Sistema Classificador, Recomanador & Anàlisi de Dades

Ferran Vera Filella

Resum - SplitIt és una aplicació de gestió de despeses personals i grupals. S'ha desenvolupat un sistema classificador i recomanador. El sistema classificador ha de ser capaç de classificar els tiquets pujats pels clients a l'aplicació SplitIt. Es duen a terme un conjunt de investigacions i avaluacions per determinar el millor model. Gràcies a la classificació de tiquets es poden agrupar els clients segons els seus gustos, d'aquesta manera i amb el desenvolupament del sistema recomanador es poden generar recomanacions personalitzades pels usuaris de l'aplicació.

Paraules claus - Machine Learning, classificador, recomanador, anàlisis, ETL.

Abstract - SplitIt is a personal and group expense management application. A classifier and recommendation system has been developed. The classification system must be able to classify tickets uploaded by customers to the SplitIt application. A set of investigations and evaluations are performed to determine the best model. Thanks to the classification of the tickets, customers can be grouped according to their tastes, in this way and with the development of the recommendation system, personalized recommendations can be generated for the users of the application.

Key words - Machine Learning, classifier, recommender, analysis, ETL.

1. Introducció

En algunes ocasions al repartir unes despeses entre varies persones es converteix en una tasca complexa i feixuga, ja que el ticket pot tenir molts items i s'ha de repartir de forma no equitativa, en funció del que ha consumit cada persona.

Així doncs neix SplitIt, que té l'objectiu de donar solució a aquestes situacions i moltes més. SplitIt és una aplicació que mitjançant un sistema d'Intel·ligència artificial de lectura de tiquets i amb funcionalitat de xarxa social, permet dividir de manera còmoda i ràpida qualsevol tiquet o despesa creant un subtiquet per cada integrant del grup.

Fig 1: Funcionament SplitIt

L'aplicació consta d'una interfície d'usuari *'user friendly'* que permet de manera intuitiva navegar per l'aplicació i fer anar les seves funcionalitats.

A més a més, SplitIt pot gestionar despeses personals ademés de grupals. Es poden introduir les despeses en forma de tiquet o manualment i automàticament, gràcies a l'ajuda de la intel·ligència artificial, la qual agrupa les despeses en diferents grups, per veure i accedir-hi de manera més clara i senzilla, i tenir-ne un millor control.

1.1 Diagrama de flux funcional de l'aplicació

Per poder entendre quin és el workflow de l'aplicació en un cas d'ús de negoci es necessita saber quines són les característiques funcionals d'aquesta aplicació.

Primer es disposa d'un diagrama de flux simplificat (Annexe - Figura 2) que permet veure les característiques principals de l'aplicació. A aquesta s'hi accedeix mitjançant una url web desde qualsevol dispositiu. Un cop s'ha accedit s'introdueixen les credencials de l'usuari, en el cas de que no sigui usuari encara s'ha de registrar, i un cop registrat, apareix una pestanya **Home**, des d'on es pot accedir o bé a les dades personals o carregar un ticket.

Una vegada registrat es pot cercar a amics per nom d'usuari i ser agregats. A partir d'aquí ja es pot crear un grup, que és amb el que es divideixen les despeses a posteriori i es poden crear tants grups com es vulguin. En funció del context es tria un grup o un altre per repartir les comptes (Annex - Figura 3).

A continuació només fa falta introduir la informació del que s'ha consumit: ja sigui en un supermercat, un restaurant o una factura. Mitjançant un telèfon o un ordinador es captura la informació del tiquet, o bé amb una fotografia feta des de la mateixa aplicació o pujant a l'aplicació l'arxiu que conté el tiquet, i es processa mitjançant un algorisme d'intel·ligència artificial que analitzarà tots els elements d'aquest: lloc, elements, preus i unitats (Annex - Figura 4).

Es veu una nova pàgina on es troben tots els elements del tiquet desglossats. Per cada element del tiquet, es selecciona quina o quines persones l'han consumit i es reparteix el seu preu entre aquestes.

Finalment, els membres del grup que hagin consumit, independentment, poden veure des de la seva aplicació quins són els elements que se'ls ha assignat i a qui han de pagar l'import corresponent (Annex - Figura 5).

2. Funcionalitats de l'aplicació:

- **Registre:** Els nous usuaris han de crear un usuari, introduint les seves dades correctament en una pestanya de registre, per tal de poder accedir a l'aplicació.
- **Login:** Per tal de poder accedir a l'aplicació és necessari introduir les credencials d'usuari (nom d'usuari i contrasenya) i que aquestes coincideixin amb les d'un usuari registrat a la base de dades.
- **Creació d'amistats:** Els usuaris poden buscar i afegir a altres usuaris registrats en l'aplicació per poder interactuar amb ells posteriorment.
- **Creació de grups:** Els usuaris poden crear grups d'usuaris amb les seves amistats.
- **Pujada d'imatges de tiquets:** Els usuaris han de pujar imatges de tiquets a l'aplicació des del dispositiu o poden fer una foto d'un tiquet que pren l'aplicació automàticament.
- **Assignació de grup:** L'aplicació permet a l'usuari que ha pujat un tiquet triar en quin grup es gestiona la despesa per poder dividir-lo entre aquests usuaris.
- **Lectura dels elements dels tiquets:** L'aplicació pot processar imatges de tiquets per tal de detectar elements com els productes, els preus, el nombre d'elements el lloc i hora de compra, etc.
- **Assignació d'elements del tiquet a usuaris:** L'aplicació permet a l'usuari que ha pujat el tiquet assignar quin element correspon a cada usuari.
- **Gestió de comptes personals:** L'aplicació permet fer diferents anàlisis sobre les despeses d'un usuari: per àmbit, dates, per grup d'amics, etc.
- **Estadístiques:** Es mostren una sèrie de gràfics amb informació sobre el tipus de restaurants visitats, consum per cada tipus de restaurant, i les recomanacions de restaurant per l'usuari.

3. Objectius globals

- Creació d'una aplicació web capaç de realitzar totes les funcionalitats descrites
- Compartir despeses de ticket
- Creació de recomanacions per a usuaris.
- Creació de promocions per a restaurants, botigues, etc.
- Oferir perfils d'usuaris a restaurants, botigues, etc.

4. Planificació

El treball es realitza mitjançant una metodologia agile, que és una forma que proporciona rapidesa i flexibilitat en el desenvolupament de projectes.

L'eina utilitzada per portar a cap la metodologia agile ha estat Microsoft Teams, ja que ens permet assignar tasques, posar dates límits i fer-ne el seguiment, afegint els comentaris sobre el grau de desenvolupament. Cal dir que la metodologia

agile és una guia on s'intenten complir tots els terminis tal com estan definits, i com això no acostuma a passar, es deixa un marge de temps per si alguna tasca ha necessitat més temps del previst.

A part de la metodologia agile, s'ha utilitzat també una eina per realitzar diagrames de Gantt per a planificar el projecte, el qual proporciona una vista general de les tasques programades, de manera que tots els components del grup saben les tasques que han de completar i quina és la data límit. Es mostra la data d'inici i final d'un projecte i totes les tasques que hi ha dins, aquesta informació és la que s'ha introduït a la secció de ***tasks*** de **Microsoft Teams.**

S'han realitzat dos diagrames de Gantt: un comú per a tots els integrants del grup, ja que hi ha tasques que es fan de manera conjunta entre tots, i un altre individual, amb les tasques específiques per cada membre (veure Figura 6 i 7 de l'annex).

