Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il management

Analisi predittiva sul prezzo dei videogiochi con algoritmi di Machine Learning: Steam dataset

Relatore:
Elena Loli Piccolomini

Presentata da:
Alessandro Capotosti

Sessione unica
2022/2023

Introduzione

Il settore videoludico ha subito un importante cambiamento e si è adattato a nuove realtà durante e dopo gli avvenimenti della pandemia COVID-19. Ad alimentare questa rapida crescita economica e d'interesse nel settore negli anni di riferimento 2020 e 2021[7] hanno contribuito i protocolli di sicurezza come il 'lockdown' messi in atto dai vari governi mondiali i quali hanno indirettamente causato un aumento sostanziale di videogiocatori, contribuendo alla trasformazione dell'industria videoludica verso la distribuzione digitale e spingendo il consumatore verso l'acquisto di licenze digitali, ossia videogiochi acquistabili tramite apposite piattaforme e non più come semplici dischi fisici. A svettare su questo mercato digitale, si individua la piattaforma Steam, rilasciata da Valve nel 2003 e ad oggi leader nel settore di vendita e distribuzione di videogiochi digitali su computer. Steam è un marketplace contenente migliaia di videogiochi ed ha registrato un picco approssimativo di 33 milioni di utenti online simultaneamente nel 2023[6] con una stima di circa 400 milioni di videogiochi venduti[8]. Una delle nozioni più importanti da considerare nel settore videoludico è il pesante rincaro che un videogioco acquisisce per via dei costi di produzione e di logistica richiesti per portare il videogioco dal publisher al consumatore; dei costi aggiuntivi che spingono i publisher a preferire l'opzione di vendita tramite licenza digitale, evitando quando possibile quelli che possono essere i costi relativi alla produzione,

assemblaggio della scatola, logistica, trasporti, etc.. con il fine di aumentare i margini di guadagno[31]. Nonstante questo possa sembrare il capolinea del discorso, ciò che implica il passaggio verso l'acquisto di videogiochi digitali piuttosto che dischi fisici è anche l'aggiunta di una nuova figura nel settore dello sviluppo software: lo sviluppatore Indie nasce proprio perché non c'è più l'obbligo di investire grandi somme di denaro per quella che è la parte produttiva e logistica dietro a un videogioco ma si hanno degli sviluppatori in grado di distribuire il software pagando piccole commissioni ai proprietari di piattaforme ed eventuali pubblicità[12]. L'altra figura che beneficia di più del discorso è invece il consumatore. Prima di Steam infatti, un videogioco era inteso solo ed esclusivamente come un software prodotto da una grande azienda acquistabile e reperibile in un negozio fisico a un prezzo prefissato dal mercato: la possibilità di ridurre quei costi implica maggior concorrenza, e questa maggior concorrenza ha portato ad un vero e proprio mercato dove i prezzi non sono più dettati necessariamente dal mercato, ma si individuano giochi a tutte fasce di prezzo poiché sviluppatori Indie con risorse più limitate tenderanno a vendere il proprio prodotto a costi più ridotti dei vari titoli già popolari e prodotti da software house più grandi economicamente, famose e strutturate. L'ultima nozione importante è che questo tipo di vendita e distribuzione digitalizzata ha dato vita anche a un nuovo genere, quello "Free to play" (noto come modello "F2P"), ossia videogiochi completamente gratuiti con microtransizioni, nonché pagamenti opzionali che, in generale, forniscono contenuti aggiuntivi. La seguente ricerca è incentrata su un'approfondita analisi dei dati con sull'utilizzo di vari algoritmi di machine learning per allenare il nostro modello ad approssimare il prezzo di listino dei videogiochi nel marketplace considerando tutti quelli che sono i

fattori esterni al prezzo, quindi elementi come il numero di utenza, la data di rilascio, il picco di utenza concorrente, le lingue supportate per audio e dialoghi, il numero di recensioni positive/negative, il genere e molte altre ancora che approfondiremo nell'apposito capitolo. Come ogni analisi dei dati con uno sviluppo di modelli predettivi vi sarà una fase di raccolta dati, visualizzazione, pre-processing ed esplorazione dei dati, che serviranno di supporto per a migliorare il modello e cercare le correlazioni tra le varie informazioni(dati) a nostra disposizione. Una volta conclusa questa parte si passa al testing dei modelli e si rappresenta tramite l'errore la nostra precisione, ossia l'accuratezza del modello. Sarà dunque altrettanto importante aggiustare gli iperparametri e implementare diverse tecniche di machine learning per poi confrontarle e analizzare i risultati.

Contents

1 Steam, data analysis e machine learning **7**

1.1 Valve, Steam marketplace 8

1.1.1 Il peso sul mercato mondiale videoludico 8

1.2 Dataset e Steamworks API 9

1.2.1 Contenuto del dataset 9

1.3 Machine learning 10

2 Pre-processing dei dati, algoritmi supervised e addestramento **12**

2.1 Identificazione del problema 12

2.2 Python 12

2.3 Pre-processing dei dati 14

2.3.1 Dati scartati 14

2.3.2 Modellazione dei dati e varie trasformazioni 16

2.3.3 One-hot encoding 18

2.3.4 Feature engineering 19

2.4 Osservazioni sulla target variable 22

2.5 Multicollinearità 23

2.6 Regressione lineare multivariabile 26

2.7 Gradient boosting regressor 28
2.8 Random Forest regression 28

3 Risultati numerici **31**
3.1 Criteri di valutazione dei modelli 31
3.2 Lasso 32
3.3 Risultati 33
3.3.1 Risultati della regressione lineare 34
3.3.2 Risultati del Gradient Boosting Regressor 35
3.3.3 Risultati del Random Forest Regressor 36
3.4 Conclusioni 38

