Projecte Steam Data

Álvaro Javier Díaz Laureano

Aquest article detalla el meu treball del projecte Kaggle per al grau d'Enginyeria Informàtica a la Universitat Autònoma de Barcelona, enfocat a predir les qualificacions de jocs a la plataforma Steam utilitzant dades de la pàgina oficial Kaggle. La metodologia inclou exploració de dades, anàlisi de correlacions i tractament de variables, amb un èmfasi especial en la neteja i preprocessament de dades, incloent-hi la codificació i normalització. S'analitzen diversos models d'aprenentatge automàtic, com regressor Lineal, Ridge, Random Forest i Gradient Boosting, destacant els dos últims com els més efectius. Les conclusions indiquen que és difícil establir relacions fortes entre les dades disponibles i les qualificacions dels jocs, i cal ressaltar la influència del nombre de valoracions en la precisió de les prediccions.

1 INTRODUCCIÓ

Aquest article tractarà sobre el meu treball fet del projecte Kaggle que correspon al grau d'Enginyeria Informàtica a la Universitat Autònoma de Barcelona. Específicament, el meu treball tracta sobre l'anàlisi d'una base de dades de Steam, plataforma de distribució digital de videojocs. La base de dades l'he extret de la pàgina oficial de Kaggle, Steam Store Games [1]. Les dades proporcionades ens ofereixen informació bàsica del joc com el nom, els gèneres i el preu, entre moltes altres. La motivació d'aquest projecte serà tractar de predir el "rating" de cada joc en base a les dades proporcionades. Analitzaré els resultats per determinar si és possible fer aquesta predicció o no i finalitzaré amb les conclusions extretes d'aquest resultats. L'ordre del procés serà el mateix que el que he portat a terme en el meu Jupyter Notebook, on segueixo una sèrie de procediments els quals explicaré als següents punts.

2 METODOLOGIA

Abans de obtenir els resultats finals, s'han hagut de realitzar una sèrie de passos, totalment necessaris, per poder obtenir una bona comprensió de les dades que ens porta a fer adequadament el procediment i, així, obtenir els millors resultats possibles.

Aquests passos mencionats són, per exemple, l'exploració de dades, anàlisi de correlacions, tractament de variables categòriques, etc. Com ja he dit abans, explicaré tots els passos detingudament per fer més comprensible el treball.

Com a pas principal, l'exportació de dades obtenint una visió general i superficial. El nostre conjunt de dades, com ja he mencionat a la introducció s'exporta de la pàgina oficial de Kaggle. En aquest obtenim informació general de 27000 jocs diferents.

De primeres, ja se'ns indica que es una base de dades neta. Això vol dir que no trobarem cap "NaN" a les dades pel que no haurem de fer cap tractament simple, ni molt menys complex, dels valors "nulls" que hi puguin haver. Això, generalment, resulta beneficiós no pel fet de estalviar-te treball, sinó pel fet de no haver de realitzar un "fill" que pugui no ser del tot correcte, afegint dades no coherents amb la naturalesa i patró que segueix la resta.

Analitzant superficialment el "dataset" podem veure que, tot i tenint la informació general dels jocs, no tenim cap variable directa amb el nostre valor a predir. Com ja s'ha comentat a la introducció, la meva motivació es predir la puntuació de cada joc, en format percentatge. Encara que no tenim aquesta columna, el "dataset" ens proporciona dues relacionades amb les ressenyes positives i negatives. A partir d'aquí, es possible extreure el valor a predir, fent una divisió dels "ratings" positius entre tots els "ratings". A més, ens interessa guardar el nombre total de ressenyes que té cada joc. El principal motiu ho veurem més endavant. Una vegada obtenim aquestes columnes es pot prosseguir amb l'anàlisi profund de les dades.

