TREBALL FINAL DE GRAU

TÍTOL: Disseny, implementació i estudi d'un sistema recomanador de pel·lícules

AUTOR: RAMON MORAL, ARNAU

DATA DE PRESENTACIÓ: FEBRER, 2021

COGNOMS: RAMON MORAL **NOM: ARNAU**

TITULACIÓ: Grau en Enginyeria Informàtica

PLA: 2018

DIRECTORA: CATALÀ ROIG, NEUS

DEPARTAMENT: CS - Departament de Ciències de la Computació

TRIBUNAL

PRESIDENT	SECRETARI	VOCAL
Bernardino Casas Fernández	Jaume Baixeries Juvillà	Anna Maria Mir Serra

DATA DE LECTURA: 4 de febrer de 2021

Aquest Projecte té en compte aspectes mediambientals: Sí X No

RESUM

En aquest Treball Final de Grau estudiarem la importància i rellevància dels sistemes recomanadors i veurem els diferents algorismes que utilitzen avui dia les grans companyies tecnològiques.

L'ús dels sistemes recomanadors s'ha vist molt incrementat per l'enorme quantitat de productes i/o serveis que poden arribar a oferir les grans empreses que operen online.

La primera empresa que ens ve al cap, inevitablement, és Amazon, que ofereix més de 12 milions de productes per vendre. Sense un bon sistema recomanador de productes i d'anuncis, es faria molt difícil que a un usuari se li recomanés justament el producte que vol comprar.

Tot i que els algorismes recomanadors poden ser comuns i independents del que s'està recomanant (música, llibres, publicitat per vendre productes, articles, etc.), en aquest treball ens centrarem en un dels mercats emergents els darrers anys: les pel·lícules. Concretament, l'objectiu seran les pel·lícules ofertades per les plataformes de visualització digital, on ja es comú que tothom amb accés a Internet disposi d'almenys una subscripció a alguna d'aquestes plataformes (Netflix, HBO, Amazon Prime Video, Disney+, etc.).

A més de l'estudi de l'estat de l'art actual, en aquest projecte dissenyarem i implementarem des de zero un sistema recomanador de pel·licules que disposarà d'una interfície en la que es mostraran els resultats que podrà veure el client o l'usuari final depenent de les seves preferències o del seu perfil d'usuari. Per tant, també caldrà catalogar a cada usuari dins d'un tipus o perfil concret.

Per últim, també realitzarem una anàlisi de les dades obtingudes amb resultats reals, on es mesurarà l'impacte dels diferents paràmetres per determinar el rendiment dels algorismes usats, avaluar les diferències i treure'n conclusions.

Paraules clau:

Sistemes recomanadors	Filtratge col·laboratiu	Filtratge basat en el contingut	Filtratge basat en la popularitat
Python	Pel·lícules	Valoració de productes	Validació de mètriques
Plataformes de servei de vídeo	Algoritmes recomanadors		

RESUMEN

En este Trabajo Final de Grado estudiaremos la importancia y relevancia de los sistemas recomendadores y veremos los diferentes algoritmos que se utilizan hoy en día en las grandes compañías tecnológicas.

El uso de los sistemas recomendadores se ha visto incrementado por la enorme cantidad de productos y/o servicios que pueden llegar a ofrecer las grandes empresas que operan online.

La primera empresa que nos viene a la cabeza, inevitablemente, es Amazon, que ofrece más de 12 millones de productos en venta. Sin un buen sistema recomendador de productos y anuncios, se haría muy difícil que a un usuario se le recomiende justamente el producto que quiere comprar.

Aunque los algoritmos recomendadores pueden ser comunes e independientes de lo que se está recomendando (música, libros, publicidad para vender productos, artículos, etc), en este trabajo nos centraremos en uno de los mercados emergentes en los últimos años: las películas. Concretamente, el objetivo serán las películas ofrecidas por las plataformas de visualización digital, donde ya es común que cualquier persona con acceso a Internet disponga de al menos una suscripción a alguna de estas plataformas (Netflix, HBO, Amazon Prime Video, Disney+,etc).

Además del estudio del estado del arte actual, en este proyecto diseñaremos e implementaremos desde cero un sistema recomendador de películas que dispondrá de una interfaz en la que se mostraran resultados que podrá ver el cliente o usuario final depende de sus preferencias o de su perfil de usuario.
Por tanto, también hará falta catalogar a cada usuario dentro de un tipo de perfil concreto.

Por último, también realizaremos un análisis de los datos obtenidos con resultados reales, donde se medirá el impacto de los diferentes parámetros para determinar el rendimiento de los algoritmos usados, evaluar las diferencias y extraer conclusiones.

Paraules clau:

Sistemas recomendadores	Filtrado colaborativo	Filtrado basado en el contenido	Filtrado basado en popularidad
Python	Películas	Valoración de productos	Validación de metricas
Plataformas de servicio de video	Algoritmos recomendadores		

ABSTRACT

In this Final Degree Project we will study the importance and relevance of the recommender systems and we will see the different algorithms that are used nowadays by the big tech companies.

The use of the recommender systems has been greatly increased by the huge amount of products and services that the big companies that operate online offer.

The first company that inevitably comes to mind is Amazon, that offers more than 12 million products to sell. Without a good product and ads recommender system, it would be too difficult for a user to be recommended exactly the product they want to buy.

Although recommender algorithms can be common and independent of the item we are recommending (music, books, selling ads, articles,etc), in this project we will focus on the emergent market of these years: movies. Specifically, the goal will be the films offered by digital video platforms, where it is common for everyone with Internet access to have at least a subscription to one of these platforms (Netflix, HBO, Amazon Prime Video, Disney+,etc).

In addition to the study of the current state of the art, in this project we will design and implement from scratch a film recommendation system that will have an interface that will show the results to the customers or the end users based on their preferences or their user profile. Therefore, each user must also be cataloged within a specific type or profile.

FInally, we will also perform an analysis of the data obtained with real results, where the impact of the different parameters will be measured to determine the performance of the algorithms used, evaluate the differences and draw conclusions.

Paraules clau:

Recommender systems	Collaborative filtering	Content-based filtering	Popularity filtering
Python	Movies	Product ratings	Metrics validation
Video service platforms	Recommender algorithms		

SUMARI

SUMARI DE FIGURES I TAULES

GLOSSARI

Collaborative filtering → És un dels principals mètodes existents de recomanació. La informació s'aconsegueix a partir d'un col·lectiu d'usuaris normalment semblants a l'usuari al qual volem recomanar. Per exemple, l'empresa d'Amazon, aplica un dels mètodes de *Collaborative filtering* anomenat *item-to-item* o *item-based* que, en termes de publicitat, es solen representar com a "les persones que compren X també compren Y".

Content-based filtering → És un dels principals mètodes existents de recomanació. La informació s'aconsegueix amb el contingut que el mateix usuari consumeix. Per exemple, l'empresa Netflix, aplica el mètode de *Content-based filtering* de manera que recomana pel·lícules amb característiques similars a les ja visualitzades per l'usuari.

Knowledge-based recommender → És un dels principals mètodes existents de recomanació. Molt semblant al *Content-based*. La informació s'aconsegueix de les preferències, gustos i informació implícita de l'usuari. S'usa per recomanar productes dels quals no hi ha prou dades per usar *Content-based filtering* o quan la mesura de similitud no s'ajusta al que és vol recomanar. Per exemple, productes dels quals no hi ha prou valoracions d'usuaris o no es venen gaire, com un pis o un vehicle molt car, per exemple.

Hybrid filtering → Mètode de recomanació complex que utilitza les tècniques i aprofita avantatges de tots els demés mètodes existents. És el més usat en les grans companyies avui dia.

Popularity/Simple filtering → És el mètode de recomanació més senzill que hi ha. La informació s'extreu únicament de la popularitat dels productes o de la posició que ocupen en un rànquing de puntuació dels usuaris. Veiem aquest sistema típicament en apartats com "el més venut" o "les pel·lícules més vistes".

User rating → Puntuació que dona un usuari a un determinat producte. Normalment representat amb un valor numèric del 0 al 10, en forma "d'estrelles" de l'1 al 5, o com a valor binari mitjançant els botons de "m'agrada" i de "no m'agrada".

***IMDB* (*Internet Movie DataBase*)** → Una de les bases de dades en línia més grans i importants en el món del cinema.

Stop words → Paraules no rellevants que no es solen tenir en compte a l'hora de filtrar o comparar documents. Entre elles hi trobem articles, conjuncions, preposicions, etc.

Data set → Conjunt de dades relacionades que obtenim de diferents elements independents però que les podem manipular en un sol fitxer. El format més típic i també l'usat en aquest treball és el CSV (*Comma-Separated values*).

Keywords → Paraules clau que descriuen la trama cinematogràfica d'una pel·lícula. S'utilitzen per calcular similituds entre els documents que contenen aquestes trames. Per exemple, per a la pel·lícula de *Toy Story*, algunes paraules clau podrien ser joguina o *cowboy*.

INTRODUCCIÓ

Un sistema recomanador es defineix com un mecanisme que, mitjançant la informació dels usuaris, ofereix productes del seu interès de forma automàtica. Com a conseqüència de la gran quantitat d'informació que es maneja avui en dia, aquests sistemes són una clara necessitat de cara a oferir als usuaris un seguit de suggeriments personalitzats sobre qualsevol tipus d'element (productes, pel·lícules, llibres, articles, etc.).

La magnitud d'aquests sistemes és tan gran que els trobem en el nostre dia a dia pràcticament a qualsevol consulta que realitzem per Internet. Comprar un producte en una plataforma com Amazon, realitzar una cerca sobre algun tema concret a Google o decidir veure una pel·lícula o una altra a Netflix, nodreixen d'informació a aquests sistemes amb la finalitat de proporcionar a l'usuari una navegació més fluida i satisfactòria.

Seria impensable que un usuari a l'hora de comprar un producte, entre els milions de productes que s'ofereixen en el comerç electrònic, fos capaç de trobar el que més s'adeqüi a les seves necessitats, o de trobar un vídeo concret a Youtube entre milers de similars.

Per citar un exemple, en el cas de Netflix, la companyia assegurava l'any 2012 que el 75% dels continguts que es consumeixen els usuaris de la seva plataforma provenen d'algun tipus de recomanació. L'èxit de les recomanacions, segons la companyia, és gràcies a l'optimització contínua de l'experiència d'usuari, que es pot mesurar a partir del seu grau de satisfacció.

En general, no només els usuaris es beneficien d'aquests sistemes, també atorguen a les empreses la possibilitat d'oferir productes més variats i entendre millor què volen els usuaris i, en definitiva, vendre més.

Existeixen multitud d'algorismes d'aprenentatge automàtic. En aquest treball, realitzarem un estudi de les diferents tècniques que s'utilitzen, les analitzarem i valorarem la seva eficiència per extreure conclusions de cara a determinar si generen, o no, bones recomanacions.

1. CONTEXT I ACTORS

Abans d'entrar més en profunditat en el funcionament dels sistemes recomanadors per a pel·lícules, realitzarem una petita introducció al món dels sistemes recomanadors a nivell del seu funcionament general.

La informació de què disposen per funcionar és la següent:

- Un **conjunt d'usuaris** consumidors dels productes.
- Un **conjunt de dades** que descriguin el contingut dels productes.
- Un **conjunt de puntuacions** que s'aconseguiran d'una manera o una altra depenent de la tècnica de recomanació que s'utilitzi.

Aquests tres aspectes mencionats anteriorment els necessitarem sempre a l'hora d'implementar un sistema recomanador del tipus que sigui, ja que bàsicament responen a les tres preguntes més basiques que ens podem fer: **A qui recomanem**, **què recomanem** i en **quin ordre ho ordenem** per mostrar o valorar-ho segons ens convingui.

Respecte al **conjunt d'usuaris**, la informació pot ser obtinguda per diverses vies diferents i depenen molt de la plataforma i dels productes que es vulguin recomanar.

En el nostre cas, si ens fixem en el sector de les pel·lícules, podem imaginar-nos que en les plataformes de vídeo més exitoses d'avui dia que tots coneixem, la informació s'obté de les nostres interaccions amb el servei, és a dir, si hem vist o no una pel·lícula i si hem pitjat el botó de "m'agrada" o el de "no m'agrada".

La realitat és que aquesta informació és de molta ajuda de cara a saber si t'agradarà una pel·lícula similar a les que ja has vist i t'agraden, però la informació també pot provenir d'altres factors a l'hora de consumir el servei de vídeo, com per exemple quins dies i a quines hores normalment ho fas, des de quin dispositiu ho veus i durant quant de temps estas veient un mateix contingut.

En general, tota aquesta informació, vingui d'on vingui, s'utilitza per identificar quin **tipus d'usuari** ets.

Classificar els usuaris en diverses categories és una pràctica molt utilitzada i molt útil, en especial per als sistemes anomenats **sistemes de filtratge col·laboratiu** (***collaborative filtering***)**,** dels quals parlarem més endavant. La raó d'això és perquè, si s'aconsegueix saber amb quin tipus d'usuari s'està tractant, es poden arribar a presentar els productes de manera que resultin més atractius per a l'usuari en qüestió.

Per exemple, com es pot veure a la Figura 2.1, la plataforma Netflix disposa de 9 portades diferents per a la famosa sèrie *Stranger Things,* i escull quina ha de mostrar i quina no en base al tipus d'usuari que siguis.

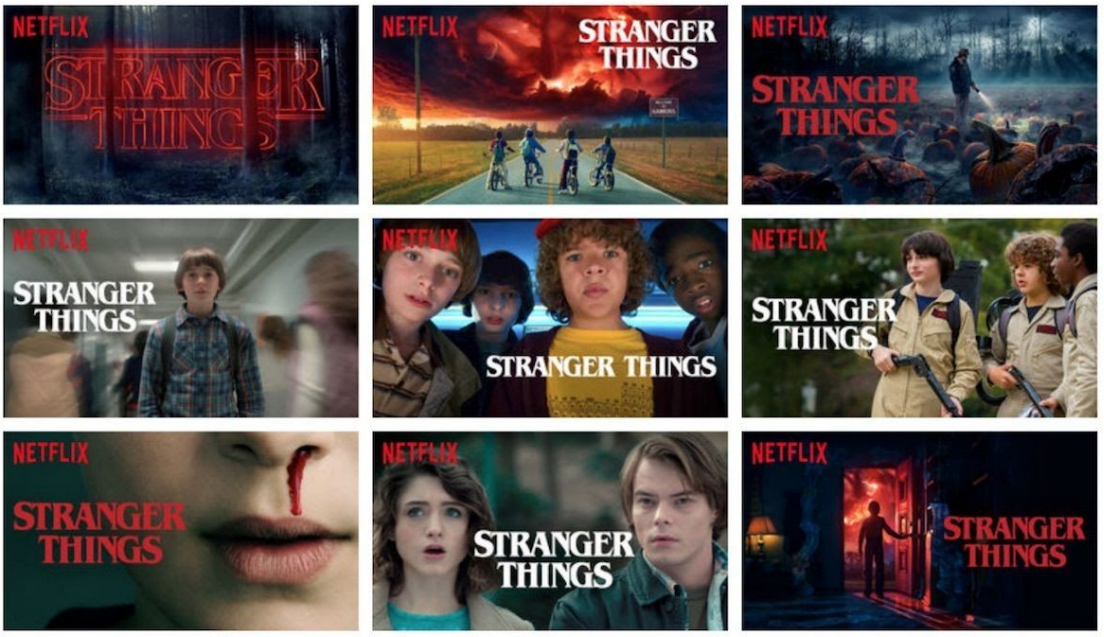


Figura 1.1 *Diverses portades de la sèrie Stranger Things disponibles a Netflix.*

Si ens fixem en les imatges de la Figura 1.1, podem associar instintivament cada una de les imatges amb un gènere de cinema determinat. Hi ha imatges que ens indueixen a pensar que pot tractar-se d'una sèrie de terror, altres potser ens recorden més a una d'infantil, de comèdia, de drama o d'investigació.

D'aquesta manera aconsegueixen que, una mateixa sèrie o pel·lícula, sent del gènere que sigui realment, resulti més atractiva a l'usuari i l'acabi veient.

Figura 1.2 *Portada de la pel·lícula Pulp Fiction mostrada en funció d'altres pel·lícules que l'usuari ha vist.*

A la Figura 1.2 podem veure un altre exemple de com el tipus d'usuari pot influir a l'hora de com mostrem les recomanacions. Per als fans d'Uma Thurman, utilitzarem la imatge superior, per als d'en John Travolta utilitzarem la imatge inferior.

Un altre aspecte, a més de la informació dels usuaris, és la informació del **conjunt dels productes** dels quals disposem, en el nostre cas, les pel·lícules.

Sembla una afirmació massa òbvia però, per recomanar qualsevol cosa, hem de conèixer bé què és aquesta cosa. A tota la informació obtinguda de les nostres pel·lícules la denominarem "contingut", nom basat en els sistemes recomanadors anomenats **sistemes basats en el contingut** ***(content-based)*** que basen la seva informació únicament en els aspectes dels mateixos productes que es recomanen.

Per l'obtenció del contingut d'una pel·lícula buscarem tota aquella informació que puguem extreure i que pugui ser interessant de cara a recomanar-la. Les característiques més típiques, en aquest domini, són:

- El/la director/a de la pel·lícula.
- El gènere de la pel·lícula
- El títol de la pel·lícula.
- Cadascun dels actors i actrius implicats en el rodatge.
- L'any d'estrena.
- Mitjana de la puntuació dels usuaris de l'1 al 10.

Aquests són simplement alguns dels paràmetres que es solen tenir en compte i els que més destaquen quan pensem en el contingut d'una pel·lícula però, de fet, podem tenir tants paràmetres com vulguem, per exemple:

- És una pel·lícula familiar?
- És una pel·lícula d'animació?
- Apareix en Leonardo DiCaprio de jove?
- És una pel·lícula on apareixen el mateix nombre d'homes que de dones?

Com podem veure, el que podem considerar com a "contingut" pot ser literalment infinit. Per això, és necessari un estudi previ a la implementació del nostre sistema recomanador sobre a quins usuaris van dirigides les recomanacions i sobre quins són els factors que més pes volem que tinguin a l'hora de recomanar, ja que per a un cert tipus de persones pot ser molt important un aspecte mentre que per a altres pot ser totalment irrellevant.

La Figura 1.3 mostra un exemple molt reduït, en el que només s'han tingut en compte dos paràmetres que per recomanar una samarreta: el color i si és de coll alt o no. En aquest cas, el valor del color es podria representar amb el seu valor RGB i, d'aquesta manera, podriem treballar amb una paleta de colors molt més àmplia, mentre que el coll és un paràmetre booleà.

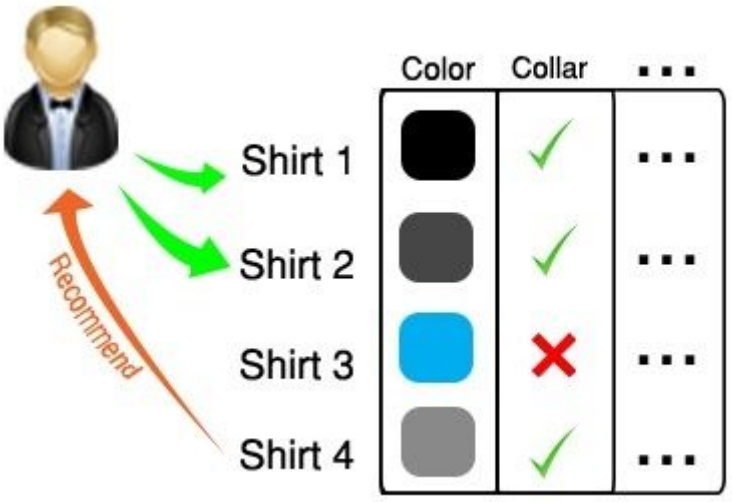


Figura 1.3 *Il·lustració dels paràmetres pel producte samarreta. S'han considerat rellevants dos aspectes: el color i si té coll o no.*

Com hem mencionat prèviament, un últim aspecte que necessiten tots els sistemes recomanadors, sigui del tipus que sigui, és un **conjunt de puntuacions**.