List of Figures

1.1 Varie tecniche di machine learning 10

2.1 Distribuzione del prezzo tra 0 e 60 23

2.2 Matrice di correlazione per Encoded Languages con lingue Europee 25

2.3 Coefficienti di correlazione tra 'Prezzo' e variabili con coefficiente maggiore di 0.1 o minore di -0.1 27

2.4 Random Forest Regression, profondità albero vs MSE . . . 30

3.1 Modelli per MSE, $RMSE$ e MAE 33

3.2 Modelli per coefficiente di determinazione R^2 34

3.3 Residui della regressione lineare 35

3.4 Distribuzione errore del Random Forest Regressor 37

3.5 Migliori 10 features dell'albero decisionale 38

Chapter 1

Steam, data analysis e machine learning

Il tema trattato da questo capitolo riguarda la piattaforma di Steam, la sua storia, il ruolo che assume oggi nel mercato globale e un'analisi dei dati in un dataset correlato con tecniche di machine learning. In generale, le analisi sui dati possono essere espresse anche come metodi per indagare, osservare dati non raffinati per trarre delle conclusioni; questo implica diversi passaggi a partire dalla selezione dei dati, la lavorazione che può essere espressa anche come una ricodifica del dato in qualcosa di più comprensibile[3], e a questa analisi si aggiungono tecniche di machine learning al fine di addestrare una macchina ad individuare delle conclusioni numeriche e statistiche.

1.1 Valve, Steam marketplace

La storia della Valve, azienda proprietaria della piattaforma Steam, inizia nel 1996 e rilascierà il primo videogioco, 'Half-Life', due anni dopo la nascita nel 1998. Dopo la pubblicazione di diversi titoli l'azienda inizia ad ingrandirsi molto rapidamente, fin quando non decide nel 12 settembre 2003 di rilasciare la propria piattaforma, appunto Steam, la quale avrà il compito di spostare il modello filosofico ed economico alla base dell'azienda due anni dopo nel 2005, non più una comunissima software house videoludica con l'intento di sviluppare e distribuire videogiochi; sarà anche un'azienda che tramite la propria piattaforma darà la possibilità a publishers e sviluppatori esterni di caricare i videogiochi direttamente sulla piattaforma Steam, trattenendo ovviamente percentuali sulle vendite, e costruendo una particolare vetrina digitale accessibile gratuitamente a tutti i videogiocatori nel mondo[15].

1.1.1 Il peso sul mercato mondiale videoludico

Il mercato videoludico oggi è noto come uno dei settori più grandi e redditizi, valutato circa 245.10 miliardi di dollari. Di questa gigantesca valutazione stimata, 25.5 miliardi di dollari sono la stima del mercato videoludico su computer[29]. Valve ha chiuso il 2023 con 8.56 miliardi di dollari in profitti[27], introiti che equivalgono a circa un terzo del mercato globale dei videogiochi su computer: non si parla quindi di un software di nicchia, ma della più grande piattaforma di videogiochi per computer sia per utenza attiva che per giro economico; di conseguenza, la ricerca sarà quindi basata su gran mole di dati molto rilevanti per il mercato globale dei videogiochi su computer.

1.2 Dataset e Steamworks API

Ogni analisi effettuata sui dati ha bisogno di un dataset, un insieme di dati raccolti in un singolo frammento, un file, o più in generale una collezione di dati trattata come singola unità dai computer[28]. Questi dati devono essere obbligatoriamente selezionabili e interpretabili, quindi non possiamo utilizzare dei dati che non rappresentino la realtà, che siano mancanti, sbagliati o semplicemente troppo difficili da comprendere. Per questa analisi il dataset viene ricavato estranedo tramite le API di Steamworks le informazioni necessarie e queste hanno un ruolo importantissimo poiché ottenere dati non è mai banale e scontato; avere la possibilità di comunicare ed estrapolare dati in questa maniera ci garantisce integrità dei dati, sicurezza, facilità, ma soprattutto fornisce ogni singola informazione che effettivamente interessa ogni singola entry del dataset. Una entry è generalmente rappresentata dalle righe di un dataset, e le sue colonne rappresentano i diversi parametri che essa assume.

1.2.1 Contenuto del dataset

Il dataset è aggiornato a marzo 2023, ed è composto da più di 85.000 entry, ognuna rappresentante un diverso titolo all'interno di Steam con 39 distinte colonne[5], ognuna delle quali rappresenta caratteristiche del videogioco. Successivamente saranno presentate nel capitolo relativo alla lavorazione dei dati le variabili che rappresentano queste colonne, e le diverse metodologie adottate per ognuna di esse.

1.3 Machine learning

Per citare il MIT: "Machine learning is a subfield of artificial intelligence that gives computers the ability to learn without explicitly being programmed"[4]. Questa definizione, tradotta in Italiano, ci dice che il machine learning è un ramo dell'intelligenza artificiale che da ai computer la capacità di imparare senza essere esplicitamente programmata. In particolare, queste tecniche sono principalmente distinte tra quattro categorie: supervised learning, unsupervised learning, semi-supervised e reinforcement learning[18]

