UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica

Sistema recomanador per a cursos online

Treball Fi de Grau

Grau en Ciència de Dades

AUTOR/A: García Cucó, Arnau

Tutor/a: Ferri Ramírez, César

Cotutor/a: Monserrat Aranda, Carlos

Director/a Experimental: GUZMAN PONCE, ANGELICA

CURS ACADÈMIC: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Sistema recomanador per a cursos en línia

TREBALL FI DE GRAU

Grau en Ciència de Dades

Autor: Arnau Garcia i Cucó

Tutors: Cèsar Ferri Ramírez, Carlos Monserrat Aranda

Directora d'experimentació: Angélica Guzmán Ponce

Curs 2022-2023

Resum

Arran la pandèmia provocada pel COVID, moltes institucions educatives i alumnes han optat per expandir-se cap a entorns de recursos d'aprenentatge oberts, com els cursos MOOC. La Universitat Politècnica de València té publicats una sèrie de cursos que no compten amb cap tipus d'ordre. Per això la tasca d'aquest projecte ha sigut realitzar un sistema recomanador que organitze els cursos de forma personalitzada per a cada usuari. El sistema consta de tres models que busquen unir forces i superar els inconvenients que cada un d'ells presenta. Els resultats són suficientment satisfactoris, tenint el model que pitjor funciona una exactitud del 45.89% davant el 13.89% de precisió d'un model aleatori. Per a l'inici en fred, hem emprat una classificació dels cursos de manera manual ja que els resultats del clústering per distàncies no eren gaire bons. Finalment, hem desenvolupat un prototip d'app que demostra com s'implementaria el model.

Paraules clau: Sistema recomanador, filtre de contingut, filtre col·laboratiu, MOOC, aprenentatge no supervisat, aprenentatge automàtic, optimització

Resumen

A raíz de la pandemia provocada por el COVID, muchas instituciones educativas y alumnos han optado por expandirse hacia entornos de recursos de aprendizaje abiertos, como los cursos MOOC. La Universidad Politécnica de Valencia tiene publicados una serie de cursos que no cuentan con ningún tipo de orden. Por eso la tarea de este proyecto ha sido realizar un sistema recomendador que organice los cursos de forma personalizada para cada usuario. El sistema consta de tres modelos que buscan aunar fuerzas y superar los inconvenientes que cada uno de ellos presenta. Los resultados son suficientemente satisfactorios, teniendo el modelo que peor funciona una exactitud del 45.89 % frente al 13.89 % de precisión de un modelo aleatorio. Para el inicio en frío, hemos utilizado una clasificación de los cursos de forma manual ya que los resultados del clústering por distancias no eran muy buenos. Por último, hemos desarrollado un prototipo de app que demuestra cómo se implementaría el modelo.

Palabras clave: Sistema recomendador, filtro de contenido, filtro colaborativo, MOOC, aprendizaje no supervisado, aprendizaje automático, optimización

Abstract

Due to the COVID pandemic, many educational institutions and students have chosen to expand into open learning resource environments, such as MOOC courses. The Polytechnic University of Valencia has published a series of courses that do not have any type of order. That is why the task of this project has been to create a recommender system that organizes these formative documents in a personalized way for each user. The system consists of three models that seek to join forces and overcome the drawbacks that each of them presents. The results are quite satisfactory, with the worst performing model having an accuracy of 45.89% compared to the 13.89% accuracy of a random model. For the cold start, we used a manual ranking of the courses since the distance clústering results were not very good. Finally, we developed a prototype app that demonstrates how the model would be implemented.

Key words: Recommender system, content filtering, collaborative filtering, MOOC, non-supervised learning, machine learning, optimization

Índex

Índex de figures

Índex de taules

CAPÍTOL 1
Introducció

Des que l'any 1983 es creara el que ara com ara coneixem com a l'Internet, el volum d'informació a la xarxa ha anat incrementant-se de forma exponencial. L'auge en llocs web des del començament del mil·lenni ha provocat que aquest volum augmente en la mateixa mesura.

Un dels problemes més greus que enfronta la informàtica és com organitzar tota aquesta informació. És clar que no es pot mostrar de forma arbitraria, ja que existeixen nombrosos biaixos que podrien afectar el nombre de visites que es produïsquen sobre un lloc web.

Per exemple, algú poc avesat en aquesta branca de la informàtica podria pensar que una bona manera de mostrar la informació seria utilitzant la popularitat. Com més popular siga un lloc web, més probable és que el lector s'hi interesse. Però aquesta opció normalment no és la més encertada, ja que l'usuari només accedirà als primers documents. D'aquesta manera, com que els primers documents tindran més visites, continuaran mostrant-se els primers, entrant així en un cercle viciós.

Un dels mètodes que s'està utilitzant per evitar aquests biaixos en el context dels agents intel·ligents [1], és el dels models recomanadors. Aquests models pretenen captar l'interés de l'usuari segons distints paràmetres per a mostrar-li els documents que li seran més rellevants.

Aquest mètode és el més fet servir en pràcticament tots els llocs web. Des de pàgines de venda en línia com Amazon fins a plataformes de material multimèdia com Netflix, tots fan ús d'algun tipus de sistema recomanador.

1.1 Motivació i objectius

Arran la pandèmia provocada per la COVID-19, moltes àrees s'han desenvolupat cap a un ecosistema en línia. L'àmbit educatiu no n'és l'excepció i és per aquesta raó que moltes institucions i alumnes han optat per expandir-se cap a entorns de recursos educacionals oberts. D'entre aquests, els cursos massius oberts en línia (MOOC) són una de les opcions preferides i la seua popularitat s'ha expandit en gran manera els darrers anys [6].

La Universitat Politècnica de València (UPV d'ara endavant) té dins del seu catàleg web una sèrie de cursos d'aquest tipus publicats en diferents plataformes, com ara pot ser EDx o COURSERA. El problema radica en el fet que aquests cursos no tenen cap mena d'estructura organitzativa dins de la pàgina web de la universitat. Per tant, la tasca que recau sobre el nostre projecte és la de crear un model o models recomanadors que estruc-

turen d'alguna manera la forma en la qual els cursos es mostren als usuaris. D'aquesta manera s'espera que la quantitat d'usuaris que s'inscriuen als cursos s'incremente.

Així doncs, podem definir l'objectiu del nostre projecte com a la creació d'una ferramenta en un període de cinc mesos en el que, amb les dades d'entrenament, obtinga una validació a l'entrenament de, almenys, el 50% d'èxit en la recomanació. Aquest objectiu es desgranarà en metes més reduïdes tal com veurem al capítol tercer.

1.2 Estructura de la memòria

Aquesta memòria es compondrà de les parts que s'especifiquen a continuació. A la primera part, ja exposada, hem donat una introducció i una motivació sobre per què hem elaborat aquest sistema recomanador.

A la segona, explicarem el context científic en el qual hem desenvolupat la nostra ferramenta. Això vol dir que explicarem què és un sistema recomanador, les seues parts i parlarem d'alguns dels treballs previs que s'han fet d'aquests sistemes recomanadors a l'àmbit educatiu establint un punt de vista crític.

A la tercera part, parlarem del context del nostre problema. Així doncs, especificarem alguns problemes i riscos ètics i legals que es poden donar sobre la ferramenta i els hi donarem solució. També parlarem sobre com hem gestionat el projecte i en quins terminis l'hem dut a terme.

A la quarta part de la memòria descriurem completament les dades. D'aquesta manera, explicarem l'origen de les dades i la seua taxonomia, realitzarem un anàlisi descriptiu sobre les mateixes i explicarem quines tècniques de enginyeria de dades hem desenvolupat per a poder dur a terme aquest projecte.

A la cinquena i sisena part, ens centrarem en els models. La diferència entre aquestes dues és que a la cinquena part ens centrarem més en els aspectes formals del model i en el perquè hem decidit emprar aquests i no altres, mentre que a la sisena descriurem els resultats, és a dir, validarem els models i enfrontarem un dels problemes més grans dels sistemes recomanadors: el Cold Start.

A la setena part parlarem del desenvolupament d'una app provisional que ens permeta utilitzar el model recomanador, especificant àrees de seguretat i de manteniment, per a poder tindre una ferramenta que puga utilitzar-se a llarg termini.

Finalment, la darrera part serà la de les conclusions, on parlarem del que hem obtés en finalitzar el projecte, del que hem aprés durant la seua realització i del treball que podríem desenvolupar en un futur.

CAPÍTOL 2
Sistemes recomanadors

Un **sistema recomanador** és una tècnica d'aprenentatge automàtic per la qual es mostra a l'usuari, a partir d'un catàleg, una sèrie d'ítems que es puguen considerar d'interés. Per a això cal realitzar una metodologia complexa que consta de tres fases fonamentals: un recolliment i neteja de les dades, una generació de candidats i una mostra de les recomanacions. Tanmateix, tot i que no és necessari, és molt recomanable realitzar també tècniques d'anàlisi per a poder descobrir patrons amagats dins les dades i poder obtindre uns millors resultats.

Cal destacar que no se solen mostrar als usuaris els candidats en el mateix ordre en que s'han generat i se solen utilitzar algoritmes de reordenació. Açò es deu al fet que aquest tipus de sistemes, no busquen dur a terme una predicció sinó una recomanació. En una predicció, s'avalua l'interés de l'usuari sobre el producte i es dóna un valor numèric a la importància que un ítem pot tindre per al client. És, per tant, molt més arriscat que una recomanació, que simplement mostra alguns dels ítems més similars a les preferències ja descrites de l'usuari.

2.1 Elements d'un sistema recomanador

Dins d'un sistema recomanador es poden distingir diferents elements.

El **domini de recomanació** indica la taxonomia de l'ítem que s'està recomanant. No és el mateix recomanar un restaurant que una pel·lícula i aquesta és una informació que s'ha de tindre en compte. A més, al domini s'inclou la intenció del recomanador. Es pretén crear un model conservador que recomane ítems que l'usuari ja haja consumit o, en canvi, es busca crear un model que s'arrisque a mostrar nous ítems al client?

La **personalització** és probablement un dels elements més importants en un sistema recomanador, ja que, com més informació es tinga de l'usuari, més es podrà adequar la recomanació a aquesta persona en concret i millor serà l'experiència del client. Tot i això, hi ha casos en els quals les circumstàncies poden impedir que el sistema tinga accés a la informació de l'usuari. És així que distingim quatre tipus fonamentals de personalització.

- En primer lloc, tenim la personalització genèrica. És el tipus de personalització que es realitza quan el sistema no té cap tipus d'informació sobre l'usuari. La recomanació que es fa en aquests casos es basa en el conjunt agregat dels usuaris.
- En segon lloc, trobem la personalització demogràfica. És la que es dóna quan el sistema té informació sobre l'usuari però no sobre les preferències. Així doncs, es realitzen el que s'anomenen recomanacions estereotipades en les quals es recomana al mateix segment de la població el mateix conjunt d'ítems.

- En tercer lloc, estan aquelles personalitzacions que són efímeres, és a dir, que no perduren en el temps. El sistema utilitzarà informació recent de l'activitat de l'usuari per a crear una recomanació. És útil en casos en els quals no es creen perfils, com per exemple en una web de compra en línia en la qual s'afegisca un producte al carret de la compra. En aquest cas, es podria recomanar a l'usuari un producte complementari al ja afegit.

- Finalment tenim la personalització persistent, que és aquella que es realitza quan el sistema té informació del perfil de l'usuari i les seues preferències. Així, es pot utilitzar l'històric de l'usuari o els seus objectius a llarg termini per elaborar una predicció amb la màxima personalització possible.

El **context** són totes aquelles circumstàncies que envolten al client quan està accedint al model recomanador. En un cas pràctic, una persona podria entrar en un lloc web buscant un viatge per vacances. Però el sistema haurà d'adequar les seues prediccions a si busca un viatge individual o en grup.

Per a obtindre les preferències d'un usuari sobre un ítem es poden recopilar dades de dues maneres diferents.

Per un costat es pot aconseguir informació directament de la persona en forma de rànquings o de valoracions dels ítems. Aquesta **informació explícita** pot ser de llarga durada (com les crítiques a una pel·lícula), o de curta durada (com els likes i dislikes d'un post a les xarxes socials). Aquest tipus de valoracions es poden donar just després d'haver consumit l'ítem (com a un vídeo de l'Internet) o un temps després (com una crítica a un hotel).

Tanmateix, aquestes crítiques no són, en alguns casos, molt correctes, ja que poden donar-se sobre ítems amb els quals l'usuari no té molta experiència (com a la valoració d'un automòbil, perquè no molta gent compra cotxes sovint) o fins i tot poden donar-se sobre ítems que l'usuari no ha consumit, en forma de bromes o crítiques nocives. A més a més, aquest tipus d'informació no sol ser precisa, puix les persones sovint canviem d'opinió o no tenim les mateixes escales. D'altra banda, haver d'avaluar ítems recurrents, com productes alimentaris a un supermercat, pot ser molest per a certs clients, que poden decidir no valorar l'ítem si no perceben cap mena de benefici en fer-ho [3].

Una altra manera de procedir és inferir la informació de l'usuari a partir del seu comportament. Aquesta **informació implícita** [2] [4] [5] es pot obtindre a partir del temps de visualització d'un ítem (com per exemple en un article de premsa) o a partir d'accions binàries, com clicks o compres entre altres. Tanmateix, aquest tipus de dades no aporten informació sobre la valoració de l'usuari respecte a l'ítem. A més a més, cal presentar els resultats de la recomanació de manera orgànica, ja que si el client no sap com coneix el sistema les seues preferències pot sentir-se inquiet, perjudicant així la seua experiència.

Dins del procés de creació d'un sistema de recomanació, hi ha diverses etapes. Recollir i netejar les dades, generar uns candidats per a cada individu i mostrar-los.

2.2 Generació de candidats

Les aproximacions que es poden donar a un model recomanador personalitzat són molt variades. Existeixen els **models basats en el coneixement**, que busquen ítems pareguts a un seleccionat per l'usuari, però alterant alguna de les seues variables, els **recomanadors basats en casos** que busquen trobar un ítem similar valorat per la persona per a predir l'avaluació del nou o fins i tot els **models basats en diàleg** que estableixen una comunicació amb el client per tal que aquest puga canviar les seues preferències i que es mostre a la persona aquells ítems més relacionats amb els seus gustos.

Tanmateix, els models de generació predominants i que nosaltres emprarem dins del nostre sistema són els de **filtre col·laboratiu** i els de **filtre basat en el contingut**.

2.2.1. Models de filtre col·laboratiu

Als models de **filtre col·laboratiu** la idea és utilitzar perfils similars per a captar els interessos de l'usuari. D'aquesta manera es recomanaran ítems que altres persones amb gustos semblants hagen consumit.

Per exemple, posem que tenim un usuari A que està interessat en pel·lícules dramàtiques i ja ha vist *Titànic* i *La llista de Schindler*. D'altra banda, tenim un usuari B que ha vist aquestes dues pel·lícules i *La vida és bella*. El model recomanador podria generar *La vida és bella* com a candidata per a l'usuari A.

Per a fer aquest tipus de recomanacions, el model compta amb una matriu $A \in \mathbb{R}^{mxn}$ on m és la quantitat d'usuaris i n és la quantitat d'ítems. Així, el valor $A_{i,j}$ indicarà la valoració de l'usuari i sobre l'ítem j.

El model recomanador, buscarà doncs, aprendre una matriu d'usuaris $U \in \mathbb{R}^{mxd}$ donada una matriu d'ítems $V \in \mathbb{R}^{nxd}$. D'aquesta manera s'assumeix que la matriu resultant de UV^T és una aproximació acceptable de la matriu A.

Seguidament cal que es definisca una funció objectiu que prove de minimitzar la distància entre UV^T i A. Una aproximació pot ser l'error quadràtic.

$\min Z : \sum_{i,j \in obs}(A_{i,j} - < \vec{u}_i, \vec{v}_j^T >)^2$

El problema d'aquesta aproximació radica en el fet que pot donar-se el cas que molts ítems no estiguen avaluats pls usuaris, de tal manera que les pèrdues siguen ínfimes. Així doncs, les recomanacions podrien ser molt generals.

Una altra aproximació és la de minimitzar la distància al quadrat de Frobenius.

$\min Z : ||A - UV^T||_F^2$

El problema d'aquest tipus de funció és que si la matriu no és molt gran, la solució UV^T serà propera a 0 i el model generalitzarà, obtenint així recomanacions poc personalitzades.

Per a superar ambdós models, una de les solucions donades a aquest problema és la factorització per pesos. D'aquesta manera es divideix l'objectiu en dues sumes. Per una banda, la suma d'errors quadràtics en els ítems avaluats, i d'altra banda la suma de les entrades no observades.

$\min Z : \sum_{i,j \in obs}(w_{i,j} \cdot (A_{i,j} - < \vec{u}_i, \vec{v}_j^T >)^2) + w_0 \cdot \sum_{i,j \in obs}(< \vec{u}_i, \vec{v}_j^T >)^2$

on w_0 és un hiperparàmetre que busca equilibrar la suma i $w_{i,j}$ és la funció de freqüència entre el perfil i i l'ítem j.

Aquesta funció es pot optimitzar utilitzant distints algoritmes, com el descens per gradient, l'RMSProp, l'Adam, el BFGS o el WALS.

2.2.2. Models de filtre de contingut

Els models de filtre basat en el contingut empren similituds entre ítems per a recomanar a l'usuari altres similars als que ja se sap que li agraden. Així doncs, aquest model no necessita de dades d'altres usuaris, cosa que fa que la seua escalabilitat siga molt superior. A més a més, pot capturar interessos específics d'un usuari i recomanar ítems en els que poques persones estiguen interessades.

Per a construir les preferències, es poden utilitzar les variables dels ítems, de tal manera que s'obté una matriu $V \in \mathbb{R}^{mxd}$ on m és el nombre d'ítems i d és el nombre de variables de l'ítem.

D'altra banda es crearà un vector per a l'usuari ($\vec{u} \in \mathbb{R}^m$) on es registraran les valoracions dels ítems. D'aquesta manera, es pot generar un perfil de l'usuari realitzant el producte $\vec{u} \cdot V$. El problema resultant és un assumpte de similituds, on s'haurà de trobar el vector d'ítem ($\vec{v} \in \mathbb{R}^m$) més semblant al perfil de l'usuari al que anomenarem $\vec{p} \in \mathbb{R}^m$.

Per a poder trobar aquest ítem més similar, s'utilitzen mesures de distància i aquestes poden ser de diversa natura. Les més freqüents solen ser la distancia Euclídea o la de Manhattan, però també s'empren d'altres com la de Jaccard, la del cosinus o fins i tot la de Pearson.

La decisió de quina utilitzar recaurà en el domini del recomanador. Per exemple, els ítems que apareixen freqüentment tendeixen a tindre la norma del vector embedding molt gran. Per tant, si es vol recomanar els ítems més recurrents, es pot usar una mesura com la Euclídea mentre que si es vol arriscar més i mostrar ítems nous es poden emprar d'altres com la del cosinus.

2.3 Models de recomanació en cursos en línia

Com hem explicat a la introducció, el nostre objectiu serà crear un sistema recomanador per als MOOC que la UPV té disponibles a la xarxa. Imran Uddin et al. fan a l'article *A systematic mapping review on MOOC recommender systems* [7] un resum de la literatura en aquest camp fins a l'any 2021 i plantegen una taxonomia de nou tipus de sistemes recomanadors per a MOOCs (MOOCRS).

