SISTEMA RECOMANADOR

DOCUMENTACIÓ PRIMERA ENTREGA(v1.0)

PROJECTE DE PROGRAMACIÓ

Grau en Enginyeria Informàtica

Pol Blavia Cañet(pol.blavia)
Aftab Ahmed Choudhry(aftab.ahmed.choudhry)
Marta Granero I Martí(marta.granero.i)
Miquel Umbert Bosch(miquel.umbert.bosch)
Grup 12-subgrup-prop1.3

Departament de Ciències de la Computació
Universitat Politècnica de Catalunya
Novembre, 2021

ÍNDEX DOCUMENTACIÓ

1 CASOS D'ÚS

1.1 DIAGRAMA

L'elaboració del diagrama de casos d'ús s'ha efectuat amb l'eina Visual Paradigm for UML.

1.2 DESCRIPCIÓ DETALLADA DE CADA CAS D'ÚS

CREAR PERFIL

Nom: Crear perfil
Actor: Usuari actiu
Descripció: L'usuari actiu crea el seu perfil al sistema recomanador.
Comportament:

- L'usuari prem l'opció "Crear Perfil"
- El sistema demana a l'usuari el nom d'usuari i la contrasenya que vol utilitzar, com així mateix, la confirmació la seva contrasenya
- L'usuari introdueix les dades per crear el perfil
- El sistema valora els atributs introduïts per l'usuari i crea el nou perfil de l'usuari amb els seus camps corresponents.

Errors possibles i cursos alternatius:

- Si ja existeix un usuari amb el nom d'usuari introduït, el sistema informa de l'error.
- Si el password i la confirmació no són iguals, el sistema informa de l'error.

CARREGAR PERFIL

Nom: Carregar perfil
Actor: Usuari actiu
Descripció: L'usuari, amb el seu perfil, accedeix al programal
Comportament:

- L'usuari escull l'opció "Carregar Perfil"
- El sistema demana a l'usuari el nom d'usuari i la contrasenya per iniciar sessió
- L'usuari introdueix les dades per loguejar-se
- El sistema valora els atributs introduïts per l'usuari i inicia la sessió de l'usuari dins del sistema.

Errors possibles i cursos alternatius:

- Si el nom d'usuari no està registrat en el sistema, aquest informa de l'error.
- Si el password de l'usuari és incorrecte, el sistema informa d'error de contrasenya.

ELIMINAR PERFIL

Nom: Eliminar perfil
Actor: Usuari actiu
Descripció: L'usuari esborra el seu perfil del programa
Comportament:

- L'usuari ordena al sistema, prement la tecla "Eliminar Perfil", que vol esborrar el seu perfil.
- El sistema demana a l'usuari el password per verificar la seva identitat.
- L'usuari verifica el password
- El sistema borra el perfil i totes les seves dades del programa.

Errors possibles i cursos alternatius:

- Si la contrasenya és incorrecte, el sistema informa de l'error.
- Si el password de l'usuari és incorrecte, el sistema informa d'error de contrasenya.

MODIFICAR PERFIL

Nom: Modificar perfil
Actor: Usuari actiu
Descripció: L'usuari modifica el perfil del sistema amb la nova informació
Comportament:

- L'usuari escollint l'opció "Modificar Perfil"
- El sistema dóna l'opció a l'usuari de seleccionar els camps que vol modificar del seu perfil
- L'usuari indica la nova informació.
- El sistema actualitza el perfil de l'usuari en qüestió amb els nous camps que s'han modificat.

Errors possibles i cursos alternatius:

- Si la nova informació introduïda no compleix amb els requeriments del camp, el sistema informa de l'error.

CONSULTAR PERFIL

Nom: Consultar perfil
Actor: Usuari actiu
Descripció: L'usuari actiu consulta les dades que es troben al seu perfil
Comportament:

- L'usuari decideix escollir la opció de "Consultar Perfil" per veure les seves dades
- El sistema mostra a l'usuari les dades que actualment es troben al seu perfil, i.e el nom d'usuari i la contrasenya que el sistema emmagatzema

Errors possibles i cursos alternatius: -

OBTENIR RECOMANACIÓ

Nom: Obtenir Recomanació
Actor: Usuari actiu
Descripció: L'usuari actiu obté la recomanació calculada pel programa a través d'un dels 3 mètodes possibles(Collaborative filtering,Content-based filtering o bé híbrid)
Comportament:

- L'usuari escull l'opció d'"Obtenir Recomanació" per obtenir una llista d'ítems sobre els quals pugui afegir la seva valoració
- El sistema retorna a l'usuari la corresponent llista d'ítems que composaran la recomanació

Errors possibles i cursos alternatius: -

GUARDAR RECOMANACIONS

Nom: Guardar Recomanacions
Actor: Usuari actiu
Descripció: L'usuari desitja guardar la recomanació que se li ha fet prèviament
Comportament:

- L'usuari avisa al sistema perquè guardi les recomanacions que se li han fet al perfil.
- El sistema guarda al perfil de l'usuari en qüestió amb les recomanacions que ha escollit guardar.