S'ha emprat Github com a eina de gestió de versions, així tothom té la més recent i pot treballar i fer avenços des de aquesta.

5. Objectius individuals

L'objectiu principal d'aquest projecte és la creació d'un sistema classificador i un sistema recomanador.

El sistema classificador, basat en machine learning, és capaç de classificar el tipus de restaurant al qual pertany el ticket introduït. Per entrenar els diferents models és necessària una extracció, transformació i càrrega (ETL) de les dades per ser processades de tal manera que el model pugui fer la seva funcionalitat.

És essencial tenir un sistema classificador dins de l'aplicació perquè a partir d'això es poden agrupar els clients segons els seus gustos (útil per a la posterior confecció d'un sistema recomanador), s'obtenen resums de despeses per categories, permet mostrar estadístiques a l'usuari o personalitzar les ofertes o recomanacions als clients en relació amb els seus gustos; aquest és l'propòsit principal del sistema recomanador.

5.1 Objectiu analítica de dades

Gràcies a les dades adquirides per l'aplicació es té un volum d'informació important que s'ha d'analitzar. Es pot analitzar les dades des de tres vessants: *usuaris, restaurants i business.*

L'anàlisi des de el punt de vista dels usuaris permet tenir un control exhaustiu de les seves despeses. Pot saber quin ha estat el restaurant en el qual més ha menjat, quines han estat les despeses totals per tipus de restaurant.

L'anàlisi enfocat als restaurants, gràcies a aquest es pot ajudar als restaurants afiliats, els quals reben cada primer de mes un informe (Annex - Figura 8), en el qual venen els ratings de valoració, els ratings de valoració d'aliments, els ratings de valoració de servei, quantes persones han vingut al restaurant, quantes han repetit, una llista dels 5 plats més i menys demanats per als clients durant aquell mes, i d'aquesta manera el restaurant pot prendre decisions per fer el negoci més *customer-oriented* en funció de les dades rebudes de l'aplicació.

La part de business va relacionada a persones que volen obrir un restaurant i necessiten informació per a prendre una decisió. Un empresari vol obrir un restaurant italià a la zona de la Vila Universitària i necessita saber quants clients potencials té, quina competència té i com està valorada pels usuaris.

6. Classificador

S'ha desenvolupat el sistema classificador prenent com a base els diferents tipus de restaurants que els clients freqüenten. Tot i que hi ha altres opcions, a desenvolupar en un futur, l'elecció de fer la classificació sobre els tipus de restaurant és degut al fet que ens permet gestionar més nombres d'usuaris, interactuar tant amb els usuaris com amb els restaurants i disposar de més informació per al sistema recomanador.

Classificar els tiquets segons el grup d'establiment és essencial per a l'aplicació, ja que permet tenir molta informació per analitzar, i d'aquesta manera donar un millor servei als clients de l'aplicació, com per exemple la recomanació de restaurants a cada usuari, i feedback dels clients als restaurants afiliats.

Els algorismes de classificació s'utilitzen quan el resultat desitjat és una etiqueta discreta. És a dir, són útils quan la resposta a la pregunta s'allotja dins un conjunt finit de resultat finits, com pot ser la determinació de si un correu electrònic és spam o no. En aquest cas només es tenen dues opcions i es coneix com **classificació binària.** Per altra banda, la **classificació de múltiples categories** aconsegueix categoritzar imatges, àudios, analitzar textos.

En el cas de l'aplicació es necessita una classificació **de múltiples categories,** ja que hi ha molts tipus de restaurant a classificar, japonès, italià, francès, americà…

6.1 Realització model classificador

Les eines utilitzades per a poder desenvolupar el sistema classificador han estat Apache airflow[5] per als processos ETL i Python per al desenvolupament dels models.

Gràcies al procés ETL es poden obtenir les dades i sotmetre-les a un posterior tractament ***eliminant classes, random undersampling, cleaning text, vectoritzant el text i generant nous registres.*** Un cop les dades estan netes es creen una sèrie de models, els quals s'expliquen amb profunditat durant el projecte. Per desenvolupar els diferents models s'ha fet servir jupyter-notebook[6], els quals són sotmesos a una posterior avaluació, mitjançant una sèrie de mètriques per determinar quin és el model que millor s'adapta a les necessitats de l'aplicació i integrar-lo dins d'aquesta. L'avaluació consta de l'exploració i comparació de resultats, entre els diferents models, de les següents mètriques i tècniques: ***accuracy, recall, f1-score, rendiment de temps, confusion matrix.***

6.1.1 Flux funcional del sistema classificador dins l'aplicació

Quan un consumidor fa una fotografia al seu tiquet el que l'aplicació primer fa és mitjançant un algorisme OCR extreure les dades importants del tiquet, nom del restaurant, diferents elements consumits amb els seus respectius preus, hora i dia. Un cop extreta tota aquesta informació es procedeix a classificar el tiquet segons el tipus de restaurant en el qual s'acaba de consumir i gràcies a aquesta classificació els usuaris són agrupats segons els seus gustos; en base a això el model de recomanació adreça als usuaris una sèrie de restaurants.

6.1.2 ETL

S'ha dut a terme una recerca exhaustiva per internet i contactat amb diferents empreses, en la qual s'han aconseguit 4 datasets que contenen informació sobre restaurants, usuaris, ratings dels restaurants i plats dels restaurants. Fent una sèrie d'unions entre els datasets ha permès crear de forma coherent un dataset final per a utilitzar-lo de base de dades de prova de l'aplicació. (Annex -Figura 9)

Mitjançant un procés ETL s'obtenen les dades de la base de dades d'SplitIt, i es processen per aconseguir les més importants: **name, name_restaurant, ingredients,** les quals són les encarregades de proporcionar al model les característiques necessàries per a poder classificar de manera correcta. A més a més també s'agafa la columna **cuisine**, que és el tipus de cuina i la variable a classificar per part del model.

6.1.3 Preprocessament de les dades

El processament de dades és una etapa essencial del procés de descobriment d'informació. Aquesta etapa s'encarrega de la neteja de les dades, integració, transformació i reducció per a la següent fase del model classificador.

6.1.3.1 Eliminar classes

Aquest pas simplement elimina les classes que tenen registres més baixos.

Les classes amb poca freqüència mai seran predites pel model, però com són molt petites, l'efecte en terme de precisió general serà mínim.

Es procedeix a comptar els registres i es crea un threshold (100 registres) a partir del qual si el recompte està per sota, la classe s'elimina (Figura 10).