1. El primer MOOCRS que defineixen és el que busca assessorar al client a l'hora de triar la plataforma on es troba el curs. Com que nosaltres només tenim una plataforma disponible (edX), no té cap mena de sentit que realitzem aquest tipus de recomanació.

2. El segon és el model d'aprenentatge adaptatiu, en el que es recomanen cursos adaptats per a dispositius d'aprenentatge interactiu. D'aquesta manera, el que buscarà el sistema recomanador és optimitzar el feedback dels usuaris amb les dades recollides dins de la pròpia formació. En el nostre cas, no només no tenim cursos enfocats cap a dispositius d'aprenentatge interactiu, sinó que a més no tenim gaire dades sobre els cursos, per la qual cosa aquest tipus de model és inviable.

3. En tercer lloc, trobem els sistemes d'aprenentatge personalitzat. En aquest cas, els cursos no empren tècniques didàctiques generals, sinó que es focalitzen en l'aprenentatge de cada alumne donant-los una atenció particular. Com que els nostres

cursos es basen en vídeos i exercicis generals, no considerem que el sistema a desenvolupar s'incloga en aquesta categoria.

4. Existeix un quart tipus de recomanador per a cursos MOOC que es basa en requisits. En qualsevol curs, es necessita una formació anterior per a poder entendre i aprendre la matèria que s'imparteix. Els estudiants que no compleixen aquest requisit previ, solen frustrar-se i desmotivar-se, de tal manera que no solen finalitzar la formació. Per a evitar açò, algunes plataformes, com Coursera o la pròpia edX solen agrupar els cursos de manera que en un mateix lloc es puguen trobar les distintes etapes de la formació. Així doncs, quan l'alumne/a accedeix a la plataforma, pot triar quin és el curs que millor s'ajusta al seu nivell. Aquesta és una magnífica idea que implementarem dins d'aquest projecte en l'etapa de presentació de resultats, ja que ajudarà a millorar l'experiència de l'usuari.

5. A la cinquena categoria de recomanadors, trobem aquells que empren els objectius d'aprenentatge per a orientar l'estudiant a l'hora de triar un curs. És a dir, el sistema identifica quines característiques i habilitats hauria de tindre l'alumne/a en finalitzar el curs i es recomana als alumnes que necessiten reforç en aquest aspecte [8]. D'aquesta manera, s'utilitzen dades acadèmiques dels usuaris per a orientar a l'alumnat. No obstant i com veurem una mica més endavant, aquesta no és una bona idea en el nostre cas, ja que es requereix accés a unes dades que la UPV considera confidencials.

6. Una sisena classe de sistemes recomanadors per a MOOC és utilitzar el contingut del curs per a associar cursos a individus. Aquest tipus de models solen emprar tècniques com la generació de candidats per filtre de contingut.

7. Tanmateix, hem de dir que el tipus de sistemes recomanadors per a MOOC (MOOCRS) que més avantatge té a la literatura és el que fa servir els atributs cèntrics de l'usuari per a recomanar cursos. Així, molt sovint podem trobar models de recomanació de MOOC que usen generació de candidats per filtre col·laboratiu, per regles d'associació o per grafs. Un altre dels nostres models estarà enfocat d'aquesta manera

8. Una altra de les categories en les quals divideixen els MOOCRS Imran Uddin et al. és per a recomanar recursos d'aprenentatge, tals com llibres, videos, llocs web, etc. Com que el nostre domini no és aquest, tal com veurem al següent capítol, podem dir que no ens interessa un sistema d'aquestes característiques.

9. Finalment, l'últim tipus de model recomanació que es distingeix a l'article *A systematic mapping review on MOOC recommender systems* és el recomanador social, on no es busca recomanar cursos sinó altres estudiants i companys/es que comparteixen gustos similars. D'aquesta manera es genera una mena de xarxa social on les relacions poden ser unilaterals o recíproques. En el nostre cas, no ens interessa recomanar usuaris, ja que s'espera que els estudiants utilitzen el model de recomanació de forma esporàdica, no que conformen una xarxa social on bescanviar idees i opinions sobre cursos, però és un tipus de model que podria interessar-nos per la seua versatilitat i perquè pot ajudar a millorar les recomanacions de filtre de contingut i col·laboratiu.

Després d'observar aquesta taxonomia, ens vam preguntar per quina raó hauríem de crear un sistema recomanador que forme part d'una única classe. Tots els models de recomanació tenen els seus avantatges i els seus inconvenients, per la qual cosa sembla lògic pensar que si fem ús d'una combinació de models de recomanació obtindrem un millor

resultat. Així doncs, vam decidir que acabaríem fent un model de filtre de contingut, un de filtre col·laboratiu i un de filtre social com a models principals.

CAPÍTOL 3
Anàlisi del problema

Al capítol anterior, hem definit la taxonomia del nostre sistema recomanador. Malgrat això, no hem comentat res sobre els elements que el componen.

Per començar, hem de parlar del domini. En el nostre cas volem recomanar, tal com hem dit a la introducció, MOOCs elaborats per la UPV. Dins del context, assumim que un usuari no voldrà repetir un curs que ja haja aprovat, ja que es tracta de coneixements que ja haurà aprés. Tanmateix, és molt probable que vullga accedir a segones i terceres parts d'un mateix curs. Per tant, assumirem que mantindre un registre de cursos complementaris és una molt bona idea. També entenem que un estudiant no s'inscriurà a molts cursos a la vegada, de forma que l'ús del nostre sistema recomanador es donarà de forma esporàdica i no de manera continuada.

D'altra banda, la personalització a la qual aspirem en aquest projecte serà la màxima. És a dir, volem un sistema recomanador que dispose de personalització persistent. Naturalment, no podrem tindre aquesta informació quan un usuari entre de nou al sistema. En aquest cas, que anomenarem segons la literatura Çold Start", la personalització pot ser, com a màxim, demogràfica si emprem la informació que inclou l'usuari quan es crea el perfil.

Finalment, és necessari que parlem del tipus d'informació que tenim. En el nostre cas no tenim ni informació de valoracions del curs ni informació sobre les notes acadèmiques obtingudes al mateix, encara que aquesta última tampoc la utilitzaríem tal com hem explicat a l'apartat 2.3. Per tant, no tenim altra opció que utilitzar informació implícita binària: si l'usuari s'ha inscrit al curs o no.

3.1 Anàlisi de problemes ètics i legals

Tanmateix, no podem desenvolupar aquest projecte sense tindre present els afers ètics i legals que podrien ocupar la nostra ferramenta.

El primer problema al qual ens enfrontem, és el de privacitat. Les dades que tenim pertanyen a usuaris privats i, per tant, poden no voler que es difonguen. Per això, haurem de tractar les dades amb cura, no podrem tindre accés obert a les mateixes i hauran de seguir un procés d'anonimització a l'hora de tractar-les.

Un dels altres problemes fonamentals al que ens enfrontem en crear aquesta ferramenta és el de biaix a l'hora de recomanar cursos. Cal considerar tots els cursos per igual i no recomanar més els cursos més populars, ja que això perjudicaria significativament als qui han elaborat els cursos amb menys visites. Per a solucionar aquest problema, s'ordenaran els candidats per ordre invers de popularitat. És a dir, els menys populars, se situaran primer mentre que els més populars se situaran darrere. D'aquesta manera assegurem als distints MOOCs, una igualtat d'oportunitats de ser recomanats dins del sistema.

Finalment, l'últim problema que pot sorgir amb la nostra ferramenta és el de recomanacions justes. Dins del nostre repositori de dades tenim informació que pot arribar a considerar-se sensible i pot provocar que disgreguem. Parlem d'atributs com pot ser l'edat, el gènere o la nacionalitat. Com que indubtablement utilitzar aquestes característiques va a dur el nostre model a discriminar, se'ns planteja un greu problema amb el nivell de personalització durant l'inici en fred, ja que no podrà ser demogràfica. Per aquesta raó, per als nouvinguts al sistema, es desenvoluparan recomanacions genèriques (per taxonomia, antiguitat, etc).

3.2 Anàlisi de problemes tècnics

Ara que hem estudiat els problemes de caràcter ètic i legal que poden sorgir de la creació d'un sistema recomanador, podem adreçar altres qüestions més enfocades en la part tècnica. Per començar, podem parlar de quines dades emprarem.

R. Obediat et al. a l'article *A collaborative recommendation system for online courses recommendations* [10], fan servir dades de caràcter acadèmic. Aquest treball es desenvolupa en el marc de cursos de la plataforma edX, com nosaltres. Tanmateix, cal que mencionem que l'objectiu del seu sistema recomanador no era assignar a cada usuari els cursos més interessants, sinó optimitzar el rendiment dels estudiants en els cursos. Així, es recomanaran sempre els cursos per als quals s'espere que l'usuari obtinga millors resultats. El problema que pot sorgir ací és que es genere un biaix de simplicitat, és a dir, que sempre es recomanen els cursos amb major percentatge d'aprovats. Per aquesta raó, considerem que fer ús de la nota acadèmica obtesa com a mètrica de l'interés de l'usuari per la formació no és una bona idea. Així i tot, obri un camí molt interessant en usar dades acadèmiques per a realitzar la recomanació.

A l'article *A recommender system for on-line course enrolment: an initial study*, M. P. O'-Mahony i B. Smyth [9] segueixen una idea semblant. La universitat de Dublín té un programa d'inscripció en línia i el treball elaborat per aquests dos autors consistia a realitzar un sistema recomanador que, en base als estudis cursats per l'alumne/a, elabora una sèrie de recomanacions per a les assignatures optatives. L'únic inncovenient que pot existir i en el qual ens trobem al nostre projecte, és el del conflicte legal a l'hora de creuar les dades per dur a terme aquesta millora, ja que tots els estudiants tant de la universitat com d'edX haurien d'autoritzar el tractament de les seues dades i fins i tot així, creuar

les dades d'ambdues bases de dades seria difícil. Açò és senzillament poc factible per al nostre projecte, per la qual cosa, decidírem no usar aquestes dades.

Nosaltres emprarem dades demogràfiques dels usuaris, que recollirà edX i, sobretot, dades dels cursos que hi ha disponibles. Malgrat això, aquestes dades són insuficients per a elaborar un sistema recomanador en condicions, per la qual cosa caldrà augmentar la informació disponible d'alguna manera.

R.Huang i R.Lu van desenvolupar a l'article *Research on content-based MOOC recommender model* [12] un sistema recomanador de filtre de contingut sobre les dades de la plataforma de MOOCs iCourse. El punt més interessant de la seua obra és que, com nosaltres, comptaven amb molt poca informació referent als cursos, per la qual cosa van optar per extraure més dades a partir de la descripció d'aquests. Per a fer-ho, van utilitzar una tècnica de processament de llenguatge natural àmpliament coneguda al món dels crawler web: TF-IDF. La idea d'aquest mètode és que una paraula que es repetisca molt a un article, però poc al conjunt d'articles representa el tòpic del document. Aquest tipus de tècniques són idònies per al problema d'extracció de característiques que tenim a mà, per la qual cosa utilitzarem nosaltres també algun tipus de tècnica de llenguatge natural per a inferir més atributs als cursos que estudiem.

El problema rau quan no puguem tindre informació de l'ítem o de l'usuari perquè entren de nou al sistema. En principi, no deuria d'haver-hi problema amb un nou curs, ja que, com hem vist, la informació sorgeix de la mateixa definició de característiques de la formació. Tanmateix, sí que podem tindre problemes quan provem d'inserir un estudiant de nou ingrés al sistema, car no tenim cap manera d'entendre quines seran les seues preferències. A.L.V. Pereira i E.R. Hruschka plantegen al seu article *Simultaneous co-clustering and learning to address the cold start problem in recommender systems* [14] emprar les dades demogràfiques dels usuaris per a resoldre aquest problema d'inici en fred. Tanmateix, això planteja el mateix problema que totes les dades d'aquest caràcter tenen: els resultats estaran esbiaixats. Un altre mètode usat en alguns estudis com el de K. Ji et al. [15] és emprar algoritmes que tracten les preferències de l'usuari de forma equitativa. És a dir, l'usuari no té preferències encara per la qual cosa s'assumeix que valorarà en el mateix grau tots els atributs dels cursos. El problema que sorgirà irremeiablement d'aquesta idea és que tots els nous usuaris tindran exactament la mateixa recomanació, ja que hi haurà un nivell mínim de personalització. Un altre estudi que cal que citem és *Addressing cold start in recommender systems: a semi-supervised co-training algorithm* [16], desenvolupat per M. Zhang et al., que planteja l'ús d'un algoritme semisupervisat i conscient del context anomenat CSEL. Aquest algoritme, tot i que extremadament complex en l'apartat de regressió semi-supervisada, planteja un precedent molt interessant en utilitzar informació addicional a la interna del sistema. El que nosaltres proposarem, doncs, anirà en aquesta línia i demanarem informació addicional als usuaris per a poder realitzar una millor recomanació.

Un altre problema que encara no hem tingut en compte és l'enorme quantitat de dades que es recullen als sistemes recomanadors. I és que, com hem comentat a la introducció, existeix una quantitat immensa d'usuaris cercant cursos en línia. Per això el que plantegen a l'article *MCRS: A course recommendation system for MOOCs* [11] H. Zhang et al. és implementar el model utilitzant ferramentes com Spark. És una molt bona idea, però queda molt enllà de l'abast del nostre projecte, ja que, com definirem en el pròxim capítol, nosaltres només arribarem a la realització d'un programa demostració i no implementarem el model obert al públic.

Tanmateix, altres autors també empren diferents tècniques per a reduir el cost temporal de la computació del model. Per exemple, R. Obediat et al. apliquen clústering sobre els usuaris abans d'extreure les regles d'associació [10] i Y. Pang et al. a l'article *Collabora-*

tive filtering recommendation for MOOC application [13] fan servir també contenidors amb grups d'usuaris. Ambdós estudis demostren que agrupar els usuaris fent ús d'aquestes tècniques és útil a l'hora d'obtindre bons resultats, per la qual cosa podem afirmar que és una bona idea examinar aquesta distribució dels usuaris. La única pega que podríem tindre és que dels usuaris només tenim dades demogràfiques, per la qual cosa aquests resultats és molt probable que estiguen esbiaixats per gènere, edat o país d'origen. Per aquesta raó, no usarem aquestes dades, sinó que emprarem d'altres, tal com s'explica al capítol cinqué.

3.3 Gestió del projecte

Ara parlarem de la gestió específica del nostre projecte. Com hem comentat abans, l'objectiu del nostre projecte serà la creació d'una ferramenta en un període de cinc mesos en el que, amb les dades d'entrenament, aconseguisca una validació a l'entrenament de, almenys, el 50% d'èxit en la recomanació. Per tal de fer-ho, realitzarem un total de tres models: un de filtre de contingut, un de filtre col·laboratiu i l'altre basat en recomanacions per usuaris. Tots tres, així com tot el projecte, es desenvoluparà en el llenguatge de programació Python [17] i en cada apartat s'especificarà quines llibreries s'aplicaran.

Per a assolir aquest objectiu, comptem amb dos períodes: el primer de tres mesos i el segon de dos mesos. L'equip de treball estarà format per un alumne que executarà les activitats del projecte i tres professors que supervisaran el seu desenvolupament i assesoraran en el que consideren pertinent. La metodologia que seguirem durant el primer període serà realitzar reunions setmanals englobades dins de la Càtedra d'Intel·ligència Artificial Aplicada a l'Administració Pública. Durant el segon període, procedirem d'una manera diferent: l'alumne elaborarà informes setmanals per als professors encarregats de la supervisió i es reunirà l'equip en reunions mensuals.

Respecte als entregables, a banda dels informes que ja hem mencionat anteriorment, hi haurà dues presentacions. A la primera, en acabar el primer període, exposarem el pla de projecte, la motivació i almenys un dels models que anem a utilitzar. A la segona, que es realitzarà en acabar el segon període, presentarem de nou el pla del projecte i realitzarem una demostració del potencial de la ferramenta.

Per a assolir l'objectiu, és necessari que desgranem les distintes fases que componen el projecte i definim les activitats necessàries per a poder desenvolupar-les. Així doncs, distingim fonamentalment quatre fases: una fase d'investigació, en la que aprendrem el màxim possible sobre sistemes recomanadors; una fase de pretractament de les dades, en la que les prepararem per a poder aplicar les fases següents; una fase de modelatge, en la que s'entrenaran i validaran els models encarregats de dur a terme les recomanacions.

A l'interior de cada fase, tal com es pot veure a l'apèndix A.1, trobarem que hi ha distints paquets, com la realització de l'anàlisi descriptiu de les dades, que es desglossen en tasques, com la realització d'un anàlisi univariant sobre les dades, l'aplicació de tècniques de llenguatge natural per a l'obtenció dels embeddings o l'aplicació de tècniques de reducció de variables. Cada una d'aquestes tasques tindrà uns terminis en els que s'haurà de complir. D'aquesta manera podrem definir un diagrama de Gantt (apèndix A.2) que serà clau per a la organització dels temps en el nostre projecte.

CAPÍTOL 4
Anàlisi descriptiu

Les dades que hem emprat en aquest projecte provenen, tal com hem comentat abans, de cursos MOOC en línia desenvolupats per la UPV. Aquests cursos es troben publicats a la web educativa edX, una plataforma digital per a la publicació d'aquest tipus de formacions. Tanmateix va ser la pròpia UPV qui ens va proporcionar les dades en el marc de la Càtedra d'Intel·ligència Artificial Aplicada a la Administració Pública.

D'aquesta manera, vam obtindre tres repositoris distints: un per als cursos en línia, un altre per als usuaris i un altre per a les inscripcions d'usuaris dins de cada curs. Aquest últim conjunt de dades serà el que utilitzarem per a integrar tots els repositoris en una base sòlida que ens ajude a crear el sistema recomanador.

4.1 Anàlisi univariant

Tanmateix, abans de començar a desenvolupar els models de recomanació, és necessari que analitzem les dades des d'un punt de vista tècnic. Així doncs, en els següents apartats es detalla el procés que es va realitzar per estudiar les dades, des d'un punt de vista de qualitat, fins a un anàlisi multivariant, passant per un anàlisi univariant que ens va ajudar a comprendre millor la tipologia de les dades a les quals ens enfrontàvem.

4.1.1. Anàlisi de les dades de cursos

Tal com hem explicat abans, el primer conjunt de dades conté informació referent als cursos per si mateixos. D'aquesta manera, comptem amb un repositori que inclou dades de 500 cursos diferents amb informació des de 2015 fins al 2022. Després d'eliminar redundàncies, vam constatar que hi havia 108 casos amb tretze variables i un identificador. Les variables són les següents: la plataforma de publicació, el títol de la formació, una breu descripció, el mode, la data d'inici, la data de final, les setmanes necessàries per completar el curs, les hores estimades per completar-lo, la llengua en la qual s'imparteix, la nota mínima per aprovar, un marcador intern i l'URL del curs.

La primera variable que hem estudiat és la quantitat de setmanes que s'estima durarà el curs. Aquesta variable, de tipus discret, pren valors de 0 fins a 23. Tanmateix, la seua distribució no és massa normal, ja que podem veure un pic anòmal a l'inici, és a dir, que hi ha molts cursos que duren una setmana o menys, mentre que veiem que hi ha alguns cursos amb valors molt alts, és a dir, que hi haurà comptats cursos que duraran més de vint setmanes. Fora d'això, els cursos es distribueixen d'una manera més o menys normal, tot i que és molt destacable el fet que hi ha una gran quantitat de cursos que es completen en cinc o sis setmanes.

