Sistema Recomanador de Regals per a un usuari amb certes preferències

Resum - La idea central d'aquest treball de final de grau és la de crear una aplicació web on es recomana un regal a l'usuari al inserir unes preferències. Per a realitzar això s'haurà de treballar diferents àmbits del món de la programació, de l'enginyeria del software i de l'enginyeria de dades a part d'una recerca d'informació per a consultar l'estat de l'art dels sistemes de predicció i un anàlisis dels resultats obtinguts per a concluir quin model reacciona de la millor manera davant aquestes dades.

Paraules Clau - Aplicació web, Frontend, Backend, Sistemes Recomanadors, Sistemes Predictors, Cloud, Regals, Usuaris, Preferències.

Abstract - The central idea of this final degree project is to create a web application that recommends a gift to the user based on their preferences. To achieve this, it will be necessary to work on different areas of programming, software engineering, and data engineering, in addition to conducting research to consult the state of the art of prediction systems and analyze the results obtained in order to conclude which model reacts best to this data.

Index Terms - Web Application, Frontend, Backend, Recommenders Systems, Prediction Systems, Clous, Gifts, Users, Preferences

1. Introducció - Context del treball

La majoria de les persones que han de buscar un regal sigui per un amic o una persona estimada pateixen d'indecisió o falta d'idees per a adquirir aquest regal. Amb aquest projecte se solucionen tots aquests problemes, una aplicació web on insereixes els diferents gustos del destinatari del regal i aquesta et recomana diferents productes segons altres respostes dels usuaris, per a així entre tots aconseguir el regal perfecte.

2. Objectius Inicials del Treball

Els objectius que es van plantejar a l'inici de la realització del treball es poden resumir en un, aconseguir una bona recomanació a partir de les preferències d'un usuari, però alhora aquest objectiu es pot dividir en molts diferents.
A continuació una llista de 10 objectius que vull assolir al realitzar aquest treball:

- Crear un sistema productiu de Machine Learning
- Analitzar els diferents models de Machine Learning i escollir el millor per aquest cas
- Crear un bon backend per efectuar totes les operacions entre el client i el servidor
- Crear un bon frontend de la pàgina web que sigui ràpid i minimalista
- Crear un bon sistema de Base de Dades, amb una idea clara de la seva distribució
- Documentar la feina de manera correcta
- Crear un codi senzill i comprensible, d'una manera organitzada
- Aprendre noves tecnologies i dominar-les
- Allotjar la web al cloud de manera eficient i escalable
- Obrir la web al públic i tenir uns primers usuaris satisfets

3. Estat de l'art

En l'actualitat la Intel·ligència Artificial està molt de moda i tot arreu es comenta i s'anuncia el seu ús. Diferents aplicacions van començar a usar la IA per a recomanar pel·lícules o cançons ja fa uns anys (Netflix o Spotify) i era d'esperar que algú crees una aplicació per a recomanar regals.

Hi ha diferents aplicacions en el mercat que tenen la mateixa intenció que aquest projecte, però realment cap fa servir IA sinó que recomanen els productes patrocinats per a diferents marques, un exemple d'aquest pot ser Regality, que tenen un front molt ben definit però les recomanacions mai acaben de ser convincents. O un altra aplicació, Gift Wrap, que no recomana productes patrocinats però les recomanacions no són gens exactes, a part les features per al model són a partir d'un text, no dels gustos de l'usuari.

4. Desenvolupament

Per al desenvolupament d'aquest projecte s'han tractat molts temes diferents que es dividiran en subseccions, explicant en cada subsecció les tecnologies utilitzades, com s'ha afrontat el desenvolupament i quins impediments o canvis de plans s'han realitzat.

4.1. Disseny Tècnic

Per a acomplir el projecte hi ha diferents processos a explicar i diferents dissenys a documentar, primer de tot vull explicar com el projecte en general quedarà amb tots els diferents serveis.

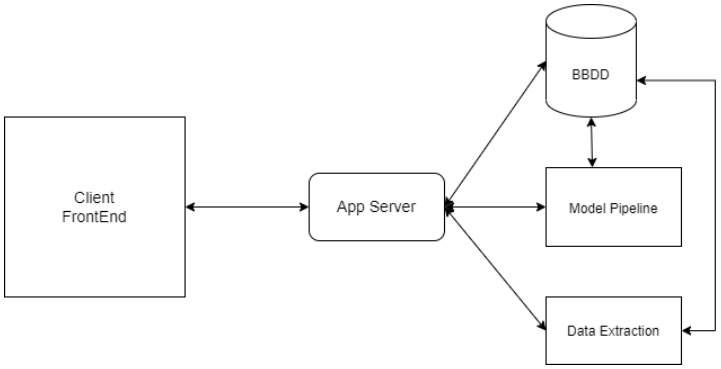


Fig. 1: Diagrama aplicació final de manera breu

En el diagrama anterior es pot observar diferents serveis interconnectats, més a l'esquerra podem observar el frontend de l'aplicació, que serà escrit amb React i serà la part visible del projecte, al centre podem observar el servidor de l'aplicació, aquest farà de façana per al frontend, és conegut com el backend també. Aquest gestionarà diferents serveis, com la Base de dades que a continuació explicarem, el pipeline per a gestionar l'entrenament i les prediccions i per últim les extraccions de les dades d'Amazon.