Les puntuacions són, en última instància, **valors numèrics que s'associen a cada producte que vulguem recomanar**, de manera que, per exemple, si una pel·lícula obté una puntuació de 250 i una altra una puntuació de 110, sabrem ordenar-les perfectament de més a menys encertada per a un usuari en concret i podrem mostrar-les en l'ordre que ens convingui perquè l'experiència d'usuari en general sigui millor. Les puntuacions són importants de cara a valorar quin és el "grau d'aparellament" entre un usuari i cada una de les pel·lícules.

Els sistemes recomanadors es poden classificar en diferents categories segons la tècnica utilitzada per fer les recomanacions. Aquestes tècniques poden estar centrades en els usuaris, els productes, les puntuacions, o bé en una combinació d'aquests. Un cop mencionades, a l'apartat 2, Estat de l'art, presentarem les tres estratègies de recomanació més conegudes:

- ***Popularity filtering***
- ***Content-based filtering***
- ***Collaborative filtering***

1.1 STAKEHOLDERS

Com a persones implicades en el projecte considerarem aquelles a les quals el projecte afecta o influeix d'alguna manera o altra. Les parts interessades del nostre projecte serien les següents:

- **Directora:** En aquests cas es tracta de la mateixa tutora del projecte, NEUS CATALÀ ROIG. El treball de direcció consisteix en guiar, aconsellar, tutoritzar i avaluar l'alumne o els alumnes que intervinguin en tot el procés de realització del treball final de grau, així com el poder exigir o demanar als desenvolupadors que el projecte agafi un rumb o altre en un moment determinat.

- **Desenvolupador:** Es tracta només d'una persona, tot i que en un projecte fora de l'àmbit acadèmic és podria tractar d'un departament o un grup d'enginyers desenvolupadors. En aquest cas, l'alumne realitzant aquest projecte, és ARNAU RAMON MORAL. El treball del desenvolupador consisteix en dissenyar, implementar i documentar tot el procés de realització del treball final de grau, incloent també reunir-se assíduament amb la tutora i directora per buscar l'aprovació i vist i plau per continuar avançant en el desenvolupament del projecte.

- **Usuaris finals (col·lectiu d'usuaris):** Es compon de tot el col·lectiu d'usuaris que facin ús de l'aplicació, ja sigui un cop acabada o mentre s'està dissenyant, implementant o documentant. Aquest grup abarcaria des dels membres del tribunal (secretari, president i vocal) aportant l'avaluació del projecte, fins a tots els alumnes i companys que han ajudat aconsellant i testejant qualsevol part del projecte. A efectes d'una aplicació comercial fora de l'àmbit acadèmic, el col·lectiu d'usuaris finals estaria format per totes les persones que fessin ús de l'aplicació final amb l'objectiu d'obtenir recomanacions de pel·lícules.

2. ESTAT DE L'ART

Els sistemes recomanadors es poden considerar un món relativament nou i en constant expansió. Sorgeix a partir del creixement exponencial de la potència de les tecnologies i la quantitat d'usuaris que comencen a consumir contingut a Internet.

Inicialment, els sistemes recomanadors repliquen un funcionament que ja existeix i s'utilitza als sectors més tradicionals del màrqueting. Un exemple clàssic podria ser el fet d'anunciar productes escolars en un lloc proper a una escola, o al mes de setembre, data d'inici de l'any escolar i quan sorgeixen les necessitats de material precisament escolar.

Tenint en compte això, els primers sistemes recomanadors, doncs, no aporten res nou ni innoven amb metodologies totalment noves, sinó que únicament es dediquen a imitar les tècniques de màrqueting ja existents a altres sectors i adaptar-los al món d'Internet.

Com ja hem comentat prèviament, el gran creixement i expansió de la tecnologia, comporta que milions d'usuaris comencen a consumir Internet i, per tant, sorgeix la necessitat d'expandir i millorar els sistemes de recomanació de cara a poder oferir la millor publicitat i contingut especialitzat per a cada usuari específic.

Un sistema recomanador bàsic, dels primers que van sorgir, és el basat en la popularitat. Les dades que es podien aconseguir sobre els usuaris, aleshores, no era gaire extensa i per la seva facilitat d'implementació i càlcul, els sistemes basats en popularitat van triomfar. La lògica darrere aquests sistemes és senzilla perquè els productes que més es consumeixen solen ser els que la gent prefereix i, per tant, els millors a recomanar i anunciar.

Amb el pas del temps, augmenta massivament la quantitat de dades que podem adquirir dels usuaris que naveguen per Internet: es normalitza el funcionament de creacions d'usuaris, l'obligació de fer *login* per visualitzar continguts de pàgines web, les eines per puntuar i indicar si un contingut és del teu gust o no, etc. En aquest context, els sistemes recomanadors estan dotats de moltes més dades de les quals anteriorment no disposaven i, per tant, es comencen a estudiar altres estratègies de recomanació més complexes com les estratègies basades en el contingut i les basades en el filtratge col·laboratiu.

Avui dia, els sistemes recomanadors segueixen en expansió continua i es comencen a usar sistemes basats en la Intel·ligència Artificial (IA) aplicant tècniques d'aprenentatge automàtic (machine learning) més avançades com l'aprenentatge profund (deep learning). L'aprenentatge profund està resultant molt útil per fer recomanacions en serveis com Youtube perquè li permet treballar a gran escala, sobre un conjunt de dades dinàmic i tractar una gran quantitat de factors externs que no es poden observar a simple vista. El sistema que usa Youtube consisteix en dues xarxes neuronals: una per generar candidats i una altra per rànquing [39].

Per exemple, amb la situació social actual respecte al virus de la Covid-19, el comportament i les conductes que tothom podia tenir com assentades i consolidades han estat modificades per complet. Un sistema recomanador basat en *collaborative-filtering*, fixant-se en la informació que prèviament els usuaris han generat, no podrà recomanar de manera tan òptima com un sistema basat en *machine learning* que, a l'estar contínuament adaptant-se, sabrà identificar que les conductes de les persones han canviat i, per tant, recomanar les noves necessitats sorgides d'un canvi social com aquest.

Aquestes tècniques, tot i que encara no son usades al 100% del sector digital, són sens dubte les més prometedores i les que marcaran el futur del món dels sistemes recomanadors.

A continuació mostrarem una explicació més detallada de cadascuna de les estratègies més utilitzades actualment en el sector digital.

2.1 POPULARITY FILTERING

Les estratègies basades en la popularitat (*Popularity filtering*) són sens dubte les estratègies menys complexes i, alhora, les més fàcils d'implementar. Això no vol dir que no siguin efectives, tot al contrari, de vegades la millor recomanació que podem fer-li a un usuari és precisament el producte més venut a nivell global que, en el nostre cas, seria la pel·lícula amb més bona valoració dels usuaris (*user rating)* del nostre catàleg, la qual és molt possible que sigui del grat de la gran majoria d'usuaris.

A la Figura 2.1 podem veure un rànquing que mostra les pel·lícules en ordre de valoració (*IMDb rating)* que han estat puntuades pels usuaris de la base de dades IMDB (*Internet Movie DataBase*) la base de dades en línia més gran i important en el món del cinema.

La millor recomanació que ens pot donar aquest sistema serà sempre la pel·lícula *Cadena perpetua* amb una valoració de 9,2 sobre 10.

Com podem observar, estem obviant totalment tant la informació dels usuaris com el contingut de la pel·lícula. Si no aprofitem els avantatges que ens donen els sistemes basats en el contingut o els de filtratge col·laboratiu, la recomanació s'acaba reduint a un simple llistat de "millors pel·lícules" a nivell global.

2.2 CONTENT-BASED FILTERING

Pel que fa a les estratègies basades en el contingut (*Content-based filtering*), com ja hem comentat breument amb anterioritat, es basen en aconseguir la informació del contingut que l'usuari mateix consumeix, en el nostre cas, el contingut de les pel·lícules. En aquest àmbit, la dades més rellevants solen ser: el gènere de la pel·lícula, el director, el repartiment d'actors principals i secundaris, i l'argument de la trama.

Aquest últim, l'argument, es representa mitjançant un conjunt de paraules clau o *keywords* que defineixen a grans trets les parts importants de la trama cinematogràfica; per exemple, per la pel·lícula de *Toy Story*, unes bones *keywords* podrien ser joguina, *cowboy* o familiar. És important remarcar que la paraula "animació´", tot i que defineix la pel·lícula bastant bé, no seria una *keyword* ja que en aquest cas correspondria al gènere.

Figura 2.1 *Rànquing de les 10 pel·lícules millor valorades de la història pels usuaris d'IMDB*.

Com podem observar a la Figura 2.2, tenim un usuari del qual coneixem que compra taronges i també coneixem les característiques d'una taronja, com per exemple que és una fruita i que és un cítric. Coneixent aquesta informació, podem deduir que l'usuari a qui estem recomanant un nou producte, és molt possible que li agradin també les llimones, ja que també són fruites i cítrics. Per tant, la recomanació final que obtindrà

l'usuari serà que li aconsellem de provar a comprar les llimones que casualment es troben en oferta, tot i que aquest últim seria més un treball de màrqueting que no del propi recomanador.

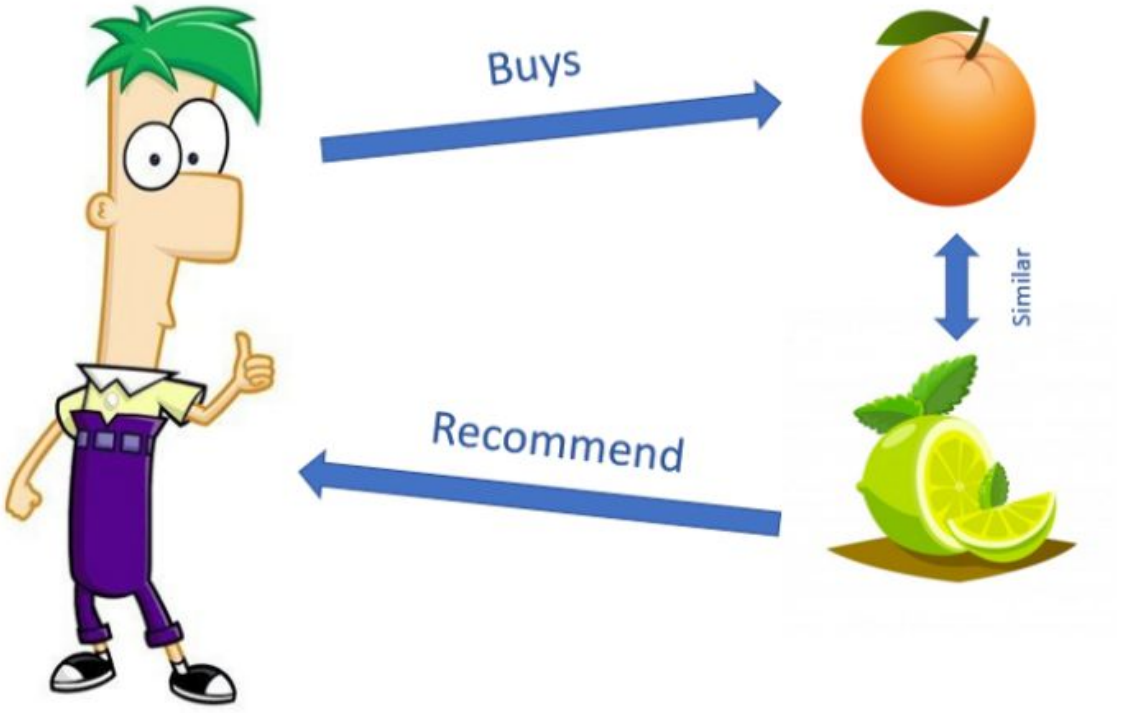


Figura 2.2 *Representació gràfica d'un sistema recomanador amb una estratègia Content-based filtering.*

En l'àmbit de les pel·lícules, no necessàriament s'ha d'haver "comprat" la pel·lícula, o ni tan sols haver-la visualitzat, per poder establir una relació d'interés entre l'usuari i la pel·lícula. Potser, l'usuari ha afegit una pel·lícula en concret a la seva llista de pel·lícules pendents de veure, o ha marcat que li agraden les pel·lícules del gènere de terror. Amb aquesta informació també es pot establir una relació entre cada pel·lícula de terror de què disposem al nostre catàleg i aquest usuari, tot i que possiblement haurem de valorar-ho amb un menor pes que si l'usuari puntua directament una pel·lícula amb una nota màxima, senyal que ens indicaria que efectivament tenim una forta relació entre aquest usuari i aquesta pel·lícula.

Per trobar similitud entre els nostres ítems, tenim els paràmetres cinematogràfics més típics ja mencionats prèviament (director, gènere, actors, etc.) però hem de tenir present que podem trobar-ne infinits, tants com es vulgui, ja que els paràmetres a considerar poden variar segons el context o el que els usuaris estiguin valorant més en una època determinada i que en altres èpoques anteriors no es tenien en compte o no se'ls donava rellevància.

Per exemple, un dels paràmetres que es té en compte avui dia de cara a determinar si una pel·lícula és similar a una altra, és si passen o no el conegut com a test de Bechdel. El test de Bechdel és un mètode per avaluar la quantitat de masclisme que pot haver-hi en una obra, ja sigui una pel·lícula, una obra de teatre, una novel·la, etc. Es prenen criteris com el fet que apareguin la mateixa quantitat de personatges femenins i masculins o la importància que poden rebre els personatges segons el seu gènere, entre d'altres.

La Figura 2.3 mostra un exemple en el que podem veure els paràmetres que poden ser tinguts en compte per recomanar una pel·lícula. En aquest cas, el valor de cada paràmetre és un booleà.

Animated	Yes	Yes	No	No
Marvel	No	No	Yes	Yes
Super Villain	No	Yes	Yes	Yes

Figura 2.3 *Il·lustració dels paràmetres d'una pel·lícula amb diversos aspectes que s'han considerat rellevants.*

Si ens fixem en els paràmetres de la Figura 2.3 i decidim que només aquests 3 aspectes seran els rellevants (si és d'animació, si és de Marvel i si hi apareix un personatge super malvat), podem observar que hi ha una clara relació de similitud entre les 4 pel·lícules mostrades.

Si determinem que cada paràmetre booleà que coincideixi entre una pel·lícula i una altra sumarà 1 punt a un comptador que determinarà la similitud entre les dues pel·lícules, obtindrem uns valors numèrics comparables entre si que ens indicaran el grau de semblança dels dos ítems.

Per exemple, entre la pel·lícula d'*Inside Out* i la dels *Minions*, trobem 2 paràmetres iguals entre si (animació i Marvel), per tant, podem determinar que sobre un màxim de 3 punts, aquestes dues pel·lícules obtenen una puntuació de similitud de 2 punts, el que vindria a ser una semblança prou elevada per considerar recomanar a usuaris que hagin visualitzat *Inside Out* la pel·lícula dels *Minions* i viceversa.

El mateix cas s'aplica a la comparació entre les dues pel·lícules de Marvel, tot i que aquestes coincideixen en tots 3 paràmetres, per tant, la recomanació tindrà molt més pes i serà més aconsellable.

Si, per altra banda, comparem la mateixa pel·lícula *Inside Out* amb qualsevol de les de Marvel, veiem que cap camp coincideix, per tant, tindrem 0 punts de similitud, o el que és el mateix, no recomanarem *Inside out* als usuaris que els agraden les pel·lícules de Marvel i viceversa.

2.3 COLLABORATIVE FILTERING

Pel que fa a les estratègies basades en la "col·laboració entre usuaris" (*Collaborative filtering*), com ja hem comentat breument amb anterioritat, es basen en extreure la informació necessària per fer una recomanació a partir dels propis usuaris, en el nostre cas, dels consumidors de les pel·lícules.

Aquesta és sens dubte l'estratègia més complexa de totes les comentades anteriorment, no tan sols per la dificultat d'implementació, sinó també la necessitat de tenir un conjunt d'informacions de suficients usuaris perquè les recomanacions siguin efectives i valorables.

Tot i basar-se en l'extracció d'informació dels propis usuaris per igual, podem distingir entre dues fonts d'informació diferenciades. Segons la font d'informació, podrem basar les nostres recomanacions amb una estratègia o una altra d'acord al que interessi al nostre recomanador en particular. Les dues estratègies basades en *collaborative filtering* són: estratègia **item-based** i estratègia **user-based**.

2.3.1 ITEM-BASED

Aquesta estratègia de la tècnica *collaborative filtering* consisteix en predir quin ítem "s'ajusta" més per ser recomanat tenint en compte la similitud entre els ítems que han estat puntuats pels usuaris.

A diferència del mètode *content-based*, aquest no es fixa en la pròpia informació del ítem, és a dir, no comprovem el gènere o el director de la pel·lícula per exemple, si no que ens fixem, entre d'altres aspectes, en si als usuaris als qui els ha agradat una pel·lícula determinada X també els agrada una altra pel·lícula Y, i aquesta coincidència es produeix amb molta freqüència (a molts usuaris que els ha agradat X, també els agrada Y). Si això succeeix, podem determinar que aquestes dues pel·lícules tenen una similitud elevada ja que els usuaris així ho han decidit amb les seves valoracions.

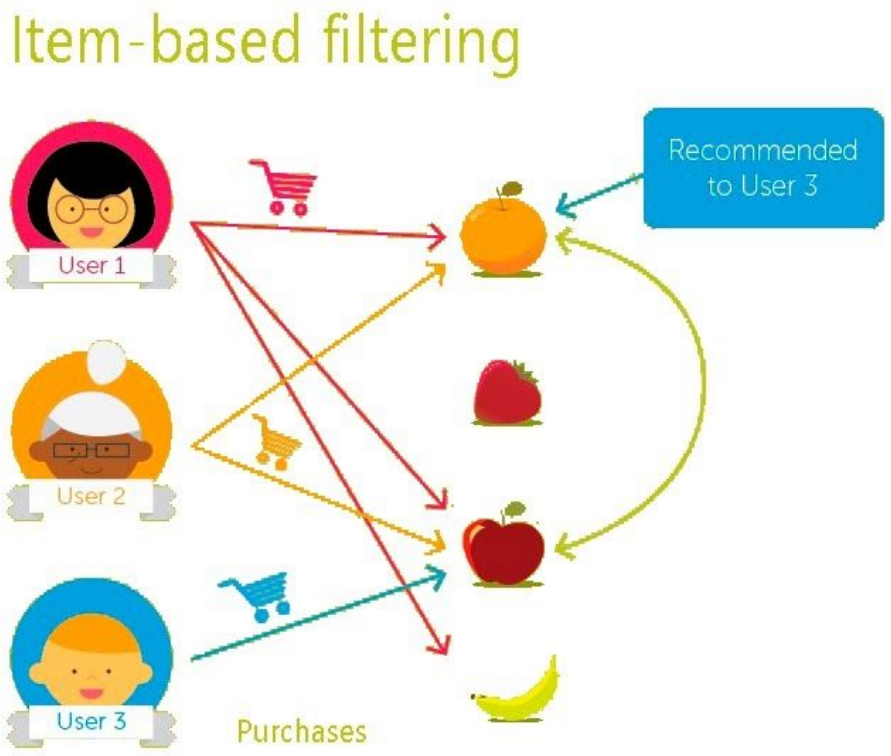


Figura 2.4 *Il·lustració d'un exemple de recomanació basada en item-based collaborative filtering.*

Com podem observar a la Figura 2.4, coneixent que l'usuari número 3 compra pomes, li recomanem les taronges, no pas perquè siguin fruites semblants, sinó perquè podem observar que els usuaris que compren pomes, també compren taronges, establint una alta similitud i relació entre les pomes i les taronges.

Ens fixem que la **relació** que s'està establint és **d'un ítem amb un ítem**, fet que dona nom a l'estratègia mateix i que molts cops també es coneix amb el nom *item-to-item*.