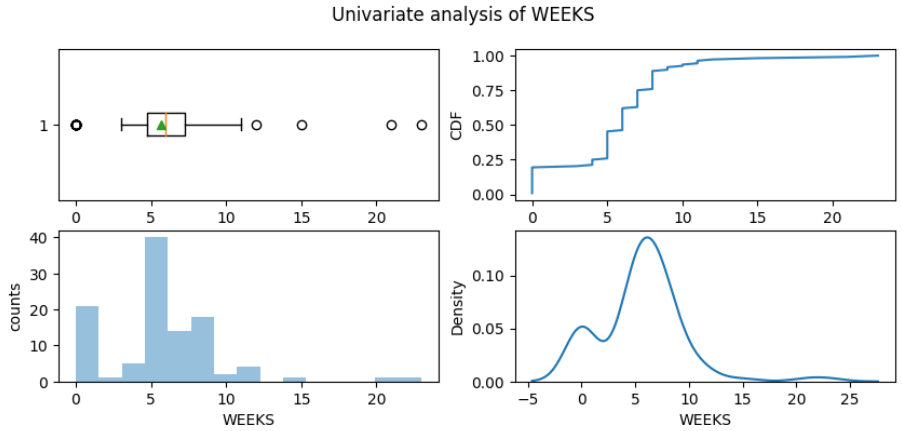

Figura 4.1: anàlisi unviariant sobre les setmanes necessàries per completar el curs.

Però no podríem estudiar la duració del curs tenint en compte únicament la informació de les setmanes. Per aquesta raó estudiarem també la quantitat d'hores que s'estima necessàries per a completar cada formació. En aquest cas veiem que hi ha una distribució molt pareguda a la de setmanes, tot i que en una escala diferent. La distribució d'aquestes dades no segueix una campana de Gauss, sinó que s'observa un alt pic a l'inici, és a dir, una gran part dels cursos requereixen menys de vint hores per a completar-se, mentre a la dreta de la mitjana (de vint-i-cinc hores) trobem alguns valors molt alts que arriben fins a les noranta-huit hores.

Un altre estudi que es pot fer sobre el temps, és el de l'evolució de la quantitat de cursos actius. Per a fer-ho, hem comptabilitzat la suma acumulada dels cursos que ja han començat i li hem restat la suma acumulada dels cursos que han acabat, obtenint així la quantitat de cursos de la UPV actius a edX en cada moment.

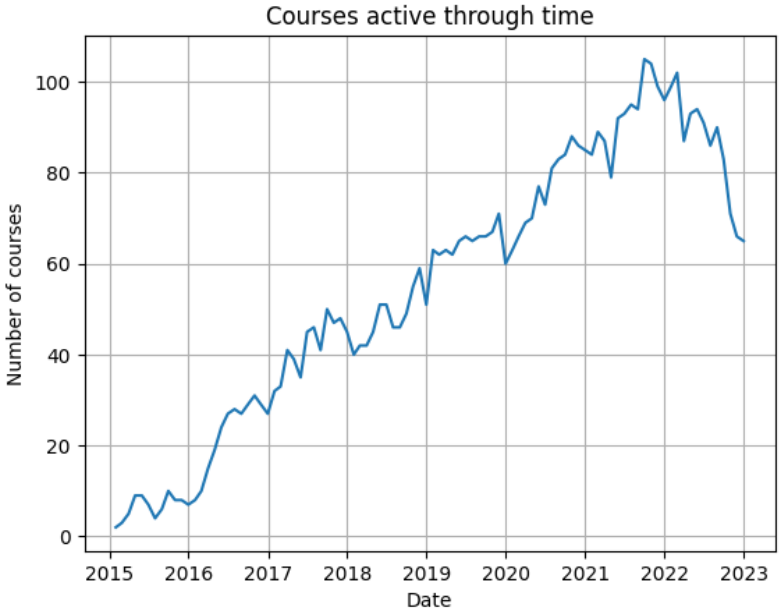

Figura 4.2: anàlisi de la quantitat de cursos actius a través del temps.

La gràfica 4.2 mostra que existeix un augment de la quantitat de cursos oberts fins a l'any 2021. A partir de setembre d'aquest mateix any, el nombre de cursos oberts tendeix a declinar, especialment durant el darrer quadrimestre de 2023.

Respecte a la nota mínima necessària per aprovar el curs, en estudiar-la vam poder comprovar que hi havia un valor summament aberrant, 65535. Com aquest valor era tan estrany vam decidir eliminar-lo. En fer-ho, vam comprovar que la resta de valors podien ser 0.8 o 80. Aquests valors són tan similars perquè són exactament el mateix: una puntuació del 80% o del 0.8. Per aquesta raó vam decidir eliminar aquesta variable que no aportava cap tipus d'informació.

Finalment i quant a la llengua, els cursos s'imparteixen fonamentalment en castellà (setanta-un cursos) tot i que hi ha una minoria de cursos que s'imparteixen en anglés (setze cursos). Dels altres no tenim informació.

Segurament el lector es pregunte per què no hem estudiat les variables "plataforma", "mode"o "marcador intern". Respecte a la primera característica, no la vam tindre en compte perquè tots els valors eren el mateix: edX. Com que era una variable que no aportava informació va ser suprimida. D'altra banda, en la figura 4.3, s'observa com els altres dos atributs no contenen informació. Per tant, també seran eliminades.

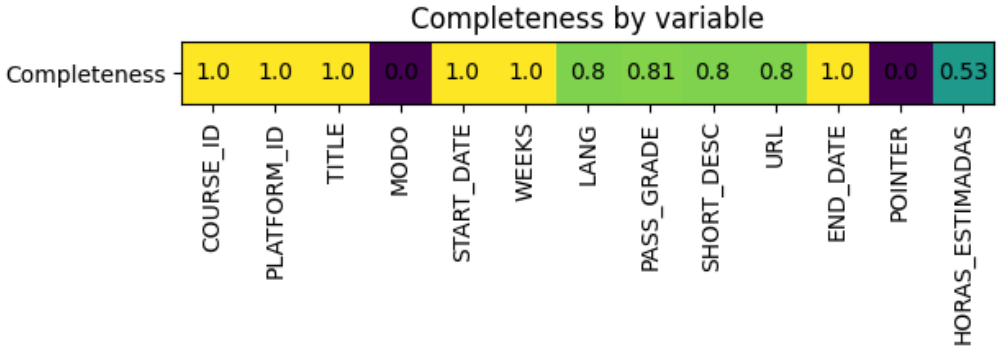

Figura 4.3: anàlisi de completitud sobre les dades dels cursos.

Una altra observació que podem fer és que hi ha pocs valors a la variable que indica les hores estimades per al curs i que les variables *llengua*, *descripció*, *nota mínima* i *URL* tot i tenir més valors, també compten amb alguna dada faltant.

4.1.2. Anàlisi de les dades d'usuari

D'altra banda, i com hem explicat abans, també comptem amb informació referent als usuaris. Aquestes dades han sigut anonimitzades per part del departament de sistemes abans que nosaltres les rebérem. Així doncs, trobem que aquest repositori contenia un total de cinc variables i un identificador. Aquestes variables són totes de caràcter opcional durant la creació de perfil i són: l'any de naixement, el gènere, el nivell d'estudis, el país d'origen i la llengua parlada.

Tanmateix, com que és informació introduïda pel client, trobem nombrosos valors estranys. Especialment, es dónen en variables que l'usuari introdueix de forma lliure, com l'any de naixement, on trobem valors anteriors a l'any 1900 o en idioma, on trobem tota classe de valors, des de llengües inventades fins a llenguatges de programació.

Comencem analitzant les dades de l'edat de naixement. El rang de les dades recollides va de 1893 fins al 2021. És més que evident que els dos extrems són impossibles o, com a mínim, molt poc probables i això ens indica que hi haurà dades incorrectes. La resta de dades es distribueixen donant una cua per la esquerra, és a dir, que la major part dels valors es trobarà al voltant de 1990, però hi haurà valors molt menors, desplaçant la mitja fins al 1987 (tres anys respecte a la mediana).

Quant al gènere, s'observa una majoria d'homes (53.31%) sobre un gran percentatge de dones (46.08%). També cal recalcar la presència de persones amb gènere no definit (0.62%).

Respecte al nivell d'estudis, comprovem que hi ha una majoria d'estudiants que ha completat un grau (b). El segueixen alumnes que han completat el batxillerat (hs) i persones que han acabat un màster (m). També es pot observar una certa quantitat d'usuaris amb un postgrau (a) i cal esmentar als estudiants que han completat només la secundària (Jhs)

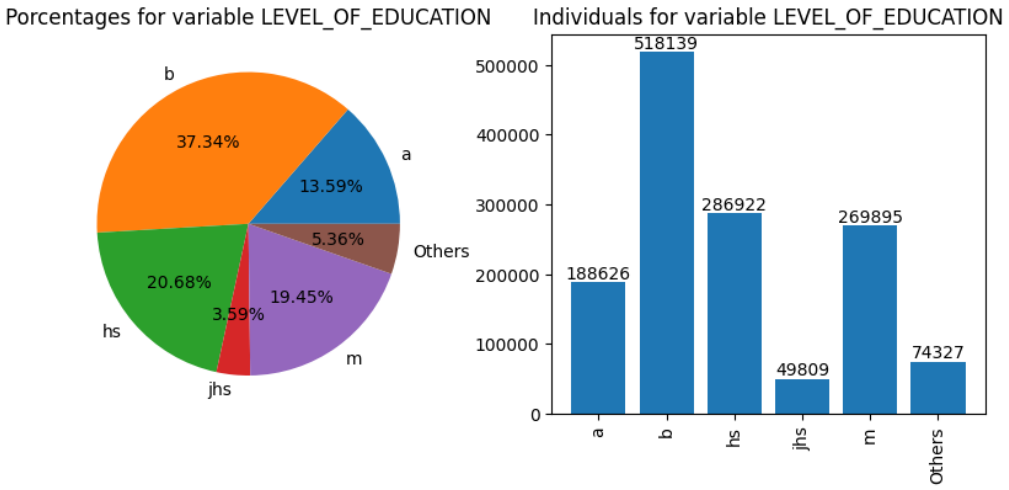

Figura 4.4: anàlisi categòric del nivell d'estudis dels alumnes.

Finalment, si realitzem un anàlisi sobre els idiomes que parlen els alumnes, podem detectar que en la seua gran majoria parlen castellà i/o anglés. Aquestes dues llengües estan molt per damunt de la quantitat de parlants del següent idioma: el portugués. El segueix el francés i molt per darrere, altres llengües com l'italià, el català, l'alemany o el rus.

Tanmateix, aquestes dades s'han d'agafar en cura, ja que, com s'observa a la figura 4.4, hi ha una gran quantitat de dades faltants. En tot cas, com que les variables que hem explicat es poden considerar sensibles i poden generar biaixos de caràcter sexista, per edat o per estudis, seran variables que no inclourem en el sistema recomanador.

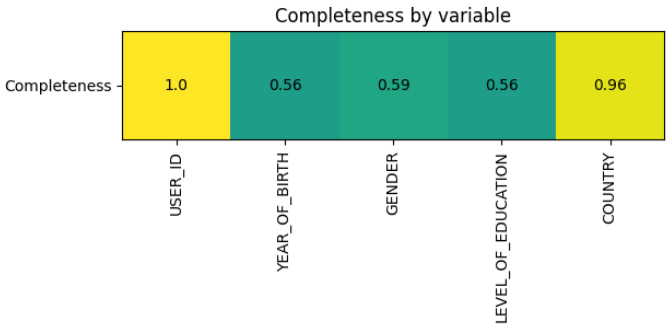

Figura 4.5: anàlisi de completitud sobre les dades dels usuaris.

4.1.3. Anàlisi de les dades d'inscripcions.

Finalment, estudiarem les inscripcions dels usuaris. Tal com hem comentat abans, aquest repositori és el que utilitzarem per a integrar les dades dels usuaris i les dels cursos, però

açò no implica que el repositori no continga informació per si mateix. De fet, el repositori conté dos identificadors i tres variables: la data d'inscripció al curs, si s'ha completat o no i el mode en el qual s'ha donat la inscripció.

Si comprovem la quantitat de cursos que estan encara actius, és a dir, que no s'han completat, observem que són prop del 90% dels casos, mentre que l'altre 10% són els qui si que han acabat la formació. D'aquestes inscripcions, el 88% simplement l'han auditat, és a dir, s'han inscrit. Un 9.29% l'han auditat amb honor o, dit d'altra manera, l'han aprovat, mentre que només un 2.57% han utilitzat càmera i document d'identitat per a identificar-se a l'hora de fer l'examen.

També es pot establir un anàlisi per a comprovar l'evolució de les inscripcions als cursos de la UPV a través dels anys. D'aquesta manera establim un eix cronològic per a veure quants individus s'inscriuen a cursos cada any.

De tota manera aquest anàlisi no és realista, ja que la quantitat de cursos disponibles també varia a través dels anys. Observem, doncs, com existeix una tendència ascendent fins al 2020, moment en el qual el nombre d'inscripcions augmenten increïblement degut a la pandèmia de la SARS COVID-19. Tanmateix, a partir de la meitat d'aquest mateix any, el nombre d'inscripcions tendeixen a baixar fins al 2022.

De tota manera aquestes dades no són realistes, ja que, com hem vist abans, el nombre de cursos varia amb el temps. Per tant, és convenient realitzar una ponderació

Això no obstant, el més interessant ací és estudiar les dades integrades i com afecten unes variables a altres. És a dir, fer un anàlisi multivariant. Per a això, el primer que farem, serà comparar quants usuaris hi ha en cada curs per tal de veure si les dades estan balancejades o no.

Tal com s'observa a la figura 4.6, els cursos amb més participació són fonamentalment els de llengües (anglés i castellà) i els de ferramentes de productivitat (office, word, etc). Destaquen sobretot els cursos introductoris al castellà i a l'excel, que en aquest ordre són els més populars amb molta diferència. D'altra banda, els cursos menys populars són aquells més enfocats a l'apartat tècnic: ciències de materials, enginyeria d'aliments, Ciència de dades...

A la inversa, és a dir, veient la quantitat de cursos que consumeix cada usuari, la distribució és també extremadament asimètrica, sent així que la majoria dels usuaris s'hauran consumit a un curs i la quantitat d'usuaris anirà reduint-se si anem augmentant la quantitat de cursos que han consumit.

4.2 Anàlisi multivariant

Quan parlem d'anàlisi multivariant, cal tindre en compte que les dades dels usuaris no són gaire fiables, tal com hem comentat a la secció anterior. Açò pot provocar que les conclusions extretes d'aquest conjunt d'informació no siguen correctes. Per aquesta raó intentarem evitar les dades d'usuari i considerarem un altre tipus de dades.

Tal com veurem al capítol sisé, un expert va realitzar una classificació per als cursos. Tenint aquesta informació en ment, ens disposem a estudiar si hi ha alguna categoria de cursos que requerisca un major temps.

Així doncs, per a estudiar si el temps fins a completar els cursos i les categories estan relacionats, provarem de desenvolupar un test d'ANOVA [18]. Abans de continuar, cal que expliquem que moltes categories tenien menys de tres cursos. Com que per a desenvolupar un test adequat se'n necessiten més de tres, no considerarem aquestes for-

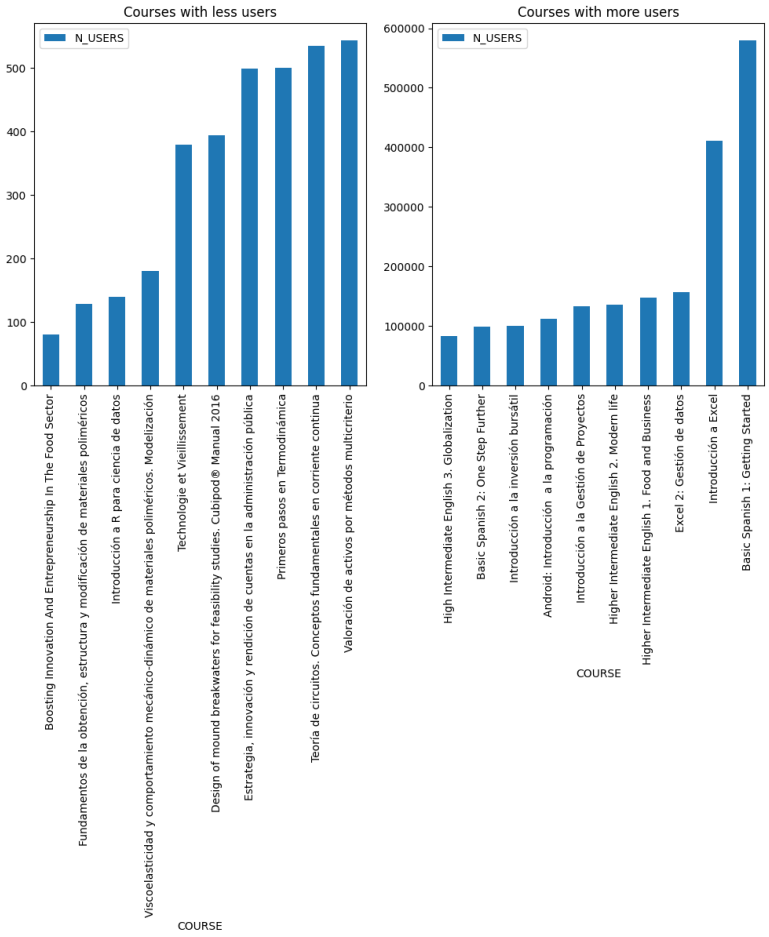

Figura 4.6: quantitat d'usuaris per curs.

macions. Per tant, vam acabar tenint tretze categories, de les quals vam eliminar aquelles amb menys de cinc cursos, de manera que ens van restar solament set.

Si observem la figura 4.7, podem comprovar que no hi ha normalitat en molts casos i tampoc no hi ha homoscedasticitat. En conseqüència, és impossible elaborar un ANOVA vàlid. Tanmateix, sí que podem aplicar un test de Kruskal [19], que es basa en la mediana, una mètrica molt més robusta que no el promig, que empra el test ANOVA.

Així, una vegada aplicat, obtenim un valor de l'estadístic de 9.42, superior a 1.96, per la qual cosa podem dir que no es pot confirmar que hi haja diferències significatives entre les medianes. Tanmateix, el fet que hi haja heteroscedasticitat sí que és molt significatiu, ja que significa que hi ha algunes categories amb cursos que varien molt en temps de realització, com els cursos d'idiomes, i altres, com els de matemàtiques, que estan molt acotats.

