

Sistema de recomanació

Tercer lliurament (3.0)

Projectes de programació (PROP)

Grau en Enginyeria Informàtica - FIB - UPC

Curs 2021-2022, Quadrimestre de Tardor

Equip 4.2

Ferran De La Varga Antoja (ferran.de.la.varga)

Alexandru Dumitru Maroz (alexandru.dumitru)

Pablo José Galván Calderón (pablo.jose.galvan)

Pol Rivero Sallent (pol.rivero)

Índex

1. Repartiment de classes	2
2. Manual d'usuari	3
2.1 Conceptes bàsics	3
2.2 Inicialització del sistema	4
2.3 Crear/eliminar usuaris i ítems	6
2.4 Obténir informació sobre usuaris i ítems	9
2.5 Configuració i avaluació del model	10
2.5.1 Algorismes de recomanació	10
2.5.2 Funcions de distància	11
2.5.3 Pesos dels atributs	12
2.5.4 Avaluar model	13
2.6 Crear un atribut derivat	15
2.7 Guardar i restaurar estat del sistema	18
3. Testing del sistema	19
3.1 Recomanació basada en K-Means i Slope1	21
3.2 Recomanació basada en KNearestNeighbours	22
3.2 Recomanació híbrida	23

1. Repartiment de classes

- **Value (+fills):** Pol Rivero
- **Attribute (+fills):** Pol Rivero
- **Operator (+fills):** Pol Rivero
- **DistanceStrategy (+fills):** Pol Rivero
- **ModelCtrl:** Pol Rivero
- **ClusterCtrl:** Alexandru Dumitru
- **Cluster (+ClusterMember):** Alexandru Dumitru
- **User:** Pol Rivero
- **Rating:** Pol Rivero
- **Item:** Pol Rivero
- **KNNItem:** Pablo José Galván
- **Recommendation (KNN):** Pablo José Galván
- **Recommendation (Slope1):** Ferran de la Varga
- **Recommendation (Mixt):** Pablo José Galván
- **Capa dades:** Pol Rivero
- **Capa de presentació:** Alexandru Dumitru
- **Driver:** Ferran de la Varga

Cal tenir en compte que aquest repartiment de classes no necessàriament és representatiu de la feina que ha fet cada persona, ja que no té en compte la redacció de la documentació o altres elements.

2. Manual d'usuari

2.1 Conceptes bàsics

Aquest programa consisteix en un motor de recomanacions, que un conjunt d'usuaris pot utilitzar per obtenir recomanacions personalitzades d'entre un conjunt de ítems.

Existeix un usuari administrador, que s'encarrega de configurar el programa. Aquest manual va dedicat principalment a l'administrador, ja que la majoria de funcionalitats del sistema estan reservades només per aquest usuari.

En el sistema (també anomenat “el model”) existeixen “**ítems**”, que són entitats genèriques que poden ser recomanades a un usuari; “**atributs**”, que són les propietats que tenen tots els ítems; i “**valors**”, que són instàncies concretes d'un atribut.

Per exemple, si es vol utilitzar el sistema per recomanar pel·lícules, un *ítem* podria ser la pel·lícula Titanic. En el model podrien existir 2 *atributs*: “Títol” i “Duració”, que en aquest ítem poden prendre els *valors* “Titanic” i “120”, respectivament.

El sistema també guarda quins **usuaris** hi ha registrats, i quines **valoracions** (ratings) d'ítems han fet (quines puntuacions han donat a cada ítem).

En quant als *atributs*, és convenient saber que poden ser de 3 tipus possibles: atributs **booleans** (per a valors del tipus TRUE/FALSE), atributs **numèrics** (nombres enters i amb decimals) i atributs **categòrics** (strings i llistes d'etiquetes). A l'exemple anterior, “Títol” seria un atribut categòric i “Duració” seria un atribut numèric.

Cada atribut també té associat un **pes** (weight), que determina quanta importància tenen (en els algorismes de recomanació) els valors dels ítems en aquest atribut.

2.2 Inicialització del sistema

Per executar el programa, cal situar-se dins la carpeta principal i fer servir la comanda `make gui` des de la terminal.

S'obrirà una finestra de login. Per entrar com a usuari cal introduir l'ID de l'usuari; si és el primer cop que s'inicialitza el sistema, s'haurà d'introduir en el camp la paraula "admin" per entrar com a administrador i configurar com serà el sistema.

El primer cop que l'administrador accedeix al sistema, es demana la ruta d'un fitxer de configuració (extensió `.info`) on hi haurà la configuració de com ha de ser el sistema i el nom dels fitxers `.csv` on hi haurà les dades. A la carpeta DATA s'inclouen alguns datasets amb aquests fitxers de configuració ja creats.

Un exemple de fitxer de configuració pot ser:

ITEMS=items.csv

RATINGS=ratings.db.csv

- Nom del fitxers on hi ha la informació de tots els ítems i les valoracions, respectivament (paths relatius al directori on es troba el fitxer de configuració).

ID_NAME=id

- Nom de la columna (del fitxer d'ítems) que s'utilitzarà com a identificador dels ítems.

MAX_RATING=5.0

- Màxim rating que pot tenir un ítem.

USERID=userId

ITEMID=itemId

RATING=rating

- Noms de les columnes del fitxer de ratings

KNOWN=ratings.test.known.csv

UNKNOWN=ratings.test.unknown.csv

- Aquests 2 paràmetres són opcionals i només s'utilitzen en cas que es vulgui fer testing del sistema. Contenen els paths dels fitxers de known (queries) i unknown ("autèntiques valoracions" per avaluar el resultat de les queries).

ALGORITHM=KNearestNeighbours

DISTANCE=Euclidean

- Algorisme i funció de distància predeterminats pel càlcul de recomanacions. Els algorismes són KmeansSlopeOne, KNearestNeighbours i Hybrid. Les distàncies són: Euclidean, Manhattan, Average i AverageSquared. Més endavant es poden canviar a la pestanya "Configure Model".

TITLE=title

OVERVIEW=overview

AVGRATING=vote_average

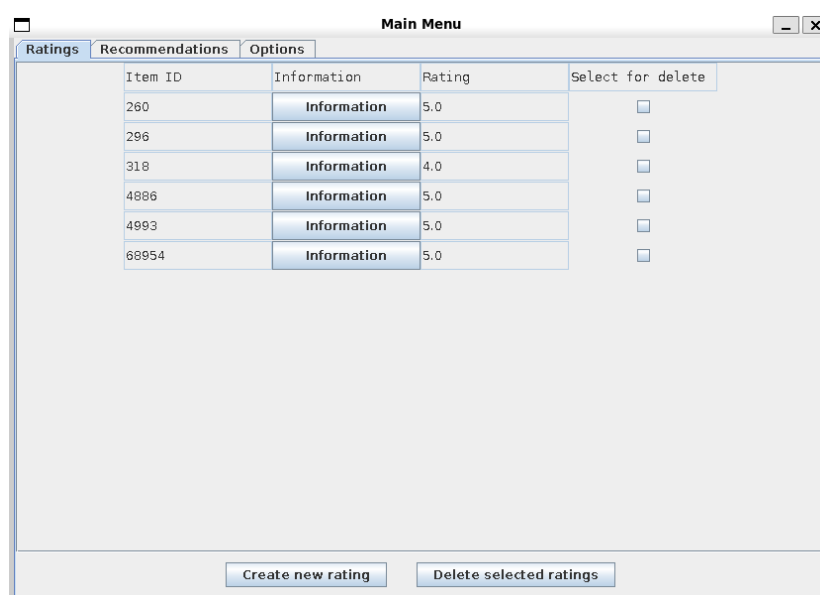
IMG_URL=poster_path

- Nom de les columnes (del fitxer d'ítems) que cal utilitzar per a mostrar informació dels ítems en un format amigable pels usuaris (en lloc de mostrar únicament el ItemID).