Respecte a la Figura 2.5, podem extreure les mateixes conclusions que amb la Figura 2.4 però, en aquest cas, podem veure que el sistema de puntuacions que hem emprat determina que, finalment, la pel·lícula dels *Avengers* ha estat la recomanada amb 2 punts, per sobre de *Ant-man* i la dels *Minions*.

Ens fixem que només extraiem informació dels usuaris als que també els ha agradat la pel·lícula d'*Inside Out*, igual que a l'usuari a qui li estem fent la recomanació. Aquests dos usuaris són en Jason i la Sarah.

Jason	Yes	Yes	Yes	Yes
Andi	No	Yes	No	Yes
Sarah	Yes	No	Yes	No
Sam	No	No	Yes	Yes
Scaz	Yes	??	??	??
		1 vote	2 votes	1 vote

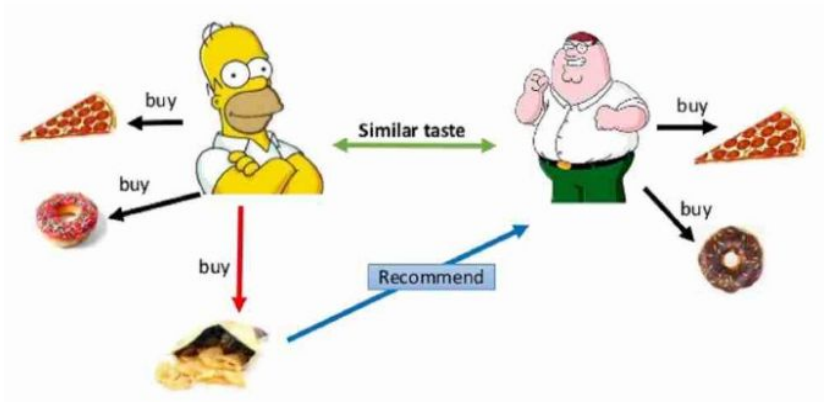
Figura 2.5 *Il·lustració de les votacions de diferents usuaris per a diverses pel·lícules.*

2.3.2 USER-BASED

Aquesta estratègia, dins la tècnica de recomanació de *collaborative filtering,* consisteix en predir quin ítem "s'ajusta" més per ser recomanat tenint en compte la similitud entre els usuaris entre si, és a dir, entre un usuari i tota la resta d'usuaris que ens interessi valorar o que siguin similars.

Per fer ús d'aquesta estratègia, és molt important conèixer la informació dels usuaris mateixos, es a dir, els seus gustos. Existeixen diversos mètodes per classificar un usuari en un tipus o un altre, segons la seva edat, el seu sexe, la seva situació laboral, etc. Aquests, però, són mètodes que requereixen de la confirmació de l'usuari perquè les seves dades puguin ser utilitzades i, en molts casos, atempten directament la privacitat dels usuaris mateixos.

En el nostre treball, reduirem el càlcul de les similituds entre els usuaris a únicament els seus gustos. És a dir, l'usuari A i l'usuari B seran molt semblants o del mateix tipus si els agraden les mateixes pel·lícules, obviant que puguin pertànyer a situacions i/o edats totalment diferents i que, en el cas d'una gran empresa, potser aquestes altres característiques també es tindrien en compte.



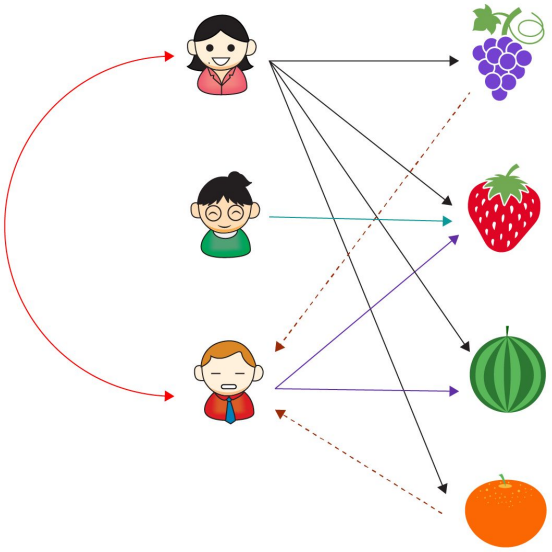


Figura 2.6 *Exemple 1 user based collaborative filtering*. **Figura 2.7** *Exemple 2 user based collaborative filtering*.

La Figura 2.6 i la Figura 2.7 il·lustren el mateix concepte. Ambdues mostren exemples d'una recomanació basada en *user based collaborative filtering*.

Pel que fa a la Figura 2.6, podem observar com els dos usuaris comparteixen gustos ja que els agraden tant la pizza com els donuts, fent que els agrupem dins el mateix tipus d'usuari. Per tant, si un d'aquests usuaris compra també patates fregides de bossa, aquest producte serà recomanat a l'altre o a altres possibles usuaris semblants.

A la Figura 2.7 podem veure un exemple on l'usuari de més a baix se li ha recomanat comprar raïm i taronges perquè comparteix una similitud alta amb l'usuari de més a dalt; podem dir que, a grans trets, són usuaris similars. En canvi, l'usuari del mig no s'assembla a cap altre i això provoca que, amb una estratègia *user based,* no disposem de la informació adequada per recomanar-li res.

Aquesta il·lustració ens ajuda a visualitzar que la **relació** s'estableix **d'un usuari a un usuari**, fet que dona nom a la estratègia mateixa, molts cops també anomenada *user-to-user*.

2.4 COLLABORATIVE VS CONTENT-BASED FILTERING

Per concloure, podem dir que la principal diferència entre ambdues estratègies es basa en d'on s'obté la informació necessària per realitzar la millor recomanació possible.

Per als sistemes on la informació dels usuaris sigui escassa i la dels ítems mateixos sigui molt extensa, estarà clar que amb una estratègia *content-based* aconseguirem uns resultats més acurats que amb *collaborative filtering.* Exactament igual pel cas contrari,

quan la informació dels ítems sigui escassa però no la dels usuaris, la prioritat serà l'estratègia *collaborative filtering* per sobre de *content-based*.

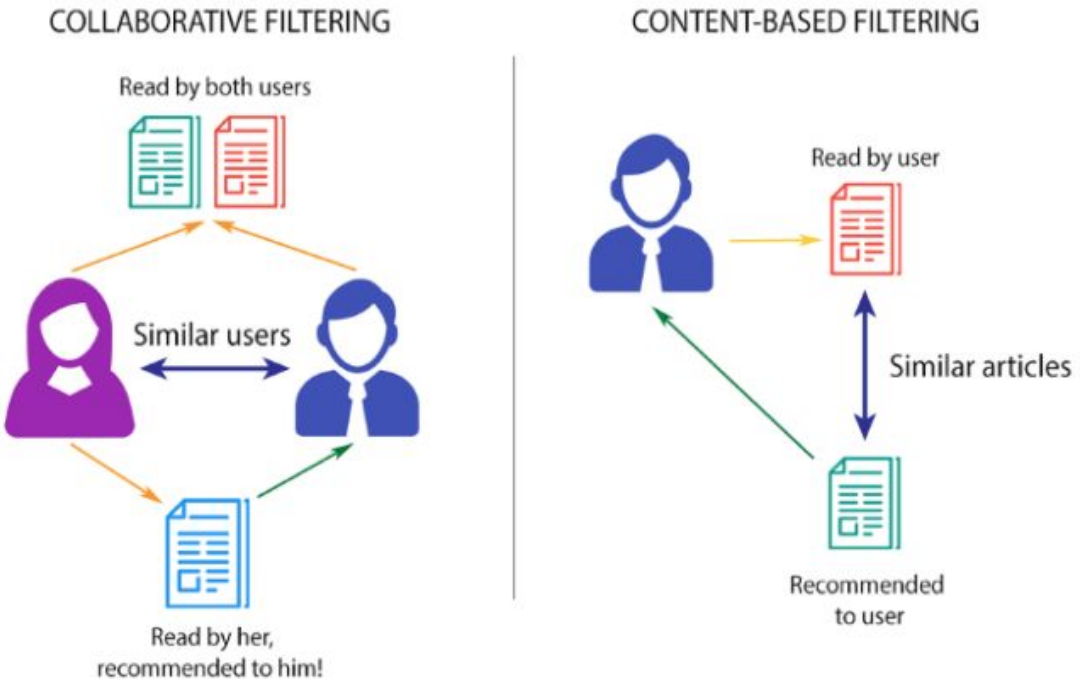


Figura 2.8 *Il·lustració que mostra les dues estratègies: collaborative filtering i content-based filtering.*

A partir d'aquí, poden aparèixer reptes molt complicats, com per exemple: Com tracten un cas d'un matrimoni que comparteix el mateix compte de Netflix? Seria possible detectar els gustos d'una de les persones de la parella tenint en compte que a una li agraden les pel·lícules d'acció i a l'altra les pel·lícules romàntiques? És a partir de situacions com les mostrades en aquests exemples que sorgeixen els sistemes híbrids.

Com hem pogut veure, tant les estratègies basades en contingut com les basades en usuaris tenen els seus avantatges i inconvenients. És precisament per aquest motiu que es plantegen sistemes de recomanació més complexos, anomenats **sistemes de recomanació híbrids** (*Hybrid recommender systems)*, que no segueixen una estratègia fixada sinó que van alternant l'obtenció d'informació segons els requeriments de cada cas concret.

La majoria d'empreses grans, avui dia, usen *Hybrid recommender systems* ja que, en algun aspecte o altre, acaben aprofitant els avantatges dels sistemes basats en el contingut (*Content-based systems*), els basats en el filtratge col·laboratiu (*Collaborative filtering systems*) i els sistemes de popularitat (*Popularity filtering*). En qualsevol cas, sempre podem descompondre una estratègia molt més complexa en qualsevol de les 3 principals ja esmentades.

3. FORMULACIÓ DEL PROBLEMA

3.1 OBJECTIU

L'objectiu d'aquest projecte és el disseny i la implementació d'un sistema recomanador a partir de l'estudi i l'anàlisi de les diferents tècniques de recomanació existents. Com a consideracions prèvies, realitzarem un estudi de l'estat de l'art de les tècniques de recomanació més esteses i ens centrarem exclusivament en un domini: les pel·lícules.

L'objectiu de dissenyar i implementar un sistema recomanador de pel·lícules és el poder entendre la *pipeline* de processos que intervenen en la creació d'un sistema com aquest. Des de com s'obtenen i es tracten les dades de les quals disposem inicialment (obtenció, estructura, format, etc.) fins arribar a obtenir una recomanació. D'aquesta manera podrem manipular les dades (característiques i mètriques, per exemple) a plaer i extreure conclusions segons el resultat per a cada una de les estratègies existents.

Un cop construït un sistema recomanador per a cada una de les 3 principals estratègies, de cara a la validació, realitzarem diverses proves modificant variables i paràmetres amb l'objectiu de mesurar l'eficiència de cada un dels algorismes i veure com es comporten en funció de les dades usades, la quantitat d'usuaris, els tipus d'usuaris, etc.

Dins del possible, com a aportació pròpia al món dels sistemes recomanadors, aplicarem canvis subtils a les fórmules més usades per demostrar si hi ha alguna casuística en concret on, el canvi proposat, funciona millor que el sistema original i tradicionalment usat.

Finalment, per poder veure i utilitzar el sistema d'una manera més pràctica i visual, dissenyarem una petita interfície gràfica on puguem aplicar les mateixes comandes escrites pel terminal però d'una manera més interactiva. D'aquesta manera aconseguirem que per a un usuari amb un perfil menys tècnic o que no estigui introduït al món dels sistemes recomanadors i les seves estratègies, pugui entendre a nivell usuari els càlculs que s'estan realitzant i el perquè darrere de cada recomanació.

3.2 SOLUCIÓ PROPOSADA

Com a solució emprada de cara a aconseguir els reptes proposats a l'apartat 3.1 OBJECTIU, s'han proposat 3 sistemes diferenciats, un per cada tipus d'estratègia principal: *popularity based*, *content-based* i *collaborative filtering*.

Per a la realització de cadascun dels 3 sistemes, s'han emprat una sèrie de dades obtingudes de diferents fonts que es troben explicades amb més detall a l'apartat d'implementació i disseny.

Segons l'estratègia a utilitzar, la informació que cal tractar del conjunt de dades d'entrada és diferent. També varien les mètriques necessàries per calcular les similituds entre ítems o entre usuaris. Es per això que ha calgut fer un estudi previ per entendre el

comportament de cada mètrica i ser capaç de veure com introduir-hi modificacions per millorar la seva eficiència.

L'objectiu final de totes les estratègies és comú: un llistat de quines són les pel·lícules que es recomanen, mostrades en ordre de més a menys pes (de més a menys recomanables). A partir dels resultats de cada estratègia, que segurament no coincidiran, s'ha donat una explicació del perquè hi ha diferències i si es pot dir que una estratègia es "millor" que una altra.

Finalment, per poder resoldre el problema de poder veure i utilitzar el sistema d'una manera més pràctica i visual, s'ha dissenyat una petita interfície gràfica on podem aplicar les mateixes comandes escrites pel terminal però d'una manera més interactiva. Les interfícies son simples traduccions del codi Python amb l'eina `Flask` que ens ha permès mostrar d'una manera més llegible, per a usuaris que no estiguin familiaritzats amb la terminal de Ubuntu, els resultats desde un navegador mitjançant codi `HTML`.

4. GESTIÓ DEL PROJECTE

4.1 ABAST DEL PROJECTE

OBSTACLES I SOLUCIONS

Un dels principals problemes que han sorgit a l'hora de començar a donar contingut al projecte ha estat el de l'obtenció de les dades.

En un primer moment, l'objectiu era crear una pàgina web en la qual els diferents usuaris a qui els donéssim accés, poguessin introduir els seus gustos, marcar d'alguna manera el botó de "m'agrada" a unes quantes pel·lícules i, d'aquesta manera, anar enriquint la nostre pròpia base de dades. En definitiva, aconseguir un conjunt d'usuaris, un conjunt d'ítems (pel·lícules) i les seves valoracions.

Aquest objectiu va ser descartat finalment ja que els possibles inconvenients eren massa grans. Bàsicament, és necessari comptar amb un elevat nombre d'usuaris, disposats a omplir certes dades (de l'ordre de milers), i informació sobre pel·lícules (de l'ordre de centenars) perquè el conjunt de dades tingui prou entitat com per poder obtenir recomanacions.

Un cop descartada per inviabilitat l'opció d'obtenir les nostres pròpies dades, la solució emprada ha estat utilitzar dades ja creades.

Com ja s'ha comentat en apartats anteriors, la gran majoria de dades han estat extretes d'IMDB (Internet Movie DataBase), ja que disposa d'una gran varietat d'informació, així com conjunts de dades de diferents tamanys que ens aniran molt bé de cara a realitzar les proves pertinents previstes per aquest projecte.

Una altra dificultat a tenir en compte ha estat en la fase d'implementació. Les estratègies més complexes, com són la *content-based* o la *collaborative filtering*, en funció del volum de les dades amb les que treballen, requereixen molt temps d'execució.

La raó de l'elevat cost computacional és perquè hi ha càlculs que s'han d'efectuar per cadascun dels usuaris "versus" tots els altres usuaris o per cadascun dels ítems versus tots els altres ítems, o el que és el mateix, un recorregut amb cost quadràtic on depenem del nombre d'elements (ítems o usuaris). Algunes de les proves treballen amb conjunts de dades (*datasets*) que superen les 27.000 pel·lícules i els 138.000 usuaris, deixant una execució d'entre 30 i 50 segons.

Aquest temps d'execució, per un usuari que vulgui utilitzar el sistema, és un temps massa elevat. Avui dia, no estem disposats a esperar 30 segons per rebre una resposta que desitgem que sigui "automàtica". La solució proposada per superar aquest obstacle ha estat fer ús d'una llibreria anomenada `pickle` de Python, que ens permet encapsular i guardar objectes Python en una cadena de *bytes* dins d'un fitxer.

L'ús que s'ha donat a la llibreria `pickle` en aquest projecte ha estat per generar arxius on resideixen els valors de les variables molt difícils de calcular pel seu elevat cost computacional, com poden ser la similitud cosinus o la predicció dels *top-k users* d'entre altres.

La primera vegada que s'executa una prova específica sobre un conjunt de dades, generem els arxius `pickle` necessaris per guardar els càlculs que s'haurien de repetir en execucions posteriors. D'aquesta manera, la primera execució serà l'única en la que haurem d'esperar bastant però, per la resta execucions, al disposar ja dels arxius amb els valors calculats, només haurem de recuperar-los per aplicar-hi la nova prova i el seu temps d'execució s'haurà reduït significativament.

4.2 METODOLOGIA DE TREBALL

Aquest projecte ha estat organitzat en diferents fases de treball, les quals es mencionen una per una i en detall més endavant.

El treball sempre ha estat incremental i ha estat basat en la metodologia àgil Scrum, una de les més usades per la gestió de projectes a nivell mundial. La metodologia Scrum es basa principalment en la gestió del desenvolupament i creació de projectes, normalment tecnològics o de programari, i aplicacions on es tenen en compte els imprevistos o dificultats sorgides del canvi d'opinions del client final (o causes similars) que permet a l'equip entregar el producte dins dels terminis establerts.

En aquest projecte no s'ha pogut realitzar la metodologia Scrum al peu de la lletra donat que únicament comptem amb una persona com a desenvolupador i una tutora (que a part de la supervisió i proporcionar ajuda, en última instància realitza el paper de client). Tot i això, s'ha intentat replicar la forma de treballar en la mesura del possible ja que, com a desenvolupador, és la forma en la que més còmode i més acostumat a treballar em trobo. Aplicar aquesta metodologia ha estat possible gràcies a eines gratuïtes, com Trello, que ens ha permès la creació de tasques amb un responsable, un termini concret i una temàtica. A l'apartat següent 4.4.3, comentarem amb més detall l'ús que s'ha específicament a Trello i a cadascun dels programaris utilitzats a nivell de gestió del projecte.

Les diferents fases del projecte han estat:

- **Fase de documentació i reunions de control (Fase General)**

  Aquesta fase abasta tota la part de creació del document de la memòria del treball, les hores de consultes i les reunions entre desenvolupador (estudiant) i "client" (tutora). Per raons òbvies, aquesta fase no pot ser ubicada a cap moment específic ni numerada ja que es realitza al llarg de tot el període disponible per a la realització del TFG (un quadrimestre complet).

- **Fase Inicial (Fase 1)**

  Aquesta fase comprèn els primers passos del projecte fora de la documentació. És la fase on es marquen els objectius i es defineix la planificació de l'equip (en aquest cas, només de l'únic desenvolupador). A més, aquesta fase inclou el procés de prematrícula del projecte, on l'estudiant ha de definir i resumir a grans trets el que vol fer perquè el director accepti la proposta i tutoritzi el treball.

- **Fase d'Anàlisi (Fase 2):**

  En aquesta fase se senten les bases perquè la part del projecte més important i extensa en hores (fase de desenvolupament) pugui ser realitzada de la millor manera possible. Consisteix en realitzar un esforç d'abstracció i calcular les necessitats (hores, recursos, etc.) i assegurar que no ens sortim dels paràmetres i terminis previstos. Els projectes tutoritzats com aquest també compten amb un gran component de presa de decisions entre una o vàries persones, per tant, és molt important el paper del director del treball en aquesta fase. Per al nostre projecte, les tasques més destacables d'aquesta fase han estat l'estudi de les tècniques de recomanació i la construcció dels prototipus.

- **Fase de Desenvolupament (Fase 3):**

  És la fase principal del projecte, la més extensa en hores amb diferència (sense comptar la fase general), Forma part d'aquesta fase la realització de qualsevol disseny o implementació de codi i interfície del projecte. Són molt importants tant la definició del nostre problema com les conclusions de les anàlisis prèvies realitzades en la fase 2, de cara a que aquesta fase no es compliqui més del previst. És normal haver de tornar a la fase 2 des de la fase 3 en determinades situacions, ja que molts reptes i obstacles són molt difícils de preveure fins que no s'ha avançat en la implementació mateixa. Aquesta pràctica, però, s'ha d'evitar al màxim sentant unes bones bases en les fases anteriors. Per al nostre projecte, les tasques més destacables d'aquesta fase han estat l'obtenció de les dades, la implementació a nivell de codi Python i la validació dels resultats de cada una de les proves realitzades amb diferents paràmetres.