A continuació ens centrarem en la tecnologia usada per a la base de dades, com bé sabem podem escollir entre una base de dades relacional i no relacional, cada una té els seus avantatges, i desavantatges, després d'un anàlisis de les dues tecnologies he decidit decantar-me per a una base de dades NoSQL, això ha sigut, ja que en el paper citat en la bibliografia de Ruihan Wang exposa les diferents millores de les bases de dades no relacionals enfront de les relacionals, una d'elles i més important és la velocitat de lectura de dades. Com en el projecte gairebé no haurem d'escriure dades això forma un gran punt a favor, a més a més, l'estructura en JSON de les bases de dades no relacionals, més en concret de MongoDB és molt adaptable i ens serà fàcil d'aplicar en el projecte. L'estructura d'aquesta serà la següent:

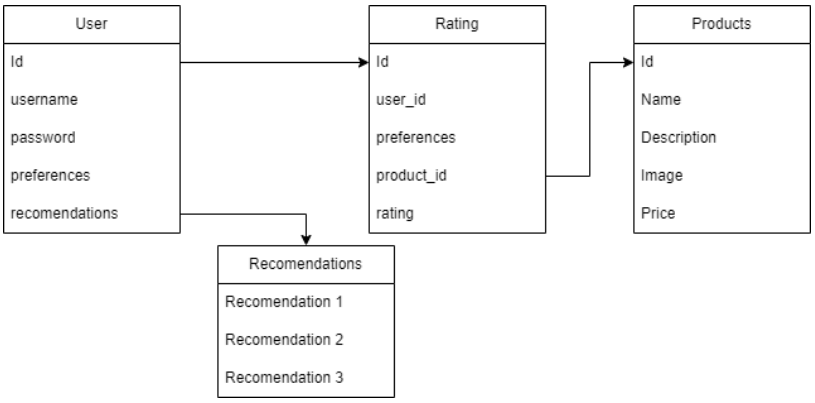


Fig.2: Disseny Base de Dades NoSQL

Per últim en aquest apartat s'explica com serà la pàgina web, com desenvoluparem el frontend, i això ho faré amb uns wireframes esquemàtics.

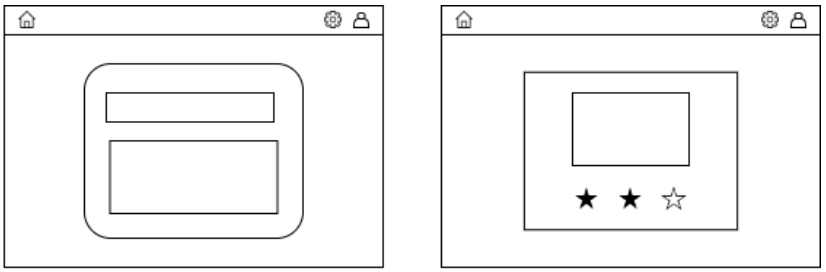


Fig. 3: Wireframe Rating Products Web App

En la figura 3 podem observar un petit wireframe representant com serà la pàgina per a alimentar la base de dades i donar un rating a cada producte. A la primera finestra serà on l'usuari entrarà les seves preferències i en el segon li apareixeran un número x de productes i li posarà una puntuació amb estrelles entre 0 i 5.

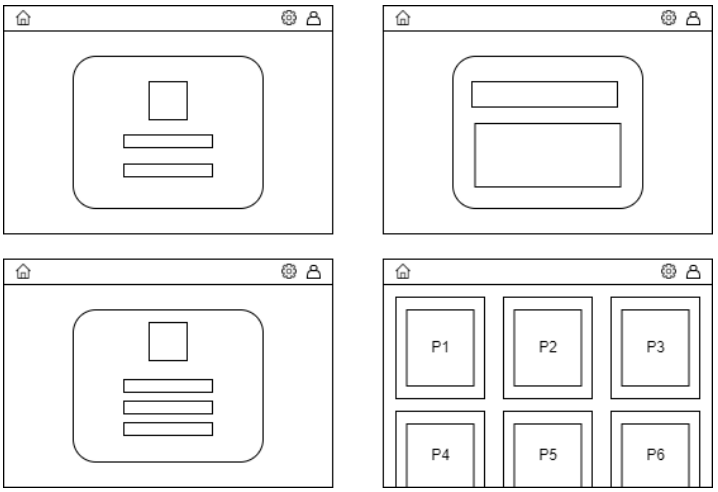


Fig. 4: Wireframe App final

En la figura 4 podem observar el wireframe per a l'app final. Aquest és format per 4 diferents pàgines, la pàgina per a iniciar sessió, la primera començant per l'esquerra, a sota trobem el wireframe per a la pàgina de registre, molt semblant a la d'inici de sessió amb la diferència de tenir un input més. A dalt a la dreta tenim la mateixa pàgina per a inserir les preferències i per últim hi ha la pàgina on podrem analitzar els resultats de la predicció en ordre de major a menor confiança.