Un cop l'administrador ha seleccionat el fitxer de configuració, el model s'inicialitza i s'obre el menú principal. Es poden consultar els ítems, usuaris i atributs que formen el sistema en forma de llistes (cal seleccionar a les pestanyes de la part superior quina llista es vol consultar).

Ara que l'administrador ha inicialitzat el sistema, els usuaris el poden començar a utilitzar. Per iniciar sessió com a usuari, es pot anar a la pestanya "Users" i copiar el UserID d'un dels usuaris. A continuació, cal seleccionar el botó de Logout (a la pestanya "Options") per tornar a la pestanya de login i enganxar el UserID.

El menú principal d'un usuari és molt més limitat que el de l'administrador. L'únic que un usuari pot fer és consultar i modificar els seus ratings a ítems, obtenir recomanacions (a partir dels paràmetres que l'administrador ha configurat i l'usuari no pot canviar) i evidentment fer logout.

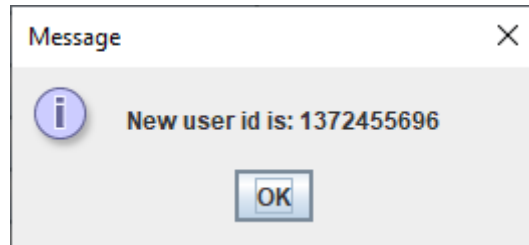


Menú principal d'usuari

Donat que les funcionalitats accessibles a un usuari són un subconjunt de les que són accessibles a un administrador (el qual pot, fins i tot, crear i esborrar ratings fets per qualsevol usuari del sistema), la resta d'aquest manual es centrarà exclusivament en l'ús del sistema des del punt de vista de l'administrador.

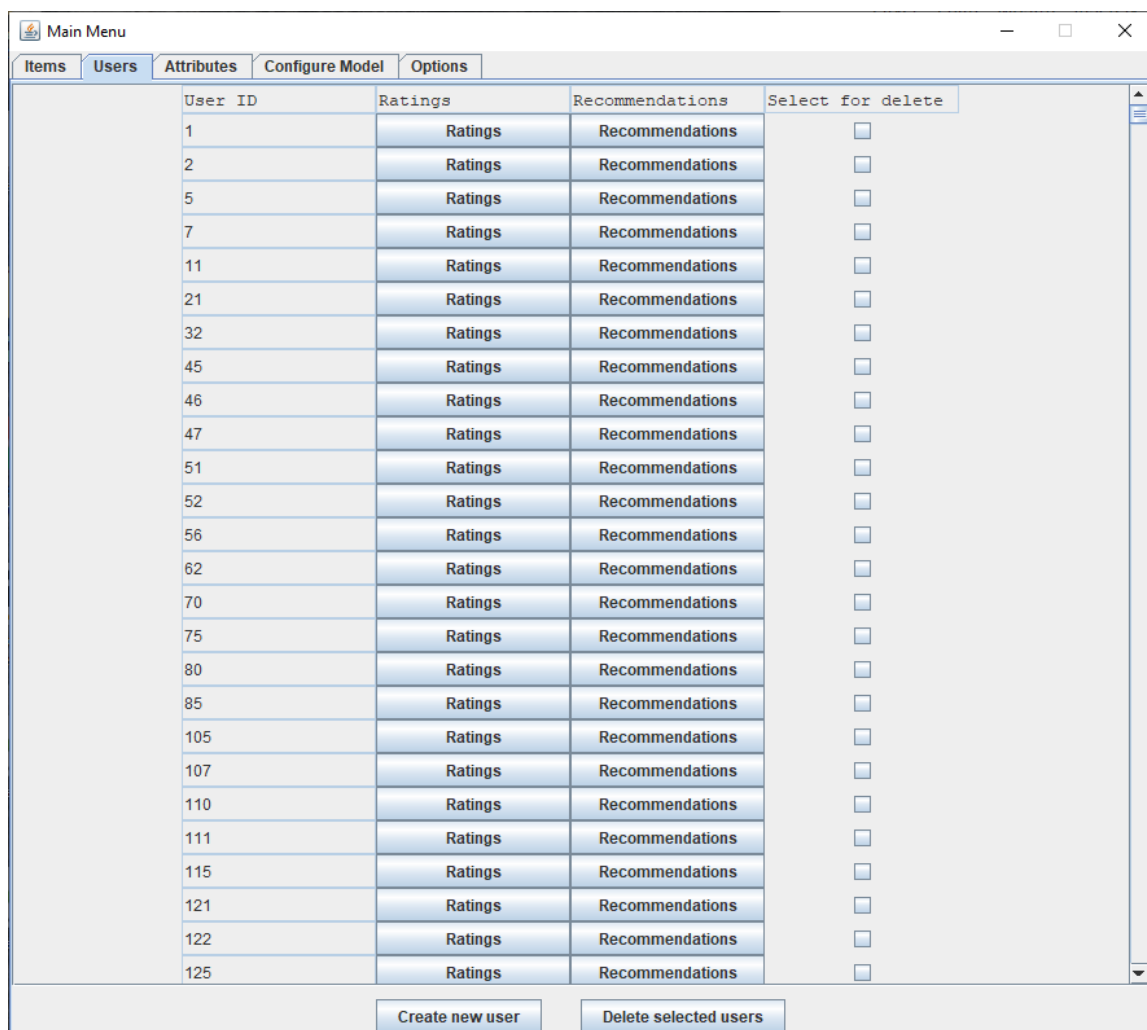
2.3 Crear/eliminar usuarios i ítems

Per crear un nou usuari, l'administrador ha de navegar a la pestanya "Users" i fer clic al botó "Create new user". Es crearà un nou usuari amb un nou id aleatori (no existent) que serà mostrat per pantalla. Des d'aquest moment el id mostrat passarà a ser vàlid en la pantalla de login i podrà començar a crear valoracions i rebre recomanacions.