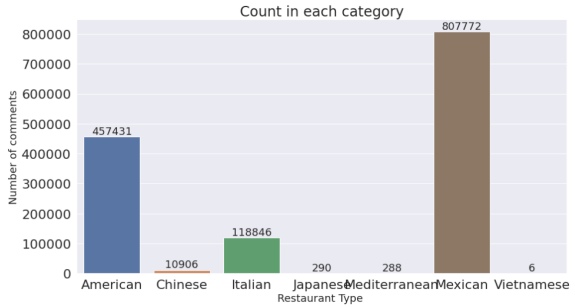


Figura 10: Recompte dels tipus de restaurant

6.1.3.2 Random undersampling

Random undersampling consisteix en seleccionar de manera aleatòria els exemples de la classe majoritària per eliminar-los del conjunt de dades d'entrenament. Això té efectes de reduir el nombre d'exemples de la classe majoritària en la versió transformada del conjunt de dades d'entrenament. Aquest procés pot repetir-se fins que s'aconsegueixi la distribució desitjada, com un nombre igual d'exemples per a cada classe.

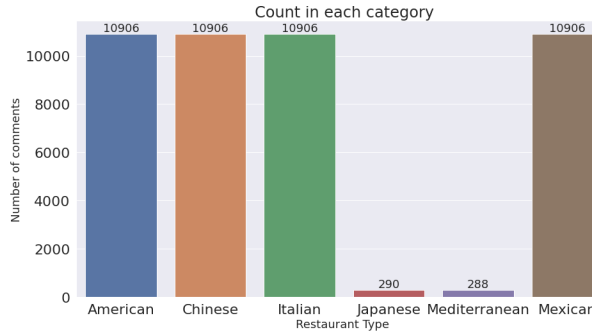


Figura 11: Random Undersampling

6.1.3.3 Cleaning text

Després de veure una mostra aleatòria de les dades del text, hi ha alguns elements específics que s'han de resoldre, així com aplicar alguns passos estàndard del processament del llenguatge natural: *eliminar caràcters especials com (|r, |n), tokenitzar el text, eliminar caràcters non-ascii, convertir el text en minúscules, eliminar puntuació i obtenir l'arrel de cada paraula.*

Tokenitzar el text: Consisteix essencialment en dividir una frase, oració, paràgraf o document de text complet en unitats més petites, com paraules o termes individuals. Cada una d'aquestes unitats més petites es denomina **token**.

Arrel de cada paraula: És el procés d'eliminar una part d'una paraula, o de reduir una paraula a la seva arrel.

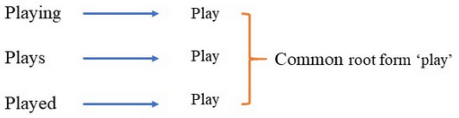


Figura 12: Procés de triar l'arrel.

6.1.3.4 Vectorització del text

Quan el text ja és net s'ha de vectoritzar, això significa codificar les paraules com números per al seu ús com a entrada a l'algorisme de machine learning, per fer la vectorització s'utilitza la llibreria **scikit-learn** té tres tipus de conversor de text a vector: *CountVectorizer, TfidfVectorizer i HashingVectorizer;* l'utilitzat en l'aplicació és **TfidfVectorizer.**

TF-IDF, és un mètode per calcular les freqüències de les paraules.

- **Terme Freqüència (TF):** Recompte de quantitat de vegades que cada paraula donada apareix dins d'un document.
- **Freqüència inversa de documents (IDF):** És una mesura logarítmica que indica si el terme és comú o rar en tots els documents.

TF-IDF són puntuacions de freqüència de paraules que tracten de ressaltar les paraules que són més interessants. **TfidfVectorizer** tokenitza documents, aprèn vocabulari, calcula les ponderacions inverses de freqüència de documents i permet codificar nous documents.

6.1.3.5 Generació de nous registres

L'últim pas per a les dades d'entrenament és utilitzar el mètode de sobremostreig SMOTE per augmentar el recompte de les classes amb poques entrades mitjançant la creació de valors sintètics basats en un K-Nearest-Neighbours (Figura 13). Després d'aquest últim pas les classes estarien equilibrades i les dades llestes per a la seva modelització.

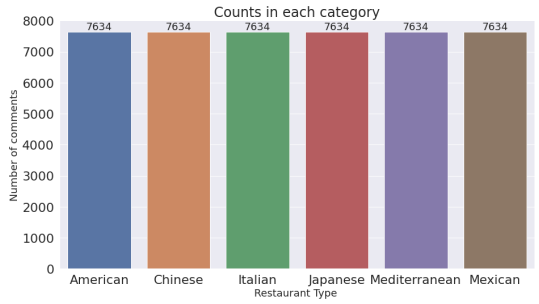


Figura 13: Classes equilibrades

6.2 Models

Després d'una exhaustiva investigació s'ha arribat a la conclusió que els següents models són els que millor s'adapten al problema esmentat anteriorment en el punt 6, classificar **múltiples categories.**

- **Naïve Bayes Clasification[1]: És** una família de simples classificadors probabilístics basat en l'aplicació de Bayes *teorema amb forts (Naïve) supòsits d'independència entre les característiques.*

$$P(A|B) = P(B|A) * P(A) / P(B)$$

- **SVC:** SVC (Support Vector Classifier) és un classificador basat en un hiperplà que, encara que no separi perfectament les classes, és robust i té una alta capacitat predictiva a l'aplicar-lo a noves observacions (menys problemes d'overfitting). Per fer-ho possible en lloc de buscar el marge de classificació més ample possible que aconsegueix que les observacions estiguin al costat correcte del marge; es permet que certes observacions estiguin al costat incorrecte del marge o inclús del hiperpla (Annex - Figura 14).
- **K-Nearest Neighbors[2]:** És un mètode que simplement busca, en les observacions més properes a la que s'està tractant, classificar el punt d'interès basat en la majoria de dades que l'envolten (Annex - Figura 15).
- **Decision Trees[3]:** En aquesta tècnica es divideixen les dades en dos o més conjunts homogenis basats en el diferenciador més significatiu en les variables d'entrada. Identifica la variable més significativa i el seu valor que proporciona els millors conjunts homogenis de població. Totes les variables d'entrada i tots els punts de divisió possibles s'avaluen i es tria el que millor resultat té (Annex - Figura 16).
- **Random Forest[4]:** S'executen diversos algorismes d'arbre de decisions en lloc d'un sol. Per classificar un nou objecte basat en atributs, cada arbre de decisió dona una classificació i finalment la decisió amb major "vots" és la predicció de l'algorisme. Pot gestionar milers de variables d'entrada i identificar les variables més significatives (Annex - Figura 17).