4.2. Desenvolupament per a l'extracció de dades

El sistema d'extracció de dades per a aquest projecte no és massa complicat ni extens, però és realment important, ja que tenir una bona base de dades de productes significa un millor entrenament i predicció pel model IA.

El primer que es va realitzar durant aquesta èpica va ser investigar les tecnologies actuals i diferents mètodes per a extreure les dades dels productes d'Amazon. Primerament, vaig pensar que existiria un API creada pel mateix Amazon d'on podríem extreure la informació necessària, però això no és així, Amazon no ens dona cap opció fàcil per a conèixer els seus productes. Un altra opció investigada va ser la de fer web scrapping, però la seva dificultat i el temps que triga van ser les raons per a no abordar aquesta opció.

Finalment, l'eina que s'ha utilitzat ha sigut un API de Rainforest, un servei de pago que permet, mitjançant peticions API, guardar els productes seleccionats. Al no haver de dur a terme gaires peticions el cost d'aquesta plataforma ha sigut 0 de moment, ja que hem realitzat les peticions per a obtenir les dades dels productes més venuts d'algunes categories, com per exemple, guitarres, electrònica o cafeteres.

Aquests productes els guardem a la base de dades com bé es va assenyalar a l'entrega inicial, en un document de MongoDB amb el seu títol, descripció, preu, imatge i id.

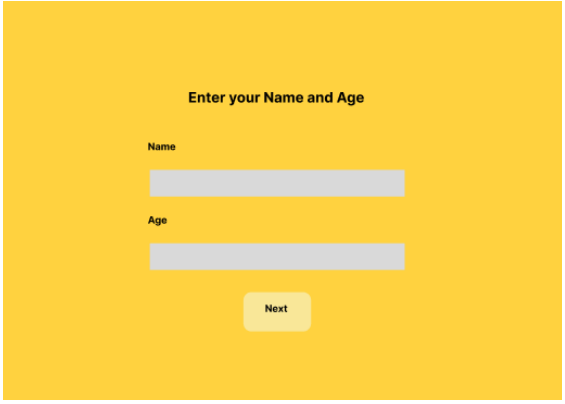
Un cop la base de dades ja té els seus respectius productes toca posar-los una qualificació, i això es farà amb la Rating App.

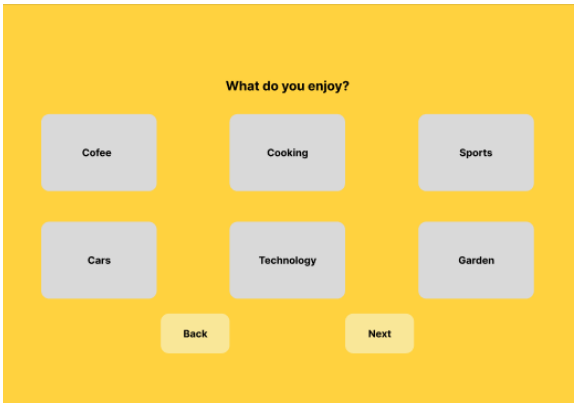
4.3. Desenvolupament pàgina Ratings App

La idea d'aquesta primera app és la d'alimentar la base de dades, és a dir, les preferències de l'usuari no afectaran les recomanacions, aquests són productes aleatoris per a veure de l'1 al 5 quan l'hi agrada a l'usuari, donant així un rating a cada producte disponible.

El desenvolupament tant d'aquesta pàgina com de tots els frontends del projecte s'han realitzat amb React, una tecnologia que desconeixia el seu funcionament a l'iniciar el desenvolupament, però que a base de documentació i prova i error s'ha arribat al resultat esperat.

Partint dels wireframes documentats anteriorment (Fig. 3) i de dur a terme un esquema més real de l'aplicació desitjada amb Figma (Fig. 4) es va desenvolupar tot el frontend de l'aplicació. El desenvolupament amb React no va suposar gran problema comparat amb el desenvolupament dels estils amb CSS, aquest desenvolupament va afectar el roadmap inicial del projecte, afegint una setmana més de l'esperat en el desenvolupament d'aquesta aplicació.