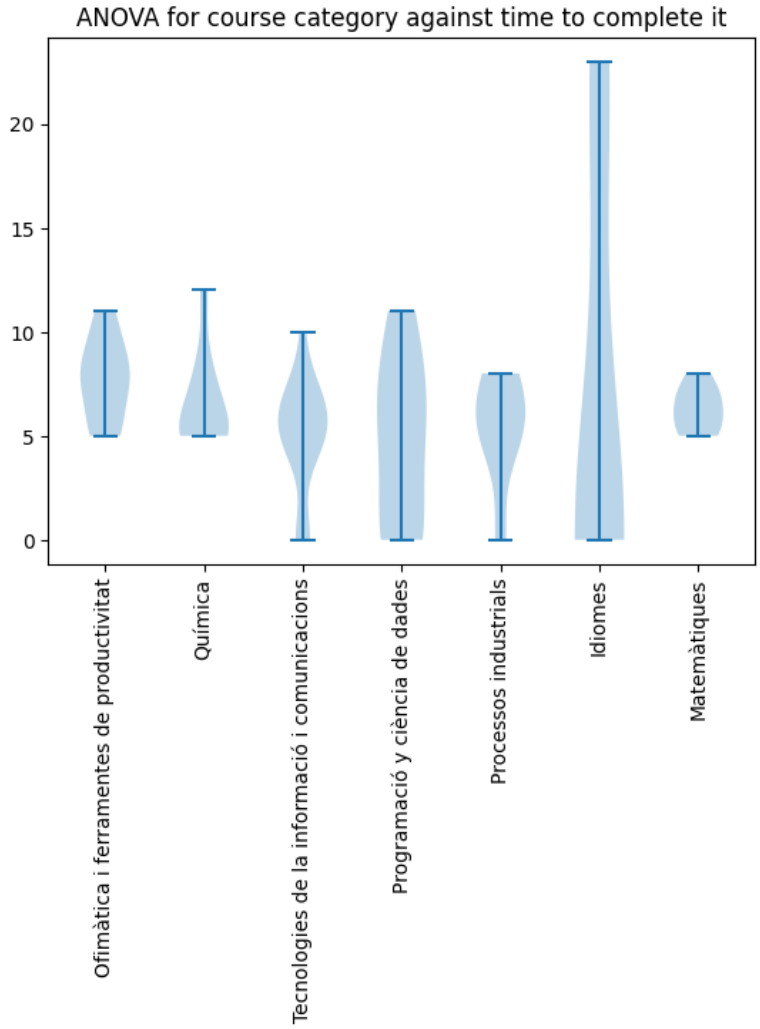

Figura 4.7: gràfic de violí per a cada categoria de més de 5 cursos.

CAPÍTOL 5
Metodologia

Ara que ja hem parlat una mica de les dades i hem observat les seues distribucions i relacions entre les diferents varibles, podem començar a crear un model per a realitzar la generació de candidats.

5.1 Model de filtre de contingut

El primer model que emprarem serà el de generació de candidats per filtre de contingut. Tal com hem explicat al segon capítol, la idea serà recomanar a l'usuari ítems similars als que ja ha consumit.

El primer pas per a desenvolupar aquesta ferramenta serà l'extracció de característiques. Com ja hem vist abans, la informació que defineix els cursos no és suficient per a elaborar un model adequat. Per aquesta raó utilitzarem alguna tècnica de processament de llenguatge natural per a facilitar l'obtenció de la informació.

Però per a això primer hem de realitzar un preprocessat del text que, en el nostre cas serà la concatenació del títol i la descripció del curs. Cal que diguem que aquest text no estava sempre en el mateix idioma. Hem trobat que alguns cursos tenien el text escrit en castellà, altres el tenien redactat en anglés i d'altres el tenien en francés. Davant aquest problema teníem dos camins a seguir: podíem desenvolupar tres models embeddings distints (un per a cada llengua) o podíem desenvolupar un model conjunt traduint els textos a l'idioma més freqüent.

El problema de crear diferents models per a vectoritzar el text és que no es manté un mateix context vectorial. És a dir, tot i que hi haja dos cursos amb textos amb un significat molt semblant, és probable que, com que estan escrits en llengües diferents, si un estudiant consumeix un curs, el model no li recomane l'altre.

A causa d'açò i perquè realment teníem pocs cursos (108 en total), vam decidir que era una bona idea traduir el text. Per això vam emprar una llibreria pròpia de Google [20] que ens permetia convertir el text a l'idioma que preferirem. Com els textos en castellà semblaven ser els predominants, vam decidir que utilitzaríem aquesta com a llengua vehicular per als textos dels cursos.

D'aquesta manera i després d'haver convertit el text a una única llengua, podíem començar a processar el text fent ús de la llibreria NLTK [22], especialitzada en el processament de textos. El primer procés que vam aplicar va ser la tokenització [21], és a dir, convertir cada paraula en una unitat que després utilitzaríem per a realitzar la vectorització. Seguidament, vam convertir aquests tokens a minúscules, vam eliminar signes d'accentuació i etiquetes al text per a mantindre una coherència entre paraules iguals i

vam eliminar les stopwords, és a dir, aquelles paraules que tenen poc valor dins el text, com determinants, preposicions o conjuncions. Després vam lematitzar els tokens [21], és a dir en vam extraure les arrels per a eliminar part paraules prescindibles del corpus. Dit d'altra manera, com que menjava i menjaves són paraules amb un significant molt similar, però en vectoritzar-les no quedarà igual, les lematitzarem de tal manera que l'arrel menja- serà igual per a totes dues.

Després d'efectuar aquest pretractament, era hora d'aplicar alguna tècnica de vectorització sobre el text. Seguint la literatura, la idea més òbvia era emprar TF-IDF [23]. Aquesta tècnica aplica a cada paraula del text un pes de la manera següent:

$$TF - IDF_{t,d} = TF_{t,d} \cdot IDF_t = \frac{freq(t,d)}{max\{freq(t,d) : t \in d\}} \cdot \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

on t és la paraula i d és el document. És a dir, es calcula la relació entre la freqüència de la paraula al document i la freqüència màxima d'aquesta paraula a un document del corpus, i es multiplica pel logaritme de la relació entre el nombre de documents i el nombre de documents en els quals apareix la paraula. D'aquesta manera, com més aparega una paraula al document i menys aparega a la resta de documents, major serà el valor que prenga el seu pes. Així doncs, aquesta paraula pot passar a considerar-se com un dels tòpics del text. Aquesta tècnica ens és molt útil en el nostre cas, ja que ens permet observar quins temes seran els més identificatius de cada curs. Per a evitar inserir al vocabulari paraules sense cap pes, eliminarem aquelles paraules que no tenen en cap document una freqüència d'aparició al text més de l'1%.

El problema que pot tindre aquesta tècnica és que confonga paraules. Per exemple, podem tindre un curs que siga de llenguatges de programació i un altre que explique la gramàtica del llenguatge en castellà. El model no sabrà distingir entre els dos l·llenguatgesï això provocarà confusions i errors. Per solucionar aquest inconvenient, es poden utilitzar els n-grames. És a dir, en compte de fer servir els tokens per a calcular el TF-IDF, es poden considerar un conjunt de n tokens escrits l'un darrere de l'altre. En el nostre cas, vam emprar bigrames (dos tokens) i trigrames (tres tokens) a més dels tokens únics.

Tanmateix, poden continuar existint problemes de confusió entre paraules, per la qual cosa, podem fer ús d'una altra tècnica de word embedding, que capten la intencionalitat del text i solen emprar-se en anàlisi de sentiments.

En el nostre cas, vam usar l'algoritme word2vec [24] de la llibreria Gensim [25]. Per a triar la grandària dels embeddings, vam seguir els passos indicats per K. Patel i P. Bhattacharyya a l'article *Towards lower bounds on number of dimensions for word embeddings* [26]. És a dir, primer vam crear una matriu amb les concurrències de les paraules. Seguidament, vam obtindre una matriu de cosinus utilitzant els valors de les concurrències de cada curs. És a dir, per a cada línia de la matriu de concurrències, vam aplicar la similitud de cosinus amb les altres línies, assolint així una matriu quadrada simètrica. Després vam usar aquesta matriu per a genererar un graf [27] i vam aconseguir el seu màxim clique, és a dir, el seu màxim subgraf complet. Una vegada obtés, K. Pattel i P. Bhattacharyya afirmen que le seu nombre de nodes serà el màxim nombre de parelles de punts equidistants, que en el nostre cas són vint-i-dos. Finalment, vam fer servir la taula creada per A. Barg i W. Yu a *New bounds for equiangular lines* [28], obtenint que el nombre mínim de dimensions dels vectors havia de ser tretze.

Una vegada determinat el nombre de dimensions per a cada vector, vam aplicar el model d'embedding a les paraules. Però, tanmateix, això ens deixava un framework asimètric, car no tots els títols tenien el mateix nombre de paraules. Per això vam emprar

la tècnica de padding per afegir 0 a l'esquerra, de manera que ens va quedar una mida de vector per curs de cent dèsset dimensions, ja que el nombre màxim de paraules a un títol era de nou.

Amb això, vam aconseguir un total de sis-centes setanta-huit variables, una xifra fins i tot una mica excessiva i que després caldria reduir (com veurem més endavant). Així i tot, amb això no havíem encara acabat de gestionar la extracció de característiques. I és que com que compararem vectors, és convenient no tindre valors massa distints en els atributs de cada observació, ja que aquestes diferències poden causar que una variable siga més important per al model que una altra. Com que totes les variables han sigut extretes de la mateixa manera, no té massa sentit que existisca un biaix en l'ús d'aquestes. Per aquesta raó, s'estandarditzarà cada una de les variables. És a dir, cada variable i $\vec{v}_i$ que serà el vector columna de $V \in \mathbb{R}^{mxd}$ es definirà com a:

$$\vec{v}_i' = \frac{\vec{v}_i - \overline{\vec{v}_i}}{\sigma_{\vec{v}_i}}$$

Una vegada ja teníem els vectors adequats per a cada curs, era hora de passar al modelatge com a tal. Per a fer-lo vam decidir que, tal com s'indica a la literatura, s'integraran les preferències dels usuaris segons la combinació linial de valors als cursos que han consumit. És a dir, considerant que $V \in \mathbb{R}^{nxd}$ és la matriu de vectors per a cada curs, on n és el nombre de cursos i d el nombre de variables i que $\vec{u}_i \in \mathbb{R}^n$ el vector de decisió de l'usuari i on $u_{i,j}$ és un nombre binari que pren el valor 1 quan l'usuari ha consumit el curs i un valor 0 quan l'usuari no l'ha consumit, es calcula el perfil de gustos de cada usuari com a:

$$\vec{p}_i = \vec{u}_i \cdot V$$

Una vegada generat aquest perfil, és hora d'aplicar el modelatge per a obtindre els candidats. Anem a buscar els cursos que l'usuari no haja consumit i tinguen major similitud amb el vector perfil de l'usuari. D'aquesta manera, el que haurem d'optimitzar en aquest model (i que farem a la validació) serà la funció de similitud emprada. Com que s'espera mostrar un total de cinc cursos a la pàgina web, hem decidit que el tamany de la mostra de candidats serà de 15 cursos, que s'organitzaran segons l'ordre invers de popularitat.

5.2 Models de filtre col·laboratiu

Quant als models de filtre col·laboratiu cal que recordem el volum de les dades. I és que tenim un total de cent huit cursos i dos milions i mig d'usuaris. Per tant, caldrà aprendre una matriu $V \in \mathbb{R}^{108xd}$ i una matriu $U \in \mathbb{R}^{2481987xd}$. La primera idea que vam tindre va ser la d'utilitzar les dades de cursos de tal manera que d fóra la quantitat de variables de cursos. Així doncs, solament caldria aprendre la matriu dels usuaris, una matriu de format 2481987x679. Aquesta matriu és, evidentment, inassumible de calcular per a nosaltres, ja que no tenim capacitat de computació suficient. Per això cal reduir el tamany d'aquesta matriu. Existeixen dues maneres de fer-ho: reduir el nombre d'usuaris o reduir el nombre de variables de cursos.

Per reduir el nombre d'usuaris, el primer que podem fer és eliminar aquells usuaris que són redundants. És a dir, agrupar els usuaris que han consumit els mateixos cursos com a un sol usuari. D'aquesta manera podem reduir el volum d'usuaris a només huitanta mil perfils.

D'altra banda, també podem reduir el nombre de variables dels cursos. Açò tindria sentit, ja que si s'observa la matriu de correlació (que no posarem ací a causa del seu gran tamany i la seua baixa interpretabilitat), hi ha un total de quatre mil casos on la correlació entre dues variables és superior al 70%. Així doncs, podem aplicar nombroses tècniques, com la descomposició en valors singulars, la factorització per variables no negatives o l'anàlisi de components principals.

D'aquesta manera, vam aplicar primerament la tècnica de descomposició en valors singulars utilitzant dos algoritmes (arpack i randomized) i valorant una quantitat de components des de dues fins a seixanta. Els resultats de l'anàlisi van demostrar que amb l'algoritme arpack i un total de 60 components, s'explicava un 89.65% de la variança.

Per aplicar la factorització per matrius no-negatives, primer necessitàvem tindre tota la matriu dels cursos de valors positius. Per tant, vam sumar-li a tota la matriu el seu mínim valor, de tal manera que vam obtindre una matriu en què el valor mínim seria 0. Després vam aplicar la factorització per matrius no-negatives i vam obtindre que amb seixanta components, s'obtenia un error de reconstrucció de 90.52.

Finalment, la darrera tècnica que vam aplicar va ser la descomposició en components principals [29]. D'aquesta forma vam avaluar la tècnica emprant diferents algoritmes (automàtic, matriu de covariança completa, arpack i randomized) i diferent nombre de components principals (des de dues fins a seixanta). Els resultats que vam obtindre van ser que amb 60 components i l'algoritme arpack s'obtenia el mínim error, explicant el 89.65% de la variança.

Així doncs, vam decidir, després de veure aquests resultats, aplicar una descomposició per valors singulars. D'aquesta manera, el tamany de la matriu a predir seria de $U \in \mathbb{R}^{16225x60}$, uns valors molt més assequibles.

D'aquesta manera caldrà optimitzar la funció següent:

$$Z : \sum_{i=1}^{n}(\sum_{j=1}^{m}(A_{i,j} \cdot (A_{i,j} - < \vec{u}_i, \vec{v}_j^T >)^2)) + w_0 \cdot \sum_{i=1}^{n}(\sum_{j=1}^{m}((< \vec{u}_i, \vec{v}_j^T >)^2))$$

$$Z : \sum_{i=1}^{n}(\sum_{j=1}^{m}(A_{i,j} \cdot (A_{i,j} - \sum_{k=1}^{d} u_{i,k} \cdot v_{j,k})^2)) + w_0 \cdot \sum_{i=1}^{n}(\sum_{j=1}^{m}((\sum_{k=1}^{d} u_{i,k} \cdot v_{j,k})^2))$$

Per a això emprarem l'algoritme Adam [30]. Aquest algoritme consisteix a creuar les millors característiques de l'algoritme de descens per gradient amb momentum i el RMSProp. Però primer, haurem de descobrir com podem calcular la derivada parcial per a cada paràmetre $u_{i,k}$. D'aquesta manera, podem comprendre que la funció Z es pot calcular com un sumatori de les funcions f i g que definim a continuació.

$$Z = f + g$$

$$f : \sum_{i=1}^{n}(\sum_{j=1}^{m}(A_{i,j} \cdot (A_{i,j} - \sum_{k=1}^{d} u_{i,k} \cdot v_{j,k})^2))$$

$$g : w_0 \cdot \sum_{i=1}^{n}(\sum_{j=1}^{m}((\sum_{k=1}^{d} u_{i,k} \cdot v_{j,k})^2))$$

Així doncs, podem definir la derivada de Z com el sumatori de les derivades de f i de g. Consegüentment, obtenim que la derivada de f seria la que segueix:

$$\frac{\partial f}{\partial u_{i,k}} = \sum_{j=1}^{m} A_{i,j} \cdot \frac{\partial((A_{i,j} - \sum_{k=1}^{d} u_{i,k} \cdot v_{j,k})^2)}{\partial u_{i,j}}$$

$$\frac{\partial f}{\partial u_{i,k}} = \sum_{j=1}^{m} A_{i,j} \cdot 2 \cdot (A_{i,j} - \sum_{h=1}^{d} u_{i,h} \cdot v_{j,h}) \cdot (-v_{j,k})$$

D'altra banda es pot definir la derivada de g com a:

$$\frac{\partial g}{\partial u_{i,k}} = w_0 \cdot \frac{\partial \sum_{i=1}^{n}(\sum_{j=1}^{m}((\sum_{k=1}^{d} u_{i,k} \cdot v_{j,k})^2))}{\partial u_{i,k}}$$

$$\frac{\partial g}{\partial u_{i,k}} = w_0 \cdot \sum_{j=1}^{m} \frac{\partial(\sum_{k=1}^{d}(u_{i,k} \cdot v_{j,k}))^2}{\partial u_{i,k}}$$

$$\frac{\partial g}{\partial u_{i,k}} = w_0 \cdot \sum_{j=1}^{m}(2 \cdot \sum_{h=1}^{d}(u_{i,h} \cdot v_{j,h}) \cdot v_{j,k})$$

D'aquesta manera obtenim que la derivada parcial de Z seria de la següent forma

$$\frac{\partial Z}{\partial u_{i,k}} = \sum_{j=1}^{m}(A_{i,j} \cdot 2 \cdot (A_{i,j} - \sum_{h=1}^{d} u_{i,h} \cdot v_{j,h}) \cdot (-v_{j,k})) + w_0 \cdot \sum_{j=1}^{m}(2 \cdot \sum_{h=1}^{d}(u_{i,h} \cdot v_{j,h}) \cdot v_{j,k})$$

$$\frac{\partial Z}{\partial u_{i,k}} = \sum_{j=1}^{m}(A_{i,j} \cdot 2 \cdot (A_{i,j} - < \vec{u}_i, \vec{v}_j^T >) \cdot (-v_{j,k}) + w_0 \cdot 2 \cdot < \vec{u}_i, \vec{v}j^T > \cdot v_{j,k})$$

Una vegada hem comprobat quina és la funció gradient, apliquem l'algoritme ADAM. Aquest algoritme aplica el següent:

$$W_{i,k}^{(t)} = \beta W_{i,k}^{(t-1)} + (1-\beta)\nabla Z(u_{i,k}^{(t)})$$

$$corr(W_{i,k}^{(t)}) = \frac{W_{i,k}^{(t)}}{1-\beta^t}$$

$$S_{i,k}^{(t)} = \rho S_{i,k}^{(t-1)} + (1-\rho)(\nabla Z(u_{i,k}^{(t)}) \cdot \nabla Z(u_{i,k}^{t}))$$

$$corr(S_{i,k}^{(t)}) = \frac{S_{i,k}^{(t)}}{1-\rho^t}$$

$$u_{i,k}^{(t+1)} = u_{i,k}^{(t)} - \alpha \frac{corr(W_{i,k}^{(t)})}{\epsilon + \sqrt{corr(S_{i,k}^{(t)})}}$$

on u_t és el valor predit per a la matriu u en la iteració t, $W_{i,k}$ i $S_{i,k}$ seran valors auxiliars, mentre que α, β i ρ són hiperparàmetres que s'optimitzaran per mitjà d'una búsqueda per graella.

La matriu U resultant serà la que utilitzem per a trobar similituds. És a dir, cercarem les distàncies entre el vector $\vec{u}_i$ i els cursos de la matriu $V \in \mathbb{R}^{mxd}$ i ens quedarem amb la mínima.

5.3 Model de recomanació d'usuaris

Finalment, el darrer model que vam decidir implementar va ser un model de cerca per usuaris. Comprenem que dins del sistema, encara que no els puguem identificar per mitjà de les seues característiques pròpies, existeixen usuaris semblants amb gustos similars. Per aquesta raó vam decidir emprar un model que trobara aquestes semblances entre usuaris, de manera que es mostraren a l'usuari els cursos que els perfils més similars hagueren consumit.

Ara la qüestió radica en quines dades d'usuari utilitzar. D'una banda, podem fer servir les dades d'inscripció de cada usuari. L'altra opció era usar els perfils extrets en el model de filtre de contingut o en el model de filtre col·laboratiu.