Tots aquests models son entrenats pel mateix dataset i gràcies a una sèrie de mètriques, el model que millor s'adapti a les necessitats de l'aplicació serà l'escollit.

	class	Accuracy
0	MultinomialNB	0.997285
1	SVC	0.997662
2	DecisionTree	0.903401
3	KNeighbors	0.994948
4	RandomForest	0.914788

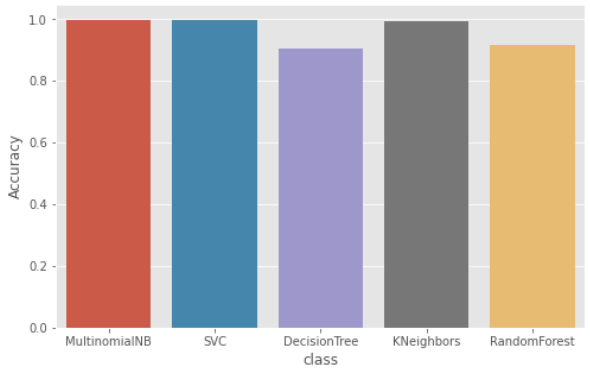


Figura 18: Comparació de models

Com es pot veure en la imatge anterior (Figura 18), hi ha tres models que clarament tenen una molt bona puntuació, ***SVC, MultinomialNB i KNeighbors***, amb un 99,76%, 99,72% i 99,49% de precisió respectivament. Com que hi ha tres models que clarament són molt bons pel que fa a precisió s'ha decidit realitzar una comparació de rendiment de temps, ja que trigar el menor temps possible en executar-se és un requisit valorat pels usuaris.

	class	time
0	MultinomialNB	0.163229
1	SVC	183.098894
2	DecisionTree	1.641165
3	KNeighbors	39.505436
4	RandomForest	0.414955

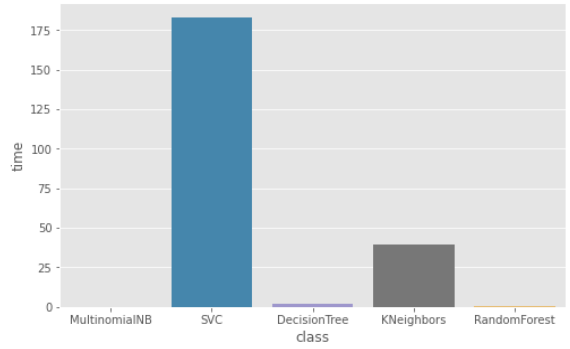


Figura 19: Comparació rendiment

Com es pot observar en la Figura 19, hi ha una gran diferència en el rendiment de temps entre els tres models que millors resultats d'accuracy donen, ***SVC, MultinomialNB, KNeighbors.*** El model basat en Multinomial Naive Bayes triga a executar-se 0,1632 segons, mentre que KNeighbors triga 39,50 segons i SVC 183.09 segons. Gràcies a aquesta anàlisi s'ha conclòs que el model que millor s'adapta als requisits de l'aplicació és el **MultinomialNB,** tot i no tenir el millor accuracy té una característica molt important per l'aplicació: **rapidesa.**

6.2.1 Avaluació del model

Un cop triat el model que millor s'ajusta als requisits, es procedeix a la seva avaluació. Per a realitzar-la s'utilitza el dataset de test, el dataset inicial es divideix en dos: dataset d'entrenament consisteix en el 80% de les dades i el dataset de test les dades restants.

L'avaluació consisteix en un conjunt de mètriques que donen certa informació sobre el model.

	precision	recall	f1-score	support
Italian	1.00	0.99	0.99	3272
Mexican	1.00	1.00	1.00	3272
Chinese	1.00	1.00	1.00	3272
American	1.00	1.00	1.00	87
Japanese	0.98	1.00	0.99	86
Mediterranean	0.99	1.00	0.99	3272
accuracy			1.00	13261
macro avg	0.99	1.00	1.00	13261
weighted avg	1.00	1.00	1.00	13261

Figura 20: Mètriques

Com es pot apreciar en la figura anterior (Figura 20) el model té un molt bon rendiment, quasi del 100%, cada classe té la seva puntuació de cada mètrica, d'aquesta manera es pot saber en quina classe s'equivoca més o menys.

Precision serveix per medir la qualitat del model. Ens indica el nombre de prediccions positives que ha realitzat: *True Positives / (True Positives + False Positives).*

Recall és una mesura que calcula quants dels casos positius ha predit correctament el classificador sobre tots els casos positius de les dades, *True Positives /(True Positives + False Negative).*

F1-score combina les mesures de precision i recall en un únic valor, *2 * ((Precision * Recall) / (Precision + Recall)).*

Además s'ha desenvolupat una **confusion matrix** (Figura 21)**,** la qual és utilitzada per avaluar el rendiment del model. La matriu compara els valors objectius reals amb els valors predits pel model. Això ens dona una visió holística de com de bé està funcionant el nostre model de classificació i quin tipus d'errors comet.

Com es pot observar en la figura següent el model emprat en l'aplicació té un molt bon rendiment. Es pot veure en quines classes el model no acaba de predir de manera correcta i s'equivoca, quan el model ha de predir **Italian** de 3272 mostres en classifica malament 33, les quals les classifica com a **Mediterranean,** la qual cosa té bastant de sentit, quan el model ha de classificar **Mediterranean,** d'un total de 3272 mostres en classifica malament 3, de les quals 1 la classifica com **Italian** i les altres dues com a **Japanese.**

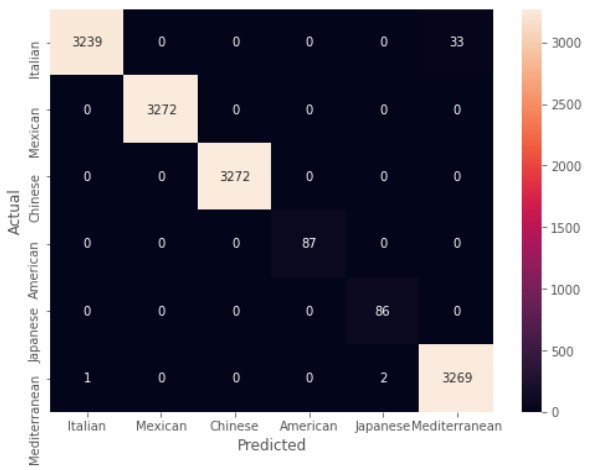


Figura 21: Confusion Matrix

7. Recomanador

Una característica distintiva que té l'aplicació és que és capaç de recomanar als clients quin tipus de restaurant han de tastar, gràcies al desenvolupament d'un sistema recomanador. Cal remarcar la importància del sistema classificador en el procés de recomanació, tal com s'ha explicat en l'apartat del classificador, ja que aquest classifica els restaurants visitats i permet al recomanador agrupar els usuaris segons els seus gustos i poder suggerir restaurants.

Hi ha diversos objectius per tal de justificar la utilització d'un sistema recomanador, ajudar als clients a reduir el temps de búsqueda d'un restaurant, proposar nous restaurants en funció dels seus gustos.

7.1 Sistema recomanador

Un **sistema recomanador** filtra les dades utilitzant diferents algorismes i recomana els ítems més rellevants per als usuaris. Primer capta el comportament d'un client en el passat i, en base a això, recomana productes que els usuaris podrien consumir.

Hi ha molts tipus de sistemes de recomanació, però pel nostre cas el que millor encaixa és un tipus de **Collaborative Filtering**. Fa servir la informació de 'masses' per identificar perfils similars i aprendre de les dades per recomanar productes de manera individual.