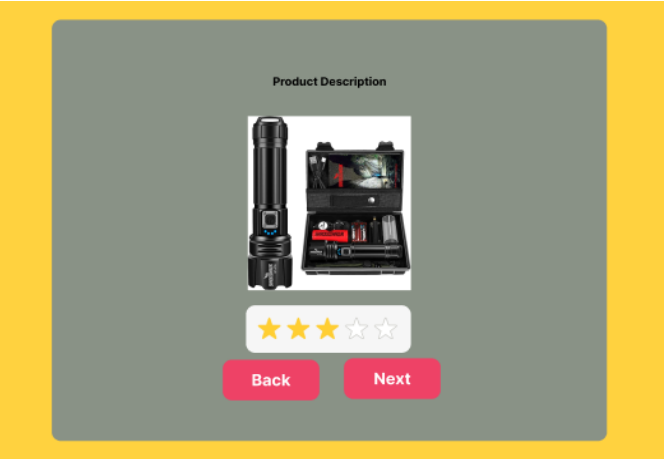


Fig. 5: Esquemes Figma Rating App

El backend d'aquesta va ser una tasca més senzilla, amb Python i la llibreria FastAPI vaig aconseguir crear diferents endpoints mantenint l'estructura i seguint els principis de disseny del REST API.

A continuació una imatge amb tots els endpoints creats per a aquesta aplicació i a continuació l'explicació de cada un d'ells.

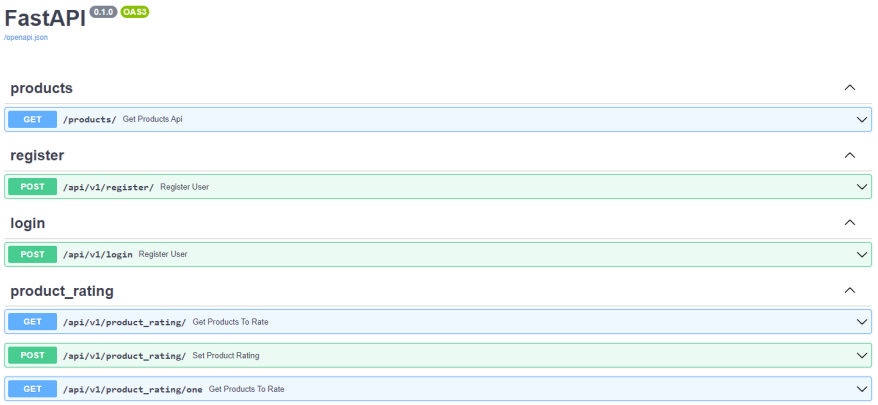


Fig. 6: Endpoints primera API

Aquests endpoints són els següents:

- GET /products: respon amb tots els productes guardats a la base de dades
- POST /register: endpoint per a registrar-se, no utilitzat en aquesta primera versió
- POST /login: endpoint per a iniciar sessió, no utilitzat en aquesta versió
- GET /product_rating: retorna 10 productes per a la pàgina de qualificació
- POST /product_rating: rep els productes amb una qualificació i els guarda a la DB

4.4. Desenvolupament model Machine Learning

Des de l'inici del projecte estava planejat realitzar i desenvolupar un sistema recomanador, tant és així que l'estat de l'art dels sistemes de filtratge col·laboratiu i segons contingut van ser estudiats. A

l'hora de desenvolupar, però es va veure que el nostre sistema no valien els models descrits anteriorment, per entendre això millor fixar-nos com tenim les dades ordenades.

Per a crear el dataframe del qual aprendrà el model s'han carregat tots els diferents documents de la base de dades amb tots els valors necessaris per a cada producte amb el seu corresponent ràting. Les columnes que formen aquest dataframe són les següents: Age, Coffee, Cooking, Sports, Cars, Technology, Garden, Asin, Rating. Diferenciant les columnes de features i la columna target, sent les features totes menys el ràting i la columna target sent el ràting, que és la columna que es vol predir.

	age	coffee	cooking	sports	cars	technology	garden	asin	category	price	rating
0	22	True	False	True	False	True	False	0	0	27.99	4
1	22	True	False	True	False	True	False	1	0	49.99	4
2	22	True	False	True	False	True	False	2	1	11.99	2
3	22	True	False	True	False	True	False	3	0	182.00	4
4	22	True	False	True	False	True	False	4	1	9.49	3
...	...	...	...	...	...	...	...	...	...	...	...
3270	28	False	False	False	True	True	False	392	7	11.00	0
3271	28	False	False	False	True	True	False	393	7	10.99	0
3272	28	False	False	False	True	True	False	394	7	5.82	0
3273	28	False	False	False	True	True	False	395	7	9.99	1
3274	28	False	False	False	True	True	False	396	7	2.99	0

Fig. 7: Estructura Dataframe

Un cop veiem com estan distribuïdes les dades es pot entendre perquè el sistema que s'ha de crear no es de recomanació, sino de predicció, de Regressió per a ser exactes. Això és perquè els models de Recomanació es basen en un usuari i els seus gustos per a predir el resultat d'un usuari diferent, en canvi en el nostre cas al no tenir els usuaris identificats només podem intentar que el model aprengui a donar un ràting a partir de les preferències de un usuari, sense saber de qui es tracta.