Al conegut apartat EDA ("exploratory data analysis"), el que fem és visualitzar variable per variable les dades contingudes en aquestes. Principalment utilitzo diagrames de barres o historiogrames, per analitzar la freqüència de cada classe de la columna en particular. Comentant-ho molt per sobre, l'anàlisi exploratori de dades ens indica els següents aspectes:

- Hi ha informació ambigua, que directament no ens serveix com pot ser Appid o Steamspy_tags que ens dona pràcticament la mateixa informació que altres columnes.
- Trobem una gran quantitat de valors únics per columnes com Name, Publisher o Developer, evidentment. Per tant, podem obviar intentar tractar amb aquest tipus de dades.
- Podem obtenir més informació en noves columnes creades a partir d'altres com Is_Free a partir de Price que ens poden ser més útils que les originals.
- La nostre variable objectiu té els valors esbiaixats cap a puntuacions altes.

Pel tema de la correlació de variables amb el nostre "target", podem deduir de primeres que pràcticament cap té relació directa. Quasi totes les dades tenen una relació molt dèbil amb el nostre objectiu, a pesar de ser estadísticament significatives. A més, no podem decidir res davant aquest resultats ja que ni tan sols s'han tractat com per considerar-los com correctes. Cal mencionar que l'anàlisi de correlacions ha variat segons si la variable era continua o categòrica. En el cas de les continua utilitzem la matriu de correlació que utilitza Pearson. Pel cas de les categòriques, a la gran majoria s'ha aplicat la correlació biserial puntual, ja que s'havia de relacionar amb el nostre "target" que es continu.

Un cop realitzat tot aquest primer anàlisi de les dades, cal realitzar la següent part més important del projecte. El pre-processament de les dades abans de avaluar models. La majoria són aspectes a tractar que s'han comentat prèviament a l'exploració de dades, de la qual s'obtenen una sèrie de conclusions que les gràfiques impreses ens permetien deduir. Aquesta sèrie de conclusions són, per exemple, que els desbalcenjos de les variables numèriques estan ocasionats per valors atípics o que, a més, s'han de normalitzar les dades per a que els models d'aprenentatge obtinguin una bona representació de les dades. Així mateix, el meu pre-processament realitza el tractament de "outliers", la codificació de les dades, la normalització i l'escalat. També realitzo un filtratge de dades per a la variable objectiu.

Començant pels valors atípics, s'analitzen les variables numèriques excepte el número de "ratings", ja que no ens interessa treure aquells valors pel factor de que són importants tants els valors alts com els baixos. Bàsicament, faig un gràfic boxplot per cadascun visualitzant i analitzant totes les dades que es consideren "outliers". Fem una observació més visual (Fig. 1).

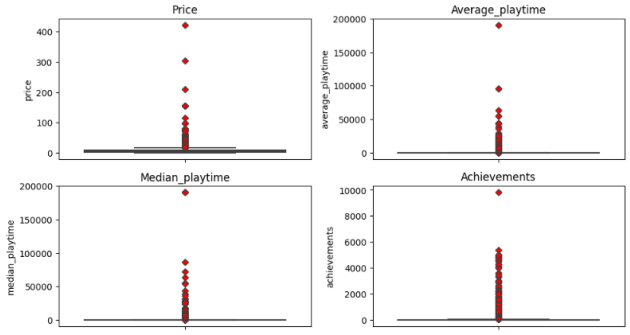

Fig.1 Boxplots de les variables numèriques.

Com podem veure a la Figura 1, totes les columnes tenen "outliers", com ja bé es podia deduir. Observem a més que les columnes Median_playtime i Average_playtime tenen una distribució molt similar. Això pot ser degut a la forta relació que prèviament ja havíem vist a l'anàlisi de correlacions, on s'observava un valor de 0.91. Pels altres dos casos hem d'analitzar millor aquesta distribució. La utilització d'un bon tractament en aquest punt és decisiu ja que aquest procés comporta la eliminació d'una gran quantitat de dades.

El procediments per l'extracció d'aquests no ha sigut una determinant i absoluta. S'han utilitzat dos mètodes totalment diferents: Rangs Interquartílics i extracció d'outliers basada en llindar fix. Fer-ho amb Rangs Interquartílics suposa, per definició, treure la meitat de les dades, pel que per alguns casos he fet una neteja arbitraria establint un llindar que consideri oportú. A tots dos casos, median i average playtime, s'ha aplicat el mateix llindar de 1000. Encara que sembli massa, la raó principal és que a partir de 1000 cap avall es on s'acumulen totes les dades. Per tots dos casos, aplicar el filtre suposa una pèrdua de 560 dades de mitjana.