Errors possibles i cursos alternatius: -

CONSULTAR RECOMANACIONS ANTERIORS

Nom: Consultar Recomanacions Anteriors
Actor: Usuari actiu
Descripció: L'usuari desitja consultar les recomanacions que se li han fet prèviament
Comportament:

- L'usuari mitjançant l'opció "Consultar Recomanacions Anteriors" demana al sistema veure les recomanacions anteriorment fetes amb un mètode en concret
- El sistema dóna al perfil de l'usuari en qüestió les recomanacions que ha guardat el sistema d'aquest

Errors possibles i cursos alternatius:

- Si l'usuari no troba cap recomanació anterior feta a l'usuari per un mètode en concret, el sistema informa de l'error a l'usuari

ESBORRAR RECOMANACIÓ

Nom: Esborrar Recomanacions
Actor: Usuari actiu
Descripció: L'usuari desitja esborrar una recomanació que se li ha fet prèviament i estava emmagatzemada
Comportament:

- L'usuari avisa al sistema perquè esborri la recomanacions de l'algorisme que se li ha fet amb anterioritat.
- El sistema borra al perfil de l'usuari en qüestió les recomanacions que ha escollit borrar.

Errors possibles i cursos alternatius: -

VALORAR ÍTEM

Nom: Valorar Ítem
Actor: Usuari actiu
Descripció: L'usuari actiu valora el conjunt d'ítems de la recomanació calculada pel programa a través d'un dels 3 mètodes possibles(Collaborative filtering,Content-based filtering o bé híbrid)
Comportament:

- L'usuari valora el conjunt d'ítems mostrats per la recomanació
- El sistema guarda la valoració dels ítems efectuada per l'usuari

Errors possibles i cursos alternatius: -

AFEGIR COMENTARIS

Nom: Afegir Comentaris
Actor: Usuari actiu
Descripció: L'usuari actiu afegeix un comentari a la valoració d'un ítem en concret
Comportament:

- L'usuari prem l'opció "Afegir Comentari"
- El sistema ofereix un camp a l'usuari per afegir el comentari que desitja sobre l'ítem
- L'usuari insereix el comentari
- El sistema guarda el comentari efectuat sobre l'ítem en concret de la recomanació

Errors possibles i cursos alternatius: -

GUARDAR VALORACIONS

Nom: Guardar Valoracions
Actor: Usuari actiu
Descripció: L'usuari actiu guarda les valoracions fetes als ítems de la recomanació
Comportament:

- L'usuari escull l'opció de "Guardar Valoracions" per guardar les reviews fetes als ítems
- El sistema guarda la llista de valoracions i la processarà consegüentment

Errors possibles i cursos alternatius:

- Si l'usuari ha oblidat valorar un ítem de la recomanació el sistema informa de l'error per la falta d'una puntuació sobre aquell ítem

CONSULTAR ESTADÍSTIQUES

Nom: Consultar Estadístiques
Actor: Usuari actiu
Descripció: L'usuari actiu decideix saber amb detall dades d'atributs dels ítems que ha fet al llarg de les valoracions i obtenir més informació sobre quins camps dels atributs dels ítems valorats
Comportament:

- L'usuari escull l'opció de "Consultar estadístiques" per veure detalls sobre els seus gustos dels ítems
- El sistema tanca el programa

Errors possibles i cursos alternatius: -

SORTIR DE L'APLICATIU

Nom: Sortir de l'aplicatiu
Actor: Usuari actiu
Descripció: L'usuari actiu tanca el programa
Comportament:

- L'usuari decideix escollir la opció de "Sortir de l'aplicatiu"
- El sistema tanca el programa i torna a la pantalla d'inici de l'aplicatiu

Errors possibles i cursos alternatius: -

CREAR ÍTEM

Nom: Crear ítem
Actor: Usuari administrador
Descripció: L'usuari administrador crea un ítem al sistema amb el respectiu identificador i els atributs que li corresponen
Comportament:

- L'usuari administrador decideix escollir l'opció de "Crear Ítem"
- El sistema li ofereix l'opció d'afegir els atributs per ítem en qüestió.
- L'usuari afegeix els atributs de l'ítem