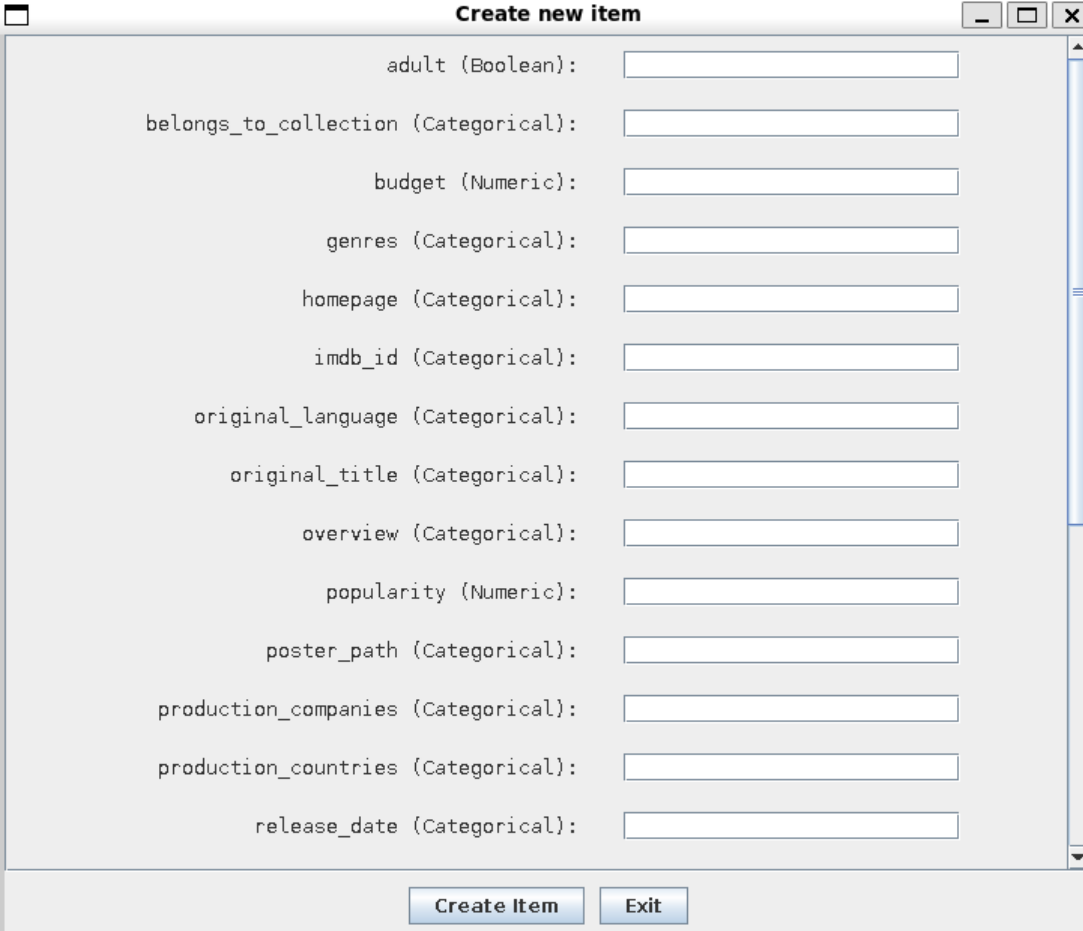
Pantalla de notificació conforme s'ha creat un nou usuari

Per eliminar usuaris, cal seleccionar-los i clicar el botó "Delete selected users". Sortirà una pantalla de confirmació i seran eliminats del sistema.



Exemple pantalla Users

Per crear un nou ítem, s'ha d'anar a la pestanya "Items" i clicar el botó "Create new item". Seguidament, apareixerà una finestra per omplir tots els atributs de l'ítem amb el tipus de valor a introduir (categòric, booleà o numèric) per a cada atribut. És obligatori introduir tots els valors dels atributs numèrics i booleans; els categòrics es poden deixar en blanc. Un cop s'ha clicat "Create", el sistema notificarà a l'administrador quin és el nou Item ID de l'ítem creat.



Create new item

adult (Boolean):

belongs_to_collection (Categorical):

budget (Numeric):

genres (Categorical):

homepage (Categorical):

imdb_id (Categorical):

original_language (Categorical):

original_title (Categorical):

overview (Categorical):

popularity (Numeric):

poster_path (Categorical):

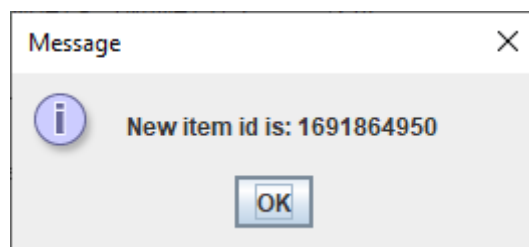
production_companies (Categorical):

production_countries (Categorical):

release_date (Categorical):

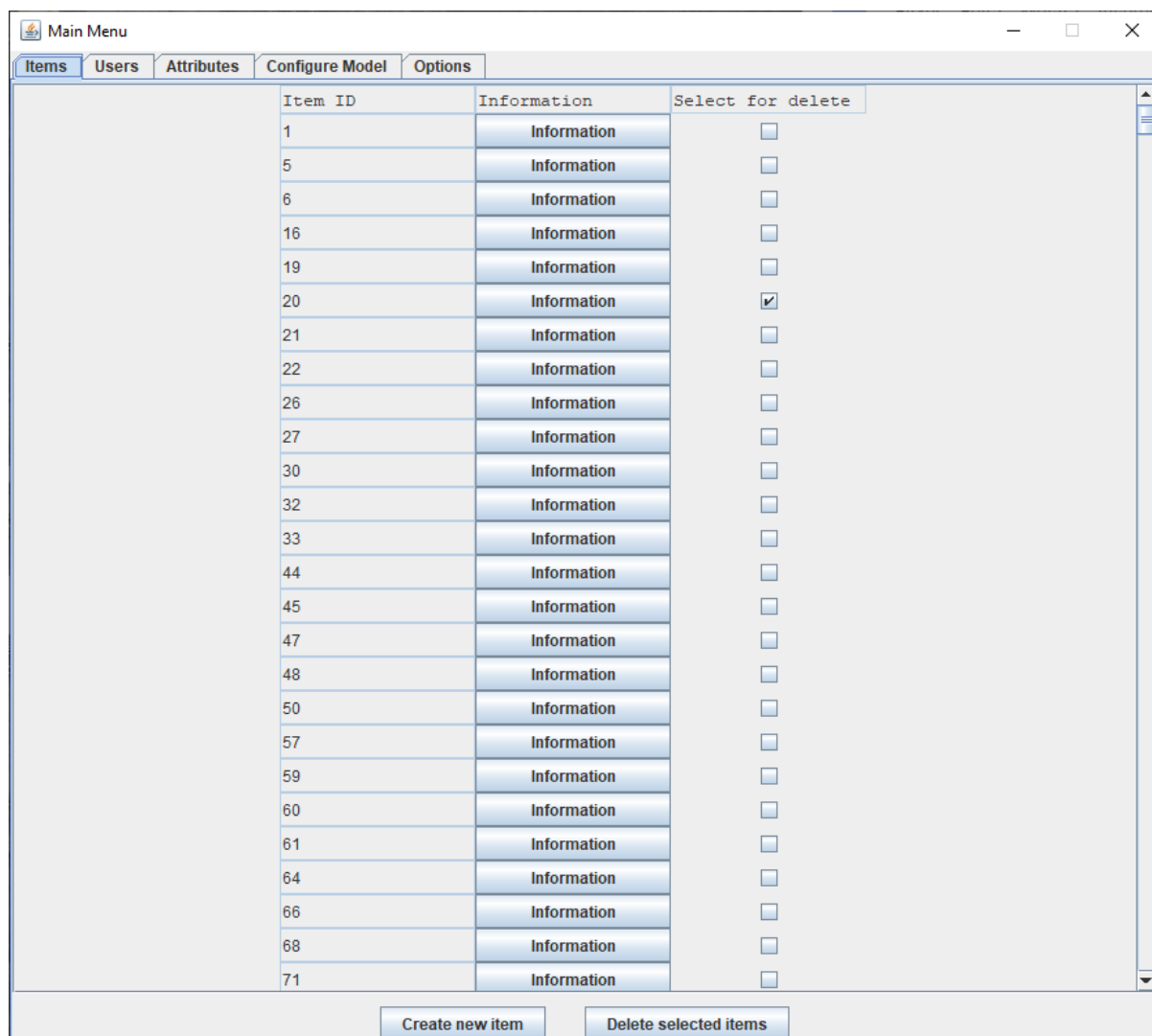
Create Item **Exit**

Exemple per introduir els camps dels atributs del nou ítem a crear

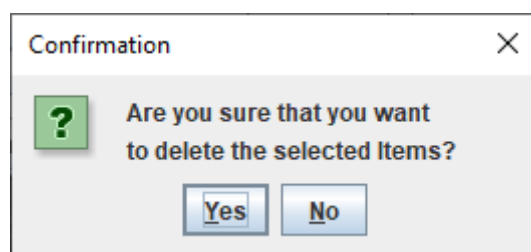


Nou ítem creat

El procés d'eliminar un ítem és el mateix que el d'usuaris però dins de la pestanya "Items" i seleccionant ítems.