El següent que s'ha realizat és el filtratge de dades al "target". L'anàlisi del principi de les dades, en aquest cas la nostra columna objectiu, ens mostra pics pronunciats tant al valor 0.0 com al 1.0. Això és degut, i es comprova al codi font, a que hi han mostres que, o tenen moltes o poques ressenyes únicament negatives, o al inrevés. Doncs el que faig amb aquestes dades és directament extreure-les perquè poden causar una mala interpretació de les dades. Fem ara una petita visualització de la columna objectiu (Fig. 2):

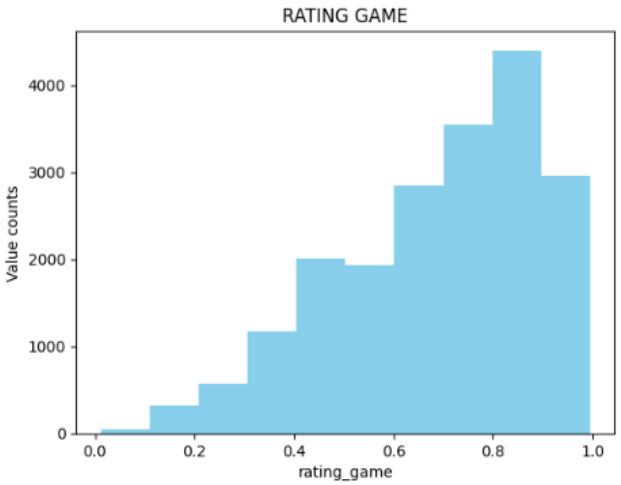

Fig. 2 Columna Target tractada filtrant valors 1's i 0's.

Continuat amb el pre-processament, el següent realitzat és el "encoding" i la normalització. A partir d'aquí ja es separen les dades "train" i "test". No fa falta explicar més que les tècniques utilitzades. Faig servir el OneHotEncoder per fer la codificació de les dades categòriques. No obstant, hi han tres variables ("platforms", "genres", "categories") que estan estructurades en mode llista. Aquestes tenen també una codificació igual que la resta però amb un procés una mica més complex, fent-ho manualment. Per altre banda, a les variables numèriques, en gran majoria desbalancejades, s'aplica una normalització logarítmica per a que tinguin una distribució normal i, posteriorment, s'escalen per a que tinguin totes el mateix tipus de dades. Una vegada realitzat això, veiem un altre cop les correlacions i veiem com han augmentat. Moltes significativament, fins un 0.1 més, però segueixen no sent valors fortament correlatius amb el nostre "target". Finalment es fa el "drop" de les columnes no desitjades i es procedeix a realitzar el model d'aprenentatge.

Aquest primer model d'aprenentatge és un regresor lineal. Comencem amb un bàsic per testejar els resultats primerament. Com a partir d'ara ja són resultats de tot el procés realitzat prèviament, els comentaré en el següent apartat.

3 Experiments, resultats i anàlisi

Com bé estava dient abans, a partir d'aquí comentaré els meus resultats i experiments portats a terme.

L'avaluació del meu model es mesurarà amb la mètrica R^2, el coeficient de determinació múltiple. Té una interpretació intuïtiva i fàcil tenint en compte que el valor màxim serà 1 i el valor mínim serà 0. Direm que si obtenim un 0.5, el 50% de la variabilitat del nostre "target" s'explica pel model. Doncs bé, el primer entrenament de tot el projecte ens dona com a resultat un 0.21 de R^2. No podem considerar-ho un bon resultat ja que no té en compte pràcticament tota la variabilitat de les nostres dades. Veiem quina és la gràfica scatterplot comparant les prediccions amb el test (Fig. 3).