Així doncs, vam elaborar un algoritme que, fent ús dels perfils dels usuaris, fóra capaç d'extraure un rànking de cursos segons la similitud entre usuaris.

```
def users_recommendations(profiles:dict, user_id:str)->dict:
    result = dict()
    for other in profiles:
        if other == user
            continue
        sim = similarity(user, other)
        if sim == None:
            continue
        else:
            for item in inscriptions[other]:
                if item not in inscriptions[user]:
                    result[item] = max([sim, result[item]])
    return result
```

Ara tan sols cal comprovar amb quin repositori de dades funciona millor: amb dades d'inscripcions, amb el perfil de filtre de contingut o amb el perfil de filtre col·laboratiu. I observar quina és la mesura de distància més adequada per a solucionar el problema

5.4 Models sense personalització

Hem comentat molt sobre els models, però la realitat ens enfrontem amb un greu problema quan no tenim dades històriques, és a dir, quan un usuari nou arriba al sistema. Per solucionar-lo, hem ideat dues estratègies: una per als usuaris que ingressen al sistema i altra per als usuaris que simplement estan de pas per la pàgina.

La primera aproximació es basa en els **models basats en el diàleg**. Tal com hem comentat al capítol tercer, M.Zhang et al. [16] plantegen un precedent foça interessant en el que introdueixen al model informació del context de l'usuari. Després el seu algoritme busca emprar aquesta informació addicional en el seu benefici. La manera d'introduir aquesta informació pot ser, tranquil·lament sol·licitant-la als clients. D'aquesta manera, es pot establir un diàleg en el qual l'estudiant participe activament en la creació del seu perfil.

En un inici, vam pensar que una bona idea seria aplicar clústering sobre les dades de cursos i preguntar a l'usuari quins serien els seus cursos preferits. D'aquesta manera, podríem aplicar tots els models que hem vist fins ara.

És a dir, si considerem $\vec{e} \in \mathbb{R}^h$ el vector de decisió on e_k prendrà valor 1 si l'estudiant té interés en les matèries del clúster, podem aplicar el filtre de contingut:

$$\vec{p} = \vec{e} \cdot C$$

on $C \in \mathbb{R}^{hxd}$ és la matriu de clústers, de tal manera $c_{i,j}$ representa el valor de la característica j del centre o centroide del clúster i el vector $\vec{p}_i$ representa el perfil generat per a l'usuari i. Amb aquest vector ja es podria elaborar un anàlisi de similitud amb els vectors donats a cada curs.

La segona aproximació és donada per a aquells usuaris que encara no s'han registrat a la pàgina. En aquests casos serà necessari aplicar un model no tan eficient quant a personalització. Proposem, per tant, tres models: un de popularitat, un aleatori i un per categories.

En el primer cas, es tractaria d'un model que cauria en un dels problemes dels quals hem parlat al llarg de tota la memòria: els biaixos de popularitat. Tanmateix, no ens preocupa aquest problema, ja que, com hem explicat a la introducció, la idea d'emprar diversos models és que siguen capaços de suplir les seues carències entre ells. D'aquesta manera, l'objectiu fonamental del model de filtre de popularitat seria atreure l'atenció dels nos usuaris. I és que s'espera que els models més populars siguen els més atractius, ja que ho han sigut fins ara. Per tant, és una bona idea fer-los servir per a recomanar a nou-vinguts.

D'altra banda, el paper del model de filtre aleatori, seria mostrar altres cursos potser no tan populars. Per tant, exclourem aquelles formacions ja recomanades al model de filtre de popularitat i extraurem aleatòriament un conjunt de cursos dels restants. D'aquesta manera, tindrem un seguit de cursos que potser podrien ser interessants per a l'usuari i que no cauran en el biaix de popularitat.

Finalment, el darrer model busca la interacció de l'usuari. Així doncs, establirà un filtre de cerca per a poder establir una comunicació amb l'usuari, de tal manera que podrà buscar d'entre les categories dels cursos per a trobar el que més li interesse.

CAPÍTOL 6
Experimentació

En el següent apartat parlarem tant de l'avaluació dels models, com del procés realitzat per aquells estudiants nou-vinguts i els resultats que hem obtés per als models.

6.1 Avaluació dels models

Per a avaluar els models hem emprat una tècnica de pseudovalidació creuada. Aquesta tècnica consisteix en observar els usuaris que han consumit més de deu cursos i aïllar aleatòriament un dels seus cursos.

Amb aquesta mostra d'usuaris que han consumit més de deu cursos (eliminant el curs aïllat), s'entrenarà el model de generació de candidats i s'observarà si dins d'aquests candidats es troba la formació que havíem aïllat primerament. Finalment, calcularem la precisió del model com a la relació entre la quantitat de vegades que el curs aïllat es troba al conjunt de candidats entre la quantitat de vegades que s'executa el model de generació de candidats.

6.1.1. Model de filtre de contingut

Tal com hem comentat al capítol cinqué, a la validació del model de generació de candidats per filtre de contingut es busca capturar les diferències a l'hora de recomanar cursos emprant distintes mètriques per a trobar similituds entre el vector de perfil creat per a l'usuari i el vector de curs.

Amb aquest motiu, hem avaluat el model fent ús de quatre mètriques de similitud distintes: la distància euclidiana, la distància de Manhattan, la distància al cosinus i la distància de Pearson.

La distància de Manhattan, també anomenada distància taxi, s'anomena així perquè està basada en la forma geomètrica en la qual solen desplaçar-se els vehicles per una gran ciutat. Es calcula com a la suma de les longituds de les projeccions del vector diferència als eixos de coordenades (figura 6.1). Més formalment es pot definir com a:

$$d_n(\vec{u}, \vec{v}) = ||\vec{u} - \vec{v}||_n = \sum_{i=1}^{n}(|u_i - v_i|)$$

on $\vec{u}$ i $\vec{v}$ són els vectors que es volen comparar.

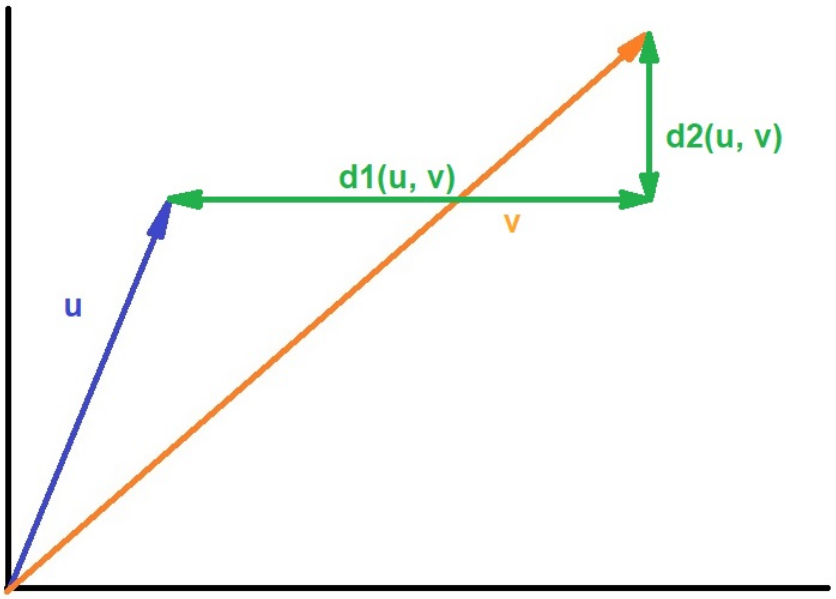

Figura 6.1: càlcul gràfic de la distància de Manhattan.

D'altre costat, la distància euclídea es calcula mitjançant el teorema de pitàgores. És, per tant, la norma del vector diferència i es calcula com s'observa a la figura 6.2 o de forma matemàtica:

$$d_n(\vec{u}, vecv) = ||\vec{u} - \vec{v}||_n = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$$

on $\vec{u}$ i $\vec{v}$ són els vectors que es volen comparar.

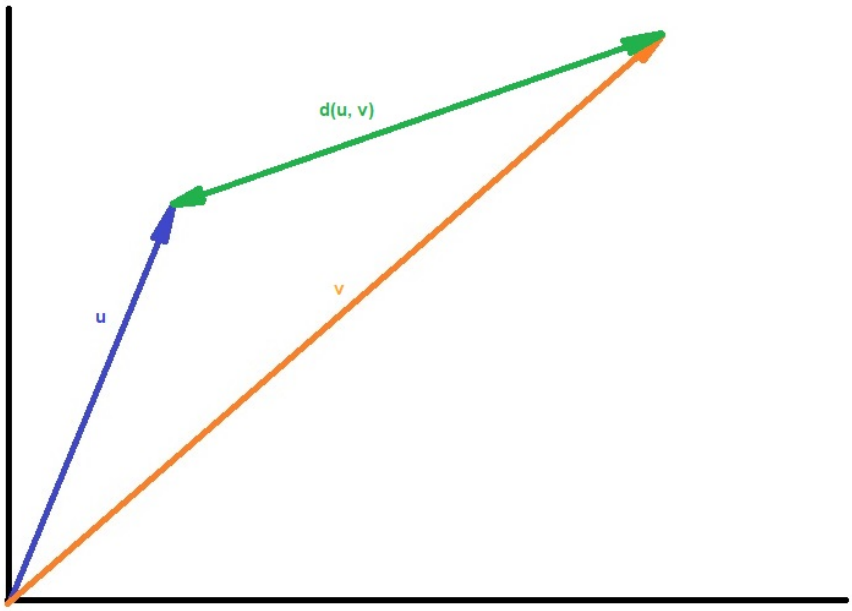

Figura 6.2: càlcul gràfic de la distància euclidiana.

Per calcular les dues similituds (la euclidiana i la de manhattan), s'ha d'invertir aquest valor. La similitud del cosinus, en canvi, no està relacionada amb la diferència entre els vectors, sinó a l'angle els vectors. Aixi doncs, si considerem la definició de producte escalar de dos vectors, podem calcular el cosinus de l'angle com s'observa a la figura 6.3, o de la següent manera:

$$< \vec{u}, \vec{v} >= ||\vec{u}|| \cdot ||\vec{v}|| \cdot \cos\sigma$$

$$\cos\sigma = \frac{< \vec{u}, \vec{v} >}{||\vec{u}|| \cdot ||\vec{v}||}$$

on $\vec{u}$ i $\vec{v}$ són els vectors que es volen comparar.

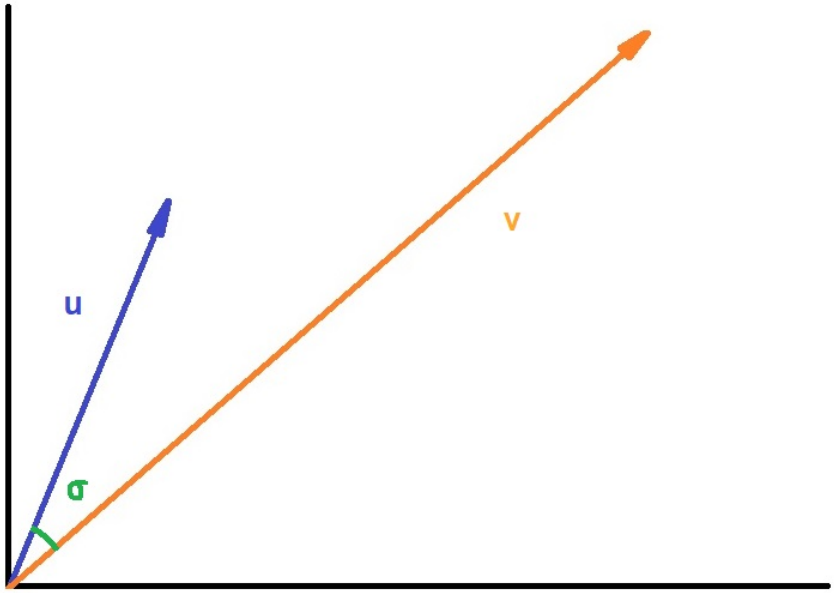

Figura 6.3: càlcul gràfic de la distància al cosinus.

Aquesta similitud té l'avantatge que tampoc no té en compte la norma del vector, com sí que fan les altres dues mesures. Una altra mètrica que no té en compte la norma dels vectors és la distància de Pearson, que es basa en el coeficient de Pearson. Així doncs, es pot calcular com a :

$$d(\vec{u}, \vec{v}) = 1 - r = 1 - \frac{cov(\vec{u}, \vec{v})}{\sigma_{\vec{u}} \cdot \sigma_{\vec{v}}}$$

De manera que la similitud de Pearson serà:

$$sim(\vec{u}, \vec{v}) = -(1 - r) = \frac{cov(\vec{u}, \vec{v})}{\sigma_{\vec{u}} \cdot \sigma_{\vec{v}}} - 1$$

Els resultats que vam obtindre en la validació eren significativament millors quan empràvem mètriques que no tenien en compte la norma del vector. Açò pot ser, segurament, resultat del qual hem explicat al capítol segon. És a dir, que el domini del sistema demana que es recomanen ítems més ajustats a les preferències i no els ítems més populars. Així doncs, per a la distància de Manhattan vam obtindre una precisió del 41.61%, mentre que per a la distància euclidiana vam obtindre una precisió del 42.29%. D'altra banda,

els models obtinguts fent servir la distància de Pearson i la similitud del cosinus tenien una precisió del 45.78% i del 45.89% respectivament, sent així aquesta última la millor distància de les estudiades. En qualsevol cas, el model resulta molt més eficient que una distribució aleatòria, per la qual obtindríem una precisió de només un 13.89%.

6.1.2. Model de filtre col·laboratiu

D'altra banda, per a avaluar el model de filtre de contingut, calia tindre en compte diversos factors per optimitzar la funció de perfilat. En primer lloc, calia optimitzar el paràmetre w_0, ja que aquest té una funció fonamental dins de la funció objectiu. En segon lloc, cal optimitzar els paràmetres α, β i ρ propis de l'algoritme ADAM que utilitzem per optimitzar la funció objectiu.

A més a més, també cal tindre en compte diferents tipus de similitud a l'hora de comparar els resultats. En aquest cas hem emprat les mateixes distàncies que en el filtre de contingut. És a dir, la similitud al cosinus, la similitud euclidiana, la de Manhattan i la similitud de Pearson.

Tot seguit, hem aplicat una optimització per graella amb distints valors per a cada paràmetre i tenint en compte les distintes distàncies. El nombre d'iteracions que hem aplicat en tot cas serà de 100, ja que el cost computacional és massa elevat per als escassos recursos amb els quals comptem.

Els resultats obtesos són que la major precisió al model es dóna quan l'hiperparàmetre α és equivalent a 0.1, quan β és igual a 0.6 i ρ és 0.8. En aquest cas, i tenint en compte que el paràmetre w_0 ha de ser 0.9, s'obté una precisió del 50% amb la distància de Pearson. Aquest valor, que és molt superior al model aleatori (13.89%), és també major al del model de filtre de contingut (45.89%), per la qual cosa, podríem argumentar que serà major. Tanmateix, és necessari que esmentem que aquesta precisió podria augmentar si tinguérem més iteracions, ja que en cap cas el model convergeix.

6.1.3. Model de recomanacions d'usuari

Per avaluar el model de recomanacions d'usuaris vam emprar diverses aproximacions. Tal com hem comentat en el capítol anterior, la nostra idea era utilitzar tant la matriu d'inscripcions, com les matrius de perfils generades al model de filtre de contingut i de filtre col·laboratiu. Tanmateix, generar aquesta última matriu és extremadament costós, ja que cal aplicar un seguit d'iteracions de l'algoritme ADAM.

Així doncs, hem seguit dues aproximacions principals: usar la matriu d'inscripcions i usar la matriu generada al model de filtre de contingut.

En el primer cas hem aplicat l'algoritme descrit al capítol cinqué, amb una lleugera variació: a la comparativa entre els dos usuaris, només hem tingut en compte aquells cursos que apareixien en almenys un dels usuaris. És a dir, si considerem U_i el conjunt de cursos de l'usuari i i U_j el conjunt de cursos de l'usuari j, s'hauran de complir les següents expressions per a poder comparar els dos usuaris (i i j):

$$|U_i \cap U_j| \geq 1$$
$$|U_i - U_j| \geq 0$$

Així doncs, només compararem aquells cursos que estiguen inclosos dins del subconjunt $U_i \cup U_j$ de manera, el codi quedarà de la següent manera:

```
def similarity(inscriptions:dict, user1_id:id, user2_id:id, metric:function)->
    float:
    shared = set(inscriptions[user1_id]).intersection(set(inscriptions[user2_id
        ]))
    union = set(inscriptions[user1_id]).union(set(inscriptions[user2_id]))

    if len(shared) == 0: return inf

    profile1 = []
    profile2 = []

    for course in union:
        if course in shared:
            profile1.append(1)
            profile2.append(1)
        elif curso in inscriptions[user1_id]:
            profile1.append(1)
            profile2.append(0)
        else:
            profile1.append(0)
            profiel2.append(1)
    return metric(profile1, profile2)
```

Amb açò aconseguim, per un costat, eliminar aquells usuaris que no comparteixen cap curs, i per l'altre, reduir àmpliament el cost computacional, ja que els vectors amb els quals calcular la distància són molt menors.

Ara que ja hem comentat açò, cal que expliquem quines mesures de semblança hem usat en aquest cas. De nou, hem utilitzat quatre: la similitud euclidiana, la de Manhattan, la del cosinus i la de Jaccard. Aquesta última, no havia sigut emprada fins ara perquè no compara vectors, sinó conjunts i és precisament aquesta característica per la qual anem a utilitzar-la en aquest cas. Així doncs, la similitud de Jaccard es calcula com la relació entre el cardinal de la intersecció dels dos conjunts i el de la unió dels dos conjunts. De manera més formal seria:

$$d(U,V) = \frac{|U \cap V|}{|U \cup V|}$$

Els resultats obtesos emprant aquest algoritme són excel·lents. De fet, millors que tots els que haviem calculat fins ara. Així doncs obtenim que la precisió de la distància al cosinus és de 72.07%, sent així el millor model, la precisió del model amb la similitud de Jaccard és de 72% i les precissions dels models amb les distàncies euclidiana i de Manhattan són una mica inferiors, de 70.99% ambdues.

D'altra banda, per a l'altra aproximació, se'ns plantejava un problema greu, com és la complexitat temporal. Com que les matrius són molt grans, el cost computacional era massa elevat i ja no podíem reduir els vectors com en el cas anterior. Tanmateix sí que podíem dividir l'espai vectorial per a simplificar la nostra tasca.

J.L. Bentley definia al seu article de 1975 *Multidimensional binary search trees used for associative searching* [31] un mètode anomenat arbres k-dimensionals (k-d trees d'ara endavant). Aquest mètode consisteix en enmagatzemar les observacions com a nodes d'un arbre, de tal manera que cada subdivisió de les branques de l'arbre representa una divisió dins l'espai vectorial. D'aquesta manera s'aconsegueix reduir el temps de busqueda de $O(n)$ on n és el nombre d'usuaris (en el nostre cas $O(80000)$) a $O(n^{\frac{k-t}{k}})$ (en el nostre cas $O(80000^{\frac{15-1}{15}}) \approx O(37689)$).

Tanmateix, existeix un problema en l'aplicació dels k-d trees al nostre problema i és que en última instància no estem cercant usuaris, sinó cursos. Així doncs, extraure k usuaris no sempre significarà que extraurem k cursos. Per a solucionar aquest problema existeixen dues solucions. La primera és aplicar k-d trees de forma recursiva. És a dir, provar amb una k de tamany 15, una k de tamany 30, etc. El problema d'açò és que establir la consulta sobre l'arbre pot ser una mica costós en espai quan hi ha moltes observacions, com és el cas. D'altra banda podem creuar les dues opcions, o el que és el mateix, emprar k-d trees quan siga viable i no emprar-lo quan no optimitze l'espai.