- **Fase Final (Fase 4):**

  Aquesta fase inclou tota la part no mencionada anteriorment. La seva principal funció és enllestir seccions que no hagin pogut acabar-se prèviament per motiu d'obstacles imprevistos o bé per revisions posteriors. Es produeixen, la gran majoria, els dies previs al lliurament del treball mateix i solen ser tasques més superficials com el maquetar la documentació, revisar els estils, moure explicacions d'un apartat a un altre, etc.

4.2.1 EINES DE SEGUIMENT I MÈTODES DE VALIDACIÓ

Les eines emprades per al seguiment i manteniment del projecte han estat, única i exclusivament, eines en línia permetent el distanciament amb motiu de la situació actual pel virus de la Covid-19. En el cas que s'hagués requerit alguna mena de dispositiu o aula física, les complicacions haguessin estat elevades per raons sanitàries. Per sort, això no ha estat necessari i aquesta situació excepcional a l'hora de realitzar un treball com aquest, ha afectat menys del que podria semblar en un primer moment en estar tractant-se d'un projecte basat purament en software.

Com a eines software utilitzades, ens fixarem únicament en aquelles de seguiment i comunicació, per tant, obviarem les pròpies de programari:

- **Google Drive**

  Es tracta d'un servei desenvolupat per l'empresa Google d'emmagatzemament de dades a Internet que ofereix fins a 15 GB d'espai al núvol gratuïtament.

  Aquest ha estat l'espai principal de manteniment i seguiment del treball. Tots els arxius que han estat rellevants en algun moment durant la realització del treball, exceptuant els fitxers CSV que ocupen massa espai, es troben penjats al núvol en aquesta plataforma.

  Val a dir que, evidentment, els arxius han estat en tot moment accessibles i editables per l'estudiant i la directora del projecte, ja sigui alhora o no, podent visualitzar els canvis en temps real.

  Aquesta eina també ens aporta un control de versions que ens permet gestionar els arxius antics i els actuals, de manera que no ha estat necessària cap plataforma especialitzada en codi com podria ser GitHub.

  Podem trobar 3 tipus de fitxers diferenciats al Drive:

  - **Documents de text:** Inclouen documents en format pdf, txt, doc i docx. Hem treballat amb arxius com el document de la memòria, relacionats amb el seu format (plantilles que indiquen el format a seguir), transparències d'assignatures impartides per la mateixa tutora del projecte i articles extrets d'altres divulgadors/es o enginyers/es que ens aporten informació i coneixement previ sobre la matèria en qüestió.

  - **Fulls de càlcul:** Inclouen tots els documents en format xlsx (Excel). Bàsicament, són tots els arxius utilitzats per a la creació de taules i gràfiques del projecte. El fet de poder enganxar una taula a la documentació i que, en modificar l'arxiu xlsx, es modifiqui també en temps real a la mateixa documentació, és una funcionalitat que ha ajudat molt per al desenvolupament i revisió de les dades finalment mostrades. Els apunts sobre les hores i tasques en les que s'ha treballat, per tal de

poder fer un seguiment i calcular el cost final del projecte, també s'han realitzat amb arxius i taules a Excel.

- **Fitxers de codi:** Inclou tots els fitxers Python implementats, així com proves i versions de codi que han quedat obsoletes o desactualitzades. Molts d'aquests arxius de codi llegeixen d'una sèrie d'arxius CSV que no es troben penjats a Google Drive per un tema d'espai. En ser arxius massa grans, el temps de pujada i descàrrega seria farragós, i no aportaria gaire ja que, al cap i a la fi, són dades que es poden extreure de la IMDB.

- **Google Meet**

  Es tracta d'un servei de videotrucada desenvolupat per l'empresa Google totalment gratuït.

  Ha estat l'aula virtual per a totes les reunions i sessions de control del projecte. Com ja s'ha comentat prèviament, totes i cada una de les reunions han estat online mitjançant aquesta eina atesa la situació social actual del virus de la Covid-19. Gràcies a la seva funcionalitat de compartir pantalla, que permet a l'estudiant mostrar el contingut realitzat setmanalment a la tutora del projecte, el seguiment s'ha pogut fer com estava previst.

- **Correu electrònic**

  Una de les eines que sens dubte també s'ha utilitzat és el correu electrònic, mitjançant els comptes d'usuaris d'estudiants i professors oficials de la universitat UPC, @estudiantat.upc.edu i @upc.edu, respectivament.
  El seu ús no té cap misteri, simplement ha estat un eina per consultes puntuals, aclariments curts i fixació de dates per a reunions mitjançant Google Meet.

- **VirtualBox**

  Aquesta eina ha estat essencial ja que el desenvolupador necessita utilitzar tant eines pròpies del sistema operatiu Windows com del sistema operatiu Linux. La solució ha estat la utilització de l'eina VirtualBox que facilita i permet virtualitzar i parametritzar una nova màquina alternativa amb el sistema operatiu a escollir. El sistema operatiu natiu ha estat Windows 10 x64 bits i un sistema virtualitzat d'Ubuntu amb la versió 20.04.1 LTS.

Pel que fa a la validació del progrés de cada fase del projecte, hi ha hagut molt de treball autònom en el que el propi desenvolupador ha hagut de valorar per si mateix si una fase pot ser començada, si encara falta donar-li una volta o si es pot donar per acabada.

Evidentment, la paraula final és de la mateixa tutora i directora del projecte, la qual recondueix a l'estudiant en cas de no estar treballant en la direcció correcta o bé aconsella certes formes de plantejament dels problemes que, fruit de la seva experiència, coneix o pot predir i calcular molt millor. Tot i això, la llibertat de la qual ha disposat el desenvolupador/estudiant ha estat pràcticament del 100%, la tutora en cap moment ha impedit o capat cap idea que s'ha volgut implementar o que finalment sí que s'hagi acabat implementant.

Pel que fa als problemes que han sorgit, la resolució sempre ha passat perquè l'estudiant indiqués, en una reunió vía Meet o bé per correu electrònic, quin era l'obstacle o dubte trobat en un punt determinat de la implementació o documentació i, aleshores, la tutora ha pres la decisió de proporcionar a l'estudiant tota la informació que coneix sobre aquest punt i, quan ha estat necessari, ha buscat informació conjuntament amb l'estudiant

4.3 PLANIFICACIÓ TEMPORAL

TASQUES, RECURSOS I CALENDARI

La planificació temporal ha estat mesurada realitzant un sumatori de les hores emprades per l'estudiant i únic desenvolupador del projecte.

Val a dir que han estat comptabilitzades les hores de treball invertides en adquirir la informació necessària sobre el temari en qüestió de cada fase. Estem parlant de tasques com la lectura d'articles, visualitzacions de vídeos sobre la matèria, seguiment de tutorials i cerques directa o indirectament relacionades amb el treball. Donat això, el càlcul de la quantitat d'hores exactes per les tasques mencionades no ha estat del tot rigorós ja que en ocasions és difícil comptabilitzar el temps dedicat en tasques que no es basen estrictament en trobar-se davant d'un ordinador documentant o implementant.

Com a rutina utilitzada, al finalitzar cada sessió de treball, les hores emprades han estat anotades en una taula d'un fitxer Excel, cosa que ens ha ajudat a l'estimació i càlcul final d'hores invertides.

Fase	Duració	Pes (%)
Fase Inicial (Fase 1)	17 dies (19 h)	5%
Fase d'Anàlisi (Fase 2)	41 dies (114 h)	30%
Fase de Desenvolupament (Fase 3)	103 dies (133 h)	35%
Fase Final (Fase 4)	13 dies (38 h)	10%
Fase de documentació i reunions de control (Fase General)	120 dies (76 h)	20%
TOTAL:	**120 dies (380 h)**	**100%**

Taula 4.1 Sumatori i distribució de les hores emprades dividit per fases

Tasca	Duració	Pes (%)
Resum	11 dies (15 h)	1,05%
Introducció	15 dies (4 h)	3,95%
Context i actors	8 dies (30 h)	7,90%
Estat de l'art	40 dies (40 h)	10,52%
Stakeholders	1 dia (3 h)	0,80%
Formulació del problema	55 dies (41 h)	19,78%
Solució proposada	20 dies (10 h)	2,63%
Disseny i implementació	101 dies (87 h)	22,89%
Model d'avaluació	50 dies (36 h)	9,47%
Gestió del projecte	8 dies (20 h)	5,26%
Conclusions	6 dies (15 h)	3,95%
Línies futures de treball	5 dies (2 h)	0,53%
Agraïments	1 dia (1 h)	0,27%
Reunions vía Google Meet	90 dies (19 h)	5%
Tasques d'adquirir informació	115 dies (57 h)	15,84%
TOTAL:	**120 dies (380 h)**	**100%**

Taula 4.2 Sumatori i distribució de les hores emprades dividit per tasques

A la Taula 4.1 i la Taula 4.2, podem observar les diferents fases i tasques del projecte classificades per duració i pes.

El sumatori de dies d'una fase específica no ha de coincidir necessàriament amb el sumatori de dies de cada una de les tasques d'aquesta fase. Això ve donat perquè algunes tasques han estat solapades en els mateixos dies. Moltes tasques no depenen l'una de l'altra i, per tant, és perfectament factible que una tasca s'iniciï en un moment donat, es deixi en pausa, i es comenci amb una altra tasca de la mateixa o d'una altra fase, i més endavant finalitzar la primera tasca iniciada.

Pel contrari, pel que fa a les hores, el sumatori d'aquestes sí que coincideix exactament amb el sumatori d'hores de la fase en qüestió.

També ens poden fixar que no necessàriament perquè una tasca tingui una duració de

dies elevada ha de tenir també una duració d'hores elevada. La raó d'aquest comportament és perquè hi ha tasques que requereixen ser revisades i/o actualitzades poc a poc i, per tant, la seva data d'inici i data final disten molt l'una de l'altra.

4.3.1 DIAGRAMA DE GANTT

Amb el diagrama de Gantt s'ha volgut mostrar una representació gràfica de les fases on podem observar més fàcilment el *workflow* general entre fases.

- Fases: Representades amb el mateix color que a la Taula 4.1. Cada fase està composta per més d'una tasca i comprenen des de l'inici de la primera tasca fins al final de l'última tasca de cada fase.

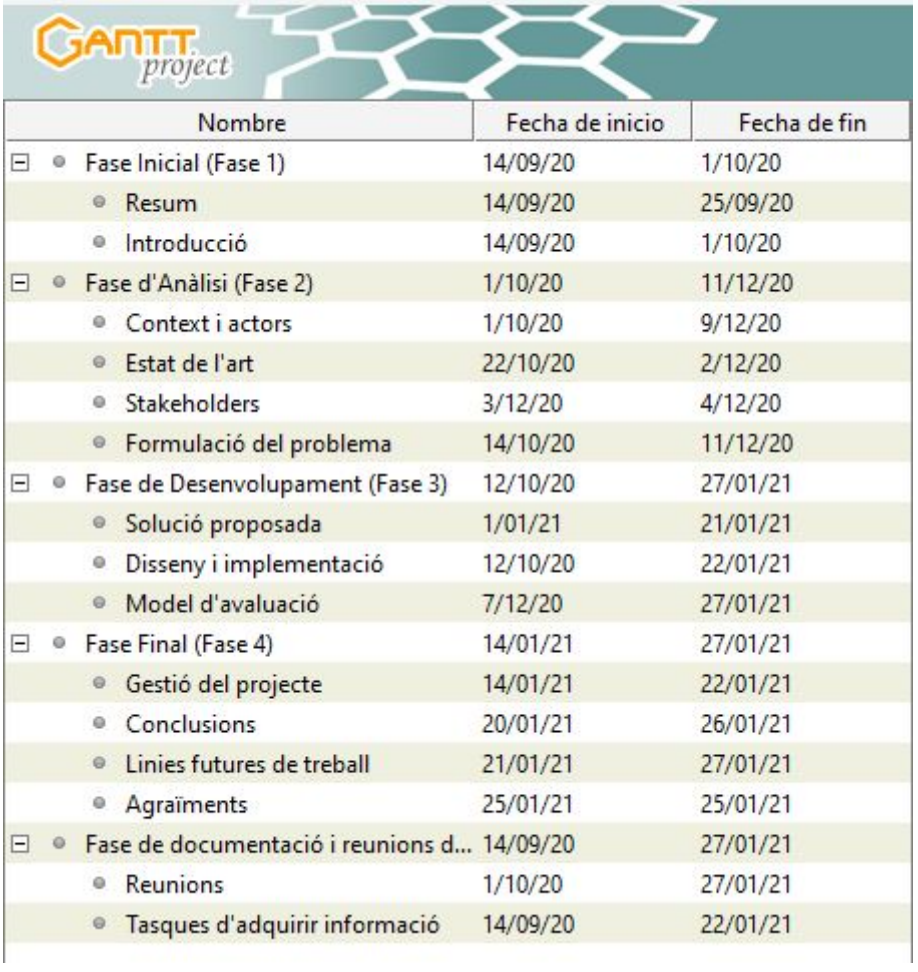
- Tasques: Representades amb el mateix color que a la Taula 4.2.

Figura 4.1 Diagrama de Gantt

Aquest mateix diagrama de Gantt el podrem trobar més ampliat al final d'aquest document, a l'Annex.

Nombre	Fecha de inicio	Fecha de fin
⊞ Fase Inicial (Fase 1)	14/09/20	1/10/20
⊞ Fase d'Anàlisi (Fase 2)	1/10/20	11/12/20
⊞ Fase de Desenvolupament (Fase 3)	12/10/20	27/01/21
⊞ Fase Final (Fase 4)	14/01/21	27/01/21
⊞ Fase de documentació i reunions d...	14/09/20	27/01/21

Figura 4.2 *Llegenda de les fases del diagrama de Gantt.*



Nombre	Fecha de inicio	Fecha de fin
⊟ Fase Inicial (Fase 1)	14/09/20	1/10/20
Resum	14/09/20	25/09/20
Introducció	14/09/20	1/10/20
⊟ Fase d'Anàlisi (Fase 2)	1/10/20	11/12/20
Context i actors	1/10/20	9/12/20
Estat de l'art	22/10/20	2/12/20
Stakeholders	3/12/20	4/12/20
Formulació del problema	14/10/20	11/12/20
⊟ Fase de Desenvolupament (Fase 3)	12/10/20	27/01/21
Solució proposada	1/01/21	21/01/21
Disseny i implementació	12/10/20	22/01/21
Model d'avaluació	7/12/20	27/01/21
⊟ Fase Final (Fase 4)	14/01/21	27/01/21
Gestió del projecte	14/01/21	22/01/21
Conclusions	20/01/21	26/01/21
Linies futures de treball	21/01/21	27/01/21
Agraïments	25/01/21	25/01/21
⊟ Fase de documentació i reunions d...	14/09/20	27/01/21
Reunions	1/10/20	27/01/21
Tasques d'adquirir informació	14/09/20	22/01/21

Figura 4.3 *Llegenda de les tasques del diagrama de Gantt.*

Per l'obtenció del diagrama i les figures s'ha utilitzat un programa de codi obert amb llicència GPL (*General Public License*), anomenat GanttProject que facilita l'administració de projectes utilitzant el diagrama de Gantt.

Aquests diagrames tenen com a objectiu exposar el temps dedicat i previst per a les diferents tasques, subtasques, activitats i fases al llarg d'un projecte en un temps determinat, però no mostra les relacions que hi pugui haver entre elles.

4.4 GESTIÓ ECONÒMICA

Per a la identificació de costos i l'estimació d'aquests, el càlcul aplicat ha estat un simple sumatori de tots els costos de recursos humans, materials (hardware) i de programari (software) utilitzats.

4.4.1 Recursos humans

A la Taula 4.3 podem observar els costos dels recursos humans del projecte.

Rol	Sou mig brut (€ / hora)	Hores emprades	Costos totals
Director de projecte	24,69 €/h	29	716,01€
Desenvolupador	13,02 €/h	380	4.947,60€
Il·lustrador	40 €/h	3	120€
TOTAL:		**412 h**	**5.783,61€**

Taula 4.3 Costos dels recursos humans del projecte

Observem 3 rols diferenciats per al nostre projecte:

- **Director del projecte** → El seu càrrec és el de coordinar, supervisar i avaluar les tasques de tots els altres rols que poden haver-hi en el projecte amb l'objectiu d'assegurar un funcionament òptim en totes les parts.
- **Desenvolupador** → És l'encarregat de dissenyar, implementar i desenvolupar totes les parts del projecte, tenint en compte sempre les restriccions i/o suggerències del director del projecte.
- **Il·lustrador** → És l'encarregat de dissenyar i crear les diferents imatges que han estat usades per al projecte. L'il·lustrador no necessàriament ha de conèixer cap aspecte tècnic sobre els sistemes recomanadors, sinó que simplement es dedica a il·lustrar la idea que el desenvolupador o el cap del projecte li indiquen, en una imatge.

4.4.2 Recursos materials

A part dels recursos humans, que suposen la part més gran del pressupost, també hem de disposar de recursos materials. La Taula 4.4 mostra el cost aproximat dels recursos materials emprats en aquest projecte.

Recurs	Preu	Quantitat	Costos totals
Ordinador mig	950€	2	1.900€
Auriculars amb micròfon	49,99€	2	99,98€
TOTAL:			**1.999,98€**

Taula 4.4 Costos dels recursos materials (hardware) del projecte

Observem 2 recursos materials:

- **Ordinador mig** → Tot el projecte es basa únicament en programari (*software*), per tant, per a la seva implantació, és necessari un ordinador amb uns requeriments mínims. Per sort, la solució proposada i els sistemes recomanadors implementats, no requereixen un ordinador amb una gran memòria. L'únic requeriment és tenir accés a programari accessible pell S.O. Windows i el mateix per Linux. S'assumeix que els ordinadors disposen de connectivitat a Internet.

- **Auriculars amb micròfon** → Donada la situació social actual respecte al virus de la Covid-19, la manera de realitzar reunions i consultes de forma remota és obligada i, per tant, disposar d'eines per poder comunicar-se virtualment és essencial.

4.4.3 Recursos de programari

L'últim tipus de cost que trobem al nostre projecte és el del programari. La gran majoria de software utilitzat ha estat completament gratuït, pel que no ha incrementat massa la suma total del pressuposts.

Recurs	Preu total
Python 3	0€
GanttProject	0€
Trello	0€
Google Meet	0€
VirtualBox	0€
Excel (Drive)	0€
Espai Google Drive (100GB)	1,99€/mes (4 mesos)
TOTAL:	**7,96€**

Taula 4.5 Costos dels recursos de programari (software) del projecte

Observem els 7 recursos de programari següents:

- **Python 3** → Llenguatge amb el que desenvoluparem la solució al problema.
- **GanttProject** → Programari per representar el diagrama de Gantt.
- **Trello** → Programari per l'organització i planificació de les tasques.
- **Google Meet** → Eina de comunicació virtual entre tutora i estudiant.
- **VirtualBox** → Programari de virtualització per poder córrer un S.O. Ubuntu.
- **Excel (Drive)** → Eina per a la realització de taules i càlculs.
- **Espai Google Drive (100 GB)** → Plataforma per emmagatzemar tots els arxius del projecte al núvol; quan l'espai a utilitzar és menor de 15 GB, aquest servei és gratuït.

