Classificació de Jocs; Detecció de Ludopatia

Gerard Burgués Llavall

Resum– La ludopatia és l'addicció patològica als jocs d'atzar i es considera una malaltia caracteritzada pel fracàs crònic i progressiu. Un 1% de la societat de l'estat espanyol la pateix. Aquest article té el propòsit d'esbrinar el comportament dels usuaris per tal de detectar a totes aquelles persones que poden estar patint ludopatia de forma inconscient. Aquest projecte comença mostrant diferents models de classificació de jocs fent ús de la descripció i instruccions de cada joc. Seguidament, es mostraran diverses maneres de predir la quantitat de diners que un usuari es gastarà al següent mes. Amb aquest procés, experts seran capaços d'identificar si els usuaris estaran creant un patró desenvolupant així una addicció.
Paraules clau– Python, Snowflake, Aprenentatge Automàtic, Classificació de text, Predicció de costos, Jocs online, Ludopatía

Abstract– Gambling addiction is a pathological addiction to gambling and is considered a disease characterized by chronic and progressive failure. 1 % of Spanish society suffers from it. This article aims to find out the behavior of users in order to detect all those people who may be unconsciously suffering from gambling . This project begins by showing different models of game classification using the description and instructions of each game. In addition, will be presented multiple ways to predict how much money a user will spend next month. With this process, experts will be able to identify if users are creating a pattern thus developing an addiction.

Keywords– Python, Snowflake, Machine Learning, Unsupervised Text classification, expenses prediction, online games, Gambling addiction

1 Introducció - Context del treball

ACTUALMENT la ludopatia forma part del dia a dia de moltes persones i s'ha de trobar la manera de combatre-la i preveure aquells casos en què desenvolupa una addicció. A continuació es pot observar en la figura 1 la representació de la quantita de transaccions realitzades cada any dins de l'aplicació slot.com .

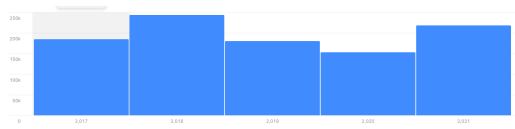

Fig. 1: Quantitat de transaccions per any

Podríem resumir que de mitjana s'han realitzat 200.000 transaccions. Assumin que hi han 15.000 usuaris actius, s'ha pogut analitzar que el 15% d'aquests gasten més de 1.000€ l'any. Mentre que un 0.4% gasten més de 10.000€ l'any, aportant unes despeses de 100.000 € aproximadament. Aquestes dades són extretes del departament de Slot.com dins la companyia Cirsa. Formen part d'una petita mostra del que passa dins l'estat espanyol.

Analitzant aquests números podem concloure que la addicció al joc es troba de forma constant dins la nostra societat ja sigui en jocs online, casinos, bingos, etc.

Fent ús de l'aprenentatge automàtic podem extreure informació dels diners que un usuari gastarà en un futur i detectar amb previsió el seu comportament davant del joc. Tenir coneixement del tipus de joc que l'usuari juga també és important, per això, es mostrarà una classificació dels diversos jocs utilitzant *Natural Language Processing* (NLP).

Per conseguir un anàlisi dels jocs i dels usuaris s'utilitzarà el llenguatge de programació Python per la realització dels models i SQL per la extracció i neteja de les dades.

• E-mail de contacte: gerard.burgues@outlook.es
• Menció realitzada: Computació
• Treball tutoritzat per: Jordi Gonzalez Sabaté (Computer Science)
• Curs 2021/22

2 ESTAT DEL ART

Els jocs online poden constituir un perill si no s'aporten eines a la societat per poder afrontar possibles dependènices [11]. Aquest document reafirma el plantejament que s'ha fet per aquest projecte. Hi ha una necessitat d'identificar quins són els usuaris que poden estar patint una addicció i ajudar-los.

Dins de la classificació a partir d'un text podem trobar diversos estudis que utilitzen l'aprenentatge supervisat per tal de crear una classificació, en el nostre cas però, ens interessa un model que utilitzi un aprenentatge no supervisat. Una possible opció seria fer ús de classes predefinides per tal d'aplicar algorismes com *TF-IDF* per l'extracció dels valors de les paraules dins dels texts i *Naive Bayes* per la classificació d'aquests . En primer lloc, es realitza la creació de diverses etiquetes on cada una d'aquestes inclouran diverses paraules que defineixen la categoria. En fer això, ja es pot considerar que fan servir un model *mid-supervised* [12]. De la mateixa manera, hi han altres mètodes que segueixen un procés semblant. Per la definició de classes s'utilitzen algorismes com Glove o Word2Vec per entrenar el model. En el cas del Glove utilitzen un model preentrenat amb dades de la Wikipedia2014 i Gigaword5. Mentre que quan es realitza l'entrenament del Word2Vec s'utilitzen tots els texts preprocessats i ajuntats com un gran *corpus* [13].

D'altre banda la regressió per predicció dels usuaris és un dels temes més importants en tots els sectors. Utilitzant un algorisme d'arbre de decisions (CART) i amb dades com el temps de joc, la quantitat de diners guanyats, la quantitat de perdúes i la quantitat de diners invertits afecten durant un periode de temps determinat [14].

Contrariament, podem trobar articles [15] que indiquen que models de forecasting com *Prophet* poden mostrar una tendència dels usuaris a partir del temps i de la etiqueta. Fer un model de forecasting et pot permetre a analitzar el comportament dels usuaris a partir del temps. Prophet et permet afegir variables per donar-li algunes característica de les dades (en el temps).

Amb tota aquesta informació s'ha decidit implementar varis algorismes per la classificació de jocs. Es farà una comparació i s'obtindrà una conclusió. D'altra banda, s'implementaran dos tipus de models de predicció. El primer, serà un model supervisat tenint en consideració les diverses característiques dels usuaris. El segon, consistirà en un model de *Time Series* (Prophet) per tal de d'esbrinar la tendència dels usuaris en el joc a partir del temps.