Per a desenvolupar el model de ML s'han seguit 3 enfocs diferents.

4.4.1. K-Nearest Neighbors

Aquest model es basa a trobar els "k" veïns més propers a un punt de dades de prova en funció d'una mesura de distància, com ara la distància euclidiana.

Un cop s'han trobat els veïns més propers, es fa la mitjana o ponderació de la variable objectiu d'aquests veïns per predir el valor del punt de dades de prova. En altres paraules, el model considera la tendència dels veïns propers per estimar el valor objectiu del punt de dades desconegut.

Per a desenvolupar el model s'ha usat la llibreria SKlearn i més especificament el model definit com KNeighborsRegressor.

Els resultats d'aquest i els altre models es mostraran a l'apartat dedicat als resultats.

4.4.2. Decision Tree Regressor

El model de regressió DecisionTreeRegressor és un algorisme d'aprenentatge automàtic que utilitza un arbre de decisions per realitzar tasques de regressió. En resum, aquest model construeix un arbre de decisions en el qual cada node representa una condició o característica del conjunt de dades.

En el procés d'entrenament, l'arbre de decisions es va dividint en branques successives en funció de les característiques del conjunt de dades. A cada node, es pren una decisió sobre quina característica proporciona la millor divisió per reduir l'error en la predicció de la variable objectiu. Aquest procés continua fins que s'arribi a un criteri d'arribada, com ara la profunditat màxima de l'arbre o el nombre mínim de mostres necessàries per crear un node terminal.

Els avantatges d'aquest model són la seva simplicitat i interpretabilitat, ja que l'arbre de decisions pot ser visualitzat i entès fàcilment. No obstant això, un possible inconvenient és la tendència a l'overfitting si l'arbre es construeix amb massa detall o si s'entrena amb conjunts de dades petits.

Per a desenvolupar el model s'ha usat la llibreria SKlearn i més específicament el model definit com a DecisionTreeRegressor dins el grup d'arbres de SKlearn.

Els resultats d'aquest i els altre models es mostraran a l'apartat dedicat als resultats.

4.4.2. Convolutional Neural Network i MLP

Per a l'últim model s'ha volgut desenvolupar desde cero un sistema de regressió amb la ajuda de la llibreria de Pytorch, per a realitzar això el model s'ha dividit en dos parts:

- CNN: 3 capes de convolució per a extreure les característiques del dataset i poder alimentar a la següent etapa del model.
  A continuació, es descriuen les capes utilitzades:
  - self.conv1: Capa de convolució 2D amb 1 canal d'entrada i ncanals de sortida. Utilitza un kernel de mida (1, 3) i un farciment (padding) de (0, 1).
  - self.conv2: Capa de convolució 2D amb n canals d'entrada i n*2 canals de sortida. Utilitza un kernel de mida (1, 3) i un farciment (padding) de (0, 1).
  - self.conv3: Capa de convolució 2D amb n*2 canals d'entrada i n*4 canals de sortida. Utilitza un kernel de mida (1, 3) i un farciment (padding) de (0, 1).
  - self.maxpool: Capa de max pooling 2D amb un kernel de mida (1, 2).
- MLP: La part de l'MLP és un regressor que processa les característiques extraïdes per la CNN. A continuació, es descriuen les capes utilitzades:
  - self.linear1: Capa totalment connectada (lineal) amb una dimensió d'entrada de 16 * 4 * ni una dimensió de sortida de 256.
  - self.linear2: Capa totalment connectada (lineal) amb una dimensió d'entrada de 256 i una dimensió de sortida de 256.
  - self.linear3: Capa totalment connectada (lineal) amb una dimensió d'entrada de 256 i una dimensió de sortida de 1.

  Finalment, es defineix una capa self.classifier que combina les capes lineals amb funcions d'activació ReLU i una funció sigmoide (nn.Sigmoid()) per a la sortida final del regressor.

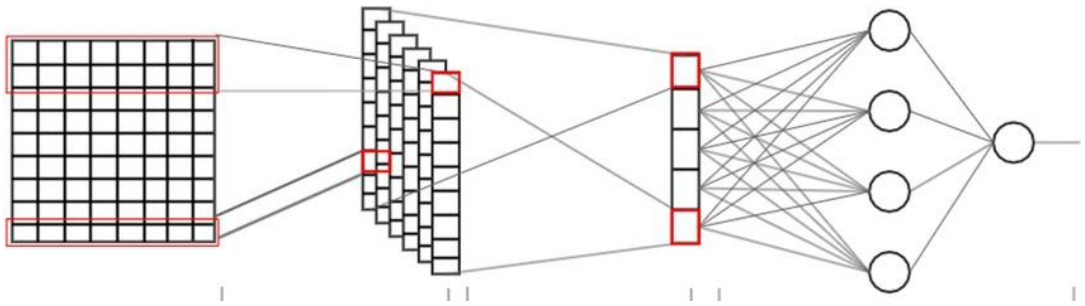


Fig. 8: Imatge semblant al CNN + MLP

4.5. Desenvolupament App final

En aquesta aplicació es plasma tot el desenvolupament del projecte sencer, s'utilitzen totes les eines documentades i desenvolupades anteriorment. La finalitat d'aquesta és la de, a partir d'unes preferències seleccionades per l'usuari, es recomani un regal adequat per l'usuari definit. El flux ideal d'un nou usuari en aquesta aplicació seria el següent:

- Si no consta d'usuari un és creat a la pàgina de registre.
- En crear l'usuari es logeja dins la pàgina directament.
- Insereix el nom i l'edat de la persona a la qui li vol realitzar el regal.
- Insereix els gustos d'aquesta mateixa persona per a qui el regal és destinada.
- El servidor utilitza el model de machine learning especificat anteriorment i et retorna un producte amb diferents opcions, mostrar un nou producte, accedir a la pàgina per a comprar aquest producte o modificar les preferències especificades al quart pas.

Per a dur a terme aquesta aplicació, igual que l'altre, s'han creat uns esquemes de Figma a partir dels Wireframes mostrats a l'inici del document. Podem observar aquests esquemes a continuació (Fig. 9).

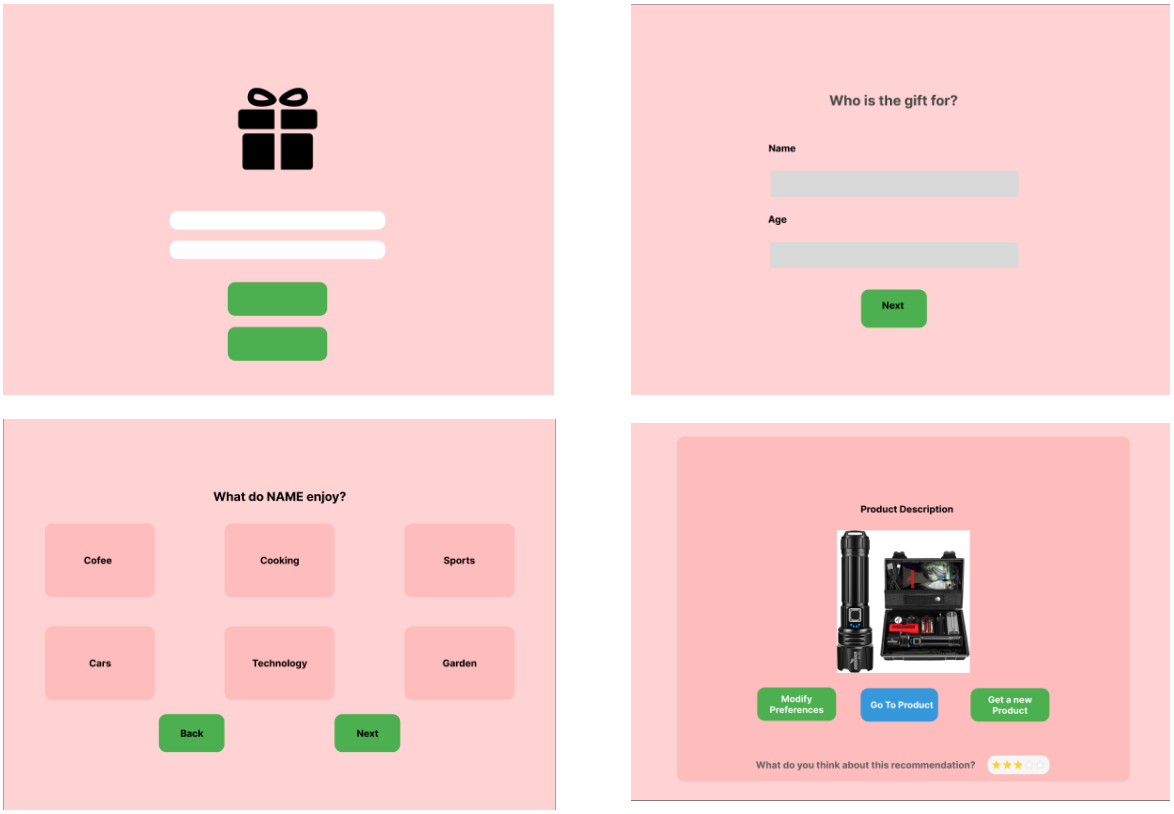


Fig. 9: Esquemes Figma App Final

Per al desenvolupament d'aquesta aplicació s'han aprofitat molts components de la Ratings App, la pàgina per a inserir el nom i la pàgina per a inserir les preferències son pràcticament iguals, s'ha creat des de zero la pàgina per a realitzar el Login i el Register, amb els seus endpoints corresponents i la pàgina per a mostrar el producte recomanat.