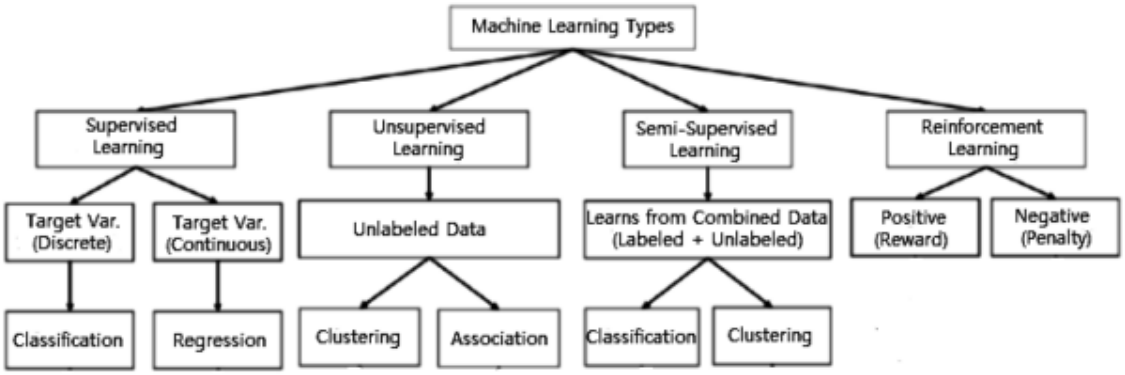

Figure 1.1: Varie tecniche di machine learning

- Supervised learning: una tecnica di machine learning con la quale apprende una funzione che mappa un input in un output basandosi su coppie di input-output[11]. Utilizza dati etichettati e una raccolta di esempi per l'addestramento per dedurre la funzione. Il supervised learning è utilizzato quando sappiamo che alcuni obiettivi possono essere identificati con un certo set di input[23]. Si distinguono in classificazione, che separa i dati in differenti classi, e in regressione che approssima i dati.

- Unsupervised learning: come suggerisce il nome, gli algoritmi di un-

supervised learning analizzano dati senza bisogno della supervisione umana, per esempio dei semplici processi data-driven[11]. I più comuni ed utilizzati sono clustering e association.

- Semi-supervised learning: può essere definito come un ibrido tra i due precedenti, perché opera sia con dati etichettati (e quindi supervisionati) che non[11]. In realtà i dati etichettati potrebbero essere rari da ottenere in particolari contesti dove i dati non etichettati sono numerosi, e questo è quando il semil-supervised learning è utile[18]. L'obiettivo principale di queste tecniche è quello di migliorare(ottimizzare e aumentare di precisione) un qualsiasi risultato ottenuto da dati etichettati, ai quali ovviamente, possiamo aggiungere dati non etichettati.

- Reinforcement learning: sono algoritmi di machine learning che forniscono alla macchina lo strumento per valutarsi in modo autonomo in uno specifico ambiente o contesto per migliorare la sua efficenza e precisione[16], quindi un approccio diretto verso l'ambiente/contesto molto presente nella robotica, in quanto tali tecniche istruiscono le macchine a valutarsi tramite un sistema di ricompensa e penalità in base alle loro performances.

Per questa ricerca si vedranno gli algoritmi supervised, in quanto l'obiettivo converge con questo specifico ramo del machine learning.

Chapter 2

Pre-processing dei dati, algoritmi supervised e addestramento

2.1 Identificazione del problema

Come anticipato nel primo capitolo, la ricerca consiste all'applicazione e valutazione di diverse tecniche di machine learning supervised. Questa valutazione sarà espressa tramite errore numerico: si vogliono addestrare e testare diversi modelli a individuare il prezzo di un videogioco, sfruttando le altre informazioni a disposizione (dati, metadati) e confrontare i risultati degli errori per stabilire quale modello, per questo specifico dataset e questa lavorazione di dati, sia più preciso e quindi con il minor margine d'errore.

2.2 Python

La ricerca è stata effettuata utilizzando Python come strumento, un linguaggio interpretato molto pratico da apprendere e relativamente veloce quando comparato ai linguaggi compilati proprio per la mancanza di compilazione.

Viene utilizzato per diverse manzioni come lo sviluppo web, l'esplorazione e manipolazione dei dati, il machine learning e altri campi dell'intelligenza artificiale. Python è molto versatile e veloce anche per quanto riguarda l'EDA, una procedura essenziale per ogni analisi sui dati. Questa pratica consiste nell'esaminare la distribuzione dei dati, individuare eventuali outliers e anomalie che potrebbero avere un impatto negativo sui risultati attesi ed è un tipo di analisi prevalentemente grafica dove i dati vengono rappresentati in relazione tra loro al fine di comprendere al meglio il ruolo di ogni variabile per il nostro problema[14]. Tra gli altri pregi del linguaggio indichiamo:

- Semplice lettura; è molto incentrato sulle keyword, le righe di codice sono più facili da distinguere ad occhio anche per la preferenza di tali keyword piuttosto che punteggiature varie.
- Open source: si può scaricare il codice sorgente, modificarlo e adattarlo alle nostre necessità.
- Portabilità; che si traduce in flessibilità, visto che tramite il suo interprete possiamo eseguire porzioni di codice Python in ogni sistema.
- Estendibilità: può essere integrato ad altri linguaggi come il C++, si possono anche aggiungere modelli di basso livello all'interprete per ottimizzare alcune particolari task di un software più esteso.
- Librerie; del linguaggio, che offre un'ampia libreria[1] con quasi ogni funzione necessaria in modo da ridurre i tempi per lo sviluppo di funzioni dedicate.

Per questa ricerca saranno utilizzate alcune tra le librerie più adottate e popolari nel campo del machine learning e dell'analisi dei dati: numpy per

i calcoli numerici e modellazione/trasformazione dei dati, matplotlib per la parte grafica relativa alla stampa delle nostre analisi ed EDA, pandas per la lettura, importazione/esportazione dei dataset e altre ancora, approfondite successivamente.[24]

2.3 Pre-processing dei dati

Il pre-processing dei dati è l'insieme di tecniche adoperate per 'pulire' i nostri dati e risolvere i problemi più comuni che possono avere: la presenza di dati ridondanti, inconsistenti, mancanti; in aggiunta, anche tecniche per la normalizzazione dei dati, discretizzazione e trasformazione sono considerate operazioni di pre-processing[2].

2.3.1 Dati scartati

Ricordiamo che il dataset presenta 39 colonne, il primo procedimento che si effettua è quello di scartare subito quelle che non possono essere utilizzate per inaffidabilità, mancanza di dati, o anche quelle variabili che non sono state codificate in quanto irrilevanti. Un resoconto delle colonne scartate e una breve motivazione:

- "Name": il nome del videogioco. Una particolarità è che si potrebbe approfondire l'esplorazione per determinate parole chiave e osservare le relazioni con il prezzo; ma sembra troppo dispendiosa la ricerca in termini di tempo e dopo aver esplorato qualche parola chiave, è stato concluso che le relazioni restano deboli e quindi la variabile è stata scartata.