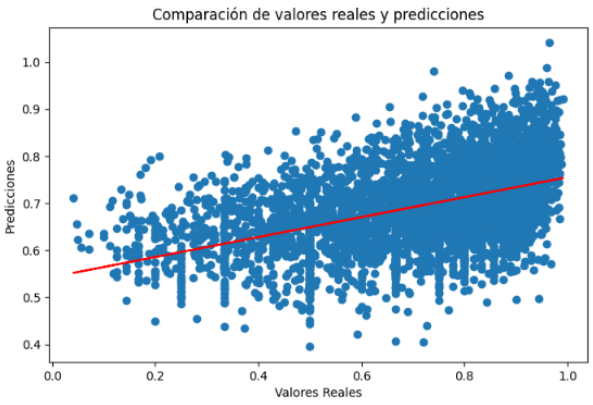

Fig. 3 Scatterplot de les prediccions i els valors reals.

Podem confirmar, doncs, que hi ha una gran variabilitat entre les dades reals i les prediccions. Unes petites comprovacions prèvies que m'ajuden a visualitzar millor com de molt varien aquestes dades és guardar en un nou "dataframe" les prediccions, els valors reals i la diferència corresponent entre tots dos. A partir d'aquest, filtro a partir de les diferències.

El filtre es basa en separar les dades amb poca variabilitat i amb molta. En aquest cas, puc considerar que 0.1 de variància no és molta, degut a les grans diferències que veiem al scatterplot. Així doncs, aplico els filtres corresponent.

Puc observar que la mostra que més varia té una diferència de 0.66 respecte el valor real. Per altre lloc, la que menys varia té un valor 0.09. Encara que aquestes dades no em diguin molt, a les taules generades puc veure una petita e inestable tendència que si podria ser decisiva. Del que parlo és bàsicament que per aquelles dades que tenen baixa variabilitat, els valors reals normalment oscil·len en "ratings" elevats, suposant més de 0.6 com elevat. Per altra banda, els jocs amb molta variabilitat tendeixen a tenir valors reals baixos, que normalment no superen els 0.2. Això ho podem associar directament al numero de "ratings" per joc. El motiu és el següent, un joc amb una puntuació baixa normalment es causada perquè les ressenyes negatives dominen contra les positives. Com bé sabem, al mon dels videojocs, que les ressenyes negatives predominin no és un cas que sovint és veu. Vull dir, si hi han casos de jocs molt dolents però no solem trobar un que tingui puntuacions tan baixes. Així doncs, aquest fet pot ser degut a dos possibles casos: un videojoc reconegut (implica moltes ressenyes) que de veritat si té aquesta puntuació o, l'altre possible cas, que es tracti de jocs poc reconeguts amb menys de 100 "ratings" d'on predominen els negatius, per una causa desconeguda.

Per confirmar la meva teoria, el que he fet bàsicament, ha sigut realitzar una gràfica que ens indica com influeix el nombre de ressenyes totals amb la diferència de les prediccions respecte els valors reals. La gràfica es la següent (Fig. 4).

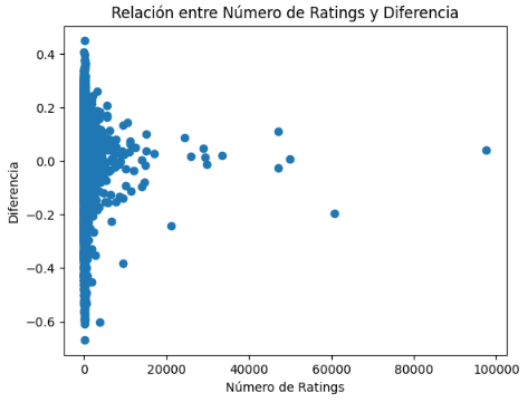

Fig. 4 Scatterplot de la relació entre la diferencia entre prediccions i valors reals amb el nombre de ratings totals.

Com esperava, hi ha una gran relació entre aquests dos factors. Quan menys nombre de "ratings" tingui el joc, major serà la diferencia normalment. Encara que no és compleix del tot perquè seguim trobant uns valors atípics però la tendència és clara i es pot confirmar.

A partir d'això, podem pensar en establir un llindar per establir una restricció que indiqui quin és el mínim de ressenyes que ha de tenir un joc per poder tenir-lo en consideració. El que he fet ha sigut veure com millora o empitjora la nostra mètrica amb diferent llindars a partir d'un gràfic (Fig. 5).