- **User-based**[7]**:** S'intenta trobar usuaris similars per a oferir items 'ben valorats' per a aquell perfil en concret. Es parla de buscar similitud entre gustos de l'usuari sobre aquests ítems, no es busquen perfils similars per ser del mateix sexe, edat o nivell educatiu. Només s'utilitzen els ítems que s'han experimentat i valorat per agrupar usuaris "semblants". La predicció d'un restaurant *u* es calcula mitjançant la suma ponderada de les classificacions d'usuaris atorgats a un restaurant *i.* La predicció *Pu, i* és definida per:

$$P_{u,i} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}}$$

On:

Pu,i és la predicció d'un restaurant

Rv,i és la classificació donada per un usuari *v* a un restaurant *i*

Su,v és la semblança entre els usuaris

Una vegada obtinguda la classificació pels usuaris en un vector de perfil hem de predir les classificacions per altres usuaris. Es segueixen els següents passos per aconseguir-ho:

1. Per les prediccions necessitem la semblança entre l'usuari *u* i *v.*
2. Trobem els restaurants classificats pels usuaris i, segons les classificacions, es calcula la correlació entre els usuaris.
3. Les prediccions es calculen utilitzant els valors de semblança. Aquest algorisme, en primer lloc, calcula la semblança entre cada usuari i, després basant-se en cada semblança, calcula les prediccions. Els usuaris que tenen major correlació tendeixen a ser més similars.
4. Sobre la base d'aquests valors de predicció, es fan les recomanacions.

7.2 Realització del model recomanador

El llenguatge de programació per desenvolupar el sistema recomanador ha estat Python, i en concret dos de les seves llibreries, *surprise*[8], la qual ha sigut molt útil per la realització del model **user-based**, i *SQLAlchemy*[9], que permet connectar-se amb la base de dades i fer un conjunt de crides per obtenir les dades necessàries, com per exemple els usuaris, quins restaurants han visitat, quins plats s'han consumit i els seus preus, la localització del restaurant, ja que s'ha de recomanar un restaurant que estigui relativament a prop de la zona en la qual el client es troba, quin tipus de restaurant és, com ja s'ha dit al principi d'aquest apartat aquesta informació vindrà donada pel sistema de classificació explicat amb anterioritat, els ratings que han donat els clients als restaurants, al seu menjar i al seu servei.

L'escalabilitat del model recomanador és una característica molt important, ja que d'aquesta manera l'usuari no s'ha d'esperar molta estona per veure les recomanacions de l'aplicació. En la figura següent (Figura 22) s'ha dut a terme una comparació de rendiment del recomanador per diferents nombres d'usuaris i ratings. Com es pot observar, a mesura que augmenten els usuaris i els seus ratings, el temps d'espera augmenta considerablement, per tant, no és viable introduir el sistema recomanador com a tal dins l'aplicació, sino executar-lo diàriament a la nit, ja que és quan hi ha menys activitat a l'aplicació, i guardar els resultats en una taula dins la base de dades, a la qual se li faran les crides necessàries per obtenir les recomanacions.

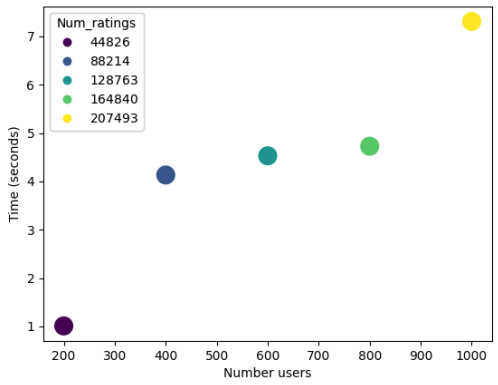


Figura 22: Comparació de rendiment en funció d'usuaris i ratings

El sistema recomanador esmentat està integrat dins l'aplicació i un cop el tiquet és introduït aquesta el classifica i seguidament a l'apartat **statistics** apareix una llista de 5 restaurants recomanats, podent el client triar el que més li crida l'atenció i guardar-ho en una llista de desitjos.

Si es va al annexe, a la figura 23, es pot veure els restaurants recomanats per l'aplicació per a un client en concret.

8. Informe de resum de l'activitat de l'usuari

Gràcies al sistema classificador els tiquets estan agrupats segons al tipus de restaurant al qual pertanyen, d'aquesta manera es poden dur a terme una sèrie d'anàlisis sobre les dades del client com s'ha mencionat amb anterioritat.

En les imatges següents (Annex - Figura 24 i 25), es poden observar dues gràfiques amb el percentatge de vegades que el client ha visitat cada tipus de restaurant i un altre sobre el percentatge resultant de la quantitat de diners gastats per a cada tipus d'establiment.

Es pot afirmar que al client clarament li agrada molt menjar en restaurants Mexicans, per tant, el més probable és que el sistema recomanador li proposi restaurants mexicans. Per altra banda es pot observar que tot i que ha visitat el mateix percentatge de vegades els restaurants de tipus americans i japonesos, el percentatge de despesa en restaurants japonesos és més del doble que en el d'americans.

9. Conclusions

En aquest projecte s'ha desenvolupat un sistema classificador de tiquets, amb una adquisició prèvia de les dades gràcies a un procés ETL, un sistema recomanador per a millor l'experiència del client amb l'aplicació, a més a més d'una anàlisi de les dades de l'usuari. Per tant, es pot afirmar que els objectius individual esmentats en el punt 5 s'han complert. El desenvolupament del sistema classificador s'ha centrat en només un àmbit, els restaurants. Seria interessant en un futur explorar altres àmbits, com pot ser supermercats, benzineres…

Referències

[1]https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[2]https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/

[3]https://www.sciencedirect.com/topics/computer-science/decision-tree-classifier

[4]https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[5]https://betterdatascience.com/apache-airflow-postgres-database/