- "AppID": codice identificativo (univoco) di ogni videogioco. Questo numero non rappresenta nulla; non è risaputo con quale algoritmo Steam associ questo numero a un videogioco, se sequenziale o randomico.

- "About the game": una variabile che contiene la descrizione del videogioco. È una stringa con informazioni generiche per ogni videogioco non molto semplice da correlare e come se non fosse già abbastanza, caratteri non codificati correttamente per via delle differenti lingue presenti e descrizioni mancanti sono la normalità per questa variabile.

- "Header image": semplice indirizzo web dove è allocata l'immagine del videogioco. Non ha valori nulli(Steam salva in automatico un immagine di default o forza lo sviluppatore ad inserire un'immagine), e sono tutti self-hosted da Steam considerato che l'indirizzi tra loro si differenziano solo per un altro codice identificativo, quindi risulta inutile per questa ricerca.

- "Score rank": due problemi, il primo è la comprensione del dato in sé in quanto i metadati a nostra disposizione non specificano cosa sia o cosa dovrebbe rappresentare. Il secondo problema è 42 non-null valori, che su un dataset di 76987 righe la rendono una variabile scarsamente popolata (sparse variable[17])

- "Screenshots", "Movies": insieme di link contenenti immagini e video del videogioco. Hanno lo stesso problema di "Header image".

- "Reviews": un altro dato sparso. Il testo è di difficile comprensione in quanto formattazione e contenuti sono difficili da contestualizzare.

2.3.2 Modellazione dei dati e varie trasformazioni

Un'altra parte di colonne è stata modellata a misura, sarà presente in questo sotto capitolo una lista delle variabili che abbiamo gestito, cercando di estrapolare più informazioni numeriche possibili per esplorare e confrontare più correlazioni possibili. Si consideri che le seguenti operazioni non sono state le uniche effettuate, i dati per tutta la durata della ricerca hanno subito trasformazioni ma vengono di seguito riassunte solamente le trasformazioni che, dopo vari processi di analisi grafica e/o numerica, hanno riscontrato i risultati migliori:

- "Release date": data di rilascio, è stata formattata e divisa in tre colonne rappresentanti giorno, mese e anno.
- "Estimated owners": questo dato è tra i più particolari e difficili da gestire, in quanto raccoglie un range stimato di possessori di un videogioco. A partire da uno dei valori più frequenti, il range 0-0 è estrapolato e rappresentato in un'altra colonna binaria dove 0 rappresenta i giochi senza possessori e 1 quelli che hanno almeno un possessore; poi per ogni range distinto, ad esempio 0-20000, è stata applicata una semplice operazione di media. È chiaro che il dato è complicato da codificare in quanto i range sono molto ampi tra loro, ma i coefficienti di correlazione che sono stati utilizzati successivamente mostrano una buona correlazione e quindi la media risulta un buon punto di partenza come approssimazione.
- "Required age": rappresenta l'età minima consigliata e da questa è stata raccolta l'osservazione per i giochi 18+, messi in una nuova colonna per distinguere i giochi consigliati ai 18+ dagli altri (ancora tramite

variabile binaria)

- "Support url", "Support email": due variabili stringhe, la prima contiene un link al sito web del supporto, la seconda un indirizzo email sempre riservato al supporto. La prima diventa binaria e assume valore 1 se possiede un indirizzo (qualsiasi stringa contenente 'www'), 0 se è nulla. "Support email" può risultare banale a prima vista, ma una rapida ricerca sul dato mostra che 23685 variabili contengono la stringa "@gmail.com", ed è quindi facile assumere che sviluppatori con un dominio gratuito potrebbero avere una forte correlazione con il prezzo del videogioco per questo salviamo il dato in una nuova colonna dove assumerà 1 se contiene la stringa menzionata precedentemente, e 0 per le altre.

- "DLC count": acronimo di Downloadable Content[19] è un conteggio di quanti contenuti aggiuntivi contiene il videogioco. Viene binarizzato in una nuova colonna come 0 se non ha DLC, 1 altrimenti. Molte di queste variabili binarie sono create mantenendo la colonna contenente la variabile continua di partenza, caso analogo per la variabile "Required age". Il motivo è cercare di individuare la combinazione migliore tra queste ed eventualmente decidere quale combinazione dare in input ai nostri modelli.

- "Windows", "Mac", "Linux": semplici booleane TRUE/FALSE rappresentati la compatibilità con il sistema operativo, si effettua una conversione numerica per renderla compatibile con i nostri modelli.

- "Metacritic url": un indirizzo link contenente la pagina sull'ononimo sito. Si distingue in binario chi ha un indirizzo link e chi altrimenti.

- "Notes": alcune note che gli sviluppatori possono aggiungere sul videogioco. Trasformata in binaria per differenziare la presenza o meno di queste note.

Tra le variabili rimanenti, a seguito di queste operazioni, vi sono variabili continue con pochi valori nulli al loro interno. I valori nulli sono stati scartati, e le uniche variabili rimanenti a questo punto sono quelle contenente diverse informazioni raggruppate. Per esempio, la variabile 'Genres' che racchiude un insieme di termini relativi al genere del videogioco; osservando la prima riga del nostro dataset vediamo che il dato è salvato come stringa "Casual,Indie,Sports". Per queste variabili sarà adottata la tecnica One-hot encoding.

