generală 6](#_Toc12014223)

[3 Implementarea sistemului de recomandări 7](#_Toc12014224)

[3.1 User-Based Collaborative filtering ( User-User) 7](#_Toc12014225)

[3.1.1 Analiza similarităților 7](#_Toc12014226)

[3.1.2 Rolul kNN (k Nearest Neighbour) 10](#_Toc12014227)

# Introducere

Aflat într-o continuă expansiune,mediul online reprezintă o scurtătură extrem de viabilă pentru cca. 4 miliarde de utilizatori oferind oportunități pentru accesarea de informație, socializare etc. Interacțiunea utilizatorilor si accesarea de informații este realizată prin intermediul aplicațiilor Web.

Motivația alegerii dezvoltării unei aplicații web pentru recomandarea de filme online vine în primul rând din pasiunea pentru vizionarea acestora,adăugând necesitatea unui ”instrument” care să vină în ajutorul utilizatorului care se intreabă ”ce film să vizionez?”. Automat aplicația va trebui să încerce să înglobeze răspunsul la această întrebare, nu în sensul de a decide in locul utilizatorului,cât de a-l ajuta prin reducerea numărului de opțiuni dintr-o gamă vastă de filme, rămânand cele mai potrivite pentru el.Acestea vor fi rezultatul sistemului de recomandări.

Tema este una de tot mai des întalnită,luând în calcul faptul că se pune foarte multe accent pe experiența trăită de utilizator în interacțiunea cu aplicațiile web. Companii precum Amazon,Linkedin,Netflix dezvoltă sisteme de recomandări în încercarea de a ajuta utilizatorii să descopere produse/obiecte/articole cât mai relevante pentru asteptările sau nevoile lor.

Pentru realizarea aplicației vor fi utilizate tehnologii de actualitate, atât pe partea de back-end, cât și pe partea de front-end.

Scopul lucrării constă in realizarea unui mediu cât mai user-friendly pentru utilizatori, și găsirea unei soluții cât mai eficientă pentru dezvoltarea sistemului de recomandări.

# Sisteme de recomandări

Pe măsura ce dimensiunea si complexitatea aplicațiilor web crește semnificativ, utilizatorul poate pierde timp căutând informațiile de care are nevoie. În același timp, utilizatorul poate să nu știe exact ce caută, sau iși poate schimba preferințele in timp. Sistemele de recomandări furnizează informații personalizate în funcție de profilul utilizatorului, învățând din interacțiunea sa cu aplicația.

## Content-based filtering

Acest sistem recomandă comparând conținutul itemelor cu profilul utilizatorului.Nu sunt implicați alți utilizatori,ci doar bazat pe preferințele proprii , algoritmul va alege articole cu conținut similar.Profilul utilizatorului este reprezentarea structurală a intereselor acestuia,folosit în recomandarea noilor articole.Procesul de recomandare consistă prin urmare în potrivirea caracteristicilor profilului cu conținutul unui articol necunoscut.Astfel, eficiența algoritmului va fi direct proporțională cu acuratețea cu care profilul reprezintă preferințele utilizatorului.

### Avantaje

* Independența față de alți utilizatori – Sistemul de recomandări bazat pe filtrarea de conținut se folosește doar de ratingurile utilizatorului în cauză pentru a modela profilul acestuia,spre deosebire de Collaborative filtering,care va fi prezentat în cele ce urmează.
* Transparență – Modul în care sistemul funcționează poate fi ușor exemplificat prin expunerea atributelor care au condus la adăgarea unui articol în lista de recomandări.Aceste atribute pot fi indicatori pentru credibilitatea sistemului.Prin comparație,sistemele colaborative nu oferă același nivel de transparență,explicația pentru un articol recomandat fiind doar faptul că alți utilizatori necunoscuți cu gusturi asemănătoare au fost interesați de același item.
* În contextul adăugării unui articol nou – Sistemul este capabil să recomande articole care n-au fost încă apreciate de alți utilizatori.În consecință nu apare problema ”primului rating”,care afectează sistemele colaborative care se bazează doar pe preferințele utilizatorului și care nu ar putea recomanda un articol nou până când nu ar strânge un anumit număr de ratinguri din partea celorlalți utilizatori.