3 OBJETIUS

Podem dividir el projecte en dos subprojectes: Classificació de jocs i Predicció del comportament dels usuaris.

3.1 Classificació de Jocs

1. Entendre el funcionament de la web (slot.com) amb la intenció d'analitzar els tipus de jocs que podem trobar.

2. Realitzar l'extracció de les dades mitjançant SQL Queries per tal de fer el processament de les dades i poder plantejar diversos models no supervisats de classificació.

3. Creació d'un algorisme on s'aplicaran diversos algorismes per tal d'aconseguir una o més categories per cada joc.

4. Proporcionar una classificació final de cada joc. Aquesta consistirà d'una taula amb dues columnes. La primera correspondrà al nom del joc i la segona contindrà una llista de les classes en què pertany.

3.2 Predicció comportament dels usuaris

- Entendre el funcionament de la web (slot.com) amb la intenció d'analitzar quines accions poden fer els usuaris per guanyar monedes i passar els nivells del joc.

- Realitzar un anàlisi mensual de les compres que els usuaris generen. Aquest ajudarà a entendre els diversos tipus d'usuaris que podem trobar depenent de la quantitat de diners que gasten cada mes.

- Creació d'un dataset fent ús de diverses taules extretes de la base de dades. S'haurà de realitzar una combinació de SQL Queries preguntant per dades especifiques dels usuaris.

- Creació de dos models d'aprenentage computacional i analitzar quin es el model que més ens convé per aprendre el comportament de l'usuari.

- Proporcionar un predicció de la quantitat de diners que gastarà un usuari el següent mes.

A fi de complir els objectiu s'ha utilitzat la metodologia scrum per tal de treballar de forma constant i evitar imprevistos. Cada dos setmanes s'ha presentat la feina desenvolupada fins al moment per tal d'aconseguir feedback i implementar millores. Per la realització de codi s'ha utilitzat Databricks que ja te incorporat Git pel control de versions.

4 CLASSIFICACIÓ DE JOCS

En aquest apartat es mostraran els diversos algorismes que s'han desenvolupat per tal de fer la clusterització de jocs mitjançant text.

Les llibreries que s'han utilitzat per la classificació de jocs són les següents: *NLTK*, *Pandas*, *Sklearn*, *Numpy*, *WordCloud*, *matplotlib.pyplot*

4.1 Dades

Per tal de crear models de NLP es necessiten diverses dades, aquestes s'obtindran d'Snowflake (eina per emmagatzemar dades). D'aquí, s'extrauran les següents columnes: Nom, Descripció i Instruccions de cada joc. Amb aquests tres camps ja es podrà duu a terme el plantejament i desenvolupament dels algorismes.

4.2 Preprocessament

Previ a realitzar els algorismes s'han realitzat una serie de processos per tal de netejar el contingut dels text.

1. Eliminar tots els elements html.
2. Convertir totes les paraules en minúscules.
3. Tokenize: Separar les paraules en tokens. En el nostre cas cada paraula serà un token
4. Eliminar "stopwords ". Paraules com: the, an, a, etc.
5. Eliminar paraules especials que no ens interessen.

A més s'ha fet un petit anàlisi de les paraules més comunes entre totes les descripcions dels jocs. En el núvol de paraules de la figura 2 s'observa que hi han bastantes paraules que apareixen constantment en tots els texts. Aquestes paraules fan referencia a jocs de casinos i bingos. Aquest anàlisi ha sigut important per posteriorment entendre els resultats dels models.

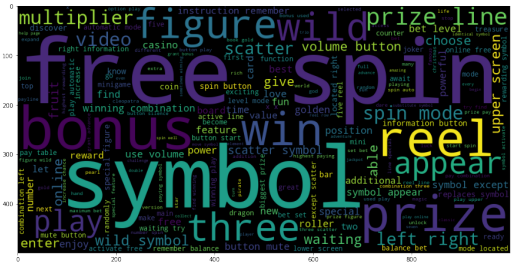

Fig. 2: Representació paraules més repetides

4.3 Algorismes

Abans de crear els diversos models primer s'ha de realitzar un pas previ. Aquest consisteix en el processament de les dades, es a dir, netejar el text per tal de que els models entenguin el contingut.

4.3.1 TF-IDF

TF-IDF (Term Frequency times Inverse Document Frequency) medeix amb quina freqüència un terme o frase apareix dins d'un document determinat i el compara amb el número de documents. Un cop s'ha obtingut el dataframe amb el valords de la paraula per cada text s'ha aplicat Kmeans per trobar els diferents clústers. Per saber el número òptim de classes s'ha utiltiza la regla del colze. Per fer una bona comparació també s'ha fet el mateix procés però previ a la crida de Kmeans s'ha realitzat una reducció de la dimensió utilitzant PCA i/o T-SNE. Es poden veure les regles dels colzes a les figures 13 i 14. A la figura 3 s'observa que no hi han clústers molt definits. Tot i que el plot no es gaire representatiu al fer l'anàlisi de texts per categories es pot veure que no hi ha una definició concreta de grups de jocs.

4.3.2 Bag of Words

BoW (Bag of Words) crea una representació del text que descriu el nombre de cops que una paraula apareix dins d'un document. Per tant, s'ha tingut en compte el nombre de

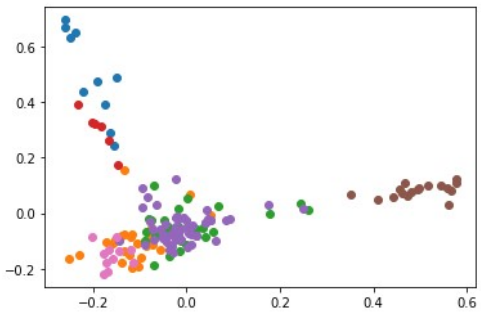

Fig. 3: Kmeans TF-IDF

cops que una paraula apareix sense tenir en consideració la seva gramàtica. Com s'ha fet anteriorment amb TF-IDF s'ha dut a terme Kmeans per aconseguir els diferents clústers dels jocs. S'ha buscat el nombre de clústers aplicant la regla del colze fent ús de reducció de dimensionalitat (PCA). Es pot observar a la figura 4 que s'obté resultats similars als de TF-IDF i, per tant, es pot concloure que aquests resultats poden ser donats per dos problemes:

Per una banda, falta de dades, és a dir, no tenim un text molt ampli ni tenim un gran dataset. D'altra banda, els texts tenen conceptes similars i per això no es pot apreciar gran diferència en crear els clústers.