- El sistema crea el pertinent ítem, identificat per un id i amb els atributs que l'usuari li ha assignat

Errors possibles i cursos alternatius:

- Si el sistema detecta que ja existeix un ítem creat amb aquest id, el sistema informa de l'error a l'usuari

CONSULTAR ÍTEM

Nom: Consultar ítem
Actor: Usuari administrador
Descripció: L'usuari administrador consulta un ítem al sistema amb el respectiu identificador
Comportament:

- L'usuari administrador decideix escollir l'opció de "Consultar Ítems"
- El sistema li ofereix l'opció de cercar l'identificador de l'ítem que vol cercar
- L'usuari escriu l'id numèric de l'ítem
- El sistema cerca el pertinent ítem, identificat per un id

Errors possibles i cursos alternatius:

- Si el sistema detecta que no existeix un ítem creat amb aquest id, el sistema informa de l'error a l'usuari, i li pregunta si vol afegir-ne un amb aquest id sol·licitat o sí volia cercar aquest ítem identificat per id

MODIFICAR ÍTEM

Nom: Modificar ítem
Actor: Usuari administrador
Descripció: L'usuari administrador modifica un ítem al sistema amb el respectiu identificador i en canvia el valor de certs atributs
Comportament:

- L'usuari administrador decideix escollir l'opció de "Modificar Ítems"
- El sistema li ofereix l'opció de cercar l'identificador de l'ítem que vol
- L'usuari escriu l'id numèric de l'ítem
- El sistema cerca el pertinent ítem, identificat per un id, i si l'ítem és al sistema, es dóna l'opció a l'usuari d'afegir valors als camps
- L'usuari modifica els camps de l'ítem identificat per id
- El sistema guarda els camps amb els canvis efectuats

Errors possibles i cursos alternatius:

- Si al punt 4, el sistema detecta que no existeix un ítem creat amb aquest id, el sistema informa de l'error a l'usuari, i li pregunta si vol afegir-ne un amb aquest id sol·licitat o altrament li demana confirmació sobre l'id de l'ítem.

ELIMINAR ÍTEM

Nom: Eliminar ítem

Actor: Usuari administrador
Descripció: L'usuari administrador esborra un ítem del sistema identificat per id
Comportament:

- L'usuari decideix escollir la opció de "Eliminar ítem"
- El sistema demana inserir l'id de l'ítem que es desitja borrar
- L'usuari insereix al camp l'id de l'ítem que se'n vol desfer
- El sistema cerca el pertinent ítem, identificat per un id, i si l'ítem és al sistema l'esborra seguidament

Errors possibles i cursos alternatius:

- Si el sistema al punt 4 no troba l'id, aquest informa a l'usuari de l'error i li demana confirmació sobre l'id de l'ítem

CONSULTAR USUARI

Nom: Consultar Usuari
Actor: Usuari administrador
Descripció: L'usuari administrador consulta un usuari al sistema mitjançant l'id de l'usuari
Comportament:

- L'usuari decideix escollir la opció de "Consultar Usuari"
- El sistema li ofereix l'opció de cercar l'identificador de l'usuari que vol cercar
- L'usuari escriu l'id numèric de l'usuari
- El sistema cerca el pertinent usuari, identificat per un id

Errors possibles i cursos alternatius:

- Si el sistema detecta que no existeix un usuari creat amb aquest id, el sistema informa de l'error a l'usuari, i li pregunta si vol afegir-ne un amb aquest id sol·licitat o li demana confirmació sobre si volia cercar aquest usuari identificat per id

CREAR RECOMANACIÓ

Nom: Crear Recomanació
Actor: Usuari administrador
Descripció: L'usuari administrador genera recomanacions que posteriorment les obtindrà l'usuari actiu a través de l'interacció amb la funcionalitat "Obtenir Recomanació"
Comportament:

- El sistema posa en marxa el sistema de recomanació un cop l'usuari administrador s'ha registrat al sistema i s'han obtingut els ítems i els ratings