### Dezavantaje

* Supra-specializarea – Acest sistem de recomandări nu oferă nici o metodă pentru găsirea neprevăzutului.Numită si problema serendipității,pune în evidență tendința sistemului de a produce recomandări cu un grad scăzut de noutate.
* Problema utilizatorului nou (Cold-Start problem) – Pentru ca sistemul să funcționeze trebuie ca mai întăi să fie creat un profil al utilizatorului,prin colectarea de ratinguri. Prin urmare , când nu sunt suficiente ratinguri disponibile,sistemul nu va putea să ofere recomandări relevante.
* Analiza limitată de conținut – Sistemul nu oferă rezultate conform așteptărilor dacă nu reușește să distingă între articolele care-i plac utilizatorului de cele care nu-i plac.Unele profile ar putea să nu înglobeze anumiți factori care influențează alegerile utilizatorului.

### Concluzie

Sistemul bazat pe filtrarea de conținut este independent de interacțiunea dintre utilizatori dar totuși necesită implicare din parte utilizatorului pentru formarea unui profil cât mai exact care să conducă la recomandări eficiente.

## Collaborative filtering

Filtrarea colaborativă este procesul care se bazează strict pe părerea comunității.Nu se ține cont de însușirile/conținutul unui anumit articol ci doar de ratingurile atribuite acestuia de către comunitate.Se formează o matrice utilizator-articol a notelor,în care fiecare linie reprezintă notele acordate de utilizator ,iar fiecare coloană reprezintă un anumit articol.La interseția dintre o linie si o coloană se va regăsi nota acordată de utilizator.

Există 2 subtipuri de Collaborative filtering : User-Based Collaborative filtering,unde în contextul recomandării unui articol sistemul va căuta utilizatori cu preferințe similare și Item-Based Collaborative filtering,unde sistemul va căuta articole cu note similare în comunitate.

### Avantaje

* Independența de conținut – Sistemul bazat pe filtrarea colaborativă ignoră proprietățile/conținutul articolelor și se focusează doar pe părerea comunității.

### Dezavantaje

* Cold-Start problem – Are nevoie de suficienți utilizatori care să acorde note pentru a putea găsi similarități.
* Problema adăugării unui articol nou – Spre deosebire de filtrarea pe bază de conținut,acest sistem nu poate recomanda un anumit articol dacă acesta nu are un număr minim de note acordate.
* User/Ratings matrix sparsity problem – Deoarce matricea user/rating este una rară ar putea deveni dificil să fie găsiți utilizatori care au acordat note acelorași articole.
* Problema popularității – Sistemul nu poate recomanda articole unui utilizator cu gusturi unice și în același timp tinde să recomande articole populare.

### Concluzie

Sistemul bazat pe filtrarea colaborativă are nevoie de o comunitate cât mai activă din perspectiva acordării de feed-back pentru articole pentru a oferi recomandări cât mai eficiente.

## Concluzie generală

Așadar ambele sisteme de recomandări au plusurile si minusurile lor în ceea ce privește eficiența rezultatelor iar o tehnică de abordare în integrarea unui asemenea sistem într o aplicație web este reprezentată de implementarea unui sistem hibrid.

# Implementarea sistemului de recomandări

## User-Based Collaborative filtering ( User-User)

Termenul de colaborativ pune în evidență din start fundamentul pe care se bazează acest tip de sistem: ”interacțiunea” colectivă. În contextul livrării unei recomandări către un utilizator vor fi căutați utilizatori cu gusturi și preferințe similare.