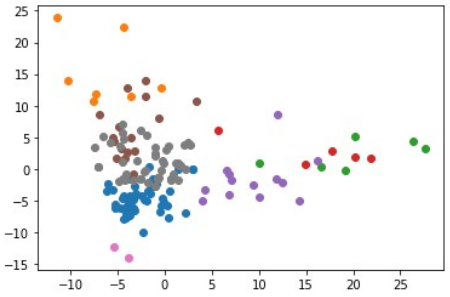

Fig. 4: Kmeans Bag of Words

4.3.3 Word2Vec

Word Embedding és una tècnica on es transforma la paraula (string) en format vector. Aquest vector està après d'una manera semblant a com ho fan les Xarxes Neurals. El vector intenta agafar diverses característiques de la paraula sobre el text total.

Aquestes característiques poden ser des de relacions semàntiques, definicions, context, etc. Amb aquests vectors es poden buscar similituds entre les paraules. Per tant, agrupar aquells texts que tinguin paraules similars. Per tal que aquest procés funcioni s'han eliminat paraules clau que formen part del 90% dels texts del dataset. Per tant, paraules com: "symbol", "reel", spin", "win", etc. No formaran part del nostre dataset. Aplicant una sèrie de processos podem veure la distància que estan les paraules entre elles a la figura 15. La figura 15 m'ha ajudat a veure quines són les paraules més influenciables dins dels texts. Aquest gràfic m'ha servit per a diversos algorismes que s'explicaran en les següents seccions: section 4.3.4 i section 4.3.5

Abans de realitzar la reducció de dimensionalitat i Kmeans s'ha hagut de fer un pas per tal d'obtenir un vector de 100 dimensions per cada text. Aquest pas s'ha aconseguit fent la suma de totes les paraules en un text i normalitzant el resultat.

S'ha de tenir en compte que ja no ens interessa Kmeans sense reducció de dimensionalitat, ja que la regla del colze mai ens dona una solució exacta. Un cop tenim Kmeans podem observar a la figura 5 que el resultat ha millorat bastant respecte als models anteriors. En fer un anàlisi de paraules en cada clúster es podria decidir que el model ho fa suficient bé per acceptar-lo, tot i això s'han trobat 2 possibles models més que poden millorar el resultat.

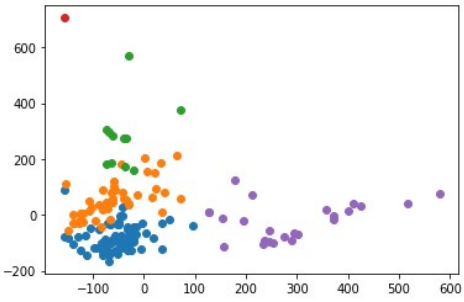

Fig. 5: Kmeans Words2Vec

4.3.4 Glove

Com també fa Word2Vec, s'han calculat els vectors de 100 dimensions utilitzant un model preEntrenat Glove [5] on per cada text es sumaran tots els vectors de les paraules i es farà una normalització grupal de tots els texts. La diferencia més gran que trobem entre Word2Vec i GloVe es que aquesta primera és un model predictiu mentre que GloVe és un çount-based"model, és a dir, conta la freqüència que veiem una paraula en un context dins d'un gran text.

Amb glove s'ha agafat el següent model preentrenat *glove_6B_100d*. Amb aquest model i eliminant paraules poc rellevants per la diferenciació dels texts hem aplicat el mateix procés de vectorització i classificació dels texts.

Un cop s'aplica Kmeans i fent ús de Plotly s'ha pogut mostrar una representació molt més exacta dels resultats. Aquest resultat s'ha pogut apreciar a la figura ...

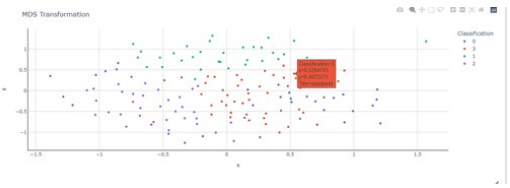

Fig. 6: Kmeans Glove

4.3.5 Label2Vec

Lbl2Vec és pot considerar un algorisme *semi-supervised* i el seu objectiu és buscar per cada text quina és la distància d'aquell respecte a les diferents classes prèviament creades.

El primer pas com diu la introducció d'aquest algorisme és la creació de diverses classes mare. En aquest cas es creen 5 classes amb les respectives paraules que la defineixen:

- Food → Banana, Juice, Lemon, Grape, Fruitful, Food, Cake, Farm, etc.
- Fantasy → Ogre, Elf, Dragon, Hobbit, Magic, Spell, Witches, Wit, etc.
- History → Egyptian, Pharao, Rome, God, Trojan, Cleopatra, Temple, etc
- Asian Culture → Asian, Japanese, Sakura, Oriental, Korean, Arabic, Shogun, Geisha, etc
- Animals → Animal, Dog, Cat, Horse, Zebra, Bear, Fish, Parrot, etc

Seguidament, s'utilitza Glove per la definició de vectors. Glove s'utilitza per tenir un dataset més gran i, per tant, poder tenir una millor representació dels vectors de 100 dimensions.