Per això cal calcular un k màxim. En el nostre cas considerarem que l'eficiència de l'algoritme serà insuficient quan el cost computacional siga $O(0.9 \cdot n)$. D'aquesta manera la deducció serà la següent:

$$
\begin{aligned}
n^{\frac{k-1}{k}} &\leq 0.9 \cdot n \rightarrow \\
\rightarrow log_n(n^{\frac{k-1}{k}}) &\leq log_n(0.9 \cdot n) \rightarrow \\
\rightarrow \frac{k-1}{k} \cdot 1 &\leq log_n(0.9) + log_n(n) \rightarrow \\
\rightarrow \frac{k-1}{k} &\leq log_n(0.9) + 1 \rightarrow \\
\rightarrow \frac{k-1}{k} - 1 &\leq log_n(0.9) \rightarrow \\
\rightarrow \frac{k-1-k}{k} &\leq log_n(0.9) \rightarrow \\
\rightarrow \frac{-1}{k} &\leq log_n(0.9) \rightarrow \\
\rightarrow \frac{1}{k} &\geq -log_n(0.9) \rightarrow \\
\rightarrow \frac{1}{-log_n(0.9)} &\geq k
\end{aligned}
$$

Així doncs, si considerem que nosaltres tenim huitanta mil casos (n), el valor k haurà de ser menor a cent set per a que la condició es complisca. Així doncs, l'algoritme que hem elaborat, queda de la següent manera:

```
1  def profiled_user_filtering(inscriptions:dict, user_id:dict, profiles:DataFrame
       , tree:KDTree, n:int, metric:function):
2      i = 1
3      result = dict()
4      while (len(keys(result)) < n) and (i<=107)):
5          result = dict()
6          d, users = tree.query(profiles, k=i*n)
7          user_candidates = zip(distance, users)
8          for (distance, other) in user_candidates:
9          if distance == 0: continue
10         for item in inscriptions[other]:
11             if item not in inscriptions[other]:
12                 result.setdefault(item, 0)
13                 result[item] = max([distance, result[item]])
14         i += 1
```

En aquest cas, només hem emprat tres mesures de distàncies, ja que l'algoritme k-d trees de la llibreria Scikit learn [32] no permet emprar distàncies que no siguen l'euclidiana, la de Manhattan o la de Chebyshev. Aquesta última, també anomenada distància

d'escacs, es calcula com a la màxima diferencia en termes absoluts entre components dels dos vectors. De manera més formal:

$$d(\vec{u}, \vec{v}) = \max_i(|u_i, v_i|)$$

No obstant això, els resultats d'aquesta aproximació no són tan bons, ja que només s'aconsegueix el 21.47% de precisió quan s'empra la similitud de Chebyshev o euclidiana, mentre que el valor es redueix quan s'usa la distància de Manhattan. Així doncs, el model final que usarem serà el que empra la matriu d'inscripcions amb la distància al cosinus com a mètrica.

6.2 Cold start

Ara parlarem de l'experimentació conduïda durant l'inici en fred. És a dir, quan s'introdueix un nou usuari. Tal com hem comentat al capítol cinqué, la idea inicial fou realitzar clústering sobre els cursos més propers, de manera que per a cada clúster es genere un vector, tal com podem veure a la figura 6.4. Aquests serien els vectors que empraríem per a computar els perfils dels usuaris i poder establir, com a mínim, el model de filtre de contingut.

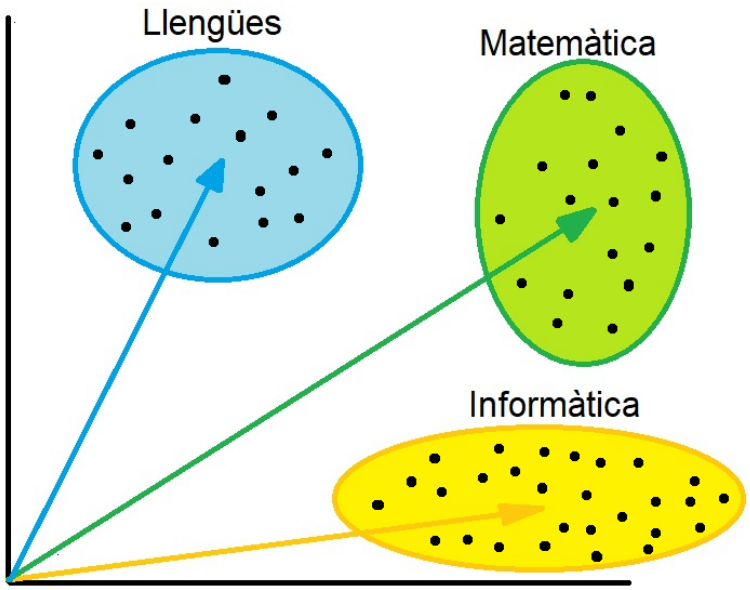

Figura 6.4: gràfica explicativa de la idea per a l'inici en fred.

Així doncs, era necessari tindre en compte només algoritmes que agrupen els ítems segons la distància. És per això que tindrem en compte l'algoritme K-means, l'algoritme K-medoides i el mètode Ward per a algoritmes jeràrquics.

En aquest primer cas, vam elaborar un anàlisi per determinar quin era el nombre de clústers òptim. No volíem que aquest nombre fóra massa gran, ja que l'usuari podria cansar-se i no completar el seu perfil. Per això, vam determinar que el nombre màxim possible de clústers seria de vint-i-cinc.

D'aquesta manera, vam aplicar una optimització per graella tenint en compte que el nombre de clústers podia anar des d'un fins a vint-i-cinc i que els algoritmes que podíem usar podien ser el complet o el d'Elkan. Aquest segon semblava treballar amb uns resultats pitjors per a les nostres dades, per la qual cosa el vam descartar. Per optimitzar el nombre de clústers, vam tindre en compte tant el coeficient de Silhouette [33], que devia ser proper a 1, com el de Davies-Bouldin [34], que deu ser mínim.

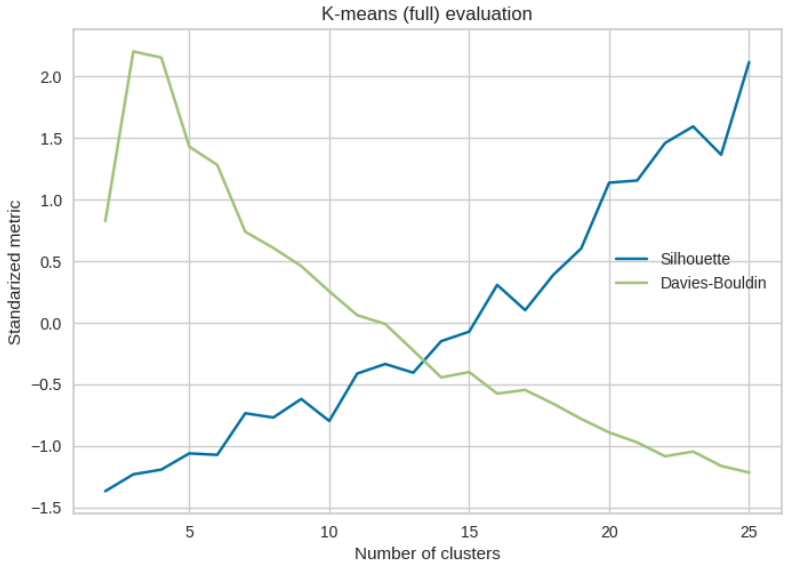

Figura 6.5: evolució de les mètriques d'avaluació de clústering estandaritzades per a l'algoritme "full"de K-means.

En conseqüència, vam obtindre la gràfica recollida a la figura 6.5, en la qual s'aprecia que amb un s setze clústers, el coeficient de Silhouette experimenta un màxim i el de Davies-Bouldin experimenta un mínim. Per aquesta raó, vam elaborar un clústering amb aquest nombre de conjunts.

Tanmateix, tal com es pot observar a la figura 6.6, el model no era gaire eficient i tot i que tenia algun clúster molt diferenciat, la realitat és que el coeficient de Silhouette promig és d'aproximadament 0.1, de tal manera que podem deduir que la major part dels clústers se superposaran.

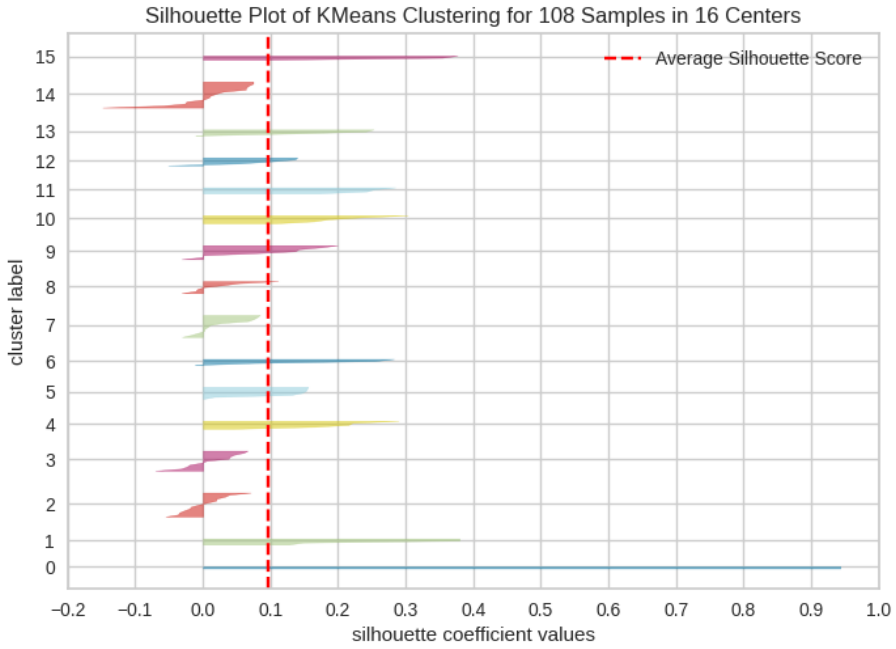

Figura 6.6: gràfica de Silhouette per al clústering en K-means "full"de setze categories.

Consegüentment, tindrem coses tan aberrants com un curs de castellà dins del conjunt de cursos de matemàtiques o un curs de màrqueting dins dels cursos catalogats com a d'edificació.

D'altra banda, vam avaluar el funcionament de l'algoritme K-medoides amb les mateixes mètriques. En aquest cas, únicament vam emprar un algoritme: el pam, ja que és un dels més precisos.

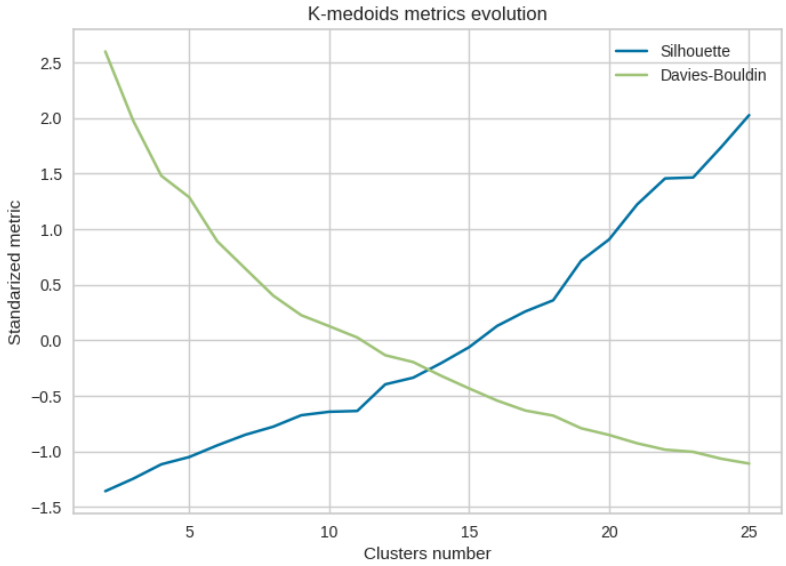

Figura 6.7: evolució de les mètriques d'avaluació de clústering estandaritzades per a l'algoritme de K-medoids.

Així doncs, vam obtindre la gràfica recollida a la figura 6.7, en la qual observem un pic en ambdues mètriques (positiu al coeficient de Silhouette i negatiu al coeficient de Davies-Bouldin) quan el nombre de clústers és de vint-i-dos.

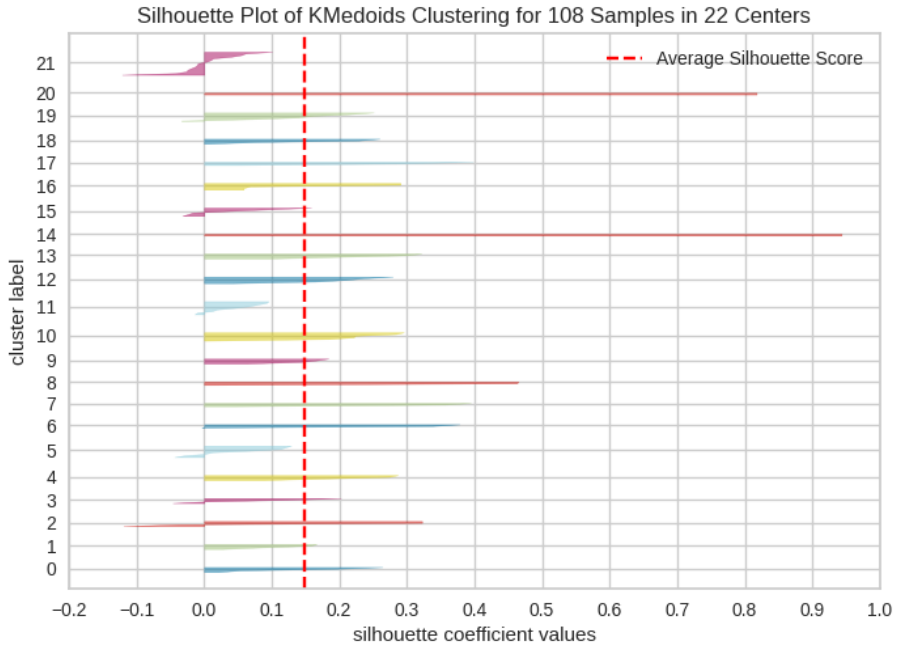

Figura 6.8: gràfica de Silhouette per al clústering en K-medoids de vint-i-dues categories.

No obstant això, els resultats, tot i que millors que els obtesos a l'algoritme de K-means, no són gaire bons. Tal com s'observa a la figura 6.8, el coeficient de Silhouette mitjà és de 0.15, un valor que ens indica que, tot i que alguns clústers de cursos són fàcilment identificables, la major part dels grups de cursos se superposen.

En tercer lloc, vam aplicar el mètode Ward per algoritmes jeràrquics. Així, vam examinar el mateix nombre de clústers: des de dos fins a vint-i-cinc, obtenint d'aquesta manera la figura 6.9.

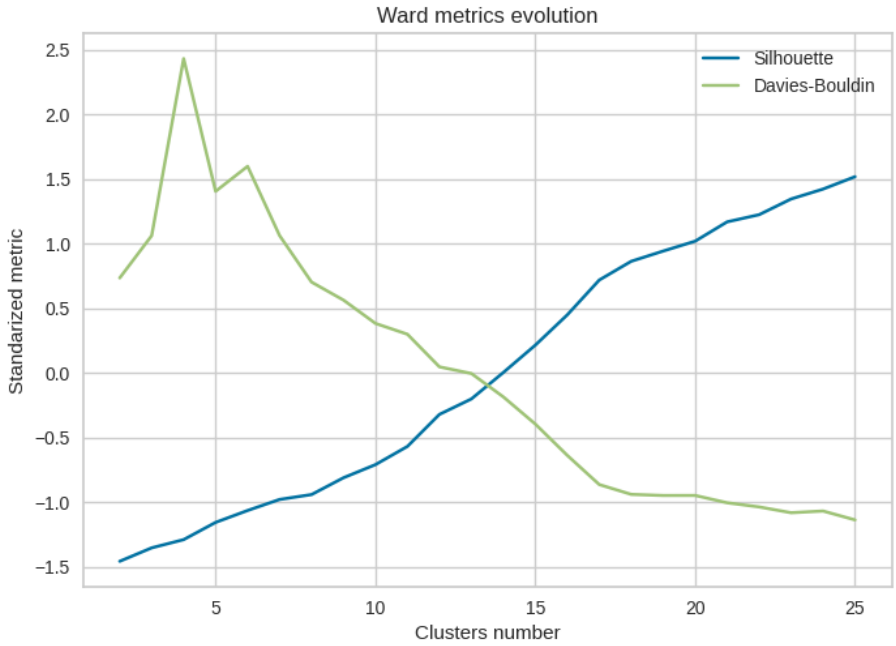

Figura 6.9: evolució de les mètriques d'avaluació de clústering estandaritzades per a l'algoritme jeràrquic de Ward.

En aquesta mateixa gràfica podem comprovar que per la regla del colze amb dèsset clústers, les dues mètriques tendeixen a estabilitzar-se. Per aquesta raó, vam decidir establir aquesta quantitat de categories de cursos. Malgrat això, els resultats obtesos no eren tan bons com havíem esperat ja que solament vam poder obtindre un coeficient de Silhouette de 0.17. La causa d'açò la trobem en la figura 6.10, en la que clarament s'aprecia que hi ha molts cursos que no poden incloure's en cap dels clústers principals.

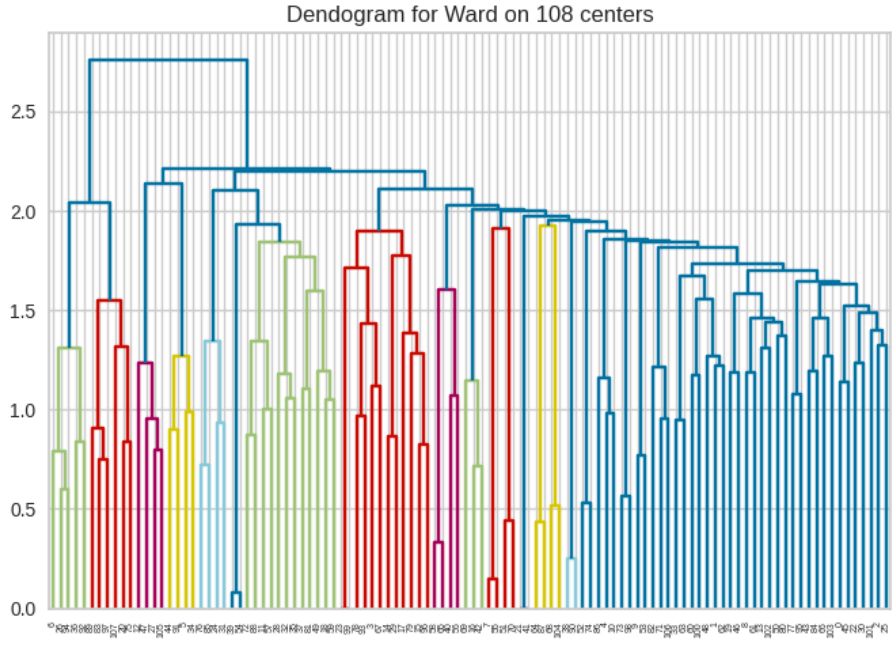

Figura 6.10: dendograma per al clústering jeràrquic de Ward.