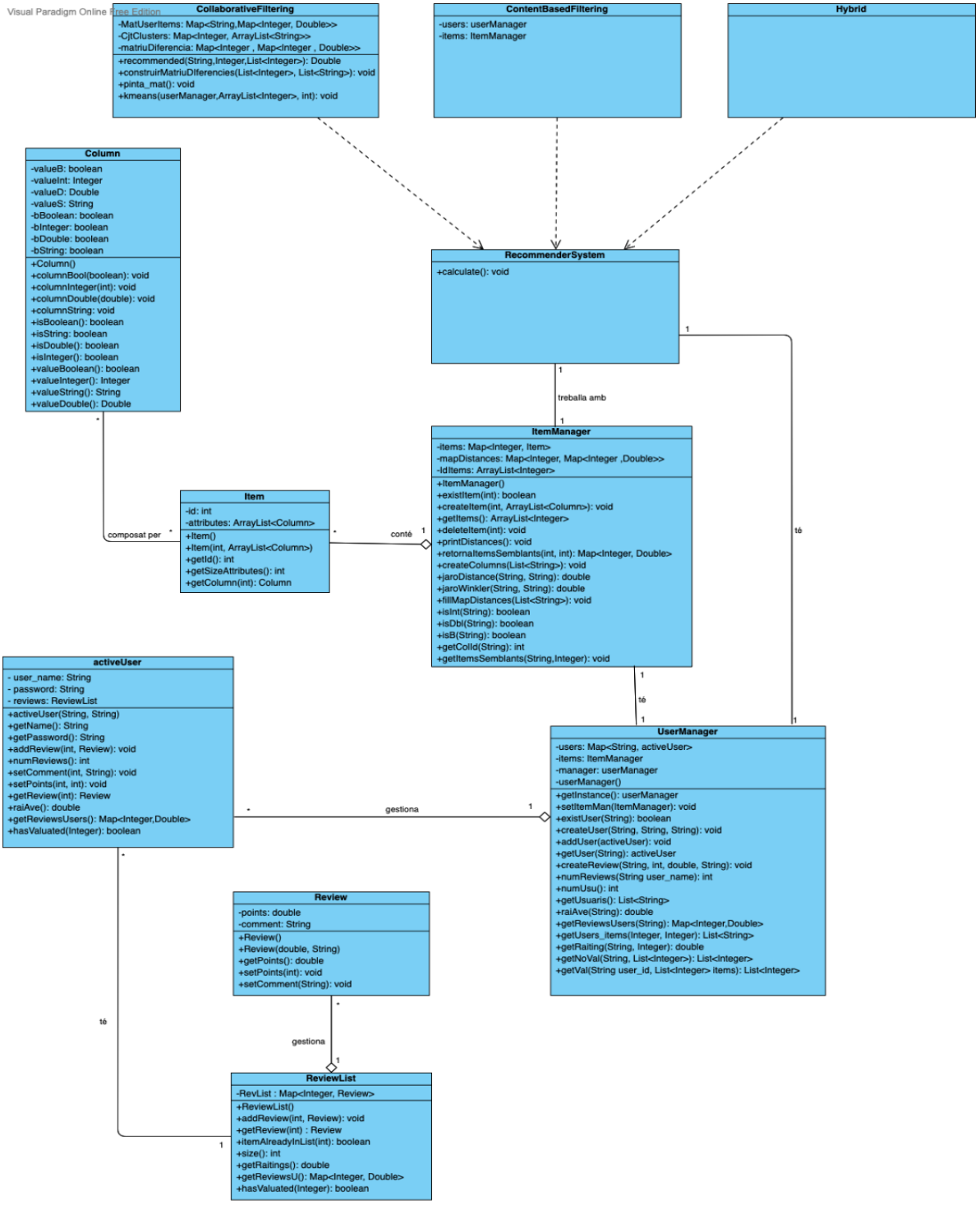
Errors possibles i cursos alternatius:

- Si el sistema detecta que no existeixen ítems o bé usuari actius per fer la recomanació, s'aborta el procés i el sistema informa de l'error