Exemple pantalla Items. Està seleccionat l'ítem 20. Si el volem esborrar cal clicar el botó "Delete selected Items"



Pantalla de confirmació d'eliminació

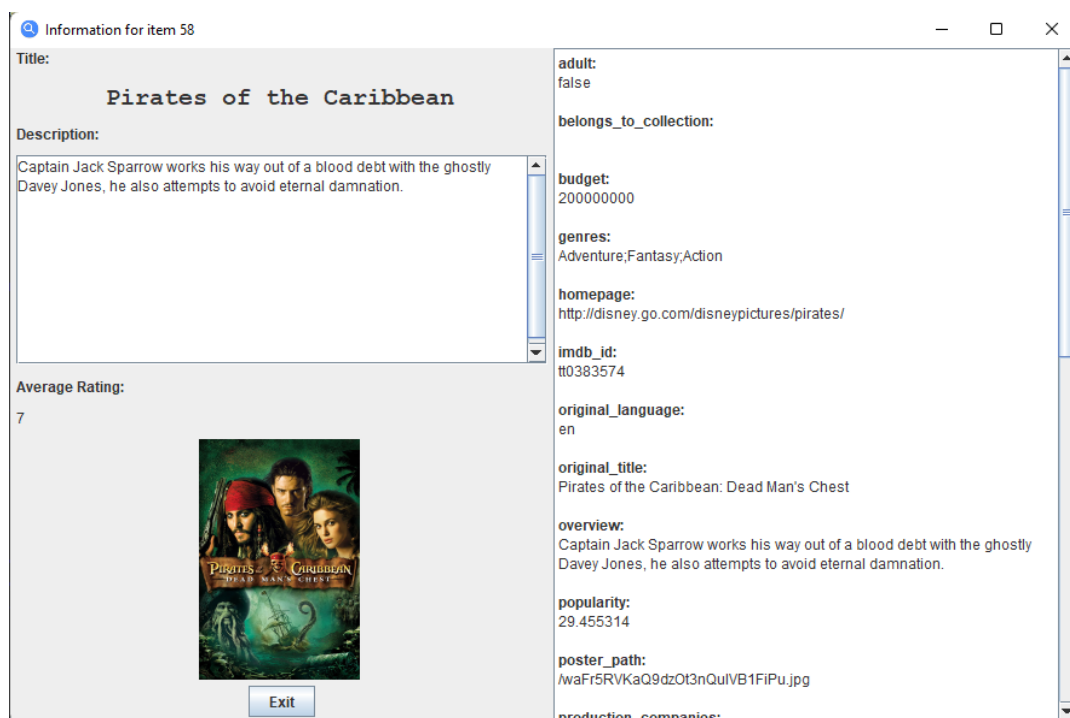
2.4 Obtenir informació sobre usuaris i ítems

A més de crear i eliminar ítems i usuaris, l'administrador també pot utilitzar els botons de la llista per obtenir informació sobre un usuari o ítem concret.

Per cada **ítem**, hi ha disponible el botó "Information", que obre una finestra d'informació detallada per a l'ítem corresponent. Actualment el sistema mostra 4 dades en aquesta finestra, que considerem que són aplicables a pràcticament tots els datasets pels quals és raonable utilitzar el nostre sistema. Les dades mostrades (per a les quals cal indicar quina columna utilitzar en el fitxer `.info`) són: títol, descripció, puntuació mitjana i imatge.

Val la pena explicar especialment el funcionament del camp d'imatge:

- Si la columna indicada conté un enllaç (URL) a una imatge, el sistema obtindrà la imatge realitzant una petició a internet i la mostrarà un cop aquesta hagi estat obtinguda.
- Si la columna no conté un enllaç, el sistema utilitzarà l'API de Google Search per realitzar una cerca de 10 imatges. Es descarten aquelles imatges no rellevants (de moment es descarten les que provenen de repositoris de Github, ja que són thumbnails autogenerades) i es mostra la imatge més rellevant no descartada.



Per cada **usuari**, l'administrador pot prémer 2 botons: "Ratings" i "Recommendations".

El botó "Ratings" mostra la llista de valoracions d'aquell usuari (equivalent a iniciar sessió com a aquell usuari i navegar a la pestanya "Ratings"). Des de la finestra de ratings es pot crear una nova valoració o eliminar valoracions existents.

El botó "Recommendations" mostra una llista ordenada amb els 20 ítems (no valorats per l'usuari) recomanats per l'algorisme.

En els dos casos, les llistes mostrades tenen botons de "Information" per obrir la finestra d'informació detallada, tal com s'ha descrit anteriorment.

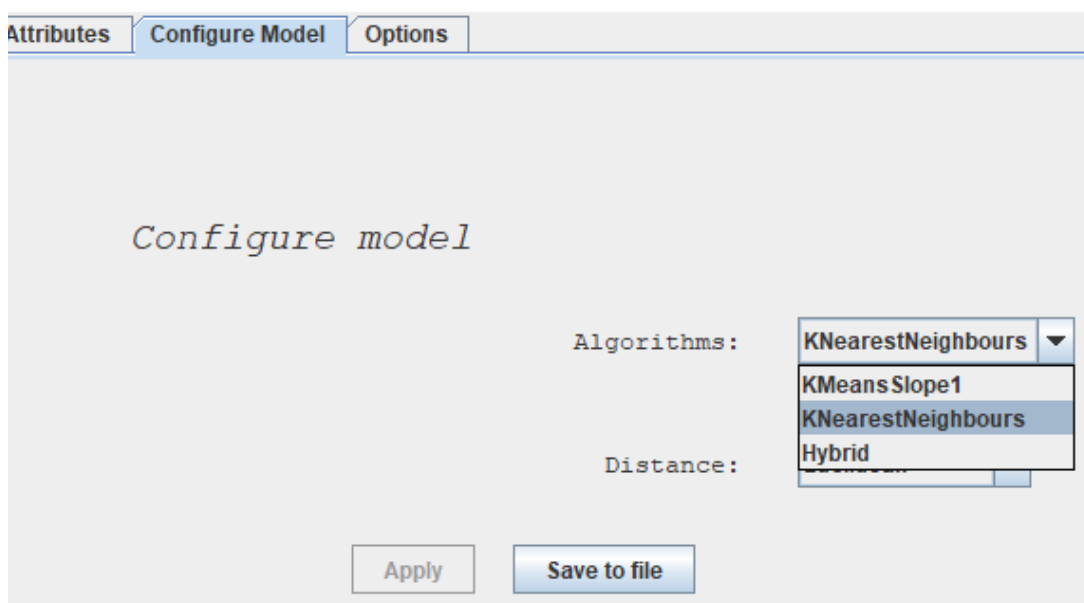
2.5 Configuració i avaluació del model

2.5.1 Algorismes de recomanació

Per saber en què consisteixen i com han estat programats els algorismes de recomanació consultar apartat 3 de la segona entrega.

Hi ha tres possibles algorismes a seguir: KmeansSlopeOne, KNearestNeighbours i Hybrid (combina KMeans + KNN).

Com s'ha explicat al punt 2.2, per inicialitzar el sistema, l'algorisme de recomanació a utilitzar s'indica al fitxer .info. Si es vol canviar posteriorment s'ha d'anar a la pestanya "Configure Model".