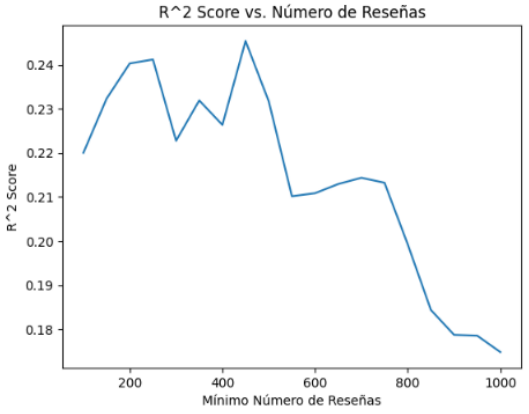

Fig. 5 Gràfic de rendiment segons el llindar de nombre total de ratings establert

Aquí podem veure que tot establint un llindar, el nostre model ha millorat, però no significativament. El millor R^2 que observem és de 0.24 que correspon a un llindar de 450 ressenyes mínimes. Això vol dir que és prediu millor quan els jocs tenen en conjunt més de 450 "ratings". No obstant, escollir aquest llindar suposa una gran pèrdua de dades, on ens quedem amb 2500 de tot el conjunt que teníem abans, que després del tractament rondava els 20000. Així doncs, la meva decisió va ser quedar-me amb 250 de llindar i poder tenir una mica més de dades amb les que entrenar.

Finalment, una vegada aplicat el filtratge de dades, ja podia entrenar diferents models i veure quin de tots em donava millors resultats. Pot ser, algun altre model trobava relacions que la regressió lineal no i em donava un R^2 considerable. Els regressors a entrenar van ser Lineal, Ridge, Random Forest i

Gradient Boosting. Per cadascun d'aquest, l'entrenament ha sigut sense paràmetres i amb paràmetres. Sense l'únic fet ha sigut un Cross Validation normal que em retorni la mitjana dels resultats de la mètrica. Pels paràmetres, he fet una cerca amb el GridSearchCV, on ja ve implementat una validació creuada. Els resultats els podem són els següents (Taula 1).

	Model	Sense hyperparametres	Amb hyperparametres
0	LR	0.250505	0.252750
1	RIDGE	0.252508	0.253438
2	RF	0.257420	0.291175
3	GBR	0.265639	0.295728

Taula 1 Resultats després de l'entrenament dels diferents models d'aprenentatge.

Podem concloure doncs, dient que el model que millor s'ajusta al nostre cas és el regressor Gradient Boosting donant uns resultats de 0.3 de R^2. Molt igualat amb el Random Forest, pel que podem escollir qualsevol dels dos. La cerca d'hiperparàmetres ha resultat útil.

Per últim, fem una avaluació final amb el test i així obtenir els resultats.

Com s'esperava, els resultats no canvien cap a bé i, fins i tot empitjora una mica.

4 Conclusions

Finalment, puc concloure diversos aspectes dels resultats obtinguts. El primer, i més important, és veure que amb les dades aportades difícilment trobarem relacions amb el que es volia predir, per no dir impossible. Sí que trobàvem petites relacions que anaven lligades al "target" però no era suficientment forta com per dictaminar completament la predicció final. Altre conclusió important a la qual he arribat al final del projecte ha sigut que trobem una forta relació en com influeix el número de "ratings" en una predicció final, ja que, quants menys tinguem, pitjor serà la predicció. Finalment dir que el "dataset" utilitzat podria haver-se orientat d'altra forma aconseguint així uns resultats molt diferents. No obstant, es casi segur que de la forma en la que lo he orientat jo, és molt probable que els resultats siguin els mateixos. Al ser un dataset on les dades ja estan molt marcades i és difícil fer un Feature Engineering, els resultats no canviarien molt.

Bibliografia

[1] Nik Davids, 2018, Steam Store Games. Kaggle. Extret de: https://www.kaggle.com/datasets/nikdavis/steam-store-games/data