4.4.4 Estimació dels costos totals

Finalment, realitzant el sumatori del total dels 3 tipus de costos que tenim en aquest projecte, tenim que el cost total del projecte és de **7.791,55€.**

Recursos	Preu total
Humans	**5.783,61€**
Materials	**1.999,98€**
De programari	**7,96€**
TOTAL:	**7.791,55€**

Taula 4.6 Cost total del projecte

5. DISSENY I IMPLEMENTACIÓ

Aquest projecte ha passat per vàries fases d'anàlisi abans de dissenyar i implementar el que finalment han estat els 3 sistemes recomanadors per separat, que donen resposta als objectius mencionats anteriorment a l'apartat 4.1 OBJECTIU.

Inicialment, la idea va ser implementar un sistema propi on poder recol·lectar les dades d'usuaris que visitessin el nostre web a base de puntuar les diferents pel·lícules que tindríem en el nostre catàleg. Després de valorar la idea i veure que no era viable atès el temps disponible per realitzar tot el projecte (d'octubre de 2020 a gener de 2021), d'entre els requeriments descartats hem pogut salvar un diagrama de classes que mostra el disseny conceptual de la informació que utilitzaria el sistema. La Figura 5.1 conté el diagrama conceptual de les dades, que mostra les entitats, atributs i relacions que es van considerar rellevants inicialment.

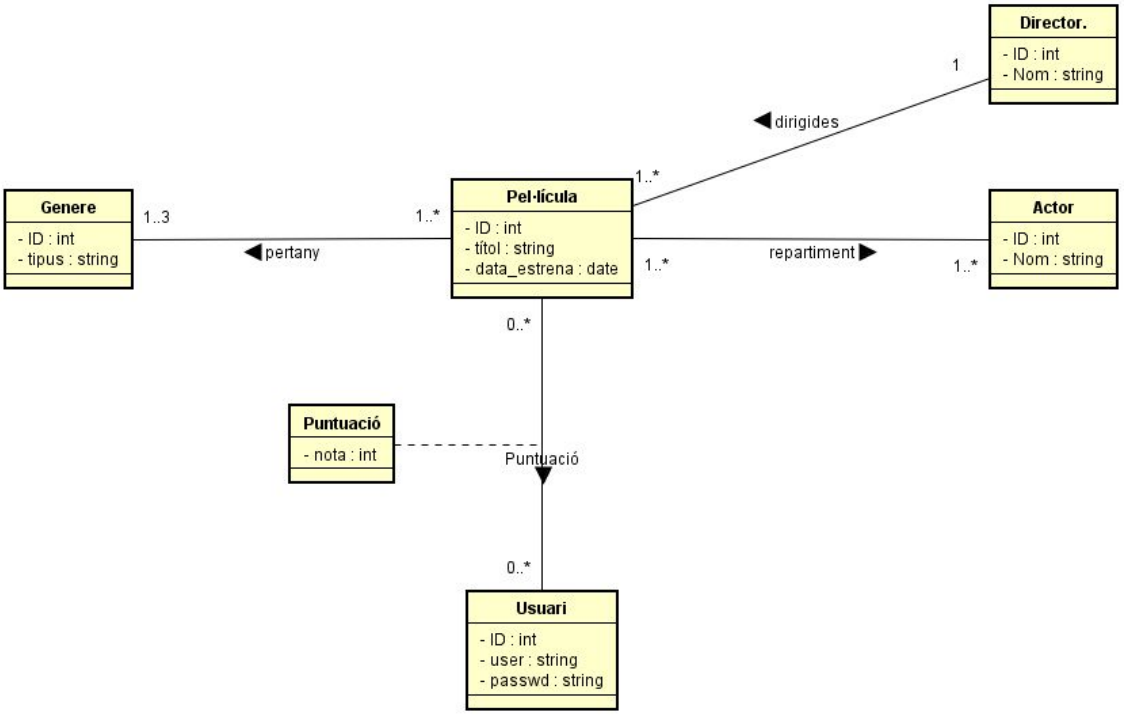


Figura 5.1 Diagrama de mòdel conceptual de les dades, inicialment plantejades pel projecte

Les entitats, atributs i relacions escollides són les que s'usen típicament a l'hora de classificar pel·lícules i, per tant, ja es tenia en compte que haurien de formar part del model.

Com podem observar, hi ha elements que no hi apareixen, com les paraules clau o *keywords*; aquestes van ser afegides més endavant ja que en un principi no es van considerar rellevants.

Pel que fa a les dades amb les que s'ha treballat, es descriuen amb profunditat a l'apartat 5.2, detallant arxiu per arxiu i quines entitats representa.

Per fer-nos una idea inicial del funcionament del conjunt de sistemes recomanadors, veurem una breu execució del menú principal que se'ns mostra a l'executar l'arxiu pare `recomanador.py`:

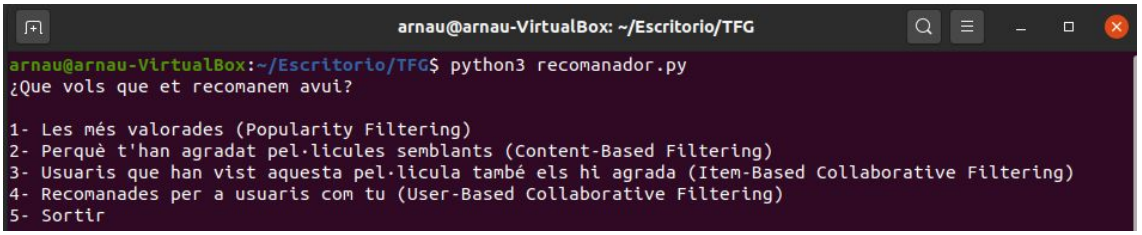


Figura 5.2 *Menú principal del sistema recomanador en un terminal d'Ubuntu.*

A la Figura 5.2 podem observar el primer l'output que se'ns mostra a mode de menú de l'execució del fitxer `recomanador.py`, que simplement fa les crides a cada una de les estratègies que es troben llistades, les quals tindran la seva pròpia lògica dins del seu propi fitxer.

5.1 SOFTWARE UTILITZAT

Per a la realització de la implementació de cadascuna de les estratègies, amb l'objectiu de mesurar l'eficiència de cada un dels algorismes i veure com es comporten, utilitzarem el llenguatge de programació Python (versió 3.8.5) amb el complement d'algunes llibreries utilitzades comunament en problemes d'aprenentatge automàtic com `pandas`, `sklearn` i `numpy`.

Figura 5.3 *Logo Python.*

Figura 5.4 *Logo NumPy.*

Figura 5.5 *Logo scikit learn.*

Figura 5.6 *Logo pandas.*

De cara a la implementació de la interfície gràfica, l'eina utilitzada ha estat `Flask`. Aquesta eina permet crear una aplicació web i traduir directament el contingut dels nostres recomanadors des del terminal a un entorn gràfic i més adaptable.

Figura 5.7 *Logo de Flask.*

5.2 DADES

La majoria de les dades utilitzades al llarg de la implementació han estat extretes de *https://grouplens.org/*. Més concretament, del projecte que duen a terme anomenat ***MovieLens***.

Es tracta d'una pàgina web que ajuda als milers d'usuaris registrats a trobar pel·lícules per veure. També duen a terme experiments en les àrees de recomanació de contingut automàtica, interfícies de recomanadors i dissenys d'interfícies d'usuaris intel·ligents, entre altres.

Els arxius o *data sets* contenen extraccions de tot tipus d'informacions recollides d'usuaris i pel·lícules al llarg del temps. Les diferenciarem segons el tamany del *data set*.

- **ml-25m.zip:** Conjunt de refència estable. Conté 25 milions de valoracions i un milió d'etiquetes[1] aplicades a 62.000 pel·lícules fetes per 162.000 usuaris. Inclou dades del que s'anomena *Tag genome*[2], amb 15 milions de valoracions de rellevància sobre 1.129 etiquetes. Publicat el desembre de 2019.

[1] Dins del context dels sistemes de recomanació, exemples d'etiquetes (*application tags*) serien: "basada en fets reals", "inspiradora", "catàstrofe", "adaptada d'un llibre", "realista", etc.
[2] El *Tag genome* és una estructura de dades que manté el pes que dona una pel·lícula a cadascuna de les etiquetes.

- **ml-latest-small.zip:** Una versió reduïda (*small*) del conjunt anterior. Conté 100.000 valoracions i 3.600 etiquetes aplicades a 9.000 pel·lícules fetes per 600 usuaris. Darrera actualització: setembre de 2018.

- **ml-100k.zip:** *MovieLens 100,000 movie ratings.* Conjunt de referència estable. Conté 100.000 valoracions fetes per 1.000 usuaris aplicades a 1.700 pel·lícules. Publicat l'abril de 1998.

- **ml-1m.zip:** *MovieLens 1M movie ratings.* Conjunt de referència estable. Conté 1 milió de valoracions fetes per 6.000 usuaris aplicades a 4.000 pel·lícules. Publicat el febrer de 2003.

- **ml-20m.zip:** *MovieLens 20M movie ratings.* Conjunt de referència estable. Conté 20 milions de valoracions i 465.000 etiquetes aplicades a 27.000 pel·lícules fetes per 138.000 usuaris. Inclou el *Tag genome* amb 12 milions de valoracions de rellevància sobre 1.100 etiquetes. Publicat l'abril de 2015; actualitzat l'octubre de 2016.

Per alguna prova en concret, s'han utilitzat diferents arxius, paràmetres o dades, però en la gran majoria podem dir que la informació usada està formada bàsicament pels següents arxius:

- **movies_metadata.csv:**
  Utilitzat per al recomanador de basat en la popularitat i el *content-based*.

  La informació més rellevant que conté és la del títol original de la pel·lícula, el resum de la trama (sinopsi) i la quantitat i la mitjana de vots comptabilitzats a TMDB (The Movie DataBase) i IMDB (Internet Movie DataBase) per a cada pel·lícula, classificada per un identificador.

  En aquest arxiu hi ha més camps que ens aporten informació a tenir en compte, però no són tan rellevants. El seu format es:

  ```
  {adult, belongs_to_collection, budget, genres, homepage, id, imdb_id, original_language, original_title, overview, popularity, poster_path, production_companies, production_countries, release_date, revenue, runtime, spoken_languages, status, tagline, title, video, vote_average, vote_count}
  ```

  Com podem veure, el fitxer original conté molta informació però aquesta només és usada per a l'estratègia *content-based*, ja que per l'estratègia basada en la popularitat només s'han tingut en compte els camps de `vote_average` i `vote_count`.

- **keywords.csv:**

Únicament utilitzat pel recomanador *content-based*.

Conté una paraula o una successió de paraules clau relacionades amb una única pel·lícula. El seu format és:

`{id, keywords}`

- **credits.csv:**
  Únicament utilitzat pel recomanador *content-based*.

  Conté informació del repartiment d'actors. El seu format és:

  `{cast, crew_id}`

  on `cast` és un camp compost format per les següents dades:

  `{cast_id, character, credit_id, gender, id, name, order, profile_path}`

- **u.data:**
  Únicament utilitzat pel recomanador *collaborative filtering*.

  Conté la informació de les valoracions de les pel·lícules per cadascun dels usuaris. El seu format és:

  `{user_id, item_id, rating, timestamp}`

5.3 RECOMANACIÓ BASADA EN LA POPULARITAT

Aquesta estratègia de recomanació és el resultat d'introduir l'opció 1 al menú principal (*Popularity filtering*). Habitualment trobarem aquest tipus de recomanacions en les plataformes comercials sota el títol de "Les més vistes", "Les que més agraden a tothom" o "Les que més nota han rebut", per exemple.

La Figura 5.8 mostra l'output (per terminal) d'una execució completa del sistema recomanador basat en la popularitat. Totes les dades d'aquesta execució han estat extretes dels fitxers explicats anteriorment, més en concret del fitxer `movies_metadata.csv`.

Aquests resultats estan basats en el càlcul del *weighted rating*, que és el mètode que s'utilitza per a qualsevol recomanació basada en la popularitat.

```
1
Buscant recommanacions...

Vot mitg:
5.618207215134185
Nombre de vots minims:
160.0
pel·licules qualificades:
(4555, 24)
                                title  vote_count  vote_average     score
314           The Shawshank Redemption      8358.0           8.5  8.445869
834                      The Godfather      6024.0           8.5  8.425439
10309      Dilwale Dulhania Le Jayenge       661.0           9.1  8.421453
12481                    The Dark Knight     12269.0           8.3  8.265477
2843                         Fight Club      9678.0           8.3  8.256385
292                        Pulp Fiction      8670.0           8.3  8.251406
522                     Schindler's List      4436.0           8.3  8.206639
23673                          Whiplash      4376.0           8.3  8.205404
5481                     Spirited Away      3968.0           8.3  8.196055
2211                  Life Is Beautiful      3643.0           8.3  8.187171
1178             The Godfather: Part II      3418.0           8.3  8.180076
1152   One Flew Over the Cuckoo's Nest      3001.0           8.3  8.164256
351                        Forrest Gump      8147.0           8.2  8.150272
1154           The Empire Strikes Back      5998.0           8.2  8.132919
1176                           Psycho      2405.0           8.3  8.132715
18465                 The Intouchables      5410.0           8.2  8.125837
40251                       Your Name.      1030.0           8.5  8.112532
289             Leon: The Professional      4293.0           8.2  8.107234
3030                     The Green Mile      4166.0           8.2  8.104511
1170                         GoodFellas      3211.0           8.2  8.077459
```

Figura 5.8 *Output de l'execució del sistema recomanador amb l'estratègia Popularity filtering.*

Abans d'explicar els aspectes més tècnics i la fórmula emprada per a les valoracions ponderades dels usuaris (*weighted ratings*), ens fixarem en les consideracions prèvies a tenir en compte perquè una pel·lícula sigui seleccionada per formar part del nostre *top*.

Si ens fixem en l'output de la Figura 5.8, abans de mostrar el rànquings de les pel·lícules, estem mostrant el nombre de vots mínims (`160.0`). Aquest és el nombre de vots que ha de rebre com a mínim una pel·lícula per ser considerada i tractada amb la nostra fórmula del *weighted ratings*. Això té una explicació molt simple i és que no té el mateix valor una pel·lícula amb una nota de 10 on només 4 usuaris l'han votada, que una pel·lícula que té una nota de 9.5 però obtinguda a partir de més de 1.000 vots d'usuaris. Clarament, la pel·lícula amb menys valoració (*rating*), en aquest cas, té molt més potencial per ser més recomanada que la primera pel·lícula tot i tenir un nota superior.

Per això, és necessari, sobretot per a les recomanacions basades en la popularitat, establir un nombre de vots mínims perquè no distorsioni el rànquing final i sigui el més fidel possible al que realment volem obtenir, que és el *top-20* de les pel·lícules amb millor puntuació del nostre catàleg de pel·lícules.

Un cop feta la criba de les pel·lícules amb el nombre de vots per sota de l'establert, podem tractar-les perquè obtinguin una puntuació o *score* final que ens servirà per indicar en quina posició de la nostra llista es situarà.

Per calcular el *weighted rating* de cada una de les pel·lícules, usem la fórmula següent:

$$WeightedRating(\mathbf{WR}) = \left(\frac{\mathbf{v}}{\mathbf{v}+\mathbf{m}} \cdot \mathbf{R}\right) + \left(\frac{\mathbf{m}}{\mathbf{v}+\mathbf{m}} \cdot \mathbf{C}\right)$$

A l'equació mostrada a dalt:

- **v** és el nombre de vots per a la pel·lícula
- **m** és el mínim de vots requerits per a ser llistada
- **R** és la puntuació mitjana de la pel·lícula
- **C** es la puntuació del vot mig d'entre totes les dades de què disposem

A nivell de codi Python, l'expressió es veu codificada com apareix a la Figura 5.9.

```
#Funció que calcula el "weighted rating" de cada pel·licula
def weighted_rating(x, m=m, C=C):
    v = x['vote_count']
    R = x['vote_average']
    # Calcul "Weighted rating" segons la formula
    return (v/(v+m) * R) + (m/(m+v) * C)
```

Figura 5.9 *Codi python de la funció que calcula el weighted rating d'una pel·lícula*.

Com a exemple, fem el càlcul amb les dades de la millor pel·lícula segons aquest recomanador i les dades d'IMDB del fitxer `movies_metadata.csv`, *The Shawshank Redemption*, títol traduït com a "Cadena perpètua".

- **v** = nombre de vots de la pel·lícula = **8358**
- **m** = mínim de vots requerits per a ser llistada = **160**
- **R** = puntuació mitjana de la pel·lícula = **8.5**
- **C** = puntuació del vot mig d'entre totes les dades de què disposem = **5.618**

Si apliquem la fórmula:

$$WR(Cadena\ perpètua) = \left(\frac{8358}{8358+160} \times 8.5\right) + \left(\frac{160}{8358+160} \times 5.618\right) = 8,445$$

Notem que obtenim un *score* de 8,445 sobre 10, igual que podem observar a la Figura 5.8 que mostra el resultat de l'execució

Aquest mètode pot produir casos curiosos on, per exemple, la pel·lícula que acabem de calcular amb un vot mig dels usuaris d'un 8,5 quedi llistada per sobre d'una amb un vot mig d'un 9,1. Aquest mateix exemple el podem observar a la mateixa figura de l'execució (Figura 5.8) si ens fixem en la tercera pel·lícula, *Dilwale Dulhania Le Jayenge*,

que acaba amb un *score* final de 8,42 quedant així per sota en el rànquing total, ja que ha estat penalitzada per comptar només amb 661 vots mentre que "Cadena perpètua" en té més de 8.000.

5.4 RECOMANACIÓ *CONTENT-BASED*

Aquesta estratègia de recomanació és el resultat d'introduir l'opció 2 al menú principal (*Content-based filtering*). Habitualment, trobarem aquestes recomanacions en les plataformes comercials sota el títol "Perquè t'han agradat pel·lícules semblants" o bé "Si t'ha agradat una pel·lícula en concret també t'agradaran aquestes".

La Figura 5.10 mostra l'output (per terminal) d'una execució completa del sistema recomanador *content-based*.

Figura 5.10 *Output de l'execució del sistema recomanador amb l'estratègia content-based sense tenir en compte tots els camps.*

A la Figura 5.10 observem la sortida d'una execució pel terminal del recomanador *content-based* després d'haver introduït pel teclat el títol de la pel·lícula *The Godfather*.

El sistema està implementat perquè ens mostri 2 outputs diferents. Per una part, inicialment, ens mostra la llista només tenint en compte l'arxiu `movies_metadata.csv`. Recordem que aquest arxiu conté dades com el gènere i el plot de la pel·lícula, entre d'altres de menys rellevància.

Per altra banda, la segona part de l'execució del sistema recomanador *content-based*, ens mostra un llistat de també 10 pel·lícules, però en aquest cas, se li sumen a l'equació els fitxers `keywords.csv`, que conté les paraules clau, i `credits.csv` que conté informació del repartiment d'actors.

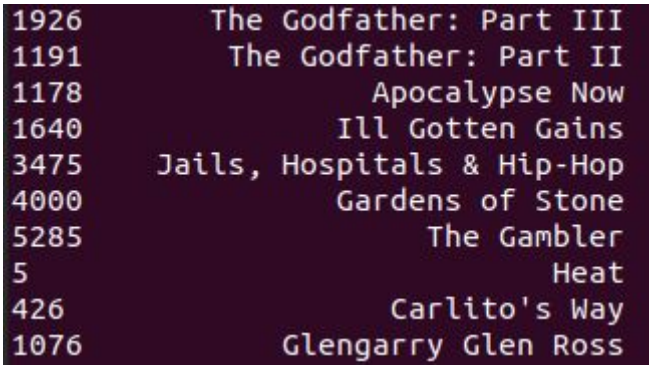


Figura 5.11 *Output de l'execució del sistema recomanador amb l'estratègia content-based tenint en compte tots els camps.*

Si fem una comparació directa entre la Figura 5.10 i la Figura 5.11, podem observar que les recomanacions canvien lleugerament. Entre els canvis més destacables, si analitzem el segon llistat que se'ns mostra, veiem que moltes de les pel·lícules comparteixen actors principals a diferència de les pel·lícules del primer llistat, que només tenen semblança de gènere i per similitud de trama.