Selecció de l'algorisme de recomanacions

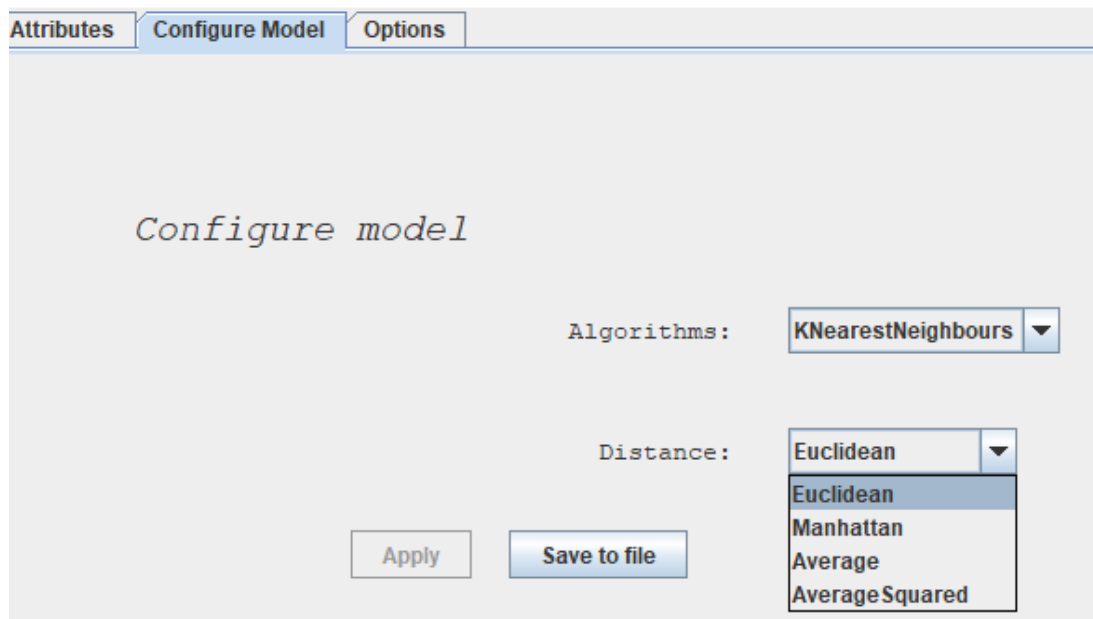
Un cop s'ha seleccionat un algorisme de recomanacions, el botó de "Apply" passa a estar disponible. Quan es prem, es guarda la configuració i totes les pròximes queries es resoldran utilitzant l'algorisme seleccionat.

2.5.2 Funcions de distància

Hi ha tres possibles funcions de distància a seguir: Euclidean, Manhattan, Average i AverageSquared.

Per saber com es calculen aquestes distàncies, cal consultar l'apartat 3.5 del segon lliurament.

Com s'ha explicat al punt 2.2, per inicialitzar el sistema, l'algorisme utilitzat pel càlcul de distàncies s'indica al .info. Si es vol canviar posteriorment, s'ha d'anar a la pestanya "Configure Model".



Canviar algorismes de distància

Igual que amb els algorismes de recomanació, cal utilitzar el botó "Apply" perquè tingui efecte.

2.5.3 Pesos dels atributs

Els pesos dels atributs són necessaris per als algorismes de recomanacions KNN i Hybrid. Si un atribut A té un pes més gran que un altre B, llavors, en el càlcul de recomanacions es tindrà més en compte l'atribut A que l'atribut B.

Per a canviar el pes d'un atribut, s'ha d'anar a la pestanya "Attributes" i modificar els camps de pesos (amb un nombre positiu). Posteriorment cal utilitzar el botó "Apply Weights", que només està disponible quan algun dels camps ha estat modificat.

Items	Users	Attributes	Configure Model	Options		
		belongs_to_collection	1.0	Categorical	Create Derived	<input type="checkbox"/>
		budget	1.0	Numeric	Create Derived	<input type="checkbox"/>
		genres	1.0	Categorical	Create Derived	<input type="checkbox"/>
		homepage	1.0	Categorical	Create Derived	<input type="checkbox"/>
		imdb_id	1.0	Categorical	Create Derived	<input type="checkbox"/>
		original_language	1.0	Categorical	Create Derived	<input type="checkbox"/>
		original_title	1.0	Categorical	Create Derived	<input type="checkbox"/>
		overview	1.0	Categorical	Create Derived	<input type="checkbox"/>
		popularity	1.0	Numeric	Create Derived	<input type="checkbox"/>
		poster_path	1.0	Categorical	Create Derived	<input type="checkbox"/>
		production_companies	1.0	Categorical	Create Derived	<input type="checkbox"/>
		production_countries	1.0	Categorical	Create Derived	<input type="checkbox"/>
		release_date	1.0	Categorical	Create Derived	<input type="checkbox"/>
		revenue	1.0	Numeric	Create Derived	<input type="checkbox"/>
		runtime	1.0	Numeric	Create Derived	<input type="checkbox"/>
		spoken_languages	1.0	Categorical	Create Derived	<input type="checkbox"/>
		status	1.0	Categorical	Create Derived	<input type="checkbox"/>
		tagline	1.0	Categorical	Create Derived	<input type="checkbox"/>
		title	1.0	Categorical	Create Derived	<input type="checkbox"/>
		video	1.0	Boolean	Create Derived	<input type="checkbox"/>
		vote_average	1.0	Numeric	Create Derived	<input type="checkbox"/>
		vote_count	1.0	Numeric	Create Derived	<input type="checkbox"/>
		keywords	50	Categorical	Create Derived	<input type="checkbox"/>

Delete selected attributes Apply Weights

Atributs dels ítems

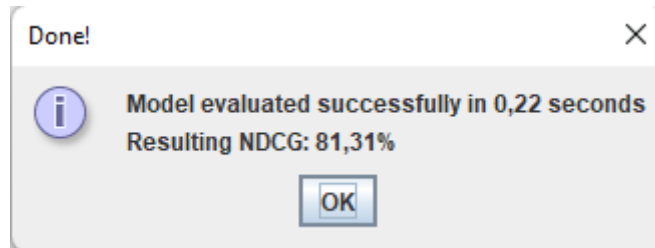
Des d'aquesta pestanya també es poden eliminar atributs (columnes) del model que no s'utilitzin. Cal tenir en compte que, des del punt de vista de l'algorisme, posar el pes d'un atribut a 0 té el mateix efecte que eliminar-lo.

Els atributs que s'utilitzen com a títol, descripció, rating i imatge en la finestra d'informació (establerts en el fitxer .info) no es poden eliminar, de forma que si no es volen utilitzar en l'algorisme caldrà posar el seu pes a 0.