Com s'observa a la figura 16 , la distancia euclidiana no es representativa per diferenciar els jocs. Aquesta no té en compte la longitud dels texts i, per tant, tots els texts que tinguin longituds semblants estaran en posicions molt pròximes. Per resoldre aquest problema s'utilitza Cosine Similarity.

Un cop tenim els vectors de cada text es mostra en un gràfic 17 els diferents jocs respecte les etiquetes que s'han generat al principi del programa.

A la figura 17 ens mostra la distància que hi ha d'un joc entre cada una de les classes predefinides. A continuació es pot observar un exemple d'un joc en concret.

El que ens indica la figura 7 es la distancia que hi ha d'un joc entre cada un de les classes predefinides.

Per tant, en l'exemple anterior es pot veure que el joc Magic_shope hauria de ser de la classe fantasia perquè és el valor més alt.

Per evitar problemes s'aplica una sèrie de llindars per saber a quantes classes un joc pertany. És a dir, un joc pot formar part com a màxim de 5 classes, però també pot no formar part de cap, això dependrà dels llindars creats.

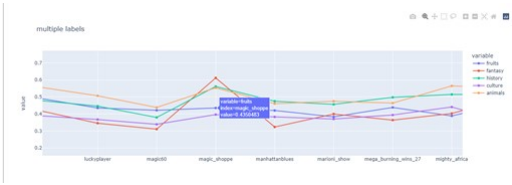

Fig. 7: Distancia Jocs entre Classes

4.4 Anàlisi de la classificació

Per tal de realitzar models de classificació amb dades no supervisades és complicat. Sempre que es vulgui implementar aquest tipus de models s'ha de tenir en compte que el percentatge d'error és més elevat i que es necessita una gran quantitat de dades.

En el cas presentat s'han pogut veure millores significatives quan s'ha aplicat un model preentrenat (Glove). En aquest s'han vist classificacions més precises, això si, s'han hagut d'eliminar les paraules més repetides per tal d'obtenir resultats realment acceptables.

Tanmateix, utilitzant un model mid-supervised s'ha assolit una classificació millor de l'esperada. Creant una combinació de Glove i Lbl2Vec es pot saber si un joc pertany a cap, una o més categories, mentre que els altres models no ens permetien fer aquesta visió.

5 Predicció del comportament dels usuaris

En aquest apartat es mostraran els dos algorismes que s'han implementat per tal d'esbrinar quants diners cada usuari gastarà al següent mes.
Les llibreries que s'han utilitzat per la classificació de jocs són les següents: *Pyplot*, *Pandas*, *Sklearn*, *Numpy*, *Seaborn*, *Pyspark*,*Scipy*,*pickle* ,*Prophet*

5.1 Dades dels Usuaris

Les dades són extretes de la base de dades d'Snowflake. S'han utilitzat majoritàriament dues taules principals:

- Taula Shopoffers: Aquest conté tots els pagaments que han fet els usuaris. Ens dona informació sobre el dia, la quantitat de diners gastats, el tipus de compra (oferta, tenda o promoció), la quantitat de monedes comprades, etc. D'aquesta taula s'han obtingut fins a un any i mig d'informació (2020-2022)
- Taula Sessions: Aquesta conté la informació del moviment de l'usuari en el joc, és a dir, el temps que juga en un joc concret, les monedes que s'ha gastat en aquell joc, si l'usuari ha completat missions, etc. D'aquesta taula s'han obtingut fins a 5 anys d'informació (2017-2022)

Per la creació del dataset final s'han utilitzat les següents característiques:

- Mes i Any de la compra o la sessió d'un joc
- Quantitat de diners gastats en aquell mes
- Quantitat de temps jugats en aquell mes
- Quantitat màxima històrica de vegades que s'han recollit el premi diari de forma seguida
- Quantitat màxima mensual que l'usuari ha recollit el premi diari de forma seguida. Aquesta informació també s'aplicaran a altres funcions del joc com són: Visualitzacions de Videos de Publicitat, La Bomba, etc.
- Quantitat mensual de vegades que s'han realitzat les missions de forma seguida. Una missió consisteix en tres reptes que l'usuari ha de completar, un cop completades l'usuari aconsegueix el premi de la missió.
- Quantitat de compres que s'han fet aquell mes, i el tipus d'aquestes. Podem trobar tres maneres de fer una compra:
    1. NotFund: Compra després de quedar-te sense monedes.
    2. Oferta: Compra a partir d'una oferta.
    3. Tienda: Compra a la tenda sense oferir una oferta.

5.2 Anàlisis de les dades

El primer que s'ha de saber sóns els diferents tipus d'usuaris que ens podem trobar. Per això el que s'ha fet és diferenciar en rangs la quantitat de diners que un usuari gasta. En la figura 8 es pot veure la quantitat d'usuaris que com a mitjana gasten entre els rangs 0-10, 10-50, etc. Aquesta informació ja ens està dient que entre les nostres dades podem trobar moltes persones amb adicció, ja que gastar-se més de 250€ en jocs en l'últim any i mig ja es considera un valor molt elevat. En l'apartat A.4 es pot trobar informació extra sobre com s'han analitzat els usuaris.

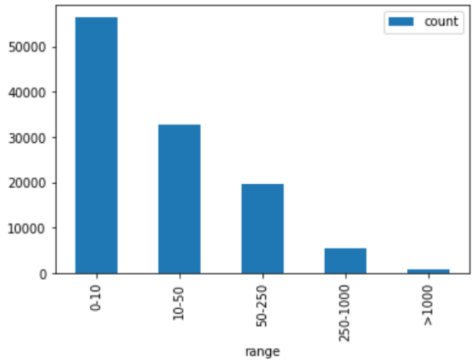

Fig. 8: Rangs entre els usuaris

Per tal de saber si hi han dependencia entre les nostres dades s'ha realitzat una matriu de confusió mostrada a la figura 9. Aquesta no es gaire rellevant ja que cada usuari es pot comportar d'una manera diferent.