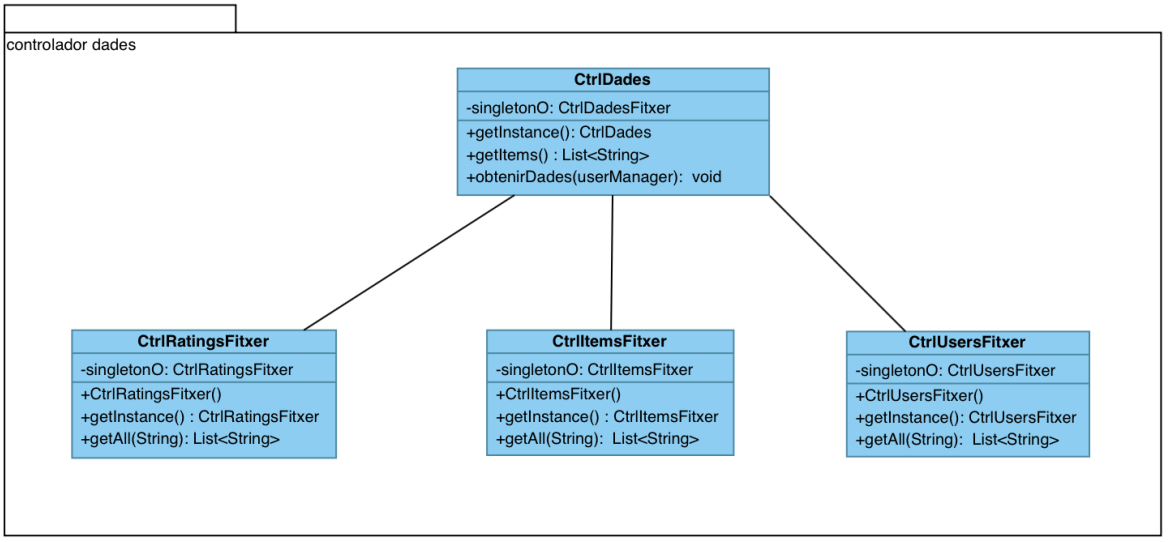
2. MODEL CONCEPTUAL DE DADES

2.1 DIAGRAMA DE CLASSES(UML)

2.1.1 UML MODEL DADES



2.1.2 CONTROLADOR DADES



2.2 DESCRIPCIÓ DELS ATRIBUTS I MÈTODES DE CADA CLASSE

Column

Estructura bàsica d'una columna. Identificada per bBoolean, bInteger, bDouble, bString. Aquests valors determinen quin és el tipus de la instància de la columna. És a dir, només un d'ells pot estar a true i determinar el tipus.

Aquesta classe està només composta de getters i setters. Uns per consultar el valor de la columna i el tipus. I uns altres per dir de quin tipus és la columna i quin valor té.

Review

Estructura bàsica d'una valoració (review). Identificada per points i comment, que determinen els punts i el comentari que hi ha associats a la valoració.

En aquesta classe només hi ha getters i setters. Uns per consultar els punts de la valoració i per fer un set de els diferents paràmetres.

ReviewList

Estructura bàsica de la llista de valoracions (reviews) d'un usuari. Identificada per RevList què és un map de les valoracions que se li ha fet a un conjunt d'ítems.

addReview: Et permet afegir a la llista de valoracions de l'usuari una nova valoració sobre un ítem donat.

itemAlreadyInList: Mira si l'usuari ja ha avaluat l'ítem donat.

getRatings: Per totes les valoracions que ha hi ha a la llista de valoracions es retorna la mitjana de puntuació.

getReviewsU: Retorna els millors tres ítems o menys que ha valorat l'usuari. El resultat està en ordre decreixent ja que s'ha ordenat prèviament el map.

`Item`

Estructura bàsica d'un ítem. Identificat per un id, l'ítem també es composa d'una llista de columnes anomenada attributes.

En aquesta classe només hi ha getters. Un per consultar l'Id, un altre per consultar la mida de la llista d'atributs i un altre per consultar la columna i-èssima.

`ItemManager`

Estructura bàsica d'un gestor d'ítems. Identificat per un map d'ítems, també té un map de distàncies entre tots els ítems i un vector amb els id dels ítems.

esxistItem: Mirar si existeix un ítem en el sistema donat un id

deleteItem: Mira d'esborrar l'ítem amb el id donat de les estructures del map items.

printDistances: Escriu totes les distàncies que hi ha entre cada parella d'ítems. Aquesta funció ens ha servit per fer el nostre propi testing i veure si les distàncies estaven ben calculades.

retornaItemsSemblants: Donat un identificador de ítem volem aconseguir mirant la matriu de distàncies els 3 elements més semblants a l'ítem donat (si no hi ha tants ítems retorna els ítems que existeixen). El resultat serà un map amb les distàncies sobre els ítems més propers.

createColumns: Ens donen una llista de strings on està tot l'input emmagatzemat. Aquesta funció ens permet generar tots els ítems del sistema amb les seves columnes i el seu tipus. L'únic cas especial és quan es llegeix la primera fila de l'input ja que té els noms dels camps de dades de les columnes i només ens fa falta saber on es reserva el id.

jaroDistance: És una funció que donat dos Strings ens pot calcular la distància entre aquests a partir de la semblança entre ells. Tan poden ser strings composats d'una paraula o de moltes.

jaroWinkler: Aquesta funció crida a jaroDistance i calcula la distància entre dos strings. Els dos métodes van lligats i el jaroWinkler prepara la distància entre els strings en diferents casos segons el resultat de jaroDistance.

fillMapDistances: En aquest mètode es crida a createColumns donada la llista de strings. Aleshores del que tracta fillMapDistances és d'omplir la matriu de distàncies per totes les parelles d'ítems del sistema. En aquesta funció és calcula la distància entre cada element segons el tipus d'aquest.

isInt: Una consultora per saber si donat un valor string realment és un integer.

isDbl: Una consultora per saber si donat un valor string realment és un double.

isB: Una consultora per saber si donat un valor string realment és un boolean.

getColId: És una funció que busca en la string que se li passi per paràmetre, a quina columna es troba la columna id i la enregistra. Això serveix per la primera fila de l'input saber a quina columna tindrem tots els id dels ítems.