2.5.4 Avaluar model

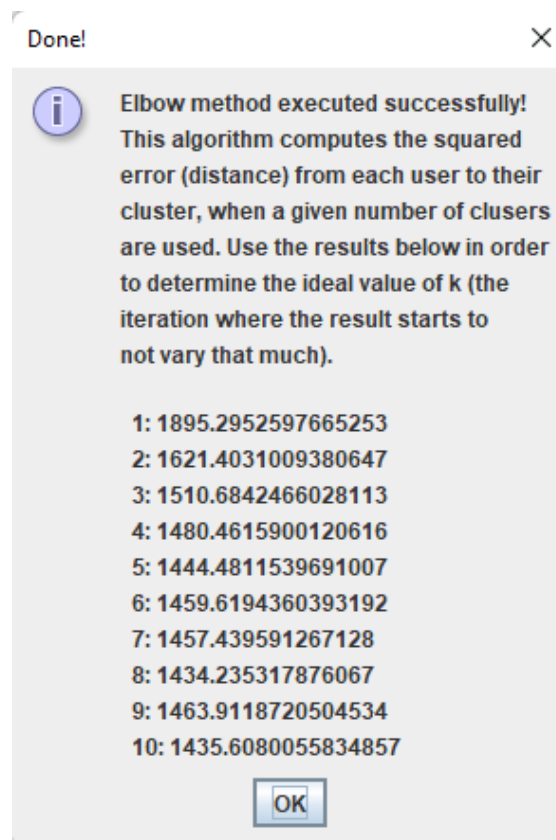
Per avaluar el model cal anar a la pestanya “Configure Model” i a l’apartat de Testing hi trobareu diferents botons:

- **Evaluate:** Avalua el model segons els conjunts Known i Unknown indicats al moment de la creació del model.



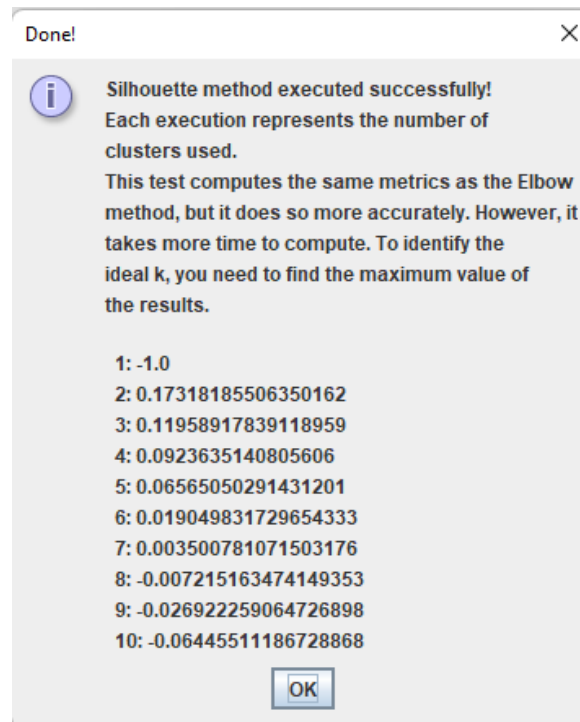
Resultat de Evaluate amb el seu Discounted Cumulative Gain

- **Test Elbow Method:** Útil per determinar el nombre òptim de clusters utilitzant el Elbow method (detectar el punt on l’error deixa de decreïxer ràpidament).



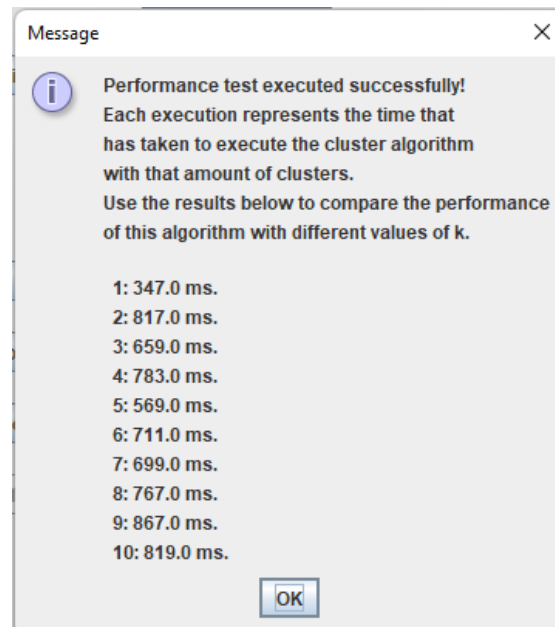
Resultat d’executar l’Elbow method

- **Test Silhouette Method:** Serveix per determinar el nombre de clusters, igual que l'Elbow method



Resultat d'executar el Silhouette Method

- **Test Performance:** Simplement executa l'algorisme amb diferents valors de k (nombre de clusters) i mostra el temps d'execució amb cada un.



Resultat d'executar el Silhouette Method

2.6 Crear un atribut derivat

El sistema permet crear un nou atribut numèric a partir d'un atribut (numèric o categòric) ja existent.

El procés és similar a configurar la *access list* d'un firewall. En primer lloc, cal introduir una llista de condicions, cada una acompanyada del valor que prendrà el nou atribut si la condició és certa. Per cada un dels ítems, les condicions es comproven de forma seqüencial i, si una retorna *true*, s'agafa el valor d'aquesta per al nou atribut. Si totes les comprovacions retornen *false*, s'utilitza el valor per defecte introduït.

Per exemple, donada una llista d'operacions:

Operador	Argument	Valor si <i>true</i>
<	10	50
<=	20	150
=	30	123
Valor per defecte: 10		

Si creem un nou atribut derivat a partir de "Atribut A", els seus valors serien:

itemId	Atribut A	Nou atribut derivat	Notes
1	5	50	Es compleix 5<10
2	10	150	Falla 10<10, es compleix 10<=20
3	20	150	Falla 20<10, es compleix 20<=20
4	30	123	Fallen els 2 primers, es compleix 30=30
5	40	10	Fallen totes les comprovacions

També es poden utilitzar els valors INVALID (per fer que el valor del nou atribut no es tingui en compte en el càlcul de distàncies) i SAME (per copiar el valor de l'atribut antic en el nou). Cal tenir en compte que els atributs creats sempre són numèrics, de forma que SAME només es pot utilitzar si l'atribut original és numèric.

Per exemple, podem modificar l'exemple anterior per utilitzar SAME i INVALID:

Operador	Argument	Valor si <i>true</i>
<	10	50
<=	20	SAME
=	30	INVALID
Valor per defecte: 10		

Si creem un nou atribut derivat a partir de “Atribut A”, els seus valors serien:

itemId	Atribut A	Nou atribut derivat	Notes
1	5	50	Es compleix $5 < 10$
2	10	10	Falla $10 < 10$, es compleix $10 \leq 20 \rightarrow A=10$
3	20	20	Falla $20 < 10$, es compleix $20 \leq 20 \rightarrow A=20$
4	30	[INVALID]	Fallen els 2 primers, es compleix $30 = 30$
5	40	10	Fallen totes les comprovacions

Cal recordar que no és possible crear un atribut derivat a partir d'un atribut booleà, ja que aquest no conté informació suficient. Ara bé, sí que és possible derivar a partir d'atributs categòrics, però l'únic operador disponible (de moment) és \in (CONTÉ).

Operador	Argument	Valor si <i>true</i>
\in	cat	10
\in	es	20
Valor per defecte: INVALID		

itemId	Atribut B	Nou atribut derivat	Notes
1	cat ; es	10	Es compleix $\text{cat} \in \{\text{cat}, \text{es}\}$
2	es ; cat	10	Es compleix $\text{cat} \in \{\text{es}, \text{cat}\}$
3	es ; en	20	Falla la primera, Es compleix $\text{es} \in \{\text{es}, \text{en}\}$
4	en	[INVALID]	Fallen totes les comprovacions

Com a exemple, ara veurem com discretitzar un atribut numèric.

Suposem que tenim un dataset de pel·lícules amb un atribut budget de tipus numèric que indica el pressupost en euros de cada pel·lícula.

El primer que cal fer és seleccionar el botó “Create Derived” per crear un atribut derivat a partir d’aquest.