2.3.3 One-hot encoding

Come già menzionato nel capitolo precedente le variabili categoriche non possono essere quantificate o interpretate dai nostri algoritmi di machine learning in quanto non rappresentano un numero e si è già vista la trasformazione per alcune di esse come ad esempio "Windows", "Linux" e "Mac", ma per altre variabili categoriche aggregate risulta più complesso estrapolare l'informazione. Una variabile categorica come 'Genres' che ha diversi valori si potrebbe rappresentare come una categorica in due modi: quello più semplice e diretto è quello di associare un numero per ogni aggregato, ad esempio la entry "Casual,Indie,Sports" potrebbe equivalere a 1, la entry "Casual,Sports" come 2, e così per il resto delle entry. La procedura non è necessariamente sbagliata, ma rappresenterebbe una realtà in parte distorta in quanto gli algoritmi non sarebbero in grado di captare quali sono le categoriche mancanti né tanto meno quelle presenti, visto che valuterà

sempre l'aggregato di questi generi e non la loro unicità. Il One-hot encoding risolve questo problema creando un vettore binario per ogni diversa categoria. Essendo codificato come un unico indice binario, si forma una matrice sparsa dove ogni riga rappresenta un'unica istanza e le altre colonne rappresentano sia la presenza che l'assenza di una specifica categoria[10]. Senza approfondire gli algoritmi adottati, in quanto le funzioni fornite da pandas non sono adatte per la lavorazione dei dati vista la formattazione interna del dato, si elencano le variabili e una breve descrizione della propria rappresentazione:

- "Supported languages": lingue di testo supportate dal videogioco.
- "Full audio languages": lingua dell'audio supportato dal videogioco.
- "Genres": genere/i del videogioco.
- "Categories": categoria/e del videogioco.
- "Tags": alcune etichette che gli sviluppatori possono associare al videogioco per semplificare la ricerca degli utenti all'interno della piattaforma.

2.3.4 Feature engineering

Tutte le variabili che sono state menzionate sono state osservate e considerate di conseguenza durante il processo di EDA e tramite rappresentazioni di distribuzione e calcoli di coefficienti di correlazione sono state selezionate e prese in considerazione alcune delle più correlate con la variabile d'interesse, il prezzo, e utilizzate per costruire nuove variabili. Questo viene definito "feature engineering", una tecnica del machine learning che sfrutta i dati esistenti per creare nuove variabili non presenti nel dataset. Può costruire

nuove variabili sia per l'apprendimento supervised che unsupervised, e il suo obiettivo è quello di semplificare e velocizzare la trasformazione dei dati mentre migliora la precisione del modello[22]. Elenchiamo quindi le nuove variabili ottenute e il procedimento con le quali sono state ricavate:

- "Achievements to playtime ratio": divisione tra il numero quantitativo della variabile 'Achievements' (obiettivi digitali che si possono raggiungere nel videogioco) e il numero di media ore giocate dai giocatori detentori del titolo (la variabile all'interno del dataset è salvata come "Average playtime forever"). Assegnamo a questa nuova colonna '0' anche quando la variabile denominatore è uguale a 0.

- "Category score": misurando i coefficienti di correlazione delle variabili codificate tramite One-hot encoding per quanto riguarda le categorie, si osservano determinate categorie con una correlazione positiva con il prezzo. Le categorie sono infatti: "SteamCloud", "SteamTradingCards", "Fullcontrollersupport", "RemotePlayonTV", "SteamWorkshop" e "RemotePlayonTablet". Il "Category score" sarà la somma di questi valori, molte di queste categorie richiedono strumenti di supporto quali controller e tablet, si ha una giusta motivazione per convincersi che l'associazione maggior dispositivi e strumenti di distribuzione sia correlata a una maggior cura nel framework di sviluppo e alla strumentazione necessaria per il testing, nonché tutti costi aggiuntivi che potrebbero giustificare l'aumento di prezzo.

- "Compatibility score": punteggio per la compatibilità sui sistemi operativi, somma tra le variabili "Mac", "Linux", "Windows".

- "Fiscal quarter": divide i giochi in base al periodo di appartenenza del

quarto fiscale[9] in modo sequenziale; 1 sono i primi tre del mese, 2 i successivi 3 e di conseguenza per rappresentare anche Q3 e Q4.

- "Languages score": stessa procedura con gli stessi procedimenti di "Category score" ma relative alla lingue testo supportate; le variabili codificate che hanno coefficienti più alti sono le seguenti: "Spanish-Spain", "German", "French", "Japanese", "Italian", "Polish", "Portuguese", "TraditionalChinese", "Russian", "Korean", "SimplifiedChinese", "Portuguese-Brazil".

- "Negative score": un punteggio per quante variabili con coefficienti negativi il videogioco contiene. Non si prendono in considerazione tutti i coefficienti negativi, ma solo quelli che potrebbero essere correlati e che hanno un peso maggiore rispetto agli altri. Conteggiato sommando le variabili: "In-AppPurchases", "Indie", "Casual" e "isGmail" quest'ultima, non presente nel dataset, è stata precedentemente ricavata ponendo a 1 tutti i videogiochi contenenti "@gmail.com" come email di supporto.

- "Trend": divisione tra "Average playtime two weeks" (tempo medio di gioco delle ultime due settimane) e "Average playtime forever".

Queste non sono le uniche variabili che sono state ricavate e analizzate, ma sono le uniche menzionate in quanto hanno riscontrato i risultati migliori sui vari modelli.

2.4 Osservazioni sulla target variable

Il dataset è stato totalmente trasformato e valutato su ogni modello in modo procedurale; ed ogni variabile nuova o insieme di variabili trasformate come le One-Hot encoded sono state osservate con i risultati del modello. Prima di procedere con gli algoritmi si osservano gli outliers per la nostra variabile 'Price': vengono contati il numero di videogiochi con valore superiore a 60 e si ottiene 209, un numero fin troppo piccolo di dati che allo stesso tempo rischia di compromettere il modello in quanto i valori hanno un prezzo che arriva intorno i 1000. Si ritiene che i videogiochi in questione siano semplicemente dei prodotti 'Premium' con un sovrapprezzo ingiustificabile e per tanto proviamo a scartarli. Un'altra cosa che si va a prendere in esame è la distribuzione del prezzo per i restanti valori rimanenti compresi tra 0-60. I valori nel dataset iniziano ad essere molto sparsi sopra il prezzo di 30 e si potrebbero valutare i modelli per le entry aventi prezzo inferiore ai 30. Questo approccio fornisce sì una miglioria negli errori, ma molti titoli sono presenti al numero '59.99', e i nostri modelli catturano questa informazione restituendo un errore medio leggermente più alto in quanto risulterà più complesso trovare il prezzo nella fascia compresa tra i 30 e i 59.99, ma riuscirà a determinare una buona parte dei videogiochi posti a 59.99, riscontrando una miglioria non indifferente nel coefficiente di determinazione a costo di errori leggermente più grandi. Si osserva la distribuzione dei prezzi nel grafico:

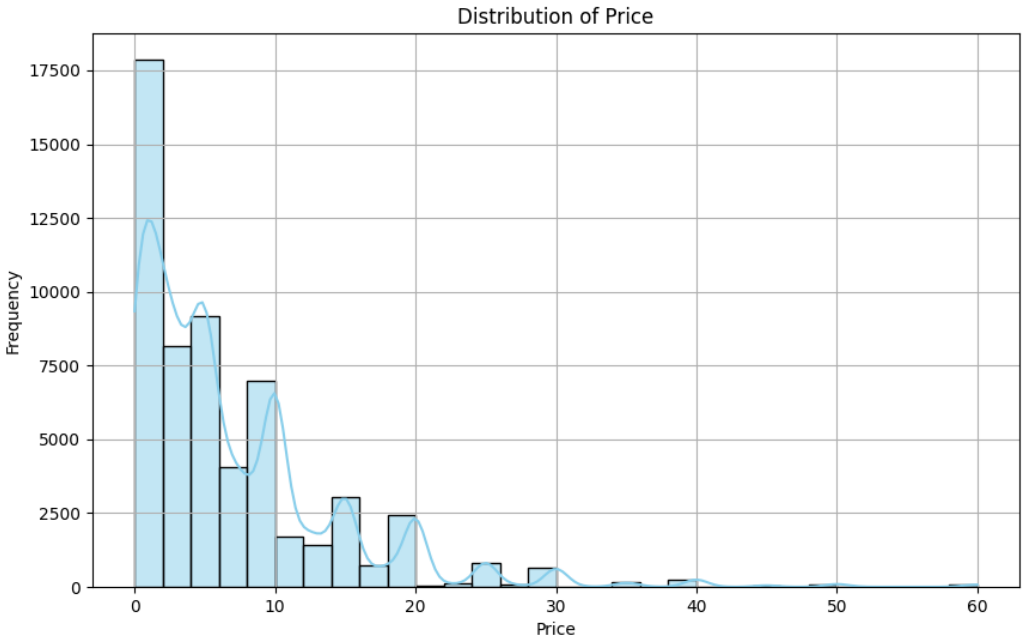

Figure 2.1: Distribuzione del prezzo tra 0 e 60

2.5 Multicollinearità

L'analisi multivariata è un metodo statistico comunemente utilizzato quando vengono considerate multiple variabili predittive per stimare l'associazione con le misurazioni dello studio. Nonostante questo, l'efficenza delle multivariabili è molto dipendente anche tra la correlazione che queste variabili predittive hanno tra di loro, in quanto i modelli assumono che questa correlazione non esista e che ogni variabile sia un assioma a sé stante. Quando queste variabili non sono indipendenti l'una tra l'altra, i problemi di collinearità o multicollinearità iniziano a crescere e come conseguenza si ha una stima dei coefficienti prevenuta che porta a una perdita di potere. La diagnosi della collinearità causa intervalli di confidenza maggiori e aumenta l'errore dei coefficienti di regressione aumentando la chance di rigetto dell'ipotesi sui test; motivo per cui anche l'aggiunta di una singola colonna

può portare a grandi differenze nei coefficienti di regressione e quindi rende l'interpretazione dei test più difficile e imprecisa.[32] Nel dataset molte variabili hanno una forte correlazione tra di loro, e sono stati testati i modelli con la rimozione di alcune di esse. Non vi sono grandi problemi in quanto la maggioranza delle variabili hanno una correlazione lieve con coefficienti di correlazione con la target variable che raramente, superano lo 0.2 nonostante la grande quantità di variabili a disposizione. Un esempio nocivo ottenuto dalla lavorazione della colonna 'Tags': alcune di queste entry non hanno etichette e quando si decodifica tramite il One-hot encoding questa variabile, si ha una nuova colonna che assume '1' quando non vi sono etichette. Il problema di questa variabile presa in esame è che rappresenta una variabile già identificata dalle altre multivariabili. Se tutte le altre etichette binarie codificate sono poste a 0 equivale a dire che la colonna priva di etichette equivale a 1, dunque l'informazione viene catturata dal modello e non necessitiamo di una ulteriore variabile. Si procede a scartare variabili di questo tipo che causano interferenza, considerando che per la natura del dataset rimarranno sempre variabili altamente correlate, ma non necessariamente causa di multicollinearità in quanto le variabili sono generalmente deboli verso la variabile da predirre e quindi migliorando i vari modelli. Un esempio di multivariabili ad alta correlazione tra loro che non impatta negativamente è quella che si ottiene della codifica delle lingue, si osserva che si ha un alto coefficiente tra alcune delle lingue Europee (Italiano, Tedesco, Spagnolo, Francese) che però fortificano i nostri modelli e non necessitano di una rimozione. Si include anche la lingua inglese nella matrice di correlazione per evidenziare una mancanza di correlazione tra questa e le restanti quattro principali lingue del continente europeo.