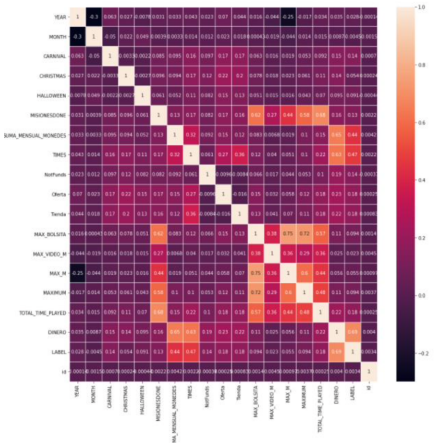

Fig. 9: Matriu de confusió entre variables

5.3 Algorismes

En aquest apartat compararem dos models molt diferents entre ells. Per una banda es realitzarà Gradient Boosting utilitzant tots els paràmetres vist en l'apartat 5.1. D'altra banda, s'utilitzara el model Prophet, en aquest només s'utilitzarà el preu gastat i el mes corresponent.

5.3.1 Gradient Boosting

Gradient Boosting està fomat per un conjunt d'arbres de decisió entrenats de forma seqüencial de tal manera que un

arbre intenta millorar els errors de l'arbre anterior. La predicció d'un nou element s'obté agregant tots els arbres que formen el model. Sent més tècnics el *Gradient Boosting* et permet crear una funció de cost sempre que aquesta sigui diferencial (La derivada de la funció és el pendent de la recta). Aquest model consta de 3 passos:

1. Ajustar el primer *weak learner* **f1**, aquest predeix la resposta **y** . Es calculen els residus d'aquest primer

$$y - f_1(x)$$

2. A continuació s'ajusta un nou model **f2** que intenta predir els errors que ha general el primer *weak learner*.

$$f_1(x) \approx y$$
$$f_2(x) \approx y - f_1(x)$$

3. Seguidament es calcularan els residus dels dos models de forma conjunta

$$y - f_1(x) - f_2(x)$$

Com s'ha realitzar anteriorment es calculen els errors causats per **f1** i que **f2** no ha sigut capaç de corregir. El tercer model **f3** intentarà corregir-los.

$$f_2(x) \approx y - f_1(x) - f_2(x)$$

4. Per finalitzar aques procés es realitzarà **N** vegades. Cada una d'aquestes iteracions intentarà solucionar els errors de l'anterior.

L'objectiu del *Gradient Boosting* és anar minimitzant els residus (error) a cada iteració.

Per tal d'executar aquest el Gradient Boosting primer s'han escollit les característiques amb més dependència cap a la variable resultant, és a dir, l'etiqueta. Després d'una anàlisi de la matriu de dependències s'ha pogut observar que altres paràmetres (mes, festivitats, mètode de compra, etc.) també han de ser important pel model, ja que el aquest hauria de tenir en consideració quan un usuari fa una compra i el perquè.

Un cop ordenades les dades s'ha dividit el data set en entrenament i test. En tenir poques dades s'ha decidit de fer el test fent ús de l'últim mes disponible (abril, 2022). Mentre que el "training"dataset pertany des del 7-2020 fins al 3-2022.

L'objectiu en aquest apartat serà predir quants diners un usuari gastarà l'abril del 2022. Per poder fer la comparació de les diverses modificacions d'aquest model s'ha utilitzat la RMSE (Error Quadràtic Mitjà). Aquest, compara el valor predit i el valor observat ja conegut. Menor és el valor calculat millor és el model tot i que cada resultat té les seves explicacions.

A l'aplicar gradient boosting s'han hagut de configurar uns hiperparàmetres. Per tal d'obtenir el millor resultat primer s'ha executat un GridRandomizeCV afegint els següents valors:

- Max_depth: [3, 6]
- learning_rate: [0.05, 0.1]
- n_estimators: [20,50]
- min_samples_split: [100,200]

Obtenint que el millor resultat amb ls següents hiperparàmetres:
Max_depth:3, learning_rate:0.05, n_estimators:50, min_samples_split:200 Aconseguint un RMSE de 41.37, mentre la mitjana de tots els diners gastats entre tots els usuaris és 198.094

Fent aquest procés amb diversos valors s'ha aconseguit analitzar els possibles paràmetres més òptims. Per tal de confirmar la solució dels resultats obtinguts al fer Randomize, s'ha generat un GridSearchCV on el millor resultat ha sigut un **RMSE de 38.57** amb els següents paràmetres:

Max_depth:3, learning_rate:0.05, n_estimators:50, min_samples_split:700

El model s'ha entrenat amb els paràmetres anteriors utilitzant tots els usuaris disponibles. El resultat que s'obté és una taula mostrada a la figura **??** que ens indica el id de l'usuari, els diners que s'ha gastat el mes 4, els diners que s'ha gastat en el mes 3, i si la quantiat de diners gastats en el mes 4 es ha incrementat respecte l'anterior Amb aquesta informació podrem saber si aquell usuari te tendència de gastar més en el següent mes. I per tant tenir informació detallada del seu procés cap a tendènica adictiva.

TAULA 1: RMSE DEPENENT DELS RANGS MODEL GRADIENT BOOSTING

ID	YEAR	MONTH	DINERO	PREDICTED MONEY
4681	2022	5	94.9	165.75
3646	2022	4	94.71	106.07

MONTH-1	DINERO_PREVIOUS_MONTH	LOWER_THAN
4	66.93	True
3	21.92	True

- ID: Identificació anònima de l'usuari
- YEAR - MONTH: Any i mes en que s'ha testejat el model
- DINERO: Diner real que s'ha gastat l'usuari
- PREDICTED: Diner predit que es gastaria
- MONTH-1: Mes anterior
- DINERO_PREVIOUS_MONTH: Diners que s'havia gastat el mes anterior
- LOWER_THAN: Comparació entre els diners predits i el mes anterior. Serà True quan el valor predit sigui superior al mes anterior.