Items	Users	Attributes	Configure Model	Options	
		Attribute	Weight	Type	Create Derived
		adult	1.0	Boolean	Create Derived
		belongs_to_collection	1.0	Categorical	Create Derived
		budget	1.0	Numeric	Create Derived
		genres	1.0	Categorical	Create Derived
		homepage	1.0	Categorical	Create Derived
		imdb_id	1.0	Categorical	Create Derived
		original_language	1.0	Categorical	Create Derived

Selecció de l’atribut a derivar

Això obre una nova finestra en la qual cal introduir el nom del nou atribut i una llista d’operacions. Es poden afegir operacions noves a la llista utilitzant el botó “Add rule”, i es poden eliminar les operacions seleccionades amb el botó “Delete selected rules”.

Observant el dataset ens adonem que algunes pel·lícules tenen un budget de 0. Segurament es tracta d’un error, de forma que la primera norma que crearem serà invalidar el nou atribut si l’antic val 0.

En segon lloc, marcarem amb un 1 les pel·lícules barates (amb un budget inferior a 1.000.000 euros), amb un 2 per les mitjanes (budget inferior o igual a 10.000.000 euros) i un 3 per les cares (totes les altres).

Finalment podem seleccionar “Create attribute” per crear l’atribut i calcular els nous valors de tots els ítems.

Operators	Arguments	Value if true	Select for delete
=	0	invalid	<input type="checkbox"/>
<	1000000	1	<input type="checkbox"/>
<=	10000000	2	<input type="checkbox"/>

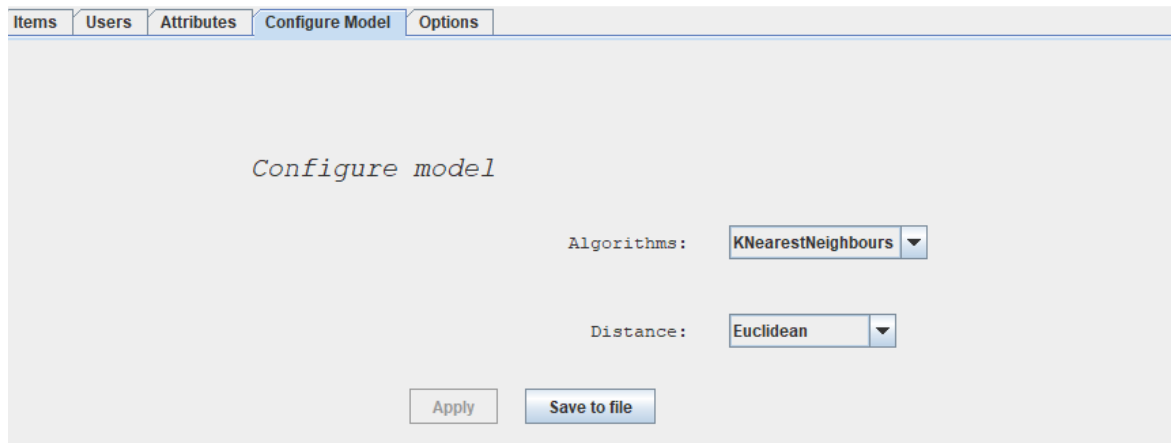
Default Value: 3

Add rule Delete selected rules Create attribute

Llista de normes després de crear les operacions.

2.7 Guardar i restaurar estat del sistema

Per a guardar l'estat del sistema (usuaris, ítems, atributs, algorismes seleccionats, pesos, etc.) s'ha d'anar a la pestanya "Configure Model" i clicar el botó "Save to file". A continuació el sistema preguntarà a quin directori es volen guardar els diferents arxius amb les dades. Recomanem seleccionar o crear un directori dins de . /DATA, tot i que no és obligatori.



Pantalla amb el botó per guardar l'estat del sistema

Els fitxers que es crearan dins del directori són els següents:

- dataset.info Fitxer de configuració del dataset.
- ratings.db.csv CSV amb les valoracions i els usuaris
- items.csv CSV amb els ítems i els atributs
- ratings.test.unknown.csv CSVs de testing. Només es copien si havien estat proporcionats originalment (ja que són opcionals).
- ratings.test.known.csv
- weights.invalid.csv CSV opcional. Conté els pesos dels atributs i una matriu de booleans que indica quins valors són invàlids (a ignorar).

Per a saber més informació de què contenen aquests fitxers, consultar el punt 2.2 (Inicialització del sistema).

Un cop s'han guardat les dades, ja es pot tancar el programa sense perdre la feina feta.

Per a restaurar el sistema des d'un estat en el que ha estat prèviament guardat, s'ha de tancar i tornar a obrir l'aplicació. En el moment de seleccionar el dataset, es selecciona el .info guardat prèviament (situat en el directori escollit), com si es tractés d'un dataset "normal" (no generat pel mateix sistema).

Per més informació dels passos a seguir per seleccionar el dataset, veure l'apartat [2.2 Inicialització del sistema](#).

3. Testing del sistema

Per avaluar els algorismes de recomanació del nostre sistema cal seleccionar un conjunt de dades, el mètode per calcular la distància entre dos ítems o dos usuaris i l'algorisme que es vol provar.

Si es volen realitzar proves del sistema amb el nostre Driver, cal donar el fitxer amb extensió *info* amb informació sobre les dades (situat al directori del conjunt de dades corresponent) i utilitzar les comandes `SET_DISTANCE` per escollir l'estratègia de càlcul de la distància, `SET_ALGORITHM` per seleccionar l'algorisme, i `EVALUATE` per avaluar l'algorisme.

D'altra banda, amb la nostra interfície gràfica cal iniciar la sessió com a l'administrador, anar a la pestanya *Configure Model*, seleccionar un algorisme i una estratègia de distància i prémer *Apply* a la secció de *Configure Model*. Un cop fet això, a la secció *Test model*, prémer el botó *Evaluate*.

Aquests tests permeten comprovar de forma senzilla que el sistema és robust i funciona correctament per a diferents entrades, a més de mostrar els NDCG de cada algorisme i els temps d'execució amb els diferents datasets.

Adicionalment, hem creat diversos programes (“*Mains*”) especialitzats per comprovar a fons diferents parts del sistema. Aquests testers es poden trobar al directori `FONT/TEST`.

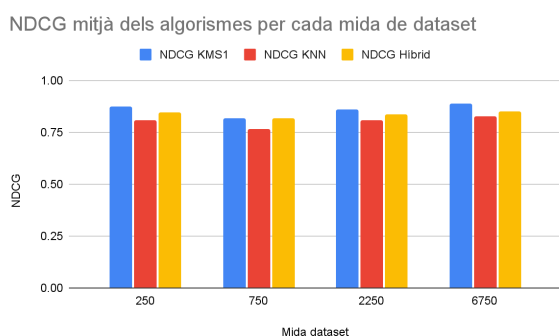
- Testers ja presents a la primera entrega: Executen els algorismes per comprovar que el clustering funciona correctament i mostren el resultat i temps d'execució per la sortida estàndard.
- `Main_TestingAllAlgorithms`: Executa els 3 algorismes utilitzant un dataset sintètic i comprova que funcionen correctament (els resultats dels algorismes són els esperats).
- `TestTimeExecution`: Executa els 3 algorismes amb les 4 funcions de distància uns quants cops i fa la mitjana del temps d'execució. Simula un ús real: les queries estan formades per *tots* els ítems que un usuari no ha valorat.

Per comprovar el funcionament i l'eficiència del nostre sistema hem utilitzat dades sobre pel·lícules (obtingudes del dataset *MovieLens*) i sobre sèries, tots dos conjunts amb diferents mides.