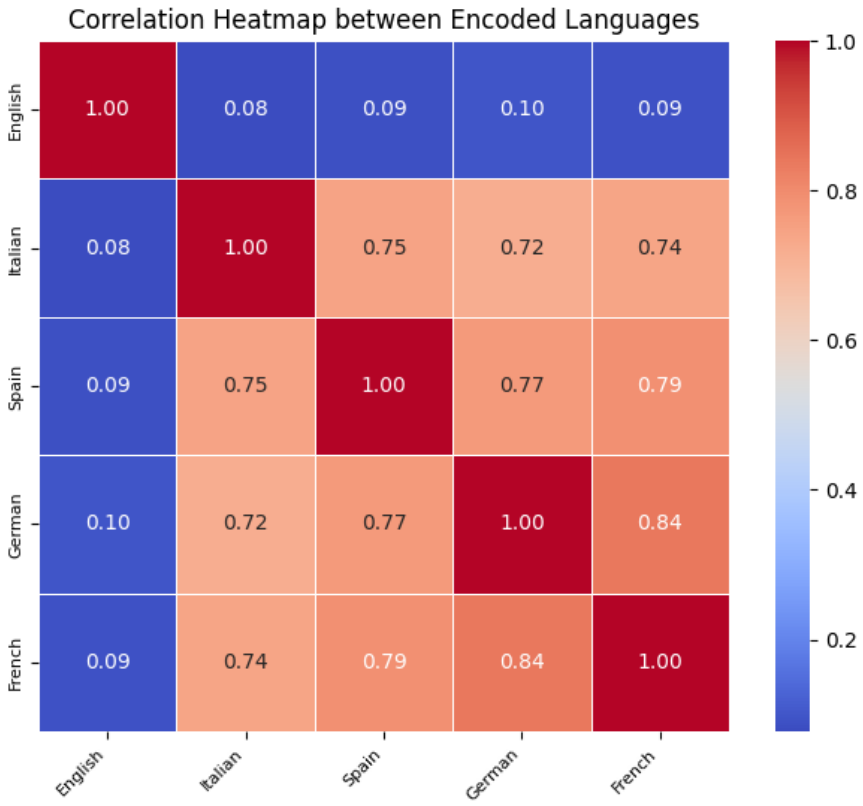

Figure 2.2: Matrice di correlazione per Encoded Languages con lingue Europee

2.6 Regressione lineare multivariabile

In molti casi, il contributo di una singola variabile indipendente non è sufficiente a spiegare la variabile da predirre. Se questo è il caso, si può effettuare una regressione lineare multivariabile per studiare l'effetto di molteplici variabili sulla variabile dipendente. In questo modello la variabile dipendente è descritta come una funzione lineare delle variabili indipendenti Xi, come:

$$Y = a + (b1 \cdot X1) + (b2 \cdot X2) + ... + (bn \cdot Xn).$$

Il modello consente la computazione di un coefficiente di regressione bi per ogni variabile indipendente Xi. Analogamente alla regressione univariata, il coefficiente di correlazione che determina complessivamente la relazione tra la variabile indipendente Xi e la variabile dipendente Y. Questo corrisponde al quadrato dei multipli dei coefficienti di correlazione, nonché la correlazione tra Y e $(b1 \cdot X1) + ... + (bn \cdot Xn)$. Un modo per condurre una regressione multivariabile è quella di includere tutte le potenziali variabili indipendenti correlate al modello. Il problema di questa metodologia è che il nummero di osservazioni (righe del dataset, entry) spesso è inferiore da quelle richieste dal modello. In generale, si vogliono avere 20 volte le osservazioni per ogni variabile indipendente sotto esame. Questa metodologia è dunque attuabile nella ricerca, ma necessita di ritocchi in quanto troppe variabili irrilevanti incluse nel modello tenderanno ovviamente a migliorare il dataset sotto esame in quanto si suggeriranno sempre dei pattern (seppur poco correlati) al modello; ma al costo di inquinare e danneggiare il modello per l'analisi di nuovi eventuali dati, visto che i coefficienti con poca correlazione tenderanno ad avere una rappresentazione errata della realtà[25].

L'immagine seguente mostra i coefficienti di correlazione con la variabile 'Price' che hanno un punteggio minore di -0.1 e maggiori di 0.1.

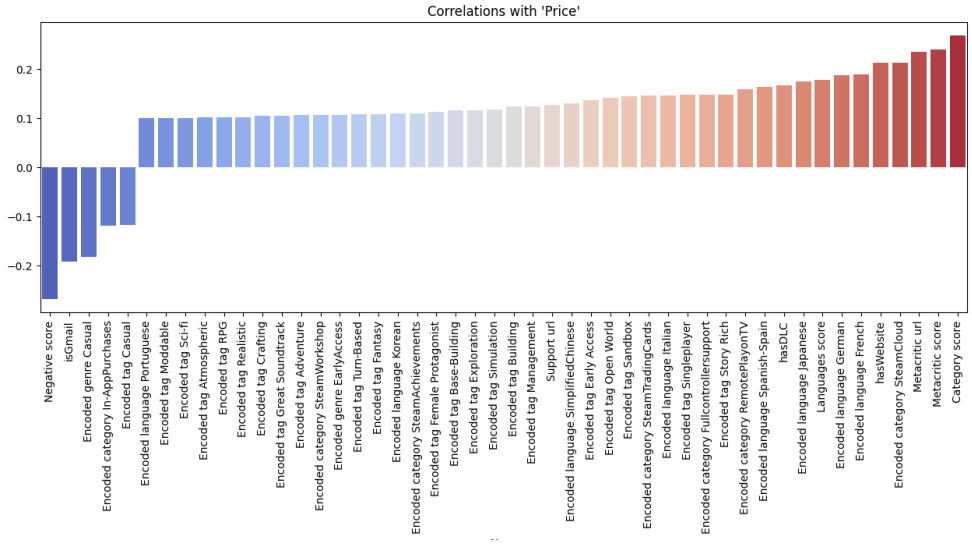

Figure 2.3: Coefficienti di correlazione tra 'Prezzo' e variabili con coefficiente maggiore di 0.1 o minore di -0.1

2.7 Gradient boosting regressor

Il Gradient Boosting Regressor (GBR) è un modello d'insieme con una collezione iterativa di modelli di alberi disposti in sequenza in modo che il successivo apprenda dagli errori del precedente. Questa tecnica di apprendimento del machine learning effettua delle predizioni utilizzando un 'boosting' (potenziamento) dell'insieme dei modelli predittivi più deboli, principalmente alberi di decisione, per addestrare un modello più robusto. Un GBR con M numeri di alberi si può scrivere come:

$$f_M(x_j) = \sum_m^M \gamma_m h_m(x_j)$$

Dove h_m è un learner debole che considerato individualmente ha uno scarso rendimento e γ_m è un fattore di scala che aggiunge il contribuito del singolo albero sul modello complessivo. GBR utilizza la funzione di perdita della discesa del gradiente per minimizzare gli errori aggiornando l'ipotesi iniziale con la nuova stima. Si ricava, un modello con la combinazione di tutte le stime preliminari con un peso relativo alla singola performance[21].