Cap d'aquestes tres solucions no era gaire bona, per la qual cosa vam decidir aplicar altres mètodes d'extracció de tòpics. La idea era que si obteníem els n-grames més significatius d'un clúster, podríem acabar aproximant quins cursos pertanyien a quins clústers. Esperàvem que aquests mètodes funcionaren pitjor que els basats en distàncies, però calia contrastar aquesta hipòtesi.

El primer mètode d'extracció de tòpics que vam implementar, va ser Latent Dirichlet Allocation. Per a això, vam aplicar una nova graella per a optimitzar els resultats. El nombre òptim de tòpics va ser vint-i-dos. El resultat, tanmateix, era molt pitjor que en qualsevol de les primeres tècniques, ja que el valor de Silhouette era de solament 0.02.

Un altre mètode era emprar matrius no-negatives. Aquest mètode de reducció de variables consisteix a partir d'una matriu A, obtenir el producte de dues matrius $W \cdot H$, de tal manera que si $A \in \mathbb{R}^{nxm}$ i volem reduir la matriu a k valors, les matius resultants seran del tipus $W \in \mathbb{R}^{mxk}$ i $H \in \mathbb{R}^{kxn}$. D'aquesta manera, si nosaltres tenim una matriu de cursos $V \in \mathbb{R}^{mxd}$, el valor $w_{i,j} \in W$ indicarà la quantitat assumida de tòpic j al document i, mentre que el valor $h_{i,j} \in H$ indicarà la rellevància de la paraula j al tòpic i. Sabent açò, podem catalogar el document en un clúster utilitzant el màxim valor per a cada fila de W, sempre tenint en compte que perdrem informació.

Així doncs, inicialment havíem de decidir quina seria la quantitat de tòpics que podíem extraure del conjunt de documents. Per a això, vam elaborar un anàlisi de coherència, que podem trobar a la figura 6.11. D'aquesta manera, vam descobrir que el nombre òptim de tòpics havia de ser quatre.

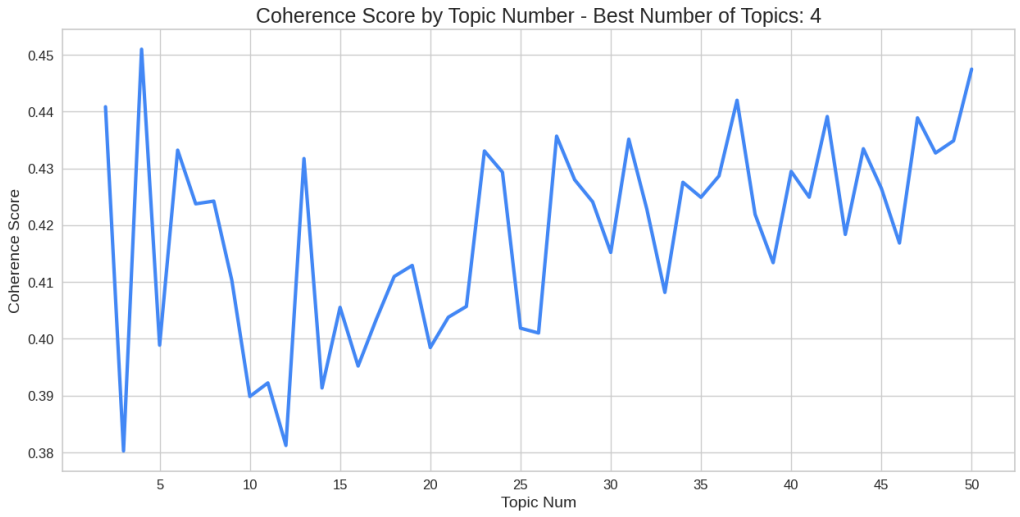

Figura 6.11: anàlisi de coherència per a la descomposició en matrius no negatives.

Després, vam aplicar una anàlisi de la diferència entre A i $W \cdot H$, per a decidir quin mode d'inici seria el més adequat, si emprant una doble descomposició de valors singulars (NNSVD), una descomposició per valors singulars amb el promig en lloc de zeros (NNSVDA), una descomposició amb valors singulars amb nombres aleatoris i no zeros (NNSVDAR), o simplement amb una iniciació aleatòria. Es va obtindre que el millor mode d'iniciació era l'NNSVDA, però el resultat del clústering tampoc no va ser gens satisfactori, assolint un valor de 0.03

Tal com hem vist, els models d'extracció de tòpics fets servir funcionen pitjor que els models de clústering, tal com havíem plantejat a la hipòtesi inicial. També hem conclòs que els cursos no es poden catalogar en clústers ben diferenciats, per la qual cosa, podem assumir que utilitzar els centres dels clústers com a vectors no és una bona idea. Tanmateix, la idea de classificació de cursos era molt bona, ja que permet que l'usuari definisca

els seus gustos sense perdre gaire temps. Per aquesta raó, un expert en aquests cursos va definir de forma manual la classificació en dènou categories.

6.3 Resultats

Així doncs, i en resum, hem obtés un sistema de recomanació per a cursos en línia compost per tres models tant quan hi ha personalització com quan no.

D'aquesta manera, dins dels models amb personalització, el que tindrà una major precisió segons el mètode amb el qual hem avaluat els models (que no té en compte el desbalanceig entre classes, és a dir, el biaix de popularitat) és el de filtre de recomanació d'usuaris, amb un valor de 72.07%. El segueix el model de filtre col·laboratiu amb una precisió de 50% i darrerament tenim el model de filtre de contingut, amb una mètrica de 45.89%.

Tanmateix, tots aquests models tenen una sèrie de fortaleses i debilitats que cal tindre en compte i que recollim a la taula 7.1.

Models amb personalització		
Filtre de contingut	Capaç de captar millor les preferències individuals de l'usuari i recomanar ítems nous. Eficient computacionalment.	Model arriscat que pot no captar bé els interressos dels usuaris.
Filtre col·laboratiu	Capta les preferències dels usuaris i està contrastat amb altres usuaris.	Molt pesat computacionalment. No sol arriscar-se a recomanar ítems nous
Filtre d'usuaris	Contrastat amb els usuaris i efectiu en ítems complementaris. Eficient computacionalment.	No recomana ítems nous i pot caure en biaixos de popularitat.
Models sense personalització		
Filtre popularitat	Capaç de cridar l'atenció. Més probable de captar els interessos de l'usuari	Cometrà sempre biaix de popularitat.
Filtre aleatori	Capaç de cridar l'atenció.	Poc probable que capte els interessos de l'usuari
Filtre contextual	Model amb personalització. Captarà l'interés de l'usuari.	Poc probable que cride inmediatament l'atenció

Taula 6.1: Avantatges i inconvenients dels models amb personalització

CAPÍTOL 7
Manteniment i desplegament dels models

En el context del desenvolupament d'un sistema recomanador per a cursos en línia de la Universitat Politècnica de València, desenvolupat per la Càtedra d'Intel·ligència Artificial Aplicada a l'Administració Pública, hem generat una aplicació model amb la llibreria Streamlit per a Python [37] per tal que servisca de futura referència per al desenvolupament del frontal.

Així doncs, en cap cas es pretén que aquesta aplicació arribe a producció, ja que no està en cap cas preparada per a tots els possibles atacs que puga rebre des de la xarxa ni tampoc està adequadament distribuïda computacionalment.

En les següents pàgines anem a explicar com hem plantejat el desplegament dels models, el manteniment i monitoratge i com hem desenvolupat l'aplicació.

7.1 Desplegament

Fins ara, hem parlat molt dels models de generació de candidats, però encara no hem parlat sobre la presentació de les recomanacions cap als usuaris. Tal com hem explicat al capítol segon, és necessari que els candidats es reordenen abans de presentar-se als usuaris. Així doncs, caldrà definir una manera de fer-ho.

En el nostre cas, i com que volem evitar els biaixos de popularitat, anem a emprar una ordenació inversa a la popularitat. És a dir, els quinze candidats generats pels models anteriors s'ordenaran per ordre d'inscripció de menor a major. Així, de la llista ordenada de candidats obtesa, sextrauran els cinc primers, que seran els que es recomanaran.

Tanmateix, amb això no hem acabat el desplegament dels models. Necessitem un sistema senzill, orgànic i segur per a mostrar les nostres recomanacions als usuaris. Així doncs, la manera més eficaç de presentar les recomanacions de forma natural, és informar a l'usuari de com s'han elaborat les recomanacions. Per a això farem servir el títol per a informar a l'usuari, tal com podem observar a les taules 7.1 i 7.2.

Model	Títol
Model de filtre de contingut	Segons els cursos que has consumit, podrien agradarte:
Model de filtre col·laboratiu	A alguns usuaris també els han agradat:
Model de filtre d'usuaris	Usuaris similars a tu han consumit els cursos:

Taula 7.1: Títols de presentació de models amb personalització

Model	Títol
Model de popularitat	Els cursos més populars:
Model aleatori	La recomanació del dia:
Model per categories	Les nostres formacions per categories

Taula 7.2: Títols de presentació de models sense personalització

Ara que ja hem definit quins són els candidats que hem de mostrar, hem de decidir com ho farem. La idea que plantegem a la demostració, és generar seqüencialment les recomanacions, l'una al damunt de l'altra amb els títols indicant quina és cada una, tal com podem veure a la figura 7.1. Per a cada curs, se li donarà un títol, una imatge i un botó d'accés al curs. Amb això pretenem que si l'usuari vol inscriure's, puga fer-ho.

Per a treure la imatge del curs, usem el seu url per a extreure informació de la pàgina web amb tècniques pròpies de la mineria de dades (web scrapping). Així, obtenim un diccionari amb l'identificador de les formacions com a clau i la imatge del curs com a valor.

El botó associat a cada formació hauria de dur a una pàgina pròpia de la universitat. Ací s'especificaran la descripció del curs, la llengua en la qual s'imparteix, el nombre de setmanes i hores esperades per a completar-lo i la nota necessària per a poder superar-lo. A més a més, s'adjuntarà la mateixa imatge abans mencionada i un botó que durà directament a la pàgina web d'edX per a poder inscriure-s'hi. Tampoc no oblidem els cursos complementaris, és a dir, aquells que formen part d'una formació més àmplia. Amb això pretenem que si l'usuari realment té interés en el tòpic, però no té el nivell necessari per a aquesta formació concreta, puga rebaixar el nivell a una altra del mateix grup de formacions.

Tanmateix, generar els models per a tots els casos és terriblement ineficient, ja que cada model requerirà un cert temps per entrenar-se. És, per tant, que emprarem tècniques de preentrenament i reentrenament, és a dir, no aplicarem directament els models, sinó que realitzarem un entrenament previ per a generar els pefils i guardarem aquests en un document json. Posteriorment, carregarem aquest document preentrenat al frontal i mostrarem les recomanacions per distància, que són molt més ràpides de calcular.

7.2 Desenvolupament de la demostració

La nostra aplicació model consta de tres pàgines principals: la pàgina d'inici, la pàgina de registre o d'inici de sessió i la pàgina de recomanacions. En aquest punt cal que expliquem que el sistema recomanador que proposem ha d'estar complimentat amb un cercador de cursos, de tal manera que els usuaris puguen cercar la formació que més els interesse.

La pàgina inicial no és més que un document en el qual s'explica quin és l'objectiu de la demostració. Salvant això, aquesta seria la pàgina a la qual podria accedir un usuari sense registrar-se. Conté les recomanacions dels models sense personalització situats en bateria, l'una sobre l'altra com es veu a la figura 7.1, de les quals parlarem una mica més endavant. D'altra banda, també inclou un enllaç per a la pàgina d'inici de sessió.

Figura 7.1: pàgina principal del prototip.

Per a construir les recomanacions, el model necessita carregar informació des d'un fitxer json. Açò planteja una vulnerabilitat, ja que si un atacant ha tingut accés a aquest fitxer d'alguna manera, el servidor podria estar carregant informació maliciosa. En un frontal adequat, aquesta informació hauria d'extraure's directament des de la connexió amb la base de dades del projecte que podria ser de SQL, de MongoDB, etc.

Tanmateix, en el marc d'aquest projecte realitzat per un estudiant, no comptàvem amb un gestor de base de dades, per la qual cosa hem emprat aquest tipus de ferramenta. No obstant això, cal recordar-ho si es desenvolupa un frontal basant-se en aquest model.

Existeixen dos formats de la pàgina d'inici de sessió: un format per a quan es crea un usuari i un format per a quan s'utilitza un ja existent. En el cas de crear un nou usuari, el sistema preguntarà a l'usuari quines característiques són les seues preferides i demanarà que es definisca un nom d'usuari i una contrasenya, que guardarà al sistema. Després, l'aplicació redirigirà a l'usuari directament cap a la pàgina principal, ja que el sistema no tindrà informació suficient per a realitzar recomanacions personalitzades. Si, en canvi, es vol iniciar sessió, el sistema comprovarà l'existència de l'usuari i la contrasenya dins de les seues parelles clau-valor. Si es troba alguna coincidència, es carregarà el seu perfil i es derivarà l'usuari a la pàgina de recomanacions personalitzades.

Figura 7.2: inici de sessió al prototip.

Parlant d'aquests usuaris i contrasenya, és necessari que ambdós estiguen codificats mitjançant algun tipus d'algoritme, ja que si no ho estan podrien produir-se atacs sobre la identitat dels usuaris. Per tant, nosaltres hem emprat l'algoritme de criptografia Fernet, que és considerat notablement robust. D'aquesta manera guardarem la informació de l'usuari (identificador i contrasenya), com a dades encriptades. Açò, a més de protegir a l'usuari de vulnerabilitats de suplantament d'identitat, també protegeix al sistema de realitzar consultes malicioses si en el futur s'implementa la càrrega de les dades des d'una base com pot ser SQL.

Finalment, tenim la pàgina que recomana els cursos de manera personalitzada. Aquesta pàgina conté informació del perfil de l'usuari, així com les seues recomanacions per-

sonalitzades en forma de bateria, mostrant la imatge i títol del mateix amb un botó que obre la pàgina de cada curs.

Dins de les pàgines dels cursos, mostrem la seua fitxa. És a dir, mostrem el títol i imatge del curs, una breu descripció, la llengua en la qual s'imparteix la qualificació necessària per a aprovar-lo i les setmanes i hores requerides per a completar la formació. També s'hi inclou un botó per a inscriure-s'hi, que redirigeix a l'usuari directament a la pàgina web d'edX i que afegeix el curs dins d'un fitxer json, de manera que tinguem constància de les actualitzacions que es donen.

Figura 7.3: fitxa dels cursos al prototip.

Per descomptat, es registren els inicis de sessió en un fitxer per a poder tindre en compte possibles atacs i errors que s'hagen comés durant l'execució. Amb això, es pretén mantindre una millora contínua de la demostració al llarg del temps.

7.3 Manteniment dels models

Tal com hem dit a la secció anterior, el frontal utilitzarà dades preentrenades, és a dir, s'entrenarà el model abans de posar-lo en pràctica amb els usuaris. Aquesta és una pràctica comuna en el sector, ja que ajuda a optimitzar el temps de computació del frontal i a protegir el model davant certs tipus d'atacs.

La major part de les escomeses que reben els sistemes recomanadors solen ser d'enverinament [35] i consisteixen a intentar trobar les fronteres dins del nostre model per a poder desplaçar-les amb mostres enverinades. És a dir, en el nostre cas, consistiria en un atacant creant usuaris falsos amb una combinació determinada de cursos que provoque que tots els models que es basen en altres usuaris, donen recomanacions errònies. Per a solucionar aquest problema, nosaltres hem formulat un pla de doble reentrenament per bateries.

Aquest reentrenament es realitzarà per bateria. Això significa que s'emmagatzemaran totes les actualitzacions d'usuaris (si un estudiant s'inscriu a un curs nou o si una nova persona entra al sistema) en una base de dades i després s'utilitzarà aquesta informació per a reentrenar els models. D'aquesta manera, resulta molt més fàcil detectar si hi ha dades enverinades en algun d'aquests conjunts al monitorar els resultats dels models reentrenats i eliminar-les.

Així doncs, al reentrenament diari els models, utilitzant els mateixos paràmetres generaran recomanacions noves per a aquells usuaris que hagen consumit un altre curs o per a aquells estudiants nou-vinguts. D'aquesta manera, s'espera que el model mantinga les recomanacions fresques.

Tanmateix, també serà necessari cada cert temps realitzar un reentrenament integral, ja que no sabem segur si els paràmetres d'optimització dels models (distàncies i hiperparàmetres) es mantindran constants al llarg del temps. Per això, proposem que aproximadament cada any o abans, si fóra necessari, es duga a terme un reentrenament integral dels models, per tal de millorar les recomanacions dels usuaris.

Una altra de les mesures que podem prendre en el nostre cas, és aplicar un màxim d'inscripcions en cursos. És a dir, podem assumir que un usuari podrà fer simultàniament certa quantitat de cursos, i que no podrà assumir-ne més per una simple raó temporal. Així, nosaltres definim que un usuari no pot estar realitzant més de deu cursos al mateix temps.

CAPÍTOL 8
Conclusions

En aquesta memòria, he explicat el procediment que hem emprat per a poder aplicar un sistema recomanador. Tal com hem explicat, considerem que resulta molt més acceptable utilitzar múltiples cursos de tal manera que es puguen complimentar els uns i als altres (taules 7.1 i 7.2).

En el nostre cas, hem aplicat tres models: un amb generació de candidats per filtre de contingut, un altre per filtre col·laboratiu i un altre que extrau les recomanacions a partir d'usuaris similars. El model que millor funcionava era el que es basava en usuaris similars per a poder recomanar cursos, que tenia una precisió del 72.07%. El seguia el model de filtre col·laboratiu amb una exactitud del 50% i ben prop es trobava el model de filtre de contingut, amb un 45.89%.

A l'experimentació hem pogut comprovar que en el nostre cas les distàncies basades en la norma d'un vector (distància euclidiana, de Manhattan...) funcionen significativament pitjor que aquells que es basen en altres distàncies.

8.1 Què hem aprés

Desenvolupant aquest projecte hem aprés tot el funcionament d'un sistema recomanador, els seus elements i la seua taxonomia. També hem adquirit coneixements sobre optimització en tècniques d'aprenentatge automàtic. Hem aprés com mantenir i monitorar un model de manera pràctica i segura i com previndre possibles atacs al sistema. També hem reforçat la nostra competència en l'extracció de característiques mitjançant tècniques de llenguatge natural i de reducció de variables emprant tècniques algebraiques.

Però també hem aprés una mica a desenvolupar una aplicació web, així com tota la càrrega de ciberseguretat que comporta aquest tipus de tasca. Hem reforçat els nostres coneixements de mineria de dades i de desenvolupament web, i hem aprés una mica de disseny web.

Malgrat això, els nostres coneixements no només s'han centrat en l'apartat tècnic. També hem aprés dels problemes de gestionar un projecte, especialment en l'apartat dels temps. Addicionalment hem adquirit un cert bagatge sobre problemes de privacitat de dades i com solucionar-los.

8.2 Idees a futur

Tot amb tot, el nostre projecte no acaba ací, ja que només és una porta cap al que pot arribar a ser.

Tal com hem dit, el model de mostra de l'aplicació no és implementable directament a la pàgina web, puix no està completament blindat davant els possibles atacs i té algunes vulnerabilitats que haurien de ser corregides.