Per a aquesta última pàgina el més complicat ha sigut el desenvolupament del seu backend, era necessari cridar al model amb totes les dades necessàries un cop entres a la pàgina. A l'inici no era clar si hi hauria problemes de concurrència, ja que diferents usuaris estarien cridant al model alhora, però això no ha suposat un gran problema, al no tenir un número molt elevat de productes la predicció del model és prou ràpida (0.9s) per a poder gestionar diferents usuaris alhora.

Els endpoints creats específicament per a aquesta pàgina han sigut els següents:

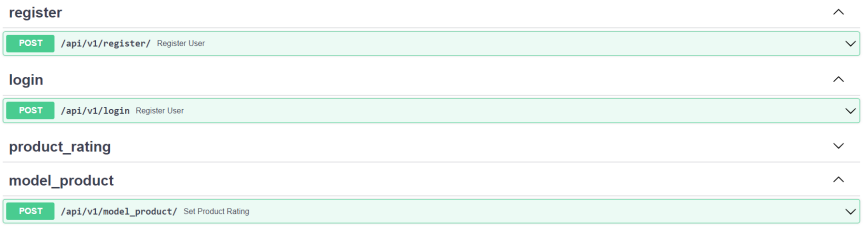


Fig. 10: Endpoints creats per a l'app final

- POST /register: endpoint per a registrar-se, no utilitzat en aquesta primera versió
- POST /login: endpoint per a iniciar sessió, no utilitzat en aquesta versió
- POST /model_product: endpoint que rep el nom, l'edat i les preferències insertades per l'usuari i retorna el producte amb totes les característiques necessàries recomanat pel model de machine learning.

4.6. Deploy projecte al Cloud

Tenia clar des de l'inici d'aquest projecte que aquesta pàgina havia de ser oberta al públic i que tothom pogués accedir, i quina millor manera de fer una pàgina accessible al públic que desplegant-la al cloud. Per a poder fer això es va haver d'investigar bastant, mirar si era una opció, mirar si es podia realitzar de manera gratuïta i senzilla, quin cloud seleccionar.
Finalment, em vaig decidir per AWS, ja que és el Cloud que més havíem vist a classe i coneixia les seves màquines gratuïtes.

Abans, però de desplegar al Cloud vaig dockeritzar els dos repositoris que tenim fins ara, el front i el back. Es va crear una imatge per a cada un i el seu docker-compose corresponent.

Un cop teníem tot això amb una màquina EC2 d'AWS vam aixecar ambdós contenidors per a poder accedir públicament a la pàgina de Rating.

Aquest és l'URL: http://13.37.207.34/

5. Resultats

La tasca a atacar per aquest projecte no era gens senzilla des del punt de vista del machine learning, després d'una gran recerca de dades no se n'han trobat cap d'interessant així que totes les dades venen per usuaris coneguts o familiars responen el qüestionari de la ratings app.

Aquestes dades rebudes de l'aplicació de rating tenen un format ordenat perquè tota la base de dades segueixi la mateixa estructura, aquesta estructura ja ha estat mencionada anteriorment. Per a realitzar una predicció l'única dada que falta és el rating, que serà el valor que intentarà predir el model.

En recollir aquestes dades de diferents usuaris sense cap coneixement del funcionament del model i omplint moltes valoracions com a 0, aquestes han resultat estar molt desequilibrades, com podem veure a la figura a continuació, el nombre de zeros era 5 cops superior a qualsevol altra dada.

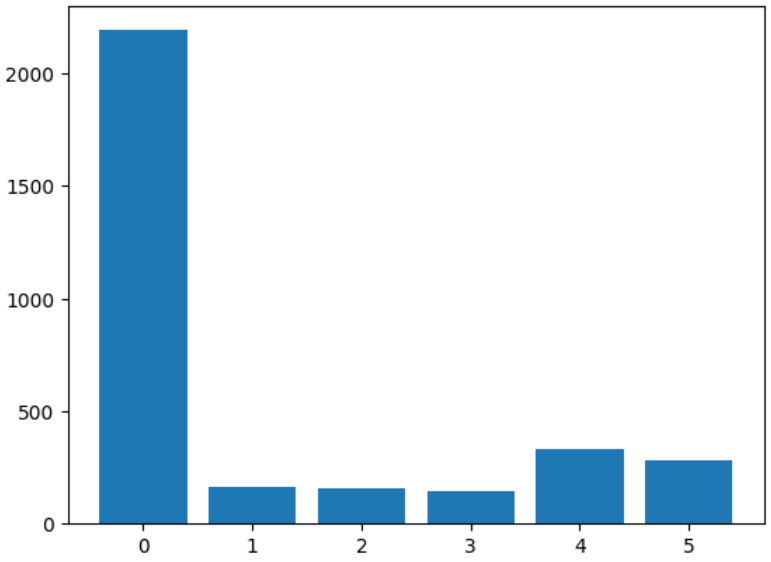


Fig. 11: Bar plot dades desequilibrades

Per a mesurar els resultats dels models desenvolupats s'ha fet servir les clàssiques mètriques per a avaluar Regressions, el MSE i el RMSE.