Per tal d'intentar saber amb més precisió el valor que gastarà l'usuari, s'ha intentat dividir els usuaris amb rangs respecte a la seva mitjana gastada total. El que ens indica la taula 2 és el RMSE respecte cada un dels rangs predeterminats. En aquesta tasca s'ha hagut de combatre l'*overfitting*

pel rang 0-10, ja que hi havia molts usuaris amb valor 0 i, per tant, les dades no estaven balancejades. Per combatreho s'han aplicat tècniques per l'obtenció del balanceig de dades i Kfold per tal de dur a terme el procés de validació.

TAULA 2: RMSE DEPENENT DELS RANGS MODEL GRADIENT BOOSTING

Rang	RMSE
0-10	32.13
10-50	21.61
50-200	66.09
200-1000	236.67
1000-10000	832.69

5.3.2 Prophet

"Time series Forecasting" [10] és un procés d'anàlisi de les dades a partir del temps i fer prediccions estratègiques. No sempre el resultat d'aquestes prediccions són exactes, tot-hi això ens ajuda a saber la tendència de l'usuari en un temps concret. Utilitzant Forecasting en aquest projecte comporta una dificultat extra ja que no tenim suficient dades per tal d'obtenir resultats més concrets. La quantitat d'anys ideal per implementar forecasting són entre 3-7 anys, en el aquest cas però, en tenim 5 Per tal d'aprofitar el màxim les nostres dades he seguit el següent procés:

- Agafar tots els usuaris que tenen com a mínim un registre de compra d'entre el 2019-2022
- Diferenciar el tipus d'usuaris per rang. Aquest rang es diferenciarà depenent de la mitjana de diners que s'han gastat entre el total de mesos que han estat actius durant els anys 2020 i 2022 (Primer registre – Present). Rangs:
  1. 0-10€
  2. 10-50€
  3. 50-200€
  4. 200-1000€
  5. 1000 - 100000
- Modificacions realitzades:
  1. Per cada rang s'ha mostrat un usuari significant per tal d'analitzar el moviment dels usuaris i el comportament de la predicció.
  2. Per cada rang s'ha realitzat la suma de diners gastats en un mes i la predicció s'ha aconseguit predient la suma de diners que tots els usuaris d'aquell rang gastaran en el mes predit.
- Funcionament :
  Facebook Prophet és un algorisme obert per tal de generar models de time series. Aquest algorisme es realment bo al tenir en compte vacances, temporades, etc.

  La clau d'aquest algorisme es la suma de tres funcions: "Growth", "seasonality", "Holidays" i l'error.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Growth Function → Modela la tendència general de les dades. Prophet pot detectar automàticament el canvi de punts o afegir aquest canvis en el model .

Seasonality Function → El model identifica la estació en que ens trobem basant-se amb els punts. Més concretamente utilitza Fourier Series (suma de varis sin i cos on cada un están multiplicats per un coeficient).

$$s(t) = \sum_{n=1}^{N} (a_n cos(\frac{2\pi nt}{P}) + b_n sin(\frac{2\pi nt}{P}))$$

Holiday Function → Aquesta funció permet ajustar al model a dates que poden ser interessants per el model. Es a dir festivitats com Halloween, Nadal, etc. En el nostre cas s'han afegit 4 rangs de dades que eren significants pel model:

1. Carnaval (21/02 - 06/03)
2. Halloween (18/10 - 31/10)
3. Nadal (13/12 - 10/01)
4. Mes més comprat (1/10 - 1/02)

Cada una d'aquestes dates (Carnaval, Halloween, Nadal) són rellevants ja que són els dies en que es realitzen ofertes als usuaris, i com a conseqüència aquests compren molt més.

- Resultats:
  A la figura 10 observem la tendència d'un usuari que forma part del rang 2. Si ens fixem amb la línia vermella podem veure que durant els primers mesos de l'any acostuma a tenir tendència a gastar per sobre la mitjana. De la mateixa manera el model apren que els últims mesos a gastat molt menys però que tot i així l'usuari gastarà durant els primers mesos. A més com s'ha dit anteriorment, se li dona més importancia en els mesos d'hivern, es per aquesta raó que el model predeix una despesa quan en realitat l'usuari no ha fet cap compra. A la figura 11 observem un usuari que forma

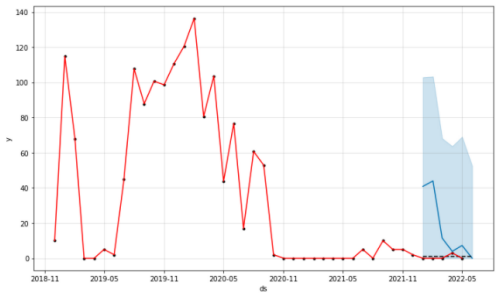

Fig. 10: Prophet predicció usuari en Rang-2

part del rang 4. Es pot observar que aquest usuari en concret no té cap tendència mensual, sinó que semrpe gasta més o menys el mateix. S'ha de tenir en compte que l'usuari és de rang 4 i per tant hi ha una gran varietat de possibilitats de la compra. En el mes 5 del 2022 el model enten que no hi haurà cap tipus de despesa, aquest esdeveniment es causat per la tendència que l'usuari tenia duran els mesos de de novembre 2019 fins

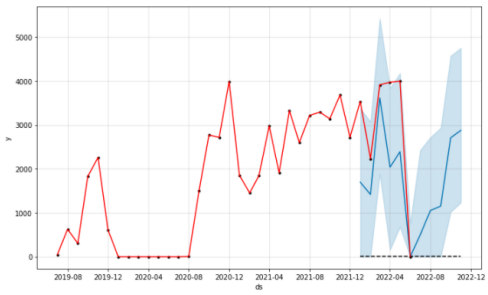

Fig. 11: Prophet predicció usuari en Rang-4

juliol del 2020. Seguidament l'usuari predeix que tornarà a gastar grans quantitats.