Un exemple és la pel·lícula *Apocalypse Now* que, mentre al primer llistat ni tan sols apareix en el *top 10*, al segon llistat apareix la tercera pel·lícula a recomanar ja que l'actor Marlon Brando apareix tant a *Apocalypse Now* com a *The Godfather*.

El segon llistat, a priori, sembla més acurat ja que tenim en compte una quantitat de camps més elevada, però estem entrant en un terreny una mica subjectiu i difícilment mesurable, ja que es podria defensar que el primer llistat és més interessant per a un usuari que no para gaire atenció als actors ni als directors de les pel·lícules i únicament busca entreteniment en pel·lícules del mateix gènere.

Donat això, i amb un bon equip de màrqueting, pot haver-hi molt de joc a l'hora de mostrar o no mostrar algunes pel·lícules, depenent sempre del tipus d'usuari amb el que estiguem tractant i les dades seves que coneguem.

Hi ha diversos mètodes per calcular les recomanacions basades en el contingut. En aquest projecte s'ha usat la fórmula següent:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}^\intercal}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \cdot \mathbf{y}^\intercal{}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{y}_i)^2}}$$

Aquesta fórmula és coneguda com la **similitud cosinus** i s'usa per calcular numèricament la similitud entre dos documents i que, aplicada al nostre context, correspon a la similitud entre dues pel·lícules.

En aquests sistemes s'utilitza la similitud cosinus ja que és relativament ràpida de calcular i hem de tenir en compte que s'ha de comparar cada pel·lícula amb tota la resta

de pel·lícules que tinguem al catàleg, per tant, estem parlant d'un recorregut amb cost temporal quadràtic.

A nivell de Python l'expressió es veu representada en el codi de la Figura 5.12.

```
# Funció que rep el titol d'una pel·licula i mostra les pel·licules més similars
def get_recommendations(title, cosine_sim=cosine_sim):
    # Agafa l'index de la pel·licula que correspon al titol
    idx = indices[title]

    # Comparem el similarity score de totes les pel·licules amb aquesta pel·licula
    sim_scores = list(enumerate(cosine_sim[idx]))

    # Ordena les pel·licules segons similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Agafem l'score de les 10 pel·licules més similars
    sim_scores = sim_scores[1:11]

    # Agafem l'index de la pel·licula
    movie_indices = [i[0] for i in sim_scores]

    # Retornem el top 10 de les pel·licules més similars
    return metadata['title'].iloc[movie_indices]

#Demanem la recomanació respecte a la pel·licula que es vulgui.
print("¿Quina pel·licula t'agrada més?")
pelicula = input()

print("Recomanacions per que t'ha agradat " + pelicula+"\n")
print(get_recommendations(pelicula))
```

Figura 5.12 *Codi Python de la funció que calcula i mostra el top 10 pel·lícules més similars.*

5.5 RECOMANACIÓ *COLLABORATIVE FILTERING*

Habitualment, trobarem aquestes recomanacions en les plataformes comercials sota el títol "Perquè ha agradat a usuaris semblants a tu" o "Usuaris similars també miren:".
Com ja hem vist amb anterioritat, sabem que els sistemes recomanadors de filtratge col·laboratiu poden utilitzar dues estratègies: *user-based* i *item-based collaborative filtering*. En qualsevol de les dues tècniques, es construeix una matriu de similitud.

Per a l'estratègia *user-based*, la matriu de similitud d'usuari es calcula a partir d'una mètrica que mesura la similitud entre dos usuaris. D'igual manera, per a l'estratègia *item-based*, la matriu conté la similitud entre un parell de pel·lícules.

Una mètrica comunament utilitzada és la similitud cosinus, mètrica que ja hem utilitzat prèviament per al càlcul de l'estratègia *content based* a l'apartat RECOMANACIÓ CONTENT-BASED. La similitud cosinus ens retorna un valor entre 0 i 1, en el nostre cas, ja que no disposem de dades amb *ratings* negatius.

Un cop tenim la nostra matriu de similitud, podem començar a predir els *ratings* que no han estat inclosos en les dades, és a dir, començar a recomanar pel·lícules a usuaris que no coneixen aquestes pel·lícules en qüestió.

Per a *user-based filtering*, la predicció que utilitzem per determinar el *rating* de l'usuari *u* cap a una película *p* ve donada per la suma de tots els vots d'altres usuaris cap a la pel·lícula *p,* on el pes de cada vot dels altres usuaris es determina per la similitud cosinus entre cada usuari amb l'usuari *u*.

A nivell de Python aquesta idea es veu representada en el codi de la Figura 5.13.

```
def fast_similarity(ratings, kind='user', epsilon=1e-9):
    # epsilon es un nombre petit que ens ajuda a manejar els casos de divisió per 0
    if kind == 'user':
        sim = ratings.dot(ratings.T) + epsilon
    elif kind == 'item':
        sim = ratings.T.dot(ratings) + epsilon
    norms = np.array([np.sqrt(np.diagonal(sim))])
    return (sim / norms / norms.T)
```

Figura 5.13 *Codi Python de la funció que calcula la similitud segons l'estratègia (paràmetre `kind`).*

Classifiquem els *top-k* usuaris o pel·lícules més semblants, sent `k` un valor parametritzable. Al llarg de les execucions, el valor de `k` varia utilitzant els valors de `k-array`, on:

```
k_array = [5, 15, 30, 50, 100, 200]
```

```
def predict_topk(ratings, similarity, kind='user', k=40):
    pred = np.zeros(ratings.shape)
    if kind == 'user':
        for i in range(ratings.shape[0]):
            top_k_users = [np.argsort(similarity[:,i])[:-k-1:-1]]
            for j in range(ratings.shape[1]):
                pred[i, j] = similarity[i, :][top_k_users].dot(ratings[:, j][top_k_users])
                pred[i, j] /= np.sum(np.abs(similarity[i, :][top_k_users]))
    if kind == 'item':
        for j in range(ratings.shape[1]):
            top_k_items = [np.argsort(similarity[:,j])[:-k-1:-1]]
            for i in range(ratings.shape[0]):
                pred[i, j] = similarity[j, :][top_k_items].dot(ratings[i, :][top_k_items].T)
                pred[i, j] /= np.sum(np.abs(similarity[j, :][top_k_items]))

    return pred
```

Figura 5.14 *Codi Python de la funció que calcula els top-k similars segons l'estratègia (paràmetre `kind`).*

Una última estratègia emprada per millorar la qualitat de les recomanacions, a part de predir els *top-k*, ha estat el tractar de manera diferent els vots provinents dels usuaris "extrems". Definim com a un usuari "extrem" aquell usuari que sempre tendeix a donar vots massa baixos o massa alts a totes les pel·lícules. Per tal de relativitzar el pes d'un usuari extrem, podem suposar que el més convenient no és la puntuació absoluta que finalment ha donat a una pel·lícula sinó "ponderar" les seves valoracions en funció de la seva mitjana de vots. Per aconseguir això, primer cal calcular la mitjana de vots de cada usuari i després recalcular cada una de les seves valoracions restant-li la mitjana dels vots que ha donat.

La Figura 5.15 conté la implementació en Python d'aquesta estratègia que té en compte el biaix dels usuaris extrems.

```python
def predict_nobias(ratings, similarity, kind='user'):
    if kind == 'user':
        user_bias = ratings.mean(axis=1)
        ratings = (ratings - user_bias[:, np.newaxis]).copy()
        pred = similarity.dot(ratings) / np.array([np.abs(similarity).sum(axis=1)]).T
        pred += user_bias[:, np.newaxis]
    elif kind == 'item':
        item_bias = ratings.mean(axis=0)
        ratings = (ratings - item_bias[np.newaxis, :]).copy()
        pred = ratings.dot(similarity) / np.array([np.abs(similarity).sum(axis=1)])
        pred += item_bias[np.newaxis, :]

    return pred

user_pred = predict_nobias(train, user_similarity, kind='user')
print('Bias-subtracted User-based CF MSE: ' + str(get_mse(user_pred, test)))

item_pred = predict_nobias(train, item_similarity, kind='item')
print('Bias-subtracted Item-based CF MSE: ' + str(get_mse(item_pred, test)))
```

Figura 5.15 *Codi Python de la funció que tracta els extrems segons l'estratègia (paràmetre `kind`).*

Finalment, fem una combinació d'ambdues estratègies, *top-k* i tractament d'extrems. El codi corresponent a aquesta solució es mostra a la Figura 5.16.

```python
def predict_topk_nobias(ratings, similarity, kind='user', k=40):
    pred = np.zeros(ratings.shape)
    if kind == 'user':
        user_bias = ratings.mean(axis=1)
        ratings = (ratings - user_bias[:, np.newaxis]).copy()
        for i in range(ratings.shape[0]):
            top_k_users = [np.argsort(similarity[:,i])[:-k-1:-1]]
            for j in range(ratings.shape[1]):
                pred[i, j] = similarity[i, :][top_k_users].dot(ratings[:, j][top_k_users])
                pred[i, j] /= np.sum(np.abs(similarity[i, :][top_k_users]))
        pred += user_bias[:, np.newaxis]
    if kind == 'item':
        item_bias = ratings.mean(axis=0)
        ratings = (ratings - item_bias[np.newaxis, :]).copy()
        for j in range(ratings.shape[1]):
            top_k_items = [np.argsort(similarity[:,j])[:-k-1:-1]]
            for i in range(ratings.shape[0]):
                pred[i, j] = similarity[j, :][top_k_items].dot(ratings[i, :][top_k_items].T)
                pred[i, j] /= np.sum(np.abs(similarity[j, :][top_k_items]))
        pred += item_bias[np.newaxis, :]

    return pred
```

Figura 5.16 *Codi Python de la funció que ajunta les dues tècniques: top-k i tractament d'extrems.*

De cara a validar si aquestes dades obtingudes són del tot acurades o no, usarem la funció `Mean Squared Error` de la llibreria `scikit-learn`. Totes les proves de validació és troben llistades a l'apartat 6.

5.6 INTERFÍCIE GRÀFICA

A nivell gràfic, s'han implementat diferents interficies amb l'eina Flask mencionada a l'apartat 5.1.

Les interfícies gràfiques són una simple traducció dels codis Python amb les mateixes funcionalitats que hem vist en els apartats que descriuen les estratègies implementades.

La diferència principal és simplement estètica. En lloc de veure els resultats via terminal, els podrem veure via web (HTML).

A l'executar el nostre recomanador, podem accedir a les interfícies a través de la URL de `localhost` de qualsevol navegador amb el port 8000.

Observem el menú principal de la Figura 5.17, que és una interfície molt senzilla que mostra les dues opcions implementades a nivell gràfic: Estratègia per popularitat i estratègia *content-based*.

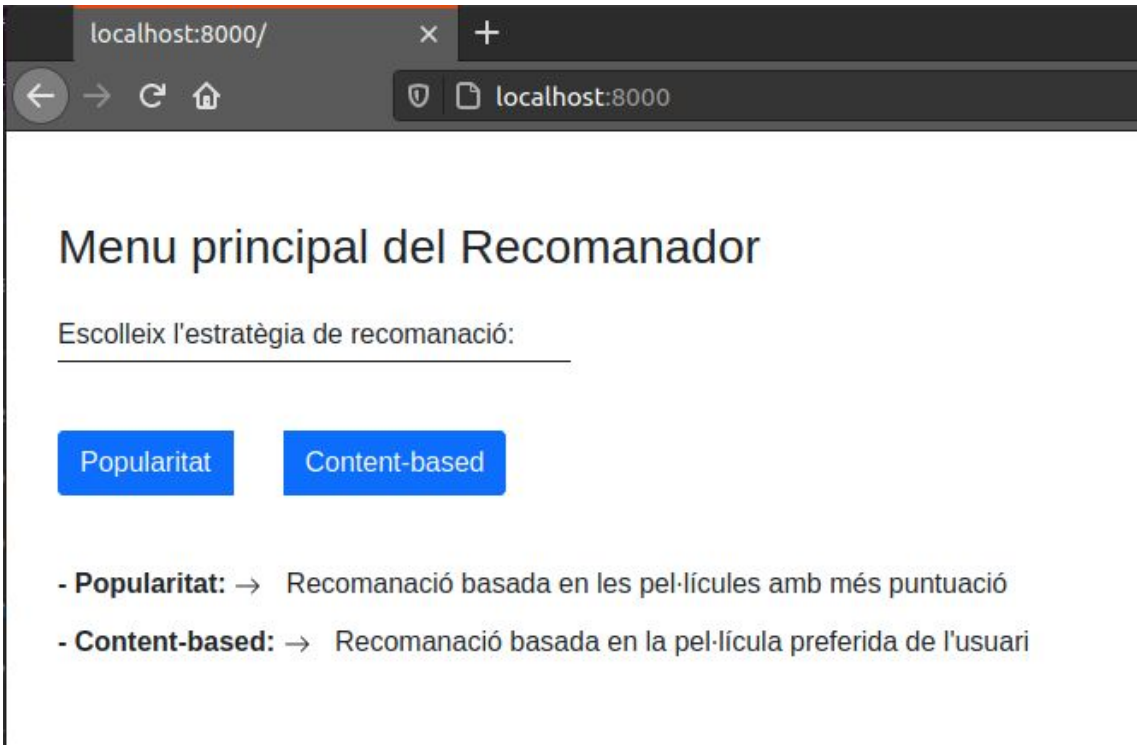
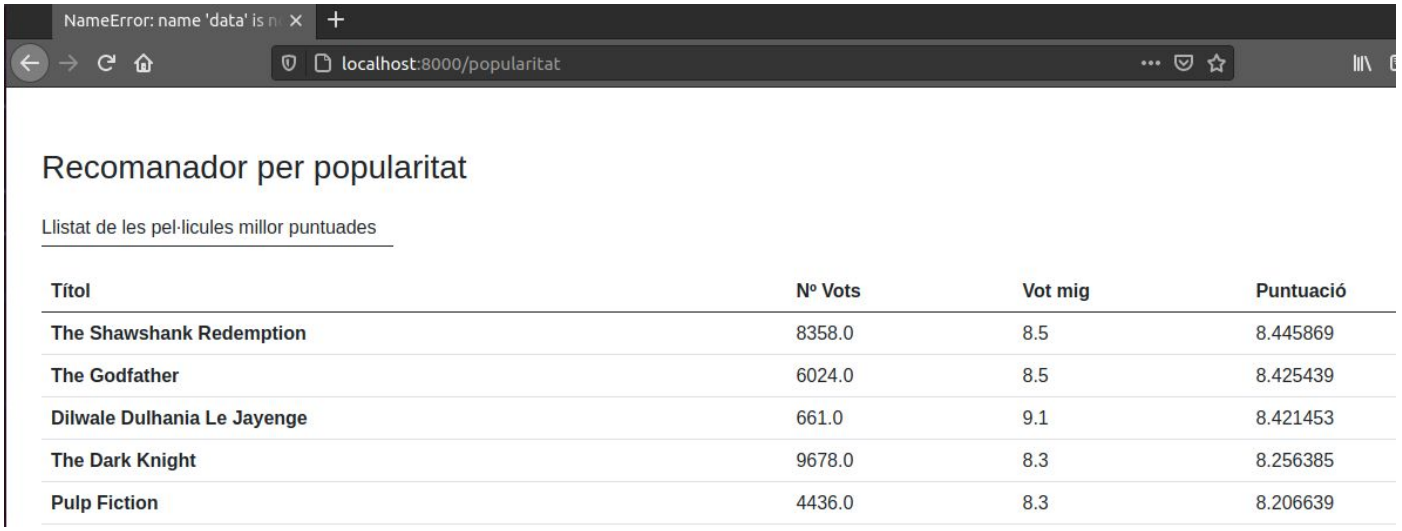


Figura 5.17 *Menú principal de la interfície gràfica.*

Prement qualsevol dels dos botons en blau, ens portarà a la pantalla corresponent del botó en qüestió per accedir a l'estratègia de recomanació seleccionada.

Al prémer el botó de `Popularitat` observem la pantalla visualitzada en la Figura 5.18.
Al prémer el botó de `Content-based` observem la pantalla visualitzada en la Figura 5.19.



Títol	Nº Vots	Vot mig	Puntuació
The Shawshank Redemption	8358.0	8.5	8.445869
The Godfather	6024.0	8.5	8.425439
Dilwale Dulhania Le Jayenge	661.0	9.1	8.421453
The Dark Knight	9678.0	8.3	8.256385
Pulp Fiction	4436.0	8.3	8.206639

Figura 5.18 *Interfície gràfica del recomanador per popularitat*.

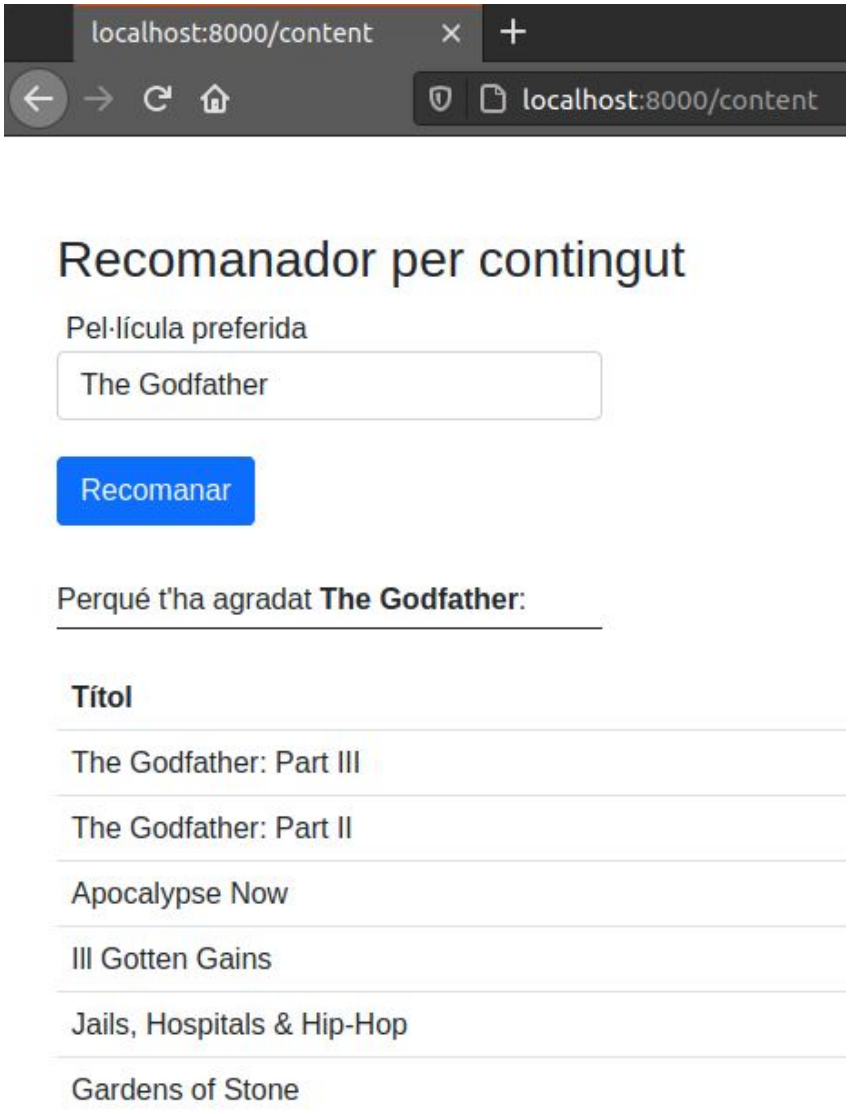


Figura 5.19 *Interfície gràfica del recomanador content-based.*

El funcionament de les interfícies és molt simple i intuïtiu. Per al recomanador *content-based* observem que podem escriure la nostra pel·lícula preferida i al prémer el botó de *Recomanar* se'ns mostra una taula amb les recomanacions basades en contingut de la pel·lícula introduïda.