2.8 Random Forest regression

Una random forest è una collezione di alberi decisionali

$$h(x : \theta_k), k = 1, ..., K$$

dove x rappresenta il vettore di input osservato di lunghezza p con un vettore casuale associato X e θ_k sono vettori casuali indipendenti e identicamente distribuiti. Si assume che i dati siano estratti in modo indipendente dalla distribuzione congiunta (X, Y) e comprende $n(p+1) - tuple(x_1, y_1), ..., (x_n, y_n)$ Per la regressione, la predizione the random forest è la media non pesata

della collezione $\overline{h}(x) = (\frac{1}{K}) \sum_{k=1}^{K} h(x; \theta_k)$. Quando $k \to \infty$, la legge dei grandi numeri assicura:

$$E_{X,Y}(Y - \overline{h}(X))^2 \to E_{X,Y}(Y - E_\theta h(X; 0))^2 \tag{1}$$

La quantità sulla destra è l'errore predetto per il random forest, indicato con PE_f^*. La convergenza ricavata in (1) implica che le random forests non sono soggette a problemi di overfit. Per definire la media dell'errore PE:

$$PE_t^* = E_\theta E_{X,Y}(Y - h(X; \theta))^2 \tag{2}$$

Assumendo che per ogni θ l'albero sia imparziale, allora:

$$PE_f^* \leq \overline{p} PE_t^* \tag{3}$$

Dove $\overline{p}$ è la correlazione pesata tra i residui $Y - h(X; \theta)$ e $Y - h(X; \theta')$ per vettori indipendenti θ, θ' L'inequalità (3) suggerisce ciò che è richiesto per una regressione con alberi decisionali randomici:

1. Bassa correlazione tra i residui dei diversi alberi della foresta;

2. Basso errore predittivo per i singoli alberi.

La regressione dunque, diminuirà l'errore del singolo albero, PE_t^* per un fattore $\overline{p}$ [26]. Per la rappresentazione, si mostra un grafico avente come asse X la profondità dell'albero e l'errore medio quadrato (MSE) nell'asse Y. L'iperparametro della profondità è posto a 20.

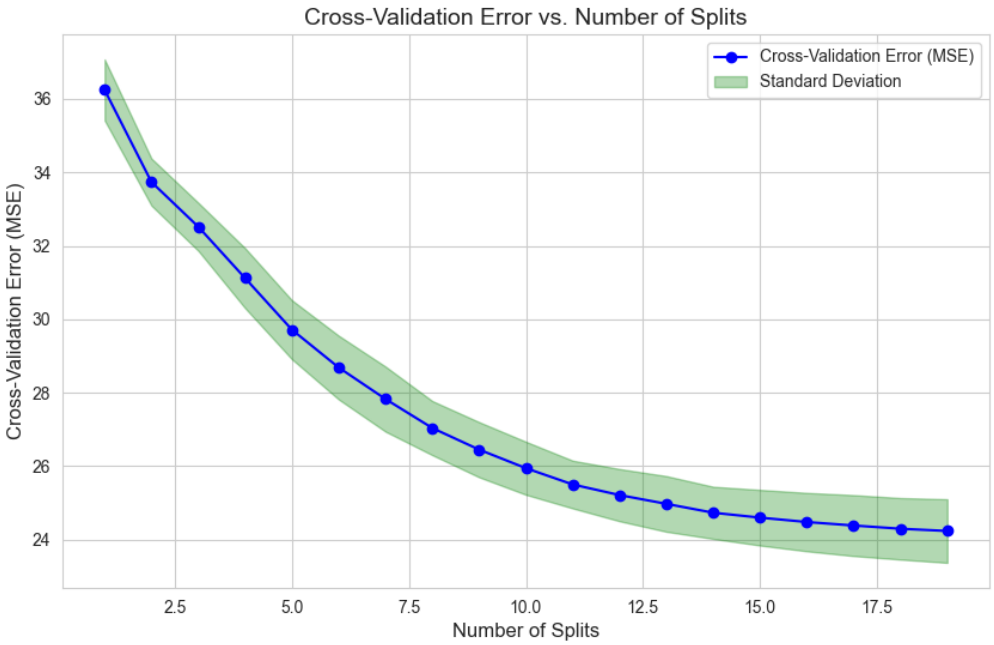

Figure 2.4: Random Forest Regression, profondità albero vs MSE

Chapter 3

Risultati numerici

3.1 Criteri di valutazione dei modelli

Per le conclusioni di questa ricerca si valuteranno i modelli in base al loro errore. Non sarà limitata al calcolo dei diversi algoritmi con uno specifico dataset, ma si applicano anche diverse tecniche di ottimizzazione e iper-parametrizzazione da confrontare. L'errore di un modello di regressione è dato dalla differenza tra i punti dei dati e la linea prodotta del modello. In particolare, consideriamo:

- MAE: Mean Absolute Error, rappresenta la media assoluta degli errori ed è espressa matematicamente come:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

- RMSE: Root Mean Squared Error, è un metro di giudizio molto utilizzato per valutare le performance di un modello perché può essere interpretato come la deviazione standard degli errori predittivi. Si scrive:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