Com s'ha dit anteriorment, el Prophet es un model per entendre la tendència dels usuaris i aprendre el seu comportament. Aquest no és un model per predir un valor resultant exacte.

A la †figura 12 es pot observar cada un dels rangs més rellevants (0-10,10-50,50-200, ¿1000) agafant dades des del 2020. La tendència de totes les gràfiques es molt semblant, per això la predicció també te característiques semblants. En primer lloc s'analitzarà la tendència general de les dades. Podem veure que en els últims mesos de l'any i principis els usuaris gasten molt més, mentre que a mesura que ens atansem a l'estiu hi ha una baixada significativa. En segon lloc podem observar que el model segueix una tendència semblant a les dades on el mes 5 es quan menys es gasta i a partir del mes 7 va augmentant significativament els diners gastats mensualment.

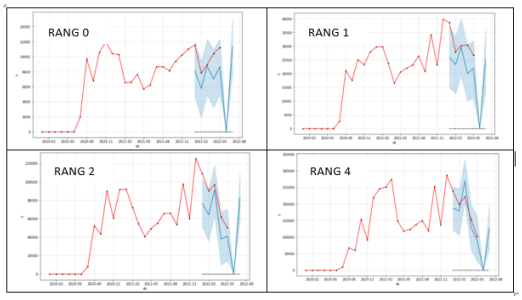

Fig. 12: Predicció suma diners gastats per rangs

5.4 Anàlisi del comportament

Cada un dels models presentats aporta gran informació sobre l'actuació dels usuaris en un futur. En el cas del Gradient Boosting ens aporta un valor exacte fent ús de variables internes mentre que el model prophet ens mostra una gràfica de la seva tendència només fent ús del temps. A més, aquest últim ens permet veure la tendència depenent del tipus d'usuari, es a dir, el rang en que pertany fent ús de la seva mitjana dels últims dos anys. En conclusió els dos models són vàlids per veure la tendència dels usuaris.

6 Conclusions

Com s'ha vist aquest treball constava de dos parts que la una sense l'altre no tindrien sentit. En primer lloc, l'anàlisi que s'ha dut a terme dels usuaris és la clau per ajudar aquelles persones que tenen una adicció. En segon lloc, veient la classificació de jocs ens serà important per a una segona versió d'aquest projecte. Sabent quins són els jocs més adictius podrem entendre encara amb més precisió el comportament de l'usuari i poder preveure amb més temps la seva reacció de cara al joc.

Per concloure, aquest projecte ha aconseguit ser la primera versió de predicció del comportament de les persones per l'ajuda d'aquestes.

Agraïments

En primer lloc, vull agrair el meu tutor Jordi Gonzalez per tota l'ajuda que m'ha donat durant el llarg d'aquest projecte. No només m'ha donat suport en conceptes tècnics sinó que m'ha donat energia i força per continuar endavant. A més, voldria agrair els meus companys de Cirsa per ajudarme amb les eines de desenvolupament i l'enteniment de les dades utilitzades. Finalment, agrair a la família pel suport extern que m'han donat.

Referències

[1] Plataforma on s'han extret les dades https://www.snowflake.com/

[2] Explicació de l'algorisme TF-IDF https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a

[3] Explicació funcionament i codi de l'algorisme Word2Vec https://radimrehurek.com/gensim/models/word2vec.html

[4] Explicació del funcionament i codi de l'algorisme Bag of Words https://www.mygreatlearning.com/blog/bag-of-words/

[5] Recull de les dades en format vector https://nlp.stanford.edu/projects/glove/

[6] Explicació general de l'algorisme Label2Vec. https://towardsdatascience.com/unsupervised-text-classification-with-lbl2vec-6c5e040354de?gi=9949214ac7d3

[7] Explicació del cosine similarity ens el texts https://www.sciencedirect.com/topics/computer-science/cosine-similarity

[8] Explicació del funcionament del Gradient Boosting Regressor https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html

[9] Explicació del funcionament de l'algorisme Prophet https://towardsdatascience.com/time-series-analysis-with-facebook-prophet-how-it-works-and-how-to-use-it-f15ecf2c0e3a

[10] Explicació de Time Series https://www.sciencedirect.com/topics/social-sciences/time-series

[11] Article d'en John Burn-Mirdoch sobre l'addicció en el joc online https://www.theguardian.com/news/datablog/2013/aug/01/uk-firm-uses-machine-learning-fight-gambling-addiction

[12] Article d'en Youngjoong Ko sobre categorització de text amb aprenentatge no supervisat http://citeseerx.ist.psu.edu/viewdoc/ citations?doi=10.1.1.107.4647

[13] Article d'en Zied Haj-Yahia, Adrien Sieg i Léa A.Deleris sobre la classificació de text mitjançant l'aprenentatge no supervisat https://aclanthology.org/P19-1036.pdf

[14] Article d'en Joan Gustafson sobre l'ús del Machine Learning per identificar la ludopatia https://www.diva-portal.org/smash/get/diva2:1356316/FULLTEXT01.pdf

[15] Article d'en Sean J. Taylor i Benjamin Letham sobre el model de forecasting Prophet https://peerj.com/preprints/3190.pdf

APÈNDIX

A.1 Regle del colze pels models de Kmeans

A continuació es mostren les regles de colze resultant amb les dues variants: fent ús del PCA 14 i sense fer-ne ús 13. Es pot veure a la figura 13 que com que no s'ha aplicat la reducció de dimensionalitat no podem obtenir una sèrie de valors, ja que gairebé no obtenim curvatura. Contràriament en aplicar PCA 14 obtenim una curvatura i, per tant, s'haurien d'agafar els valors entre 5 i 7

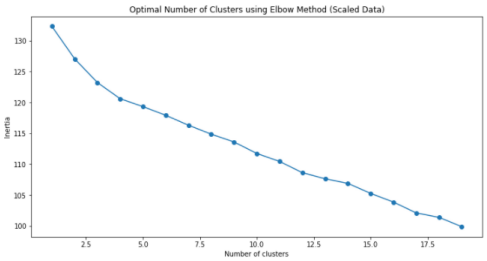

Fig. 13: Regle del colze sense reducció de dimensionalitat

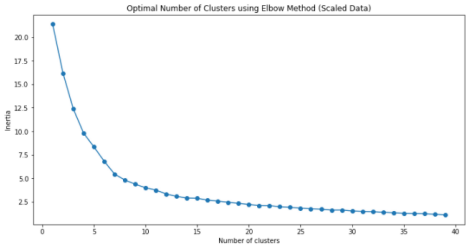

Fig. 14: Regle del colze amb reducció de dimensionalitat (PCA)

Aquest fet es troba en els models TF-IDF , Bag of Words i Word2Vec.