Per a començar, s'hauria d'aplicar una base de dades per tal d'evitar els problemes a l'hora de carregar arxius propis del servidor. També caldria revisar els inicis de sessió i que es requerira d'algun tipus de doble autenticació (via correu electrònic per exemple), per tal d'evitar atacs de suplantament d'identitat. Finalment, és necessari reforçar la seguretat del codi i revisar-lo per tal que no hi haja vulnerabilitats. Seria molt recomanable realitzar una auditoria de hacking ètic per evitar possibles problemes en un futur.

Quant als models, tot i que ja són funcionals, són millorables en molts aspectes. Per començar, només hem utilitzat sis tipus de distàncies distintes. És, per tant, convenient que s'analitze si hi ha algun altre tipus de mètrica que millore l'exactitud dels models desenvolupats.

Tampoc no hem adreçat el problema que implica l'arribada d'un nou curs. Com que els nostres models es basen en l'extracció de característiques via tècniques de processament de llenguatge natural, no podem inserir directament un ítem al model. Caldria reentrenar tots els models per a obtindre algun tipus de resultat.

Una altra forma d'abordar aquest problema podria ser la implementació d'un model neuronal que prediga aquelles característiques que no estan en el sistema en funció de les altres. D'aquesta manera, podríem allargar una mica el temps abans de realitzar un reentrenament integral.

Tanmateix, assumim que no serà necessari emprar aquest tipus de ferramentes, ja que habitualment els cursos s'obren a l'inici de l'any acadèmic, per la qual cosa amb el reentrenament anual que hem proposat, hauríem de cobrir aquest inconvenient raonablement bé.

Una altra manera de millorar el sistema, podria ser la integració de dades pròpies dels estudiants de la UPV. D'aquesta manera, els alumnes de la universitat tindrien un grau de personalització molt més elevat i les recomanacions serien molt més fidels no només als seus interessos, sinó també a la seua forma d'aprendre.

Integrar dades pròpies dels estudiants podria no només ajudar-nos a millorar el sistema recomanador, sinó que també ajudaria a millorar els mateixos cursos, ja que es podrien aplicar anàlisis sobre quins tipus d'estudiants aprenen més i millor amb certs tipus d'estímuls.

Tanmateix, realitzar aquesta integració es força complicat, car caldria demanar permís als usuaris de la universitat per a poder fer-ho i necessitaríem tindre una anonimitat completa per tal que la informació delicada d'estudiants no es difonguera. Això implicaria també un nivell de seguretat extra als sistemes.

Addicionalment, també podríem millorar el sistema inserint models de recomanació nous. Els models de recomanació basats en regles d'associació són força potents i són una molt bona manera de millorar el sistema. Caldria, això no obstant, tindre en compte el desbalanceig de les dades, ja que aquest tipus de models són extremadament susceptibles als biaixos de popularitat.

D'altra banda, els models recomanadors basats en grafs són cada vegada més utilitzats[36] per la seua versatilitat i eficiència. Implementar aquest tipus de models dins del sistema podria ser extremadament beneficiós per a l'experiència de l'usuari.

Bibliografia

[1] P. Maes. Agents that reduce work and information overload. *Readings in Human-Computer Interaction: Toward the Year 2000*, 811–821, 2014. Elsevier Science. https://doi.org/10.1016/B978-0-08-051574-8.50084-4

[2] E. R. Núñez-Valdéz, J. M. Cueva Lovelle, O. Sanjuán Martínez, V. García-Díaz, P. Ordóñez de Pablos, C. E. Montenegro Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28:4:1186–1193, juliol, 2012. https://doi.org/10.1016/j.chb.2012.02.001

[3] J. Grudin. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM*, 37:1:92–105, gener, 1994. https://doi.org/10.1145/175222.175230

[4] S. K. Lee, Y. H. Cho, S. H. Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180:11:2142–2155, juny, 2010. https://doi.org/10.1016/j.ins.2010.02.004

[5] K. Choi, D. Yoo, G. Kim, Y. Suh. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research Applications*, 11:4:309–317, agost, 2012. https://doi.org/10.1016/j.elerap.2012.02.004

[6] R. Wahid, A. Ahmi, A. S. A. F. Alam. Growth and collaboration in massive open online Courses: a bibliometric analysis. *International Review of Research in Open and Distributed Learning*, 21:4:292–322, 2020. Athabasca University Press (AU Press). https://doi.org/10.19173/irrodl.v21i4.4693

[7] I. Uddin, A. S. Imran, K. Muhammad, N. Fayyaz, M. Sajjad. A systematic mapping review on MOOC recommender systems. *IEEE Acces*, 9:118379–118405, 2021. https://doi.org/10.1109/ACCESS.2021.3101039

[8] V. R. Raghuveer, B. K. Tripathy, T. Singh, S. Khanna. Reinforcement learning approach towards effective content recommendation in MOOC environments. *Proc. IEEE Int. Conf. MOOC, Innov. Technol. Educ. (MITE)*, 285–289, desembre, 2014.

[9] M. P. O'Mahony, B. Smyth. A recommender system for on-line course enrolment: an initial study. *Proceedings of the 2007 ACM conference on Recommender systems*, 2007. https://doi.org/10.1145/1297231.1297254

[10] R. Obeidat, R. Duwairi, A. Al-Aiad. A collaborative recommendation system for online courses recommendations. *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, 49–54, 2019. https://doi.org/10.1109/Deep-ML.2019.00018

[11] H. Zhang, T. Huang, Z. Lv, S. Liu, Z. Zhou. MCRS: A course recommendation system for MOOCs. *Multomedia Tools and Applications*, 77:7051–7069, 2018. https://doi.org/10.1007/s11042-017-4620-2

[12] R. Huang, R. Lu. Research on content-based MOOC recommender model. *2018 5th International Conference on Systems and Informatics (ICSAI)*, 676–681, 2018. https://doi.org/10.1109/ICSAI.2018.8599503

[13] Y. Pang, Y. Jin, Y. Zhang, T. Zhou. Collaborative filtering recommendation for MOOC application. *Computer Applications in Engineering Education*, 25:1:120–128, gener, 2017. https://doi.org/10.1002/cae.21785

[14] A.L.V. Pereira, E.R. Hruschka. Simultaneois co-clustering and learning to address the Cold Start in recommender systems. *Knowledge-Based Systems*, 82:11–19, juliol 2015. https://doi.org/10.1016/j.knosys.2015.02.016

[15] K. Ji, H. Shen. Addressing cold-start: scalable recommendation with tags and keywords. *Knowledge-Based Systems*, 83:42–50, juliol 2015. https://doi.org/10.1016/j.knosys.2015.03.008

[16] M. Zhang, J. Tang, X. Zhang, X.Xue. Addressing cold start in recommender systems: a semi-supervised co-training algorithm. *SIGIR: The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 73–82, juliol, 2014. https://doi.org/10.1145/2600428.2609599

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

[18] F.J. Anscombe. The validity of comparative experiments. *Journal of the Royal Statistical Society. Series A (General)*, 111:3:181–211, 1948. https://doi.org/10.2307/2984159

[19] W.H. Kruskal, W.A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47:260:583–561, desembre, 1952. https://doi.org/10.2307/2280779

[20] Pàgina web oficial de la llibreria googletrans. https://pypi.org/project/googletrans/

[21] C.D. Manning, P. Raghavan, H. Schütze. Capítol 2. *The term vocabulary and posting lists*. a *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[22] Pàgina web oficial de la llibreria Naturall Language Toolkit. https://www.nltk.org/

[23] C.D. Manning, P. Raghavan, H. Schütze. Capítol 6. *Scoring, term weighting and vector space model*. a *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[24] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient estimation of word representation. *arXiv*, setembre, 2013 https://doi.org/10.48550/arXiv.1301.3781

[25] Pàgina web oficial de la llibreria gensim. https://pypi.org/project/gensim/

[26] K. Pattel, P. Bhattacharyya. Towards lower bounds on number of dimensions for word embeddings. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 31–36, novembre 2017. https://aclanthology.org/I17-2006

[27] Pègina web oficial de la llibreria NetworkX. https://networkx.org/

[28] A. Barg, W. Yu. New bounds for equiangular lines. *arXiv:1311.3219*, novembre, 2013 https://doi.org.10.48550/arXiv.1311.3219

[29] I.T. Jolliffe, J. Cadima. Principal component analysis: a review and recent developments. P*hilos Trans A Math Phys Eng Sci*, 13:374(2065), abril, 2016. https://doi.org10.1098/rsta.2015.0202

[30] D.P. Kingma, J. Ba. Adam: a method for stochastic optimization. *3rd International Conference for Learning Representations*, desembre, 2014. https://doi.org/10.48550/arXiv.1412.6980

[31] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:9:509–517, setembre, 1975. https://doi.org/10.1145/361002.361007

[32] Pàgina web oficial de la llibreria scikit-learn. https://scikit-learn.org/stable/

[33] P.J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, novembre, 1987. https://doi.org/10.1016/0377-0427(87)90125-7

[34] D.L. Davies, D.W. Bouldin. A cluster separation measure. *IEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:2:224–227, abril, 1979. https://doi.org/10.1109/TPAMI.1979.4766909

[35] Y. Deldjoo, T. Di Noia, F.A. Merra A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Computing Surveys*, 54:2:35:1–38 https://doi.org/10.1145/3439729

[36] S. Wang, L. Hu, Y. Wang, X. He, Q.Z. Sheng, M.A. Orgun, L. Cao, F. Ricci, P.S. Yu. Graph learning based recommender systems: a review. *arxiV.2105.06339*, maig, 2021. https://doi.org/10.48550/arXiv.2105.06339

[37] Pàgina oficial de la llibreria Streamlit. https://streamlit.io/

APÈNDIX A
Planificació del projecte

A.1 Estructura de desglossament de la feina

- A1 - Investigació sobre sistemes recomanadors
- F2 - Pretractament de les dades
  - P2.1 - Ingenieria de dades
    - A2.1.1 - Anàlisi de qualitat de les dades
    - A2.1.2 - Tractament de les dades
    - A2.1.3 - Supressió de dades sensibles
    - A2.1.4 - Normalització del text
    - A2.1.5 - Aplicació de tècniques de llenguatge natural
    - A2.1.6 - Integració de les dades
  - P2.2 - Anàlisi descriptiu de les dades
    - A2.2.1 - Anàlisi univariant de les dades
    - A2.2.2 - Anàlisi multivariant de les dades
    - A2.2.3 - Anàlisi temporal de les dades
- F3 - Modelatge
  - P3.1 - Model de filtre de contingut
    - A3.1.1 - Depuració de dades
    - A3.1.2 - Creació de perfils
    - A3.1.3 - Creació del model recomanador
    - A3.1.4 - Validació del model
  - P3.2 - Model de filtre col·laboratiu
    - A3.2.1 - Creació del model recomanador
    - A3.2.2 - Optimització del model recomanador
    - A3.2.3 - Validació del model recomanador
  - P.3.3 - Model de filtre basat en usuaris
    - A.3.3.1 - Creació de perfils
    - A.3.3.2 - Aplicació de tècniques de reducció de variables
    - A.3.3.3 - Implementació del model recomanador
    - A.3.3.4 - Validació del model recomanador

  - P.3.4 - Models de filtres sense recomanació
    - A.3.4.1 - Implementació del model de popularitat
    - A.3.4.2 - Implementació del model aleatori
    - A.3.5 - Implementació del Cold Start

- F4 - Creació del prototip d'aplicació
  - A.4.1 - Aprenentatge de la ferramenta Streamlit
  - A.4.2 - Implementació d'un procés de creació d'usuaris
  - P.4.3 - Implementació dels distints models
    - A.4.3.1 - Implementació del model de filtre de contingut
    - A.4.3.2 - Implementació del model de filtre col·laboratiu
    - A.4.3.3 - Implementació del model de recomanació basada en usuaris
    - A.4.3.4 - Implementació del model de filtre de popularitat
    - A.4.4 - Presentació de resultats

A.2 Diagrama de Gantt

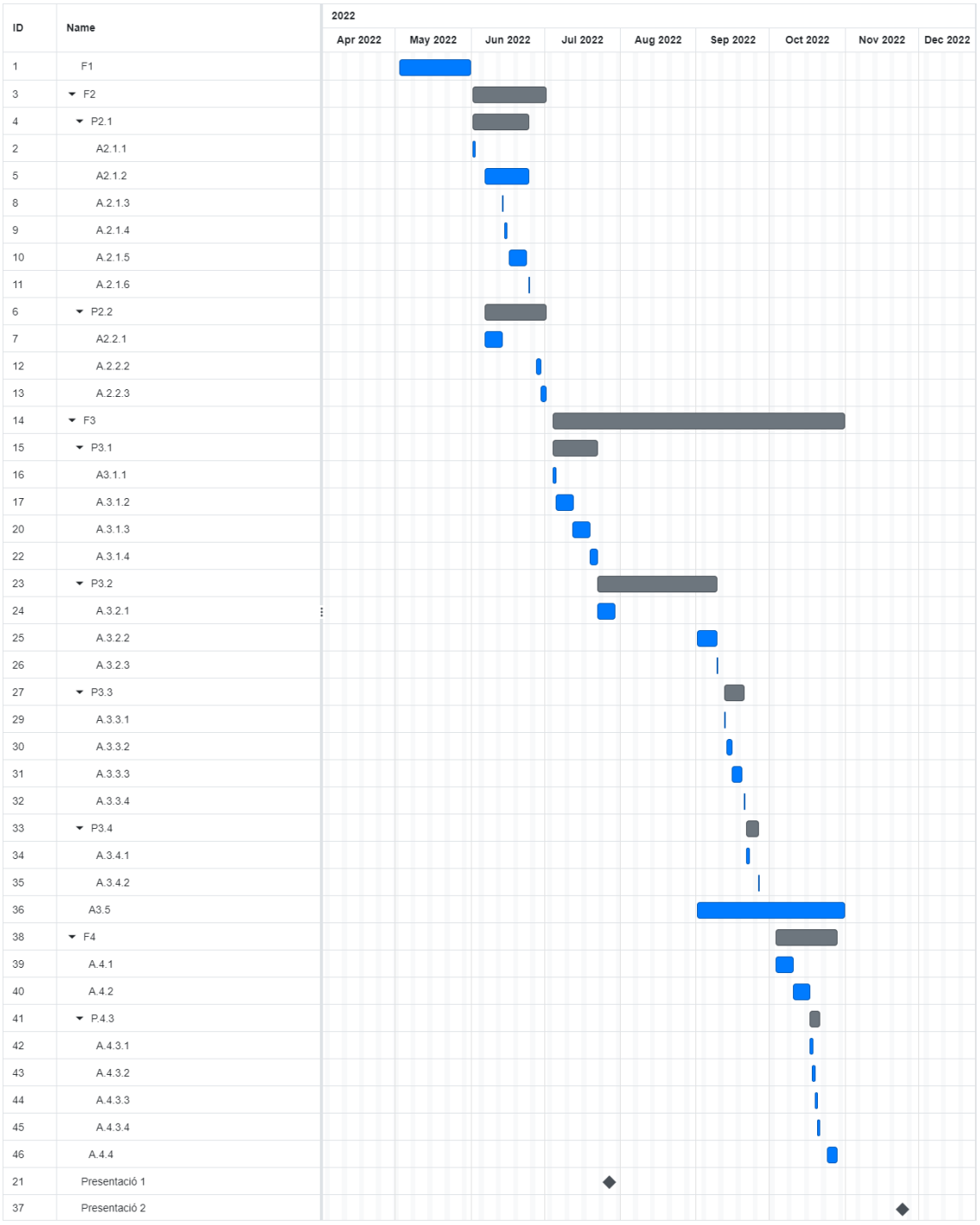

Figura A.1: diagrama de Gantt del projecte.

APÈNDIX B
Objectius de desenvolupament sostenible

En setembre de 2015, l'Organització de Nacions Unides van adoptar un conjunt d'objectius comuns a escala global per al desenvolupament que s'haurien de complir abans de l'any 2030. Aquesta Agenda 2030 compta amb dèsset punts i s'espera que el seu compliment ajude a millorar la vida de les persones a tot el món.

La UPV, com a institució socialment compromesa, té molt present el compliment d'aquesta agenda, per la qual cosa no podríem tancar la memòria d'aquest projecte sense considerar com podem afectar el seu compliment.

Així, dels dèsset objectius que conformen l'acord internacional, considerem que el nostre projecte afecta fonamentalment a cinc d'ells, tal i com es pot veure a la taula B.1.

Objetivos de Desarrolo Sostenibles.	**Alt**	**Mitjà**	**Baix**	**No Procedeix**
ODS 1. **Fi de la pobresa.**				**X**
ODS 2. **Fam zero.**				**x**
ODS 3. **Salut i benestar.**				**X**
ODS 4. **Educació de qualitat**	**X**			
ODS 5. **Igualtat de gènere.**			**X**	
ODS 6. **Aigua neta i sanejament.**				**X**
ODS 7. **Energia assequible i no contaminant.**				**X**
ODS 8. **Treball decent i creixeent econòmic.**		**X**		
ODS 9. **Indústria, innovació i infraestructures.**		**X**		
ODS 10. **Reducció de desigualtats.**			**X**	
ODS 11. **Ciutats i comunitats sostenibles.**				**X**
ODS 12. **Producció i consum responsables.**				**X**
ODS 13. **Acció pel clima.**				**X**
ODS 14. **Vida submarina.**				**X**
ODS 15. **Vida d'ecosistemes terrestres.**				**X**
ODS 16. **Pau, justícia i institutcions sòlides.**				**X**
ODS 17. **Aliances per assolir objectius.**				**X**

Taula B.1: implicació del projecte amb els Objectius de Desenvolupament Sostenibles.

En primer lloc, ens agradaria explicar sobre com el projecte ajudarà a mantindre una educació de qualitat (nové objectiu de desenvolupament sostenible). Durant la pandèmia de COVID-19, tal com hem explicat a la introducció, nombrosos usuaris s'han llençat a buscar alguna manera de formar-se per Internet. Per tant, millorar l'organització de cursos massius i oberts en línia, ajudarà que molta gent al voltant del món puga trobar més fàcilment aquelles formacions que li interessen. En conseqüència, esperem que el sistema desenvolupat durant el projecte puga ajudar a millorar l'educació de la població arreu del món.

D'altra banda, la creació i manteniment dels models de recomanació ajudaran a innovar i crearan part d'una infraestructura web per a la difusió dels cursos en línia. Aquesta infraestructura web s'ha vist que és crítica durant la pandèmia, ja que ha ajudat a digitalitzar nombrosos llocs de feina i ha millorat els sistemes de videoconferència. El nové objectiu de desenvolupament sostenible marca de gran importància reduir la bretxa digital, per la qual cosa, crear infraestructura web considerem que ajudarà a complir aquest punt.

Crear un sistema recomanador també ajudarà a la creació de llocs de treball que s'hauran de dedicar, per començar, a dissenyar un frontal per als models. A més, caldrà que certes persones es dediquen a monitorar, reentrenar i protegir els models que hem dissenyat, de manera que esperem que aquest projecte afavorisca que es complisca el huité objectiu de desenvolupament sostenible.

A més a més, tal com hem mantingut al llarg de tota la memòria, aquest projecte està molt conscienciat en els biaixos que poden aparéixer durant la creació de models utilitzant tècniques d'aprenentatge automàtic. Però com hem vist, aquests biaixos no només són sobre els ítems, sinó que poden extraure's dels usuaris en forma de models que empren dades demogràfiques. Aquests models sovint acaben reflectint els estereotips sexistes presents dins la societat heteropatriarcal en la qual ens trobem. Per tant, tindre en compte aquests biaixos ajudarà a arribar a una igualtat entre gèneres així com a reduir totes les desigualtats derivades dels prejudicis que té la societat.