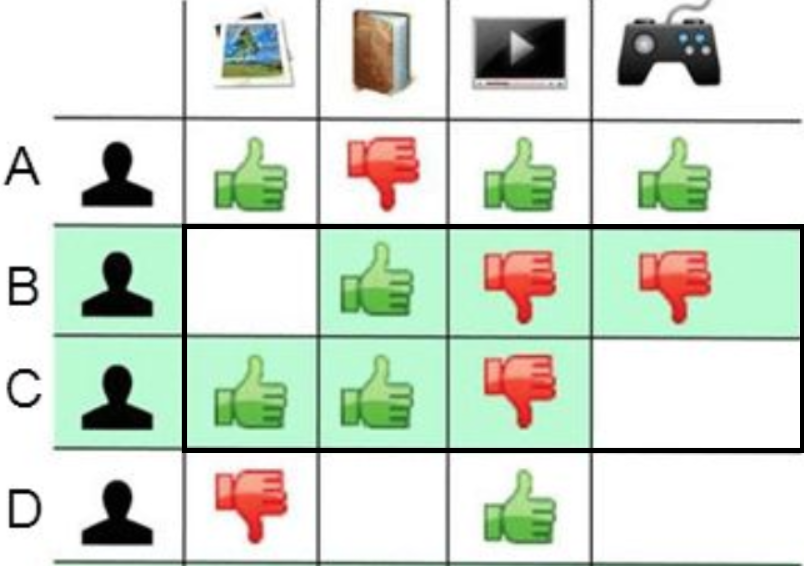


Figura 1 : similaritatea dintre utilizatori în matricea user-item

### Analiza similarităților

Analizând din perspectiva matricii user-item putem defini ca similaritatea dintre doi utilizatori interpretarea ratingurilor comune date de aceștia.Spre exemplu,în Figura 1 nivelul de similaritate dintre utilizatorii B și C este unul maxim , luând în calcul faptul că articolele pe care amândoi le-au evaluat au primit aceeași valoare.

Pentru estimarea similarității dintre utilizatori există diferite tipuri de măsurători care vor fi folosite în experimentul de testare al sistemului cu scopul de a găsi cea care se potrivește cel mai bine pentru sistemul dezvoltat.

Imaginând un spațiu 2D care reprezintă notele acordate de către utilizatori pentru două filme,pe fiecare dintre axe fiind prezente ratingurile posibile (de la 1 la 5) .

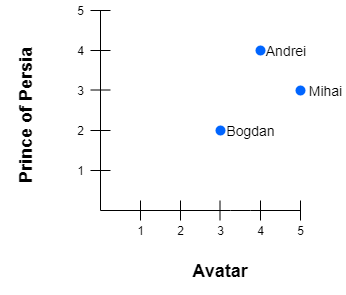
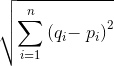


Figura 2 : spațiu 2D pentru reprezentarea ratingurilor

Distanța dintre Bogdan și Mihai va reprezenta similaritatea la nivel de preferințe dintre cei doi. Desigur, această distanță trebuie calculată într-un spațiu cu N filme , deci N-dimensional.

(1)



Aceasta poate fi calculată cu ajutorul distanței Euclidiene, unde qi si pi reprezintă ratingurile filmului comun i.

O problemă care poate fi luată în calcul este non-concordanța dintre metodele de acordare a notelor de către utilizatori.Pot fi utilizatori pentru care nota 4 înseamnă un film foarte bun și utilizatori care notează articolele care le-au plăcut cu nota maximă iar celelalte cu 4.Pot fi utilizatori care notează doar cu nota minimă si nota maximă.

Una dintre soluțiile la aceasta problemă este utilizarea coeficientului de corelație Pearson.Acesta descrie relația dintre două variabile si are o valoarea reală cuprinsă în intervalul [-1,+1] , unde -1 semnifică că una dintre variabile are tendința sa descrească când cealaltă crește, 0 faptul că nu există nici un fel de relație între cele două variabile și 1 dacă o variabilă are tendința să crească o dată cu cealaltă.

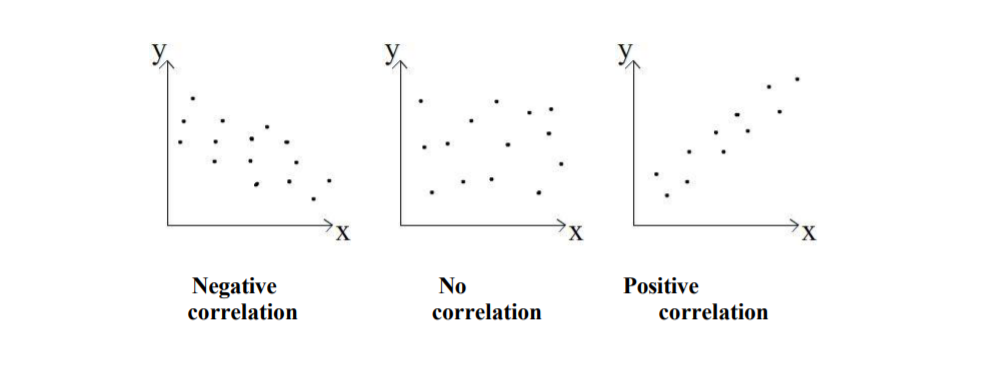
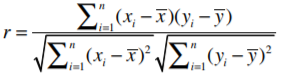


Figura 3 : corelația Pearson

(2)



(2) Reprezintă formula de obținere a coeficientului de corelatie Pearson,xi și yi reprezintă ratingurile acordate filmului comun i,iar și mediile ratingurilor acordate de utilizatorii x și y. Prin scăderea valorii medii se obține centrarea datelor,deoarce ratingurile sunt valori pozitive ,prin acest procedeu coeficientul de similaritate va avea valori cuprinse în intervalul [-1,1].