Per comparar els nostres algorismes, hem avaluat els seus DCGs normalitzats per cada conjunt de dades i estratègia de distància, i se n'ha calculat la mitjana aritmètica per cada mida del conjunt de dades per observar com varia el rendiment segons el nombre d'ítems. A més a més, hem calculat els temps d'execució mitjans de la mateixa forma, ja que també són factors importants a tenir en compte.

Segons les nostres expectatives, l'algorisme de *collaborative filtering* (K-Means + Slope1) hauria de mostrar una precisió major que el de *content-based filtering* (K-Nearest Neighbours), però també hauria de tardar més, i l'algorisme híbrid hauria de donar els millors resultats però ser el més costós en temps. A més, hi hauria d'haver una clara relació entre el temps d'execució i la mida del conjunt, però no entre la precisió i la mida.

Un cop hem obtingut el temps d'execució i NDCG de tots els algorismes i mides de dataset, hem representat els nostres resultats en forma de gràfics (nota: tots els temps estan en segons).



Gràfics dels NDCG (esquerra) i temps d'execució (dreta) dels els diferents algorismes.

Podem observar que la precisió dels algorismes no depèn gaire de la mida dels datasets, ja que es manté aproximadament constant. En canvi, sí que hi ha una diferència important entre els algorismes, ja que KMeans+Slope1 (en blau) acostuma a ser el que obté una millor puntuació, seguit per l'algorisme Híbrid (groc) i finalment KNearestNeighbours (Vermell). En el gràfic de la dreta veiem que el cost (temps d'execució) dels algorismes sí que depent fortament de la mida del dataset, i l'algorisme híbrid és més lent que els altres amb diferència.

Per treure conclusions més complexes, podem representar els NDCG obtinguts en una matriu entre l'entrada (dataset) i l'algorisme utilitzat. El color de la taula representa el NDCG com a tercera dimensió del gràfic. A sota hi ha el mateix però amb el temps d'execució.

Nota: En aquest cas, s'ha utilitzat el tester `TestTimeExecution` per automatitzar part del procés. També s'ha realitzat l'execució en un ordinador diferent, de forma que els resultats podrien no ser completament coherents amb els anteriors però la velocitat relativa (respecte els altres algorismes) hauria de ser similar.

NDCG	Movielens/250	Movielens/2250	Movielens/6750	Series/250	Series/750	Series/2250
KNN	0.827	0.828	0.829	0.79	0.765	0.791
KMS1	0.903	0.899	0.888	0.845	0.816	0.822
Hybrid	0.82	0.835	0.851	0.868	0.82	0.837

Temps (s)	Movielens/250	Movielens/2250	Movielens/6750	Series/250	Series/750	Series/2250
KNN	0.017	0.175	0.145	0.011	0.025	0.095
KMS1	0.082	1.47	2.35	0.068	0.24	0.775
Hybrid	0.418	45.41	77.15	0.125	0.716	2.63

3.1 Recomanació basada en K-Means i Slope1

En primer lloc, es pot apreciar que el DCG normalitzat d'aquest algorisme és generalment el més alt, sent l'algorisme que mostra millors resultats. Això és probablement a causa de l'àmplia informació a partir de la qual raona aquest algorisme, ja que es tenen en compte diverses valoracions de tots els usuaris del mateix clúster que l'actiu per classificar els ítems. En tenir tantes mostres a partir de les quals aprendre, és natural que els seus resultats siguin dels més precisos.

No obstant això, això també té un inconvenient: ja que s'estan avaluant moltes dades per generar una recomanació, el nombre d'usuaris del clúster i el nombre de valoracions de cadascun tenen un gran impacte en el temps que triga a executar-se l'algorisme. Per aquest motiu, tal i com es pot veure, el temps creix més amb la mida del conjunt que amb l'algorisme de *content-based filtering* (KNN), i això pot suposar un problema si utilitzem moltes dades.

D'altra banda, no s'observa ninguna relació evident entre la mida del conjunt de dades i la precisió de les recomanacions, tal i com esperàvem.

A la taula NDCG de la pàgina anterior veiem que, en general, la precisió d'aquest algorisme és comparable amb la de l'algorisme híbrid, però amb un cost temporal molt menor. A més, al dataset de Movielens la precisió d'aquest algorisme és excepcionalment bona.

3.2 Recomanació basada en KNearestNeighbours

En segon lloc, com es veu als resultats, l'algorisme de *content-based filtering* mostra un DCG normalitzat que, malgrat ser generalment acceptable, és clarament el més baix dels tres. Això era d'esperar, ja que a diferència de l'estratègia de *collaborative filtering*, aquest algorisme no pren com a referència un conjunt molt gran de dades. En concret, només genera les seves recomanacions a partir dels ítems que ha valorat l'usuari actiu, i això evidentment dona menys informació que les valoracions dels ítems de tots els usuaris del mateix clúster.

Aquesta característica, però, fa d'aquesta estratègia la més ràpida amb diferència, i no sembla ser greument afectada en termes de temps d'execució per la mida del conjunt (almenys pels conjunts sobre els quals s'han fet les valoracions, ja que sí que hi ha un impacte, encara que és molt reduït en comparació als altres algorismes). Això fa aquest algorisme molt convenient si es volen resultats amb precisió acceptable però alhora ràpids per un conjunt molt gran.

A més, en aquest cas tampoc hi sembla haver cap relació visible entre el NDCG i la mida del conjunt.

Cal tenir en compte que, a diferència de KMeans+Slope1, la precisió d'aquest algorisme depèn dels pesos que l'usuari ha establert per cada atribut i pot augmentar si es defineixen bons atributs derivats. Això significa que, si ens dediquéssim suficient temps a crear atributs derivats i afinar tots els pesos, la precisió d'aquest algorisme podria augmentar de forma significant.

3.2 Recomanació híbrida

En tercer lloc, l'algorisme híbrid és el que ha anat més en contra de les nostres suposicions inicials, particularment en quant a la precisió de les recomanacions, ja que esperàvem que el aquesta superés tant a la de l'estratègia de *content-based filtering* com a la de *collaborative filtering*, però, com es pot apreciar, generalment només supera a la primera. És possible que això sigui degut a que els usuaris dels conjunts de dades tinguin gustos amplis i variats, causant que els mètodes de *content-based filtering* siguin menys efectius en ser incapaços de predir el comportament de l'usuari actiu, ja que suposen que aquest trobarà interessants ítems semblants als que ja li agraden. Això explicaria per què els seus resultats estan entre els de K-Nearest Neighbours i els de K-Means+Slope 1, ja que en tenir una font de dades de la que aprendre més àmplia és capaç d'arribar a conclusions més precises que l'estratègia de *content-based filtering*, però sense arribar a la precisió de la de *collaborative filtering*, més adequada a aquests casos.

Respecte al temps d'execució, les nostres prediccions s'han complert, i aquest algorisme és amb molta diferència el més lent: avaluar el model per obtenir el NDCG pot arribar a tardar uns quants minuts en el cas més extrem. Això era esperat, ja que per cada usuari del clúster de l'usuari actiu (excepte aquest) itera sobre tots els ítems que ha valorat i a més compara cadascun amb tots els ítems sobre els quals generar la recomanació, mentre que l'algorisme de *content-based filtering* només fa això amb els ítems valorats per l'usuari actiu, i el de *collaborative filtering* només itera sobre els ítems valorats pels usuaris del clúster. Aquesta diferència causa que el temps d'execució variï enormement amb la mida del conjunt de dades. Malgrat ser un cost esperat, la precisió reduïda fa que aquest algorisme no sigui tan òptim com la resta.

A més, de la mateixa forma que amb els altres algorismes, la mida del conjunt no sembla tenir cap efecte significatiu en la precisió de les recomanacions.