`activeUser`

Estructura bàsica d'un usuari actiu. Identificat pel nom d'usuari i la seva contrasenya. Aquest objecte també té una llista de valoracions associada que fa referència a tots els ítems que l'usuari ha valorat.

addReview: L'usuari afegeix una valoració a un ítem a la llista de valoracions que té associada.

numReviews: Retorna la mida de quantes valoracions ha fet l'usuari actiu.

raiAve: És una funció que retorna la mitjana de les puntuacions de les valoracions que ha fet l'usuari.

`userManager`

Estructura bàsica d'un gestor d'usuaris. Identificat per un map d'usuaris.

existUser: Donat un nom d'usuari mira si existeix en el map d'usuaris.

createUser: Crea un nou usuari i l'insereix en el map d'usuaris a no ser que ja existeixi en el map.

addUser: Donat un usuari actiu s'insereix en el map d'usuaris a no ser que ja existeixi en el map.

createReview: Aquest mètode crea una valoració i la afegeix a la llista de valoracions que ha té un usuari (el del nom d'usuari que em passen per paràmetres)

numReviews: Donat un usari retorna el número de valoracions que aquest ha dut a terme.

numUsu: La funció retorna el nombre de usuaris enregistrats al sistema.

getUsuaris: Aquesta funció retorna una llista amb tots els usuaris registrats al sistema.

raiAve: Donat un nom d'un usuari retorna la puntuació mitjana de les valoracions que ha fet l'usuari.

getItemsSemblants: El mètode busca per un usuari i una k la recomanació que li farà a l'usuari. La k determina quants ítems similars s'agafen a partir dels ítems que li han agradat a l'usuari. Finalment amb els ítems similars es fa una recomanació a l'usuari de possibles ítems que li poden agradar.

`RecommendationSystem`

Estructura bàsica d'un sistema de recomanació. Aquest concepte és una interface i permet implementar totes les funcions que té, als diferents sistemes recomanadors que tenim en el projecte. Com ara: Collaborative Filtering, Content Based i Hybrid Approach. Les funcions no estan implementades aquí sinó a les classes que fan servir la interface.

`CollaborativeFiltering`

Estructura bàsica d'un sistema recomanador anomenat Collaborative Filtering. Identificat per una matriu fet amb map de maps per identificar les valoracions de tots els usuaris amb tots els ítems. També s'identifica per un conjunt de clústers (agrupacions d'usuaris).

writeCjtClusters: Aquest mètode escriu tots els usuaris que pertanyen a cada un dels clusters d'usuaris. Serveix per veure per terminal si després de fer el kmeans els conjunts d'usuaris estan ben distribuïts.

kmeans: Donada una k, el user manager i una llista de tots els identificadors de ítems. Es calcula l'algorisme kmeans. Gràcies a aquesta funció aconseguim una distribució dels usuaris en diferents clusters. Finalment aquests clústers després de les diverses iteracions del codi han d'esdevenir ser conjunts d'usuaris molt similars a partir de les valoracions que tenen els usuaris sobre tots els ítems existents.

`ContentBasedFiltering`

Estructura bàsica d'un sistema recomanador que pren el nom de Content Based Filtering. Aquesta classe pretén donar una recomanació d'un conjunt d'ítems a l'usuari actiu en funció dels ítems que li han agradat. En aquesta classe implementarem l'estratègia de K-NN. Per tant el que aplicarem en aquesta classe, és donat un ítem i un paràmetre k, retornarem els k ítems més semblants a aquest en funció de la distància entre aquests dos. Per emmagatzemar les distàncies tindrem un map de distàncies(Map<Integer, Map<Integer ,Double>>), on per cada parell d'ítems tindrem la distància entre aquests dos. Una vegada calculades les semblances entre els ítems, agafarem els k ítems amb l'item pres que té les k mínimes distàncies sobre aquest. I retornarem un map amb k ítems amb la distància respectiva a l'ítem que li passem per paràmetre a la capçalera de la funció.

Hybrid

Estructura bàsica del Sistema recomanador que dona la recomanació d'ítems per l'usuari en funció de fer la intersecció entre els ítems recomanats per l'algorisme CollaborativeFiltering i els que dóna el ContentBasedFiltering.