[6] https://arxiv.org/pdf/1804.05492.pdf

[7]https://www.aprendemachinelearning.com/sistemas-de-recomendacion/

[8]http://surpriselib.com/

[9]https://towardsdatascience.com/sqlalchemy-python-tutorial-79a577141a91

Annex

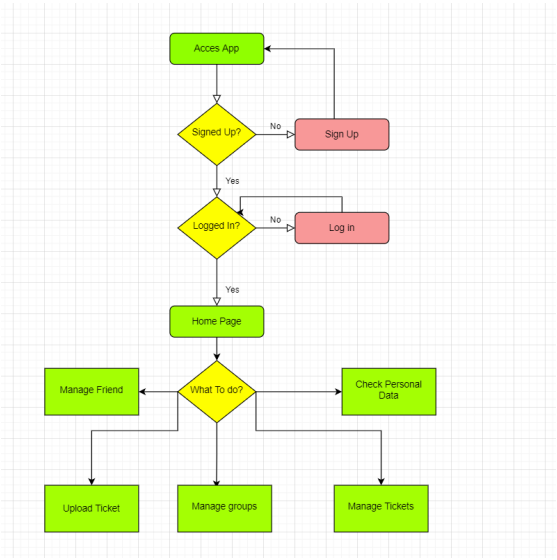


Figura 2: Diagrama de flux funcional de l'aplicació

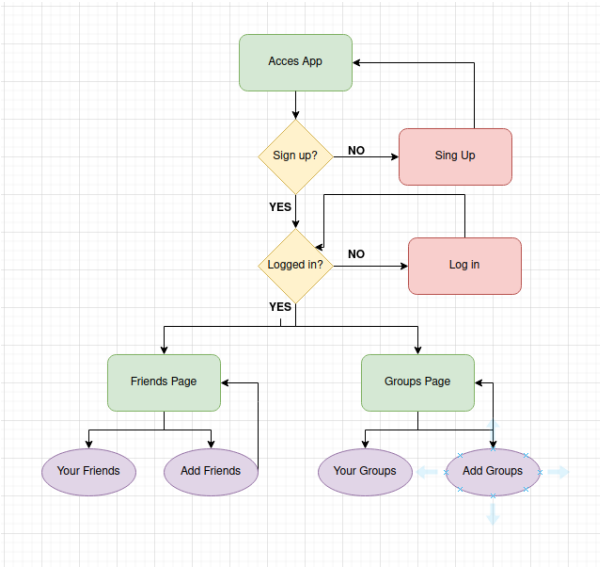


Figura 3: Diagrama de flux registre a l'aplicació

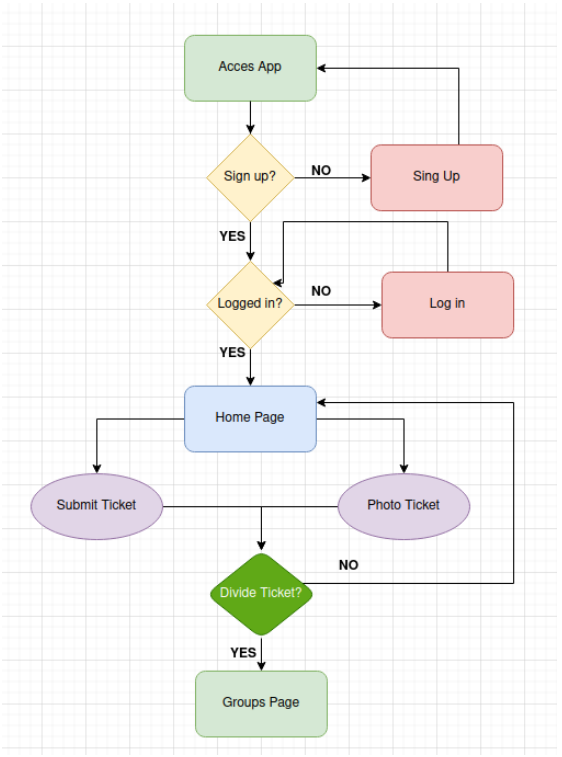


Figura 4: Diagrama de flux de pujar el ticket a l'aplicació

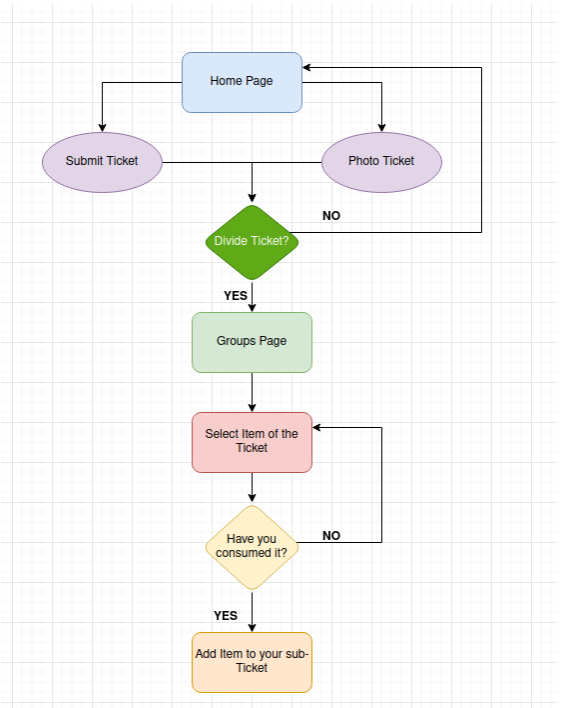


Figura 5: Diagrama de flux de mostra de tickets personals

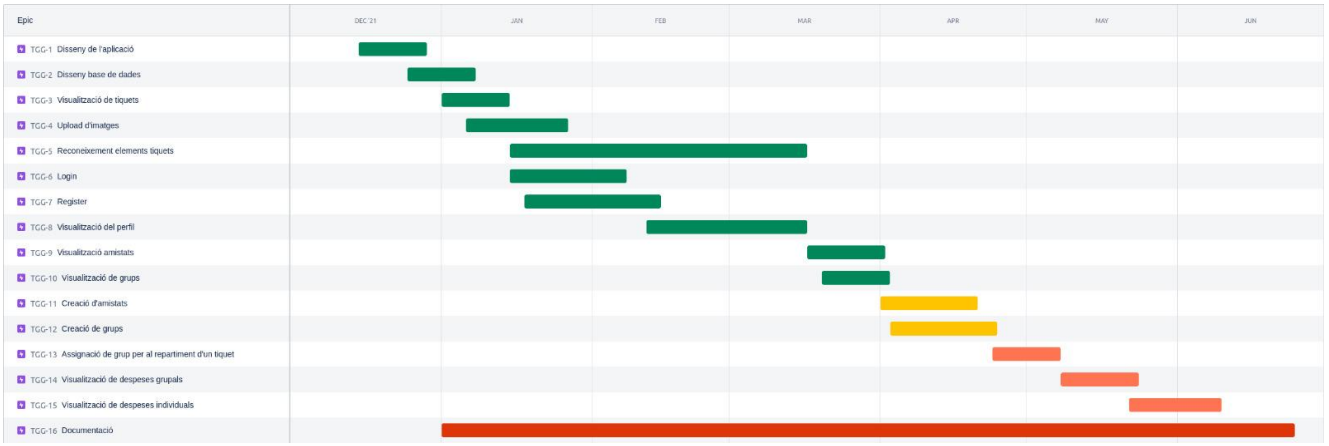


Figura 6: Diagrama de Grantt comú

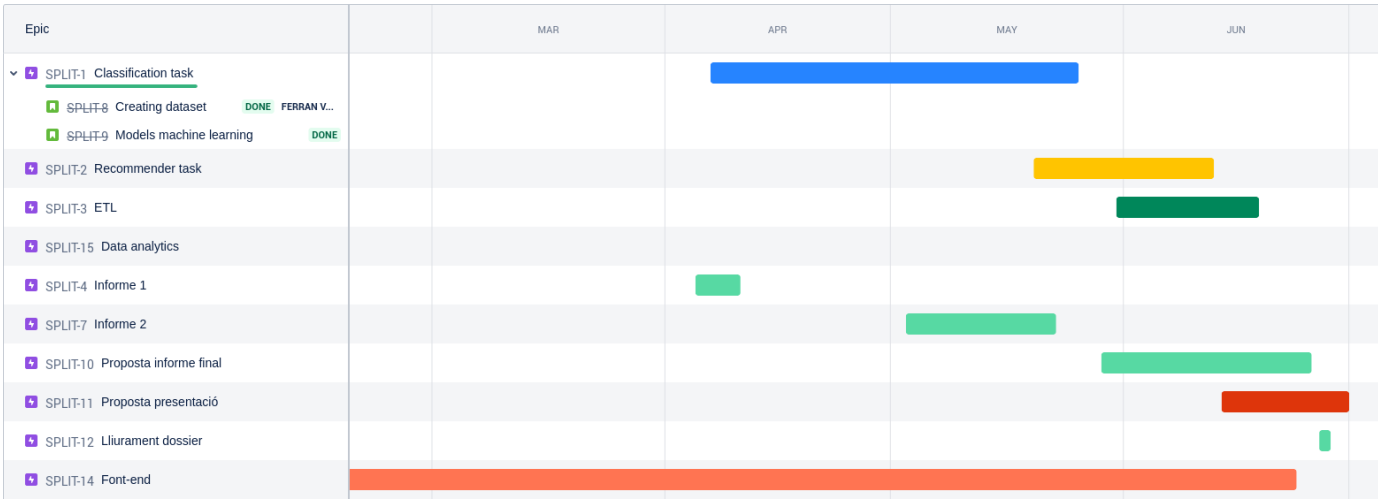
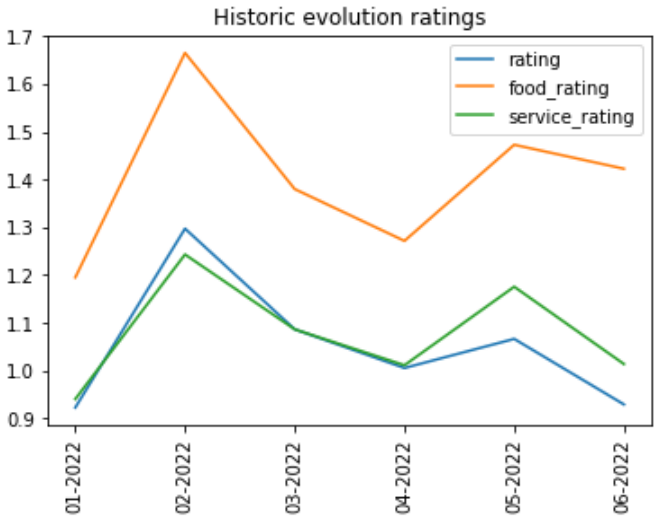


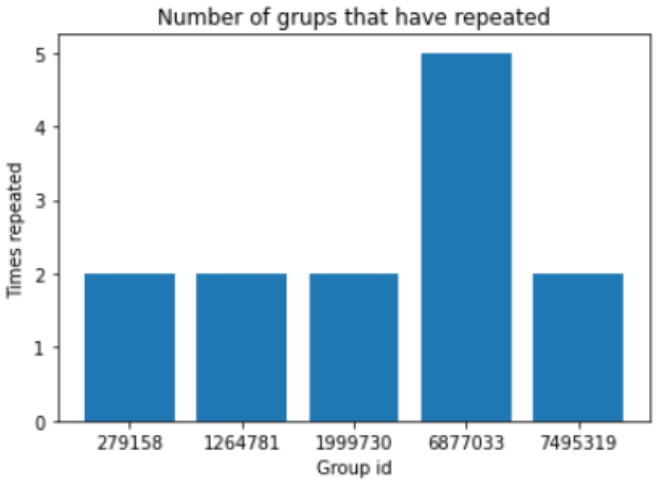
Figura 7: Diagrama de Gantt individual

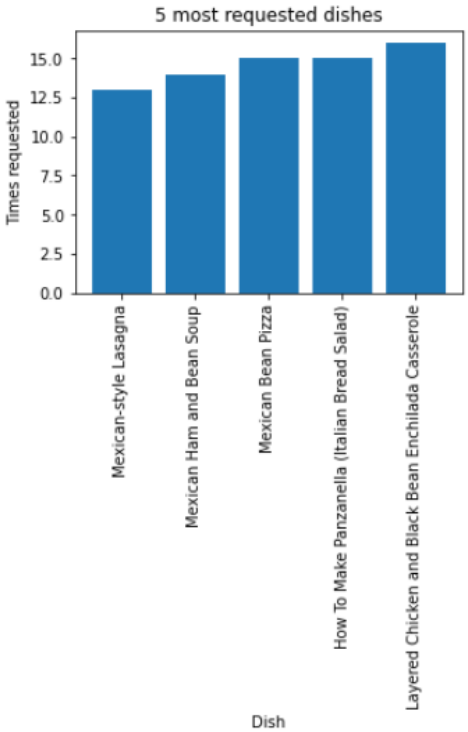
Informe pel restaurant: *Gorditas Dona Tota*
Mes: *01-05-2022*

```
RATINGS
Mean rating:  1.0891089108910892
Mean food_rating:  1.5693069306930694
Mean service_rating:  1.1782178217821782