6. MODEL D'AVALUACIÓ

VALIDACIÓ, TESTS, RESULTATS

Un cop vistes les implementacions, necessitem mesurar d'alguna manera com sabem si estem realitzant bones recomanacions i, a més, saber què podem considerar com a una bona recomanació en un primer moment.

Per determinar si una recomanació és bona o no l'hauríem de comparar amb les puntuacions que realment donen els usuaris. D'alguna manera seria mesurar l'error de la predicció feta pel sistema recomanador. Amb aquest objectiu, descriurem algunes mètriques basades en la mesura de l'error, però també s'inclouran altres mètriques per avaluar la qualitat de les recomanacions d'una forma alternativa.

Hi ha varies mètriques i fórmules que tradicionalment s'han utilitzat no només al món dels sistemes recomanadors, sinó que també al món de l'estadística, que ens permeten valorar la qualitat dels resultats obtinguts.

A continuació, fem una breu descripció de les mètriques, anomenades mètriques d'exactitud predictiva, que mesuren com de prop estan les puntuacions predites pel sistema recomanador respecte a les puntuacions reals fetes per l'usuari.

Mean Absolute Error (MAE)

És la mitjana de la diferència entre el valor predit pel recomanador i el valor donat per l'usuari (el *rating* o valoració que ha "votat" per aquella pel·lícula). Només volem saber la diferència entre el valor predit i el real.

La Figura 6.1 mostra un exemple del càlcul del valor de l'error mig absolut a partir de les taules que contenen les valoracions donades per un usuari (primera taula) i les prediccions donades per un recomanador (segona taula). Les diferències, en valor absolut, de cada valoració es troben a la tercera taula. El valor final és la mitjana de la suma d'aquestes diferències.

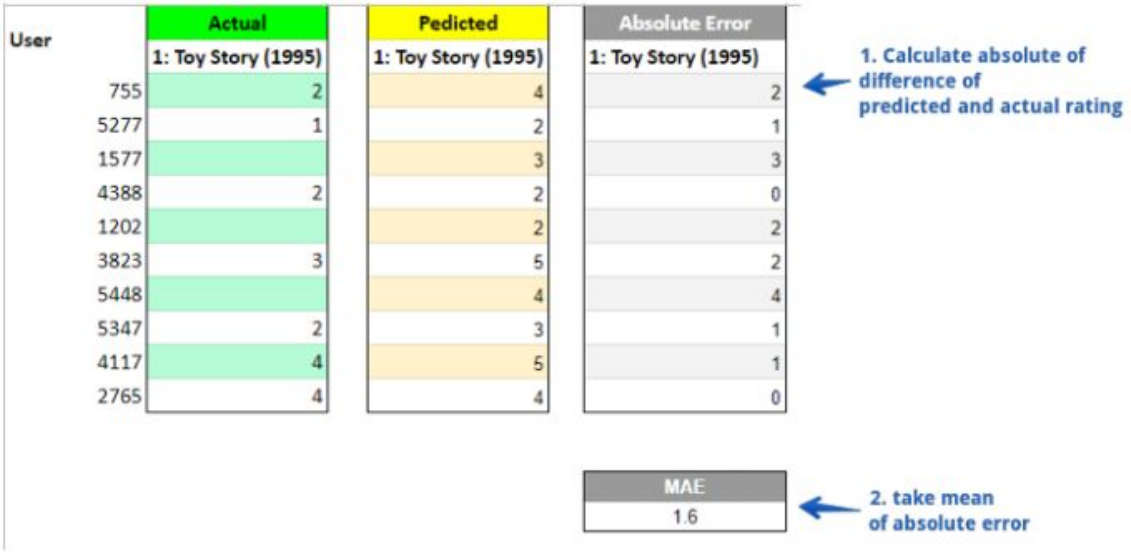


Figura 6.1 *Il·lustració en taules del càlcul error mig absolut (MAE).*

L'expressió matemàtica que calcula la mesura l'error mig absolut (MAE) és la següent:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

***Mean Squared Error* (MSE)**

L'error quadràtic mig, en anglès *Mean Squared Error*, és similar al MAE però, en aquest cas, elevem a 2 (agafem el quadrat) de l'error absolut. D'aquesta manera aconseguim que les diferències (errors) més grans tinguin encara més pes. És una manera de penalitzar justament les diferències més grans perquè s'eleven al quadrat.

L'expressió matemàtica que calcula la mesura l'error quadràtic mig (MSE) és la següent:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

La Figura 6.2 mostra un exemple del càlcul del valor de l'error mig absolut a partir de les taules que contenen les valoracions donades per un usuari (primera taula) i les prediccions donades per un recomanador (segona taula). Les diferències, en valor absolut, de cada valoració es troben a la tercera taula i en la quarta taula es troben els mateixos valors elevats al quadrat. El valor final és la mitjana de la suma d'aquestes diferències elevades al quadrat.

User	Actual	Pedicted	Absolute Error	Squared Error
	1: Toy Story (1995)	1: Toy Story (1995)	1: Toy Story (1995)	1: Toy Story (1995)
755	2	4	2	4
5277	1	2	1	1
1577		3	3	9
4388	2	2	0	0
1202		2	2	4
3823	3	5	2	4
5448		4	4	16
5347	2	3	1	1
4117	4	5	1	1
2765	4	4	0	0

MAE	MSE
1.6	4

Figura 6.2 *Il·lustració en taules càlcul de l'error quadràtic mig (MAE).*

Un cop vistes les diferents fórmules per realitzar el càlcul de l'estimació dels errors, podem determinar que MSE ens aporta uns resultats una més "fiables" que MAE, ja que penalitza més els errors.

La mètrica que hem utilitzat en a aquest projecte per al càlcul de l'error de les recomanacions ha estat el **Mean Squared Error** (MSE) o error quadràtic mig, ja que és la més estandarditzada i típicament utilitzada en aquests casos.

A continuació observarem els resultats obtinguts al aplicar MSE per a les execucions de l'estratègia *collaborative filtering* amb el conjunt de dades de l'arxiu `ml-100k`.

Hem creat una sèrie de gràfiques, amb la llibreria `matplotlib` de Python, que ens permeten mostrar a nivell visual com es comporten les dades en funció del paràmetre `k` utilitzat.

Quan parlem dels *top-k users* o *top-k items,* ens referim a *k* com al paràmetre que indica el nombre d'usuaris o ítems més similars a un usuari o ítem concret.

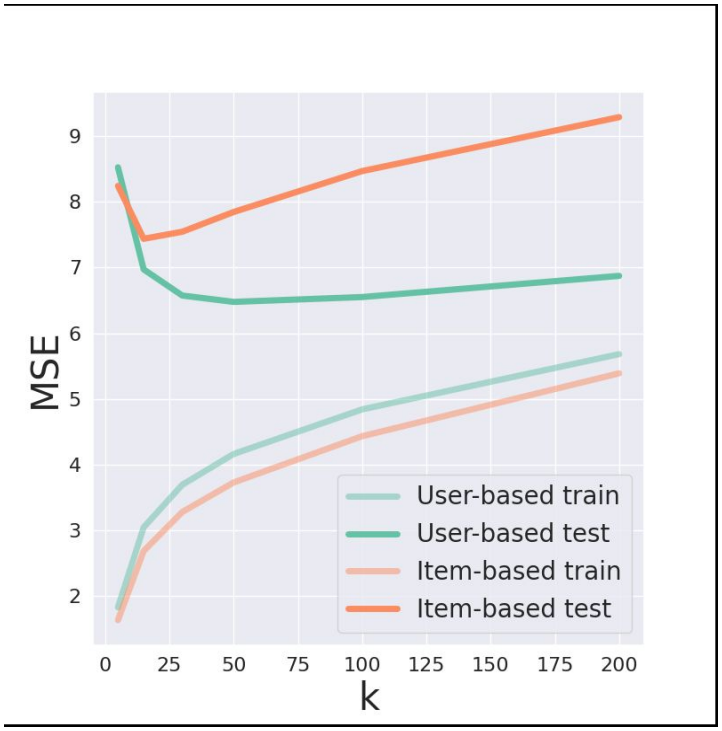


Figura 6.3 Plot del MSE aplicant estratègia top-k d'extrems

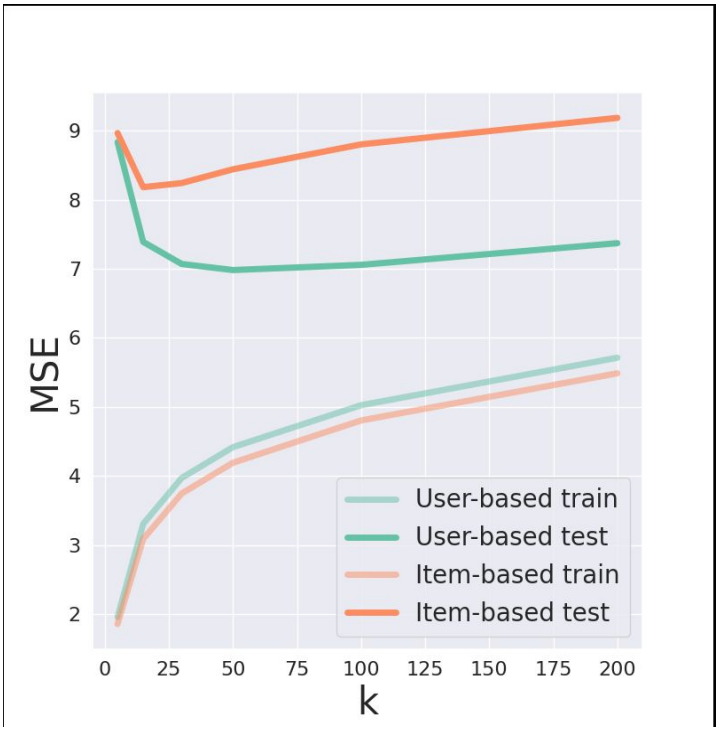


Figura 6.4 Plot del MSE aplicant estrategia top-k i tractament

A la Figura 6.3 se'ns mostra el plot del MSE tenint en compte els *top-k* users més similars a partir del conjunt de dades de l'arxiu `ml-100k`. A la figura 6.4 se'ns mostra el plot del MSE tenint en compte els *top-k* users més similars i el tractament de casos extrems a partir del conjunt de dades de l'arxiu `ml-100k`.

Com podem observar, les dades de qualsevol dels plots estan diferenciades per dos tipus de classificacions. Per una banda tenim *user-based* o *item-based* i per l'altra banda tenim *train* o *test*.

Les línies representades en verd fan referència a les dades de l'estratègia *user-based*, mentre que les representades en taronja fan referència a les dades de l'estratègia *item-based*. D'igual manera, les línies representades amb un to més fluix fan referència a les **dades d'entrenament** mentre que les línies més sòlides és refereixen a les **dades de test**.

Per entrenar un algoritme cal un conjunt de dades. Una part d'aquest conjunt serveix per entrenar l'algoritme (i construir un "model") i una altra part serveix per avaluar el rendiment de l'algoritme. Per això, el que es fa es dividir el conjunt complet en 2 subconjunts:

- **Dades d'entrenament (*training dataset*):** Subconjunt de dades que seleccionem per entrenar l'algoritme. En general, s'utilitza un 80-90% de les dades.

- **Dades de test o prova (*test dataset*):** Subconjunt de dades que seleccionem per avaluar el rendiment de l'algoritme entrenat amb les dades d'entrenament. En general, s'utilitza un 10-20% de les dades.

Ens hem d'assegurar que el conjunt de les dades de test o de prova tinguin un volum suficientment gran perquè puguin ser significatives des d'un punt de vista estadístic i que les dades d'entrenament escollides siguin representatives, és a dir, no podem escollir unes dades de prova amb caracteristiques que no coincideixin amb les dades d'entrenament.

L'objectiu és comprovar si el resultat de l'algoritme sobre les dades de prova (dades que l'algoritme no ha usat durant l'entrenament) coincideix amb el resultat correcte, que es coneix com a *gold standard*.

Tornant a les figures 6.3 i 6.4, veiem com a mesura que el nostre experiment avança, en funció del paràmetre k, les dades d'entrenament es van acostant i tendeixen a la mateixa direcció que les dades de prova.

Per tant, si ens fixem en la tendència de les dades d'entrenament, sabem que les dades obtingudes amb la millora de tractament d'extrems, tot i no semblar gaire evident, a llarg termini acabaran sent més properes a les dades de test que sense aquesta millora.

A continuació veurem eines de suport que ens ajuden a determinar, un cop vists els resultats, si hem escollit bones pel·lícules a recomanar o no.

A diferència de les mètriques d'exactitut predictiva (*MAE* i *MSE*), aquestes ens ajuden a determinar en quina mesura el recomanador ha estat útil a l'hora d'assistir a l'usuari i ha encertat recomanant certes pel·lícules i evitant d'altres

Les dues mètriques, molt utilitzades en tasques de classificació, són *Precision* i *Recall*.

Precision

És el nombre d'ítems rellevants que el sistema ha recomanat respecte el nombre total d'ítems que el sistema ha recomanat. Es centra en recomanar items rellevants assumint que hi ha d'altres que també ho són però que no es recomanaran. A l'exemple de la Figura 6.5, es pot veure que s'han recomanat tres ítems i dos d'ells són rellevants, el que dona una precisió del 66%.

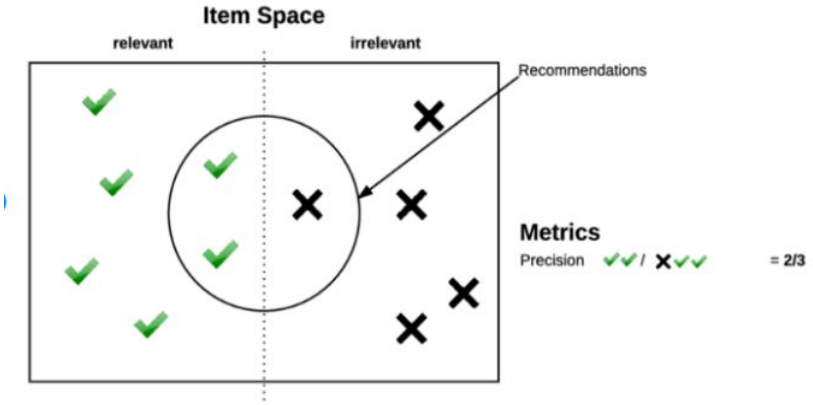


Figura 6.5 *Càlcul de la precisió.*

L'expressió matemàtica pel càlcul de la precisió és la següent:

$$Precision = \frac{nº\ de\ les\ nostres\ recomanacions\ que\ han\ estat\ rellevants}{nº\ de\ pel{\cdot}licules\ que\ hem\ recomanat}$$

Recall

És el nombre d'ítems rellevants que el sistema ha recomanat respecte el nombre total d'ítems rellevants. A l'exemple de la imatge, es pot veure que s'han recomanat dos dels sis ítems que són rellevants, el que dona un *Recall* del 33%.

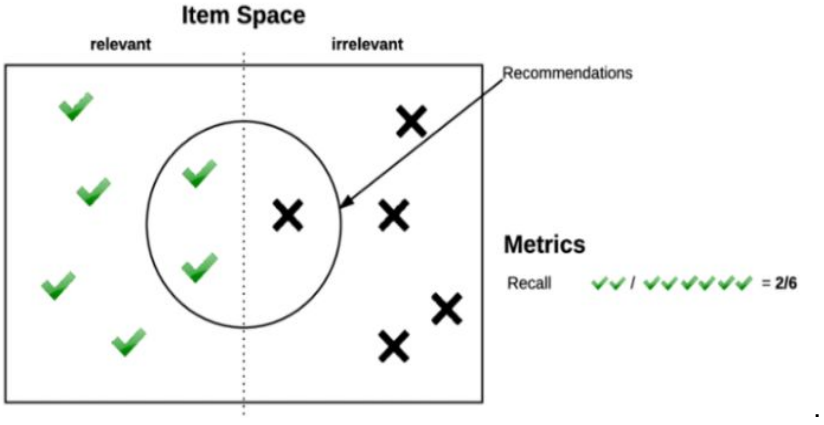


Figura 6.6 Càlcul del *recall*

L'expressió matemàtica pel càlcul del *recall* és la següent:

$$Recall = \frac{nº\ de\ les\ nostres\ recomanacions\ que\ han\ estat\ rellevants}{nº\ de\ totes\ les\ possibles\ pel{\cdot}lícules\ rellevants}$$

Eines per al rànquing dels mètodes base

Els mètodes que hem vist ens ajuden a entendre l'eficiència general del resultat que obtenim a partir del sistema recomanador, però no ens aporta informació de com els ítems han estat ordenats. Un model pot tenir un bon valor de MSE o MAE però si les primeres recomanacions no són rellevants per l'usuari, no ens és útil.

El següent mètode ens ajuda a solucionar aquest problema.

CG (*Cumulative Gain*) i DCG (*Discounted Cumulative Gain*)

Classifica entre ítems rellevants, és a dir, distingeix entre un ítem rellevant i un altre, i situa a un per sobre de l'altre, tot i que els dos no deixen de ser rellevants. Inicialment, penalitza els ítems amb menor rellevància dins dels rellevants i, finalment, dóna un valor que ens indica l'èxit de la recomanació.

Cumulative Gain només acumula el valor dels ítems rellevants independentment de la posició que ocupin en la taula, per tant, modificar la posició d'un ítem i mostrar-ho al principi de la llista o al final, no modifica el CG.

L'expressió matemàtica pel càlcul del *Cumulative Gain* (CG) és:

$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i$$

Pel contrari, *Discounted Cumulative Gain*, penalitza els ítems que apareixen més a baix a la llista. Un ítem rellevant que aparegui molt a baix a la llista, voldrà dir que el sistema recomanador no ha estat encertat.

L'expressió matemàtica que segueix pel càlcul del *Discounted Cumulative Gain* (DCG) es::

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

Per a l'exemple de la Taula 6.1, hem considerat amb una puntuació de 2 punts les pel·lícules més rellevants, amb una puntuació d'1 les pel·lícules mitjanament rellevants i amb 0 les no rellevants:

Items Ranking	Relevancy Score
Movie 1	1
Movie 3	2
Movie 2	2
Movie 5	0
Movie 4	1

Perfect Ranking	Relevancy Score
Movie 3	2
Movie 2	2
Movie 1	1
Movie 4	1
Movie 5	0

CG =	6
DCG =	12.1

CG (p) =	6
DCG (p) =	13.9

nDCG = DCG/DCG (P) =	0.87

Taula 6.1 *Càlcul del Cumulative Gain (CG) i Discounted Cumulative Gain (DCG).*

Notem que *nDCG* és un càlcul de la divisió entre el DCG obtingut i el DCG que consideraríem perfecte i ens indica un valor normalitzat entre 0 i 1.

Una altra mètrica per al rànquing de les pel·lícules és el *Mean Reciprocal Rank*.

Mean Reciprocal Rank (MRR)

Es centra en localitzar el primer ítem més rellevant. El valor MRR d'un sistema amb el primer element més rellevant a la posició 3, serà més alt que el d'un sistema amb l'element més rellevant a la posició 4. A la Taula 6.2 podem veure que els ítems rellevants són comptabilitzats segons la posició que ocupen en el rànquing.

Items Ranking	Relavent Items	Reciprocol Ranking
Movie 1	No	0
Movie 3	Yes	1/2
Movie 2	Yes	1/3
Movie 5	No	0
Movie 4	Yes	1/5

MRR =	1/2+1/3+1/5 =	1.03

Taula 6.2 *Càlcul del Mean Reciprocal Rank (MRR).*

Encara existeixen altres mètriques típicament usades per avaluar els sistemes recomanadors , tot i que depenen molt del context i el domini del sistema recomanador concrets. Tot seguit les descrivim més breument.

Novetat

En algunes plataformes, trobem en molts casos l'apartat de "novetats". En dominis com, per exemple, la música, és una bona mètrica a tenir en compte.

Diversitat

La diversitat és un altre aspecte a valorar, si hi ha una alta varietat en el nostre model, voldrà dir que sempre tindrem pel·lícules diferents per veure.