3. ESTRUCTURES DE DADES I ALGORISMES USATS PER A LA IMPLEMENTACIÓ DE LES PRINCIPALS FUNCIONALITATS

3.1 RECOMANACIÓ D'ÍTEMS PER A UN USUARI UTILITZANT COLLABORATIVE FILTERING I CONTENT-BASED FILTERING

En el sistema recomanador es vol desenvolupar un algorisme basat en els gustos d'altres usuaris sobre diferents ítems per tal de determinar les recomanacions que si li poden fer a l'usuari. Com a primera idea necessitem saber que el algorisme es basa en conjunts d'usuaris anomenats clusters, i que la recomanació a l'usuari vindrà determinada pel grup d'usuaris en el que està present l'usuari actiu.

Collaborative Filtering

K-Means

La metodologia d'estudi que es fa servir per agrupar els usuaris en clusters es basa en K-Means. Aquesta solució al problema ens diu que s'han d'inicialitzar k centroides en l'espai de manera aleatoria. Els centroides escolliran quins són els usuaris que tenen més aprop en l'espai i miraran de crear clusters d'usuaris amb aquests.

Un cop tenim k clusters a l'espai hem de fer que aquests usuaris realment estiguin amb usuaris relacionats en un mateix cluster. Els grups d'usuaris s'han creat a partir d'un centroide situat aleatòriament, per tant la única manera de fer els conjunts óptims és resituar els centroides i col·locant-los en els llocs en els que realment han d'estar.

Ara doncs per fer aquesta funcionalitat es calculen els punts mitjos de cada cluster per fer que aquests siguin els nous centroides.Un cop s'ha provocat aquesta modificació només fa falta repetir els passos anteriors per anar acurant les posicions dels centroides. És a dir que cal tornar a calcular els nous conjunts d'usuaris segons els nous centroides mirant les distàncies dels usuaris als k centroides.

Slope One

L'algorisme Slope One tracta de predir les valoracions que farà l'usuari actiu sobre ítems que encara no ha valorat. Aquesta predicció ve donada a partir d'usuaris que tenen gustos similars i han valorat diferents ítems.

Explicació de com s'ha desenvolupat l'algorisme:

La recomanació collaborative filtering està creada com a una classe que parteix de la interface RecommendationSystem. En aquesta classe tenim una funció anomenada kmeans, la funció necessita la instància de UserManager amb tots els usuaris registrats en el sistema. També necessita una llista amb tots els id dels ítems que existeixin. Finalment necessita un paràmetre k.

Per crear les dimensions en les que els usuaris estan a l'espai hem creat una estructura de dades basada en map de maps per veure la valoració que ha fet cada usuari sobre tots els ítems del sistema. El MatUserItems té el primer map que identifica el usuari amb una String que fa referència al seu nom d'usuari, el map interior té un integer per determinar el id del item i un double per dir quina valoració ha puntuat aquell usuari sobre l'ítem. Com bé hem dit abans per cada usuari es tindrà un map de tots els ítems, encara que no els hagi puntuat.

El primer pas de la funció és omplir la Matriu MatUserItems on la puntuació que ha fet un usuari sobre els ítems que no ha valorat és 2.5. Aquest paràmetre hem cregut que és el més adient per poder situar un usuari a l'espai ja que les puntuacions van de 0 a 5 i les coordenades que són nules creiem que han de tenir la puntuació mitja per no inferir directament en els extrems de la coordenada.

Fet això el següent pas consisteix en elaborar una llista ListKelem composada de maps. La llista tindrà tots els centroides (k) amb els maps que fan referència a les coordenades d'aquests centroides a l'espai. Aleshores pels k centroides setejem unes coordenades aleatòries en cada una de les dimensions.

Un cop tenim els centroides hem decidit que aplicarem l'algorisme kmeans 50 vegades per tenir un resultat prou acurat. En cada una d'aquestes iteracions primer de tot passem per tots els centroides i per cada conjunt d'usuaris del centroide. Per cada usuari mirem les seves puntuacions als ítems fent servir MatUserItems i anem calculant la mitjana en coordenades del Cluster. Quan tenim les noves coordenades li diem al centroide la seva nova posició a l'espai.

Després d'haver calculat els nous centroides s'han de assignar tots els usuaris en un cluster de nou, això ho fem calculant la a distància de cada usuari per cada cluster. Hem fet servir dos fors un a dins de l'altre per passar primer per tots els usuaris i l'interior per passar per tots els centroides.

L'algorisme ens donarà una resposta molt acurada per no dir la millor després de 50 iteracions de tal manera que els clústers estan plens d'usuaris similars.