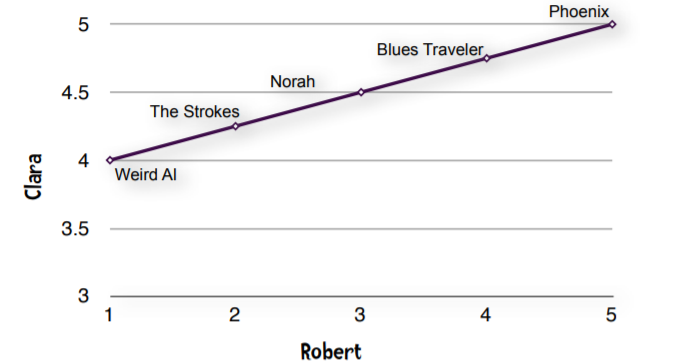
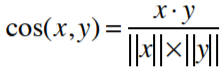


Figura 4 : corelație perfectă

În Figura 4 se poate observa o corelație perfectă între notele acordate de utilizatorii Clara și Robert.

 (3)

Prin similaritatea cosinusului (3) se calculeaza cosinusul unghiului dintre doi vectori nenuli.Valoarea numărătorului semnifică produsul scalar dintre cei doi vectori ,iar la numitor se regăsește produsul dintre lungimile(modulele) celor doi vectori date de formula(4).

modul-vector.png(4)

Se poate observa asemănarea dintre formulele de calcul ale coeficientului de similaritate Pearson (2) și similaritatea cosinusului (3),cu deosebirea că cea din urmă nu centrează datele.

likelihood1.gif(5)

loglikelihood.gif(6)

Relația (5) exprimă verosimilitatea(likelihoodul) în exprimarea parametrilor.Prin date se înțelege datele avute la dispoziție iar prin θ setul de parametri care influențează datele.Pentru ușurința calculelor se folosește log-likelihoodul deoarece logaritmul este o funcție monoton crescătoare,acest lucru însemnând că se păstrează relevanța rezultatului.

jaccardindex.png(7)

Formula (7) reprezintă modul de calculare al indexului Jaccard , cunoscut ca și raportul dintre interescția și reuniunea a două mulțimi,acestea reprezentând mulțimile ratingurilor acordate de doi utilizatori pentru care se vrea calculată similaritatea.De observat este faptul că indexul Jaccard nu ține cont de valoarea ratingurilor ci doar de prezența acestora.

### Rolul KNN (K Nearest Neighbour)

K-Nearest-Neighbour reprezintă unul din cei mai populari algoritmi de Machine Learning utilizați pentru clasificare.Acesta are un rol important și în algoritmul de filtrare colaborativă bazat pe similaritățile dintre utilizatori.Acestea sunt calculate pentru fiecare dintre utilizatori,doi câte doi ,iar la final vor fi aflate toate similaritățile dintre ei.Însă în contextul efectuării unei recomandări de către sistem ar trebui luate în calcul preferințele mai multor utilizatori similari cu cel pentru care se recomandă un articol și nu doar a celui cu nivelul de similaritate cel mai bun.Pentru acest lucru va fi utilizat algoritmul KNN , unde K va reprezenta numărul de utilizatori similari al căror preferințe vor fi luate în calcul pentru elaborarea de recomandări pentru utilizatorul curent.

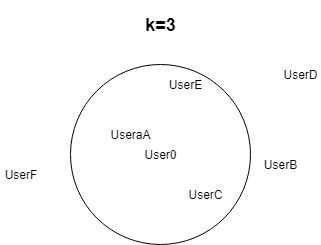
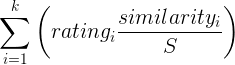


Figura : cei mai similari utilizatori pentru k=3

În Figura 5 sunt exprimate similaritățile utilizatorilor A,B,C,D,E,F cu utilizatorul notat User0.Pentru k=3 vor fi luate în calcul preferințele celor mai apropiați 3 utilizatori în elaborarea recomandărilor pentru User0.