Total number of groups that visited the restaurant:  25
```







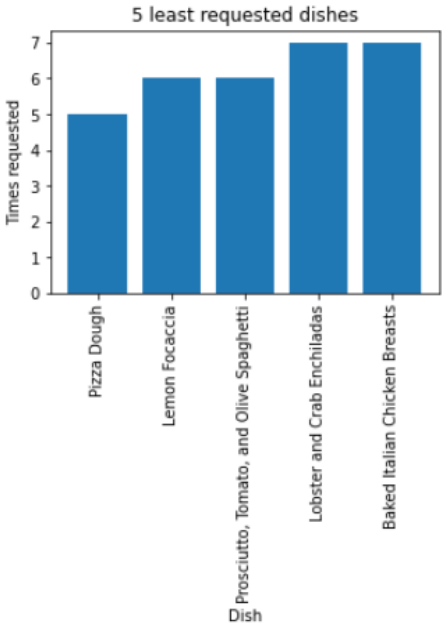


Figura 8: Informe mensual per als restaurants afiliats

	placeid	address	city	state	country	name_restaurant	course	cuisine	ingredients	name	userid	rating	food_rating	service_rating
0	134999	Revolucion	Cuernavaca	Morelos	Mexico	Kiku Cuernavaca	Soups	Japanese	['8 ounces uncooked udon noodles (thick, round...	Udon-Beef Noodle Bowl	U1093	2	2	2
1	134999	Revolucion	Cuernavaca	Morelos	Mexico	Kiku Cuernavaca	Soups	Japanese	['8 ounces uncooked udon noodles (thick, round...	Udon-Beef Noodle Bowl	U1066	1	1	1
2	134999	Revolucion	Cuernavaca	Morelos	Mexico	Kiku Cuernavaca	Soups	Japanese	['8 ounces uncooked udon noodles (thick, round...	Udon-Beef Noodle Bowl	U1040	1	1	1
3	134999	Revolucion	Cuernavaca	Morelos	Mexico	Kiku Cuernavaca	Soups	Japanese	['8 ounces uncooked udon noodles (thick, round...	Udon-Beef Noodle Bowl	U1110	2	2	2
4	134999	Revolucion	Cuernavaca	Morelos	Mexico	Kiku Cuernavaca	Soups	Japanese	['8 ounces uncooked udon noodles (thick, round...	Udon-Beef Noodle Bowl	U1121	2	2	2