A.2 Word2Vec

Com s'ha dit anteriorment, per tal de que aquest model funcioni s'han eliminat les paraules més repetides. Per tal de fer un anàlisi més especific de les paraules s'ha realitzat una gràfica 15 on cada un dels punts representa una paraula de tal manera es pot veure la distancia euclediana entre les paraules.

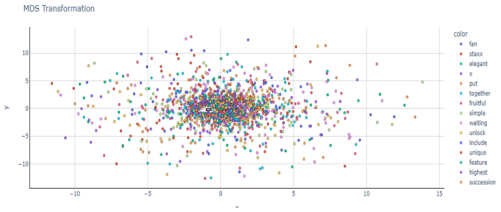

Fig. 15: Distànica Euclediana entre paraules

A.3 Lbl2Vec

Per tal de realitzar aquest algorisme primer s'ha hagut de fer un anàlisi exhaustiu de les paraules més rellevants dels

jocs. Al finalitzar cada model es realitzava un anàlisi de les paraules que apareixien en cada clúster. Aquest anàlisi m'ha ajudat a veure els diferents tipus de jocs que podem trobar dins de slots.com. Seguidament es mostra una llista de cada un dels temes amb les paraules corresponents.

- food : banana, fruits, apple, juice, watermelon, lemon, grape, fruitful, food, cake ,cookies, delicious,cook, farm, carrots
- fantasy: orgre, dragon, witch, treasure, alchemy, amulet, demon, dwarf, elf ,ghost, fairy, goblin, shaman, talisman, troll, fantasy, angels, druid, dungeons, elves, enchantment, gnome, hobbit, illusion, magic, wizard, spell, alkemor, witches, wit, lord of the rings, giants, warriors
- history: egyptian, pharao, pyramid, god ,sphinx, temple, tomb, sarcophagus, pyparus, oasis, cleopatra, rome, senate, etruscans, basin, trojan, culture, cesar , jesus, constantine, hystory, myth, mythological, talisman, arrow.
- asian culutre: asian, japanese, mahjong, sakura, buddihsm, ankh, geisha, shogun, korean, indian, mahyana, japan, oriental
- animals: animal, dog, cat, elephant, rabbit, horse, parrot, husky, zebra, falcon, bird, bear fish

La distancia entre texts es el que ens permetra diferenciar els diferents clústers i poder definir-los d'una manera clara i particular. S'observa la figura 16 que no es poden diferenciar els clústers de manera clara ja que la distancia euclediana no es representativa.

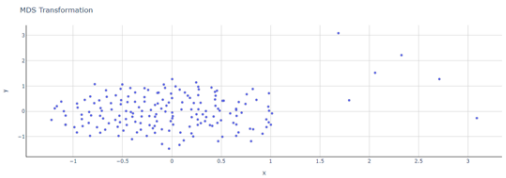

Fig. 16: Distància Euclediana entre texts

Per tal de solucionar el resultat anterior s'utilitza consine similarity. A la figura 17 s'observa la distancia de cada un dels texts en respecte a cada etiqueta.

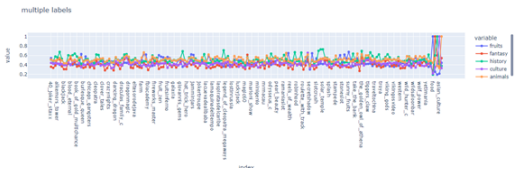

Fig. 17: Cosine Similarity entre texts

A.4 Anàlisi dels usuaris

En el següent gràfic de la figura 18 es pot observar la quantitat de cops que fan una compra cada un d'aquests grups. Cada usuari forma part d'un grup depenent de la suma total de diners i transaccions gastades durant 2 anys. S'observa que com més diners els usuaris gastin més transaccions faran.

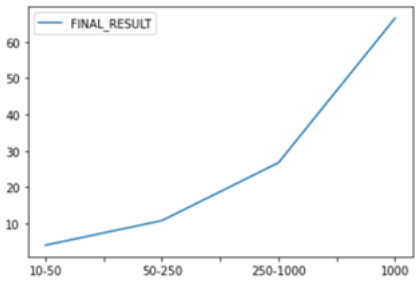

Fig. 18: Quantitat de transaccions dels usuaris depenent del rang

Seguidament es mostra a la figura 19 totes les transaccions realitzades pel grup 1 (10-50€) durant l'any 2021. S'observa que al final de l'any la quantitat d'usuaris gasten molt més que durant els mesos d'estiu. Aquesta xifra s'ha pogut contrastar no només amb el gràfic sinó calculant la quantitat de transaccions realitzades durant els últims mesos de l'any

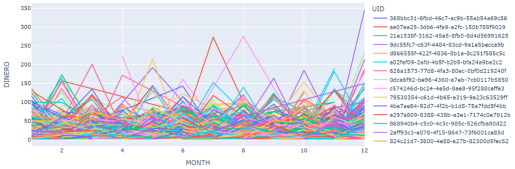

Fig. 19: Gràfic mostrant les transaccions realitzades durant el 2021