Content Based Algorithm

L'algorisme content based té una idea general per fer una recomanació a l'usuari actiu sobre els ítems que ha valorat. Es volen mirar ítems similars als que més li hagin agradat a l'usuari i fer una recomanació amb els que siguin més semblants als ítems preferits per l'usuari.

Com que la llista d'ítems del programa pot ser de totes les mides i de tota mena de dades, la nostra resolució es basa en detectar el tipus de cada columna i mirar de guardar i comparar les dades segons el seu tipus independentment del significat d'aquells valors.

Per començar l'algorisme cal llegir les dades del fitxer d'entrada que se'ns proporciona, es guarda tot l'input llegit en una llista de Strings. A partir d'aquí el nostre algorisme viatja per tots aquests Strings llegits i determina el tipus de cada columna. A més també emmagatzema la columna on es guarda el id del item, per així tenir-los localitzats i poder-los anar guardant en una llista d'enters amb els id dels ítems.

Com que les columnes volem gestionar de quin tipus són hem creat la classe columna amb la informació a partir de booleans del tipus que desenvolupa la instància concreta de la columna en el sistema. També té els diferents tipus de valors i aquests només s'omplirà el valor corresponent al tipus de la columna. Per aconseguir el tipus de cada columna s'ha utilitzat la funció parse amb el tipus que es vol analitzar.

Un cop hem arribat fins aquest punt ara cal fer el calcul de distancies entre els diferents ítems que els enregistrarem a una matriu de distàncies feta amb un map de maps. El map de maps s'estableix per comparar dos ítems (identificador del map exterior i identificador del map interior) i poder assolir la seva distància. La distància hem fet servir un algorisme bàsic en el que fem:

- Quan comparem booleans veiem que si els dos ítems tenen valors oposats es suma 1 a la distància.

- Quan comparem integers apliquem la fórmula |i1-i2|/(i1 + i2) per calcular la distància entre els ítems. En el cas en el que el divisor sigui 0 fem la mateixa operació sumant 1 al divisor.

- Quan comparem doubles apliquem la fórmula |d1-d2|/(d1 + d2) per calcular la distància entre els ítems. En el cas en el que el divisor sigui 0 fem la mateixa operació sumant 1 al divisor.

- Quan comparem strings fem servir una funció ja desenvolupada anomenada jaroWinkler que pot comparar tant strings compostos per una paraula com strings compostos de moltes.

Per finalitzar l'algoritme es busca sobre cada ítem que més li ha agradat a l'usuari els k ítems més semblants. Si la k és més gran que 3 s'agafen només els tres millors ítems amb distància menor. A partir d'aquí és fa la recomanació a l'usuari amb els ítems més semblants als que li han agradat.

4. RELACIÓ DE LES CLASSES IMPLEMENTADES PER CADA MEMBRE DEL GRUP

- Main:
    - Controladors de Dades:
        - CtrlDades: Marta
        - CtrlItemsFitxer: Marta
        - CtrlRatingsFitxer: Marta
        - CtrlUsersFitxer: Marta

- Domini:
  - Controlador Domini:
    - CtrlDomini: No implementat per aquesta entrega
  - Model:
    - Column: Pol
    - ActiveUser: Miquel
    - UserManager: Miquel
    - RecommenderSystem: Marta
    - CollaborativeFiltering: Miquel & Marta
    - ContentBasedFiltering: Pol & Aftab
    - HybridApproach: No implementat aquesta entrega
    - Item: Aftab
    - ItemManager: Aftab
    - Review: Pol
    - ReviewList: Pol
    - Avaluation: Miquel & Marta

- Testing:
  - Drivers:
    - JUnit:
      - driverActiveUser: Miquel
      - driverUserManager: Miquel
      - driverItem Miquel
      - driverItemManager Miquel
      - CollaborativeFilteringTest Aftab
  - Stubs:

5. CODI

5.1 MODEL

El codi de totes les classes i controladors a GitLab.

5.2 FUNCIONALITATS PRINCIPALS

El codi de les classes de domini associades a les funcionalitats principals dels algorismes de recomanació de l'aplicació es poden trobar al corresponent package(algoritmos) a GitLab.

6.TESTING

6.1 JOCS DE PROVA

Executables que permetin provar totes les classes implementades, mitjançant *stubs* i *drivers*.

6.2 JUNIT

Tests unitaris de JUnit per com a mínim una classe (a consensuar amb el tutor). Tests que l'equip ha fet servir per provar el seu projecte (a afegir als *drivers* i *stubs*: una cosa no exclou l'altre).