Figura 9: Dataset Final

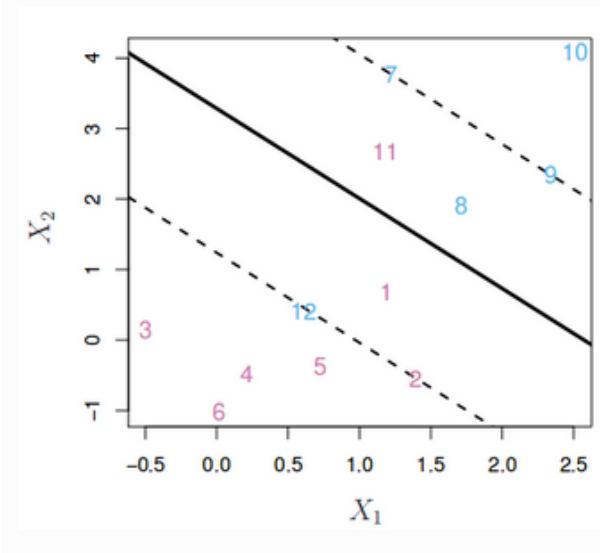


Figura 14 : Esquema SVC

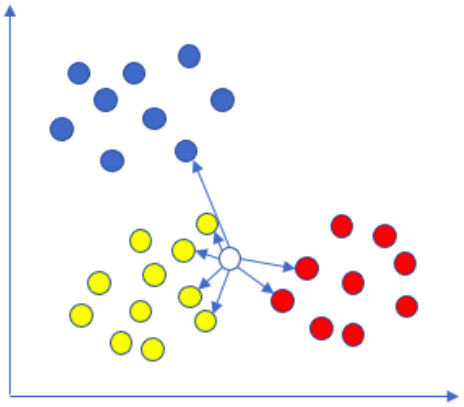


Figura 15 : Esquema KNN

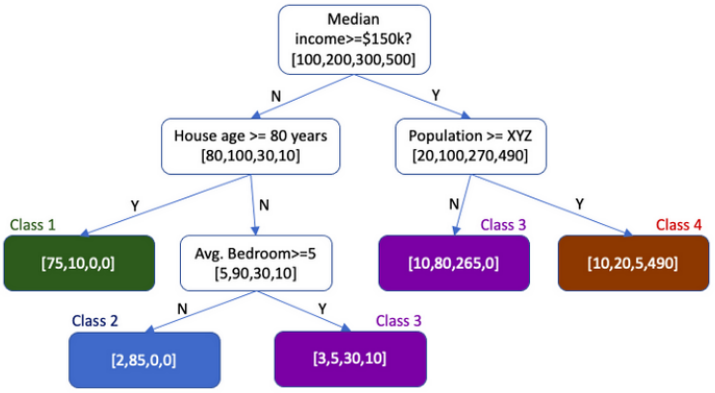


Figura 16 : Esquema Decision Tree

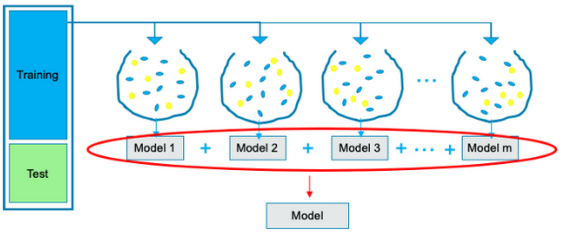


Figura 17 : Esquema Random Forest model

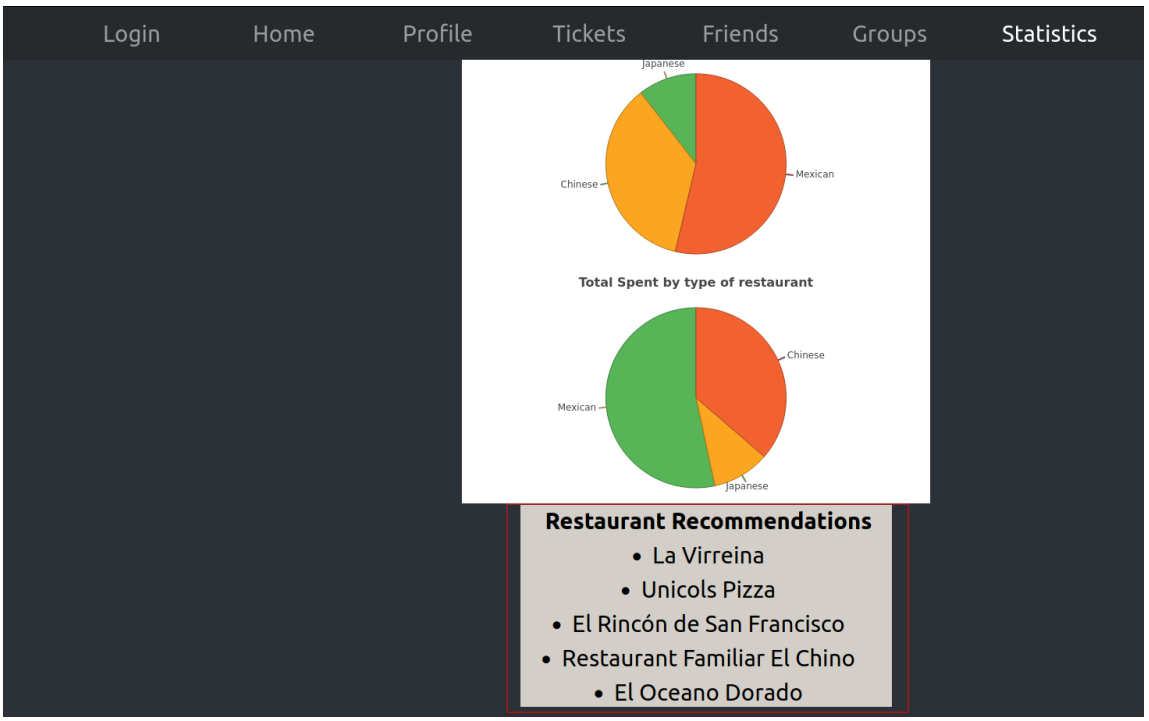


Figura 23 : Recomanacions de restaurants

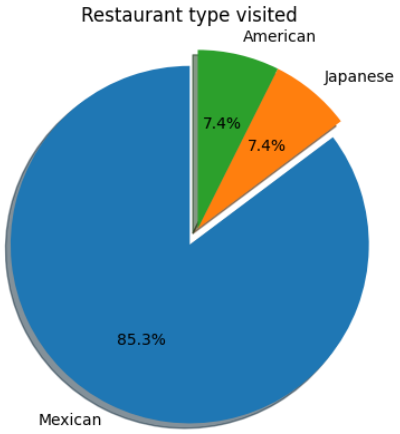


Figura 24 : Percentatge de tipus de restaurant visitat

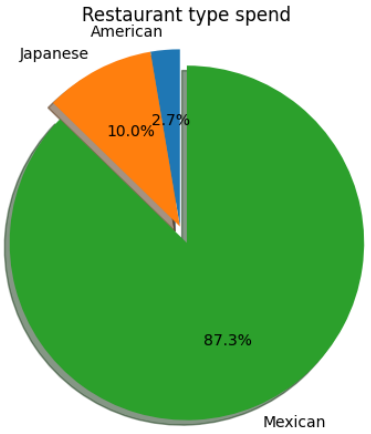


Figura 25 : Percentatge costos per tipus de restaurant