Mètriques empresarials

Sens dubte, de les més importants al món on vivim. En última instància, una companyia sempre busca un model que generi beneficis o ajudi a complir els seus objectius empresarials.

7. CONCLUSIONS

Després de tot l'estudi, la implementació i la validació de les dades, i l'anàlisi dels resultats podem dir que tots 3 algoritmes base (popularity filtering, content-based filtering i collaborative filtering) tenen la capacitat de proporcionar bones recomanacions.

El terme "bona recomanació", tot i disposar d'eines de mesura que ens indiquen els càlculs de l'error en les prediccions, dependrà molt de l'usuari que estigui sent recomanat.

Per exemple, un usuari al qual se li recomanem una sèrie de pel·lícules que s'adeqüen bastant als seus gustos però, les dues primeres pel·lícules no són exactament les que vol veure, molt possiblement acabarà pensant que la recomanació que li ha donat el sistema no es bona.

En molts casos. la tècnica a escollir per un sistema recomanador dependrà de les preguntes que ens puguem fer i sobre el que vulguem resoldre. El més important és tenir un bon coneixement del problema i pensar com el podem solucionar, ja que el millor recomanador dependrà sempre de si ens fem les preguntes correctament.

També podem determinar que, en estar tractant amb algoritmes que aprenen directament de les dades amb les quals disposem, la quantitat i qualitat de les dades influeixen directament en els resultats que s'obtenen. Donat això, els sistemes que obtenen les dades mitjançant estratègies de *deep learning* es presenten molt prometedores de cara al futur.

Si disposem d'un volum de dades suficient sobre els diferents usuaris de la nostra plataforma, una estratègia *collaborative filtering* serà sempre una garantía de que el que estem recomanant són, en última instància, les pel·lícules que els usuaris més estan consumint, i per tant, tindrem poc marge a equivocar-nos en la nostra recomanació.

Si pel contrari no disposem d'un gran volum de dades dels usuaris per alimentar el nostre sistema recomanador, no podem garantir que els resultats obtinguts siguin els esperats o desitjats amb *collaborative filtering*, per tant, haurem de buscar altres estratègies com la *content-based filtering,* on el filtratge dependrà del contingut mateix de les pel·lícules.

Tot i coneixent quin tipus d'estratègia funciona millor per cada cas, sempre serà una bona pràctica el combinar les nostres estratègies de manera que l'usuari pugui tenir suficients eleccions rellevants de pel·lícules a veure. Per exemple, una estratègia de popularitat simple, basada únicament en la puntuació que té una pel·lícula a IMDB o basada en la quantitat de "m'agrades" que hagi obtingut la pel·lícula en qüestió, dotarà al nostre sistema un salt de qualitat en quant a les recomanacions realitzades sense haver necessitat grans quantitats d'informació.

Com hem comentat amb anterioritat, per a plataformes com Netflix que disposen de molts milions d'usuaris i, una part molt elevada dels continguts que es visualitzen venen donats per les recomanacions, és imprescindible que aquests sistemes i estratègies estiguin altament depurats i siguin molt eficients, ja que hi ha una clara i directa relació entre les recomanacions i els ingressos que genera la companyia.

Veiem un exemple del sistema de recomanació de Netflix aplicat directament al meu compte personal:

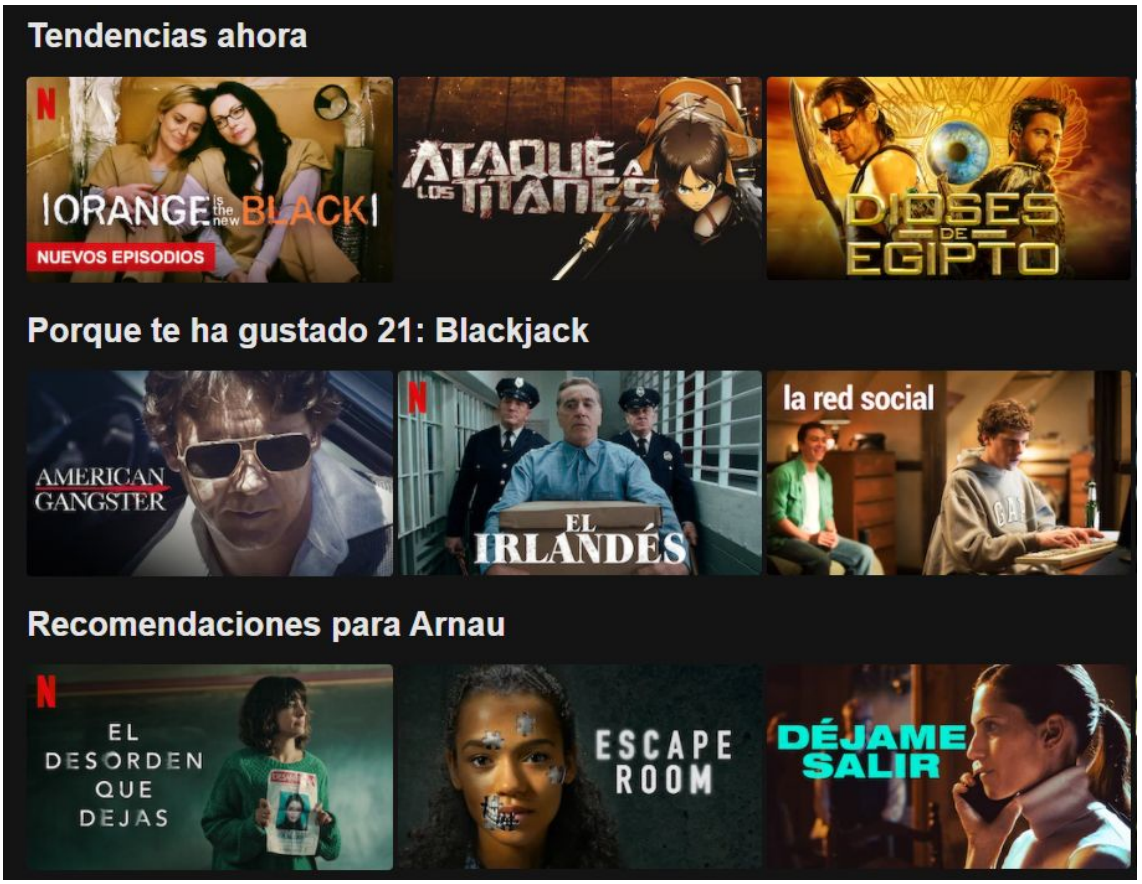


Figura 7.1 *Recomanacions del catàleg de Netflix.*

A la Figura 7.1 podem veure un clar exemple de la diversitat de resultats que ens ofereix el sistema recomanador de Netflix segons l'estratègia utilitzada.

A la part superior de la imatge de la Figura 7.1, podem observar una sèrie de recomanacions on llegim "*Tendencias ahora*". Aquestes són recomanacions basades en l'estratègia *popularity filtering*. En aquest cas, no estem parlant de les pel·lícules o sèries més populars de la història de tot el catàleg de Netflix, sinó que simplement s'està aplicant una variable de temps. D'aquesta manera tenim un indicatiu del que està sent tendència aquests dies, potser perquè han sortit nous episodis o simplement perquè s'ha posat de moda.

A la part central de la imatge de la Figura 7.1, llegim "*Porque te ha gustado 21: Blackjack*". Aquestes són recomanacions basades en l'estratègia *content-based filtering*.
Són recomanacions únicament basades en les característiques de la pel·lícula *21: Blackjack*, que jo havia marcat amb "m'agrada" i, per tant, totes les recomanacions que em fa en aquest apartat són pel·lícules que comparteixen gènere o bé actors o bé sinopsi, per exemple, amb la pel·lícula *21: Blackjack.*

Finalment a la part inferior de la imatge de la Figura 7.1, llegim "*Recomendaciones para Arnau*". Aquestes són recomanacions basades en *collaborative filtering.* Segons el criteri que hagi utilitzat Netflix, el meu compte està classificat com a cert tipus

d'usuari, per tant, les pel·lícules que m'ofereix en aquest apartat són aquelles que, seguint l'estratègia del filtratge col·laboratiu, més veuen els usuaris semblants a mi.

A nivell personal, aquest treball m'ha ajudat a entendre millor quines són les eines que les grans empreses utilitzen per a recomanar productes, pel·lícules o publicitat variada per procurar que consumeixis el producte en qüestió. Per exemple, la Figura 7.1 és una imatge extreta del meu propi compte personal de la plataforma Netflix, fet que demostra que ara sóc més conscient de l'estratègia que s'està seguint a l'hora de visualitzar un anunci o una recomanació d'una plataforma de vídeo de les que consumeixo diàriament.

Per tant, puc dir que aquest treball m'ha canviat la manera de fixar-me en la publicitat que rebo i en les recomanacions que em fan les plataformes i navegadors que uso habitualment. Suposo que com jo mateix, tots hi estem extremadament exposats diàriament en qualsevol moment que passem a Internet.

8. LÍNIES FUTURES DE TREBALL

Quan parlem de futur i de sistemes recomanadors, no hi ha una altra paraula que se'ns pugui venir al cap que no sigui l'aprenentatge profund (*deep learning).*

Sense dubte, seria molt interessant continuar en la direcció d'explorar les diferents eines que trobem en el sector de les recomanacions per a pel·lícules aplicades a la Inteligencia Artificial i, més en concret, en l'ús de les xarxes neuronals i el *deep learning*[3]*.*

Molt possiblement, sempre depenent de les dades que s'utilitzin i del cas concret, un sistema de recomanador basat en el reconeixement de patrons que aprengui i evolucioni mitjançant tècniques de *deep learning* obtindrà uns millors resultats als de les tècniques base que hem vist al llarg del projecte.

Com a possibles tasques a realitzar a partir d'aquest treball també seria molt interessant portar tots els aspectes d'aquest projecte a altres dominis diferents de les pel·lícules i les plataformes de vídeo. Per exemple, provar les mateixes tècniques presentades en aquest treball però amb dades de novel·les, articles o obres de teatre, i veure si trobem diferències significatives pel que fa a les bondat de les recomanacions. Es a dir, comprovar si, en funció del domini al que s'aplica el sistema recomanador, servirien les mateixes tècniques o no.

Un altre exemple molt interessant seria l'aplicar les estratègies de recomanació orientades cap a la venda de productes i la publicitat, com les que existeixen en qualsevol pàgina web.

En definitiva, podem trobar molts camps i aspectes per ampliar i millorar el món dels sistemes recomanadors ja que es troba en continu creixement i agafant cada cop més rellevància donada la gran dependència de les noves generacions a ser consumidors d'Internet en general.

[3] "*Deep*", en català "profund", fa referència a la profunditat o a l'elevat nombre de capes d'una xarxa neuronal.

9. AGRAÏMENTS

Vull agrair especialment a la meva tutora Neus Català Roig per tot l'esforç, l'ajuda i el constant seguiment del projecte al llarg d'aquests mesos. Estic convençut que els seus consells i treball han estat la clau per a la realització del treball en la seva totalitat.

També als meus companys de carrera Pau Olivé, per recomanar-me l'eina Flask amb la que he realitzat les interfícies web i a l'Enrique Lucena, pel seu suport al llarg del quatrimestre i l'aportació d'idees.

Per últim, agrair la realització d'algunes il·lustracions mostrades al llarg de la documentació al meu amic Daniel Prado, el qual m'ha ajudat amb els seus coneixements de disseny gràfic.

10. REFERÈNCIES

10.1 REFERÈNCIES WEB

Totes les referències web són accessibles en la data de publicació de la memòria (28 de gener de 2021).

[1] Com funcionen els sistemes de recomanació (Netflix/Amazon):
https://www.youtube.com/watch?v=n3RKsY2H-NE&ab_channel=ArtoftheProblem
[2] Divulgació respecte als sistemes recomanadors:
https://www.youtube.com/watch?v=Eeg1DEeWUjA&ab_channel=CS50
[3] Com recomana Netflix les pel·lícules? Factorització de Matrius:
https://www.youtube.com/watch?v=ZspR5PZemcs&t=1s&ab_channel=LuisSerrano
[4] Llista dels mètodes per a avaluar un sistema recomanador:
https://towardsdatascience.com/an-exhaustive-list-of-methods-to-evaluate-recommender-systems-a70c05e121de
[5] Evaluant els sistemes recomanadors | Stanford University:
https://www.youtube.com/watch?v=VZKMyTaLI00&ab_channel=ArtificialIntelligence-AllinOne
[6] Divulgació de l'avaluació dels sistemes recomanadors
https://www.youtube.com/watch?v=qG0wUgsEugw&ab_channel=WayfairDataScience
[7] Mean Average Precision (MAP) per a sistemes recomanadors.
http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html
[8] Avaluant mètriques per a sistemes recomanadors:
https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093
[9] Com construir un sistema recomanador de pel·lícules content-based:
https://towardsdatascience.com/how-to-build-a-content-based-movie-recommender-system-92352f5db7c6
[10] Raons per les que és important classificar la informació:
https://www.avepoint.com/blog/manage/5-razones-por-las-que-es-importante-clasificar-la-informacion/
[11] Adapta Netflix als teus gustos:
https://www.xatakahome.com/servicios-de-smart-tv/asi-adapta-netflix-a-tus-gustos-las-imagenes-de-presentacion-de-tus-series-favoritas
[12] Com funciona el sistema recomanador de Netflix.
https://help.netflix.com/es-es/node/100639
[13] Sistemes de recomanació amb *Machine Learning:*
https://www.aprendemachinelearning.com/sistemas-de-recomendacion/
[14] Pàgina principal d'IMDB. Llistat de les pel·lícules més votades:
https://www.imdb.com/chart/top/
[15] Construint un sistema de recomanació *collaborative filtering:*
https://realpython.com/build-recommendation-engine-collaborative-filtering/
[16] Llibreria oficial de Python. Surprise:
http://surpriselib.com/
[17] Diferencies entre collaborative filtering *item-based* i *content-based*:

https://stackoverflow.com/questions/16372191/whats-difference-between-item-based-and-content-based-collaborative-filtering
[18] Introducció als sistemes col·laboratius:
https://github.com/EthanRosenthal/DataPiques_source/blob/master/content/notebooks/2015-11-02-intro-to-collaborative-filtering.ipynb
[19] Base de Dades. GroupLens, MovieLens:
https://grouplens.org/datasets/movielens/
[20] Pàgina principal d'IMDB:
https://www.imdb.com/?ref_=nv_home
[21] Diferencies entre training data i test data:
https://lhessani-sajid.medium.com/what-is-the-difference-between-training-and-test-dataset-91308080a4e8
[22] Separació de dades (training i test):
https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data
[23] Recomanacions respecte el sistema de Netflix:
https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429
[24] Definició del Test de Bechdel:
https://ca.wikipedia.org/wiki/Test_de_Bechdel
[25] Imatge Content-based vs Collaborative filtering:
https://www.researchgate.net/figure/Content-based-filtering-vs-Collaborative-filtering-Source_fig5_323726564
[26] Pàgina web oficial de VirtualBox:
https://www.virtualbox.org/

[27] Pàgina web oficial de Trello:
https://trello.com/

[28] Definició de la metodologia Scrum:
https://ca.wikipedia.org/wiki/Scrum

[29] Pàgina web oficial de la llibreria Python Flask:
https://flask.palletsprojects.com/en/1.1.x/

[30] Cursos i exemples de llenguatge HTML:
https://www.w3schools.com/html/

[31] Descarregar boostrap i implementar-ho a un HTML:
https://getbootstrap.com/docs/3.4/components/

[32] Pàgina web oficial de GanttProject:
https://www.ganttproject.biz/

[33] Definició diagrama de Gantt:
https://es.wikipedia.org/wiki/Diagrama_de_Gantt

[34] Pàgina web oficial de Python:
https://www.python.org/

[35] Pàgina web oficial de Scikit-learn:
https://scikit-learn.org/stable/

[36] Pàgina web oficial de la llibreria Python Pandas:
https://pandas.pydata.org/

[37] Pàgina web oficial de la llibreria Python NumPy:
https://numpy.org/

[38] Pàgina web oficial de la llibreria Python matplotlib:
https://matplotlib.org/

[39] Deep Neural networks a les recomanacions de Youtube:
https://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/45530.pdf

11. ANNEX

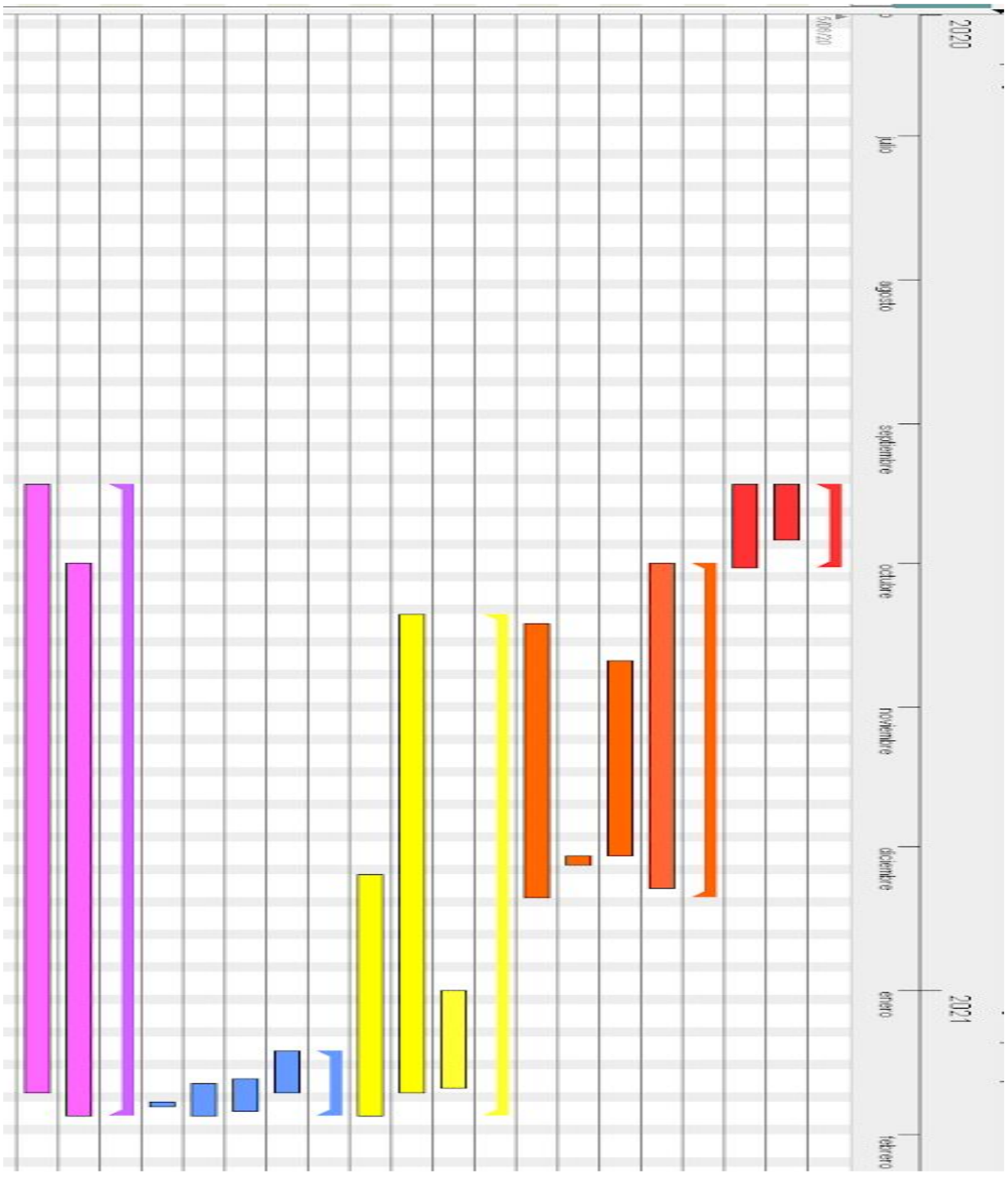


Figura 11.1 Diagrama de Gantt ampliat