	MSE	RMSE
KNN	3.167939	1.77987
DecisionTreeRegressor	1.29465	1.13782

Com podem observar, el resultat del DecisionTreeRegressor són millors respecte als del KNN, això és a causa que el DecisionTreeRegressor pot ajustar-se millor als patrons no lineals i complexos presents en les dades, com és en el nostre cas. En canvi, el KNN tendeix a funcionar millor amb dades més lineals.

A continuació també es mostra un boxplot (Fig. 12) dels errors quadràtics d'ambdós models per a les dades de test. Aquí podem tornar a analitzar que les dades predites per l'arbre són més concretes que en el cas del KNN. També podem veure que el número d'outliers és molt menor al del KNN.

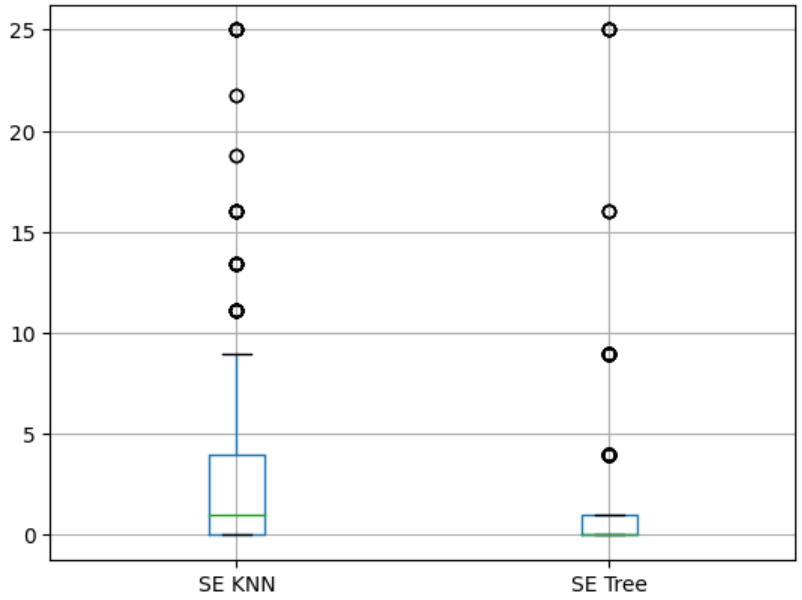


Fig. 12: Boxplots mostrant el error quadrat entre resultats

Per a tancar l'apartat dels resultats fa falta parlar del model CNN que s'ha explicat en l'apartat 4.4.2. El desenvolupament d'aquest ha sigut correcte, però el cost de les prediccions i del seu entrenament és molt alt respecte als altres dos, cosa que ens resultaria amb problemes de concurrència. A part, els resultats d'aquests no eren millors que els de l'arbre definits anteriorment.

6. Conclusions

Les conclusions del treball son les següents:

- El machine learning és una eina molt poderosa, però té com a requisit l'existència d'un nombre elevat de dades que en alguns àmbits són complicades de aconseguir.
- El Front de la pàgina web és fàcil de desenvolupar amb la quantitat de llibreries existents en el mercat, però el desenvolupament d'estils sempre és la part més complicada.
- El Backend de l'aplicació amb les llibreries actuals és molt fàcil de crear i de mantenir.
- La interpretació dels resultats dels diferents models són un desafiament important, ja que els gustos de cada usuari són únics i no objectius.
- La concurrència amb els models de machine learning al Cloud és un gran problema, és per això que tantes posicions de MLE estan sorgint actualment.
- A vegades val més ser senzill que complicar-se resultant u cas de sobre-enginyeria.
- El desplegament d'una aplicació al Cloud de manera senzilla és accessible per a tothom, però configurar tot l'entorn de manera adequada i productiva requereixen molt de temps i atenció.

7. Bibliografia

- [1] Ruihan Wang. SQL vs NoSQL.
- [2] Viplove Prakash. Web application architecture
- [3] YongKang Xing. Research and analysis Frontend Frameworks
- [4] Yugesh Verma. Flask vs Django vs FastAPI

- [5] ZoltanBettenBuk. How To Scrape Amazon Product Data
- [6] RainForest. RainForestAPI Documentation
- [7] React. React Documentation
- [8] Nikunj Mani Gupta. AWS VS AZURE VS GCP: LEADERS OF THE CLOUD RACE
- [9] analyticsindiamag. Django vs Flask vs FastAPI
- [10] Naruti Techlabs Types of recommendation systems
- [11] Vatsal Recommendation Systems Explained
- [12] CSS Documentation CSS Docs
- [13] Hagay Lupesko Introduction to Deep Learning for Recommendation Systems
- [14] Yifan Hu Collaborative Filtering for Implicit Feedback Datasets
- [15] Tensorflow Tensorflow Recommenders
- [16] Netflix Techblog https://netflixtechblog.com/