### Predicția unui rating

Luând în considerare similaritățile cu ceilalți utilizatori , precum și care sunt cele mai apropiate K similarități se poate realiza predicția ratingului unui anumit film.

(8)

Ratingul estimat al unui anumit film va fi egal cu rezultatul prezentat în formula (8),unde k reprezintă numărul de utilizatori similari luați în calcul,ratingi reprezintă ratingul acordat de utilizatorul i ,similarityi reprezintă valoare similarității dintre utilizatori,iar S reprezintă suma tuturor celor k similarități ale utilizatorilor implicați.

### Implementare

Pentru implementarea sistemului a fost folosit framework-ul de la Apache Mahout,care oferă suport pentru statistică și machine learning.



Figura 6 : dependințe Mahout

dataModel.png

Figura 7 : DataModel

DataModel stochează preferințele utilizatorilor,și poate primi ca input un fișier în care pe fiecare linie un rating este exprimat ca userID,itemID,rating , sau o bază de date la care trebuie specificate coloanele userID,itemID și rating de unde pot fi preluate datele.

usersimilarity.png

Figura 8 : UserSimilarity

UserSimilarity exprimă metoda de calculare a similarităților dintre utilizatori care primește modelul.

userneihgbourhood.png

Figura 9 : UserNeighbourHood

UserNeighbourHood primește ca input o valoare n echivalentă cu K-ul care stabilește vecinătatea de selecție a celor mai apropiate similarități,tipul de măsurare al similarităților și si modelul.

userrecommender.png

Figura 10 : UserBasedRecommender

UserBasedRecommender primește ca parametrii modelul,vecinătatea precum și modul de calculare al similarităților.

userbased_estimatepref.png

Figura 11 : predicția unui rating

Prin metoda estimatePreference care primește ca parametrii un userID si un itemID se efectuează predicția ratingului pentru itemul respectiv.

userbased_recommend.png

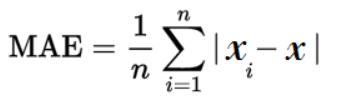
Figura 12 : elaborarea recomandărilor

Prin metoda recommend sistemul va estima preferințele pentru toate itemele cărora utilizatorul cu id-ul dat ca prim parametru nu le-a acordat o notă și va returna o listă de lugime cel mult egală cu al doilea parametru, o listă de obiecte RecommendedItem care conțin id-ul itemului respectiv și ratingul prezis ordonate descrescător în funcție de valoarea ratingului.

### Testarea sistemului

Pentru a putea stabili ce tip de similaritate precum si ce vecinătate trebuie aleasă pentru un sistem de recomandări cât mai precis din punct de vedere al filmelor recomandate trebuie efectuate câteva teste.

Pentru evaluare a fost folosit dataset-ul MovieLens 100K Dataset [[1]](#footnote-2) care prevede 5 perechi de fișiere de antrenament(80% din date) – fișiere de test(20% din date) disjuncte care pot fi utilizate pentru 5 fold cross validation[[2]](#footnote-3).

(9)

Metrica de măsurare a acurateții sistemului a fost MAE(Mean absolute error)(9).După colectarea datelor din fișieirul de antrenament au fost comparate ratingurile prezise de sistem cu cele existente în fișierul de test.Media absolută a erorii a fost calculată pentru fiecare utilizator în parte iar formula exprimă de fapt media diferențelor dintre ratingul prezis și cel adevărat.

eroare_sistem.gif(10)

Eroarea totală a sistemului este reprezentată în relația (10) și semnifică raportul dintre suma MAE a tuturor utilizatorilor și numarul de utilizatori.

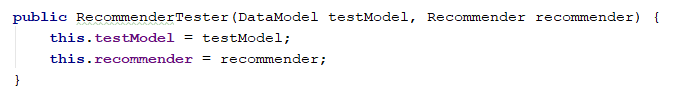


Figura 13 : constructor RecommenderTester

Testarea a fost realizată prin intermediul clasei RecommenderTester(Figura 13) care primește în constructor un set de date de test de tip DataModel și un sistem de recomandări.

1. https://grouplens.org/datasets/movielens/100k/ [↑](#footnote-ref-2)
2. Repetarea expermientului pe fiecare pereche set de antrenament-set de test și efectuarea mediei rezultatelor [↑](#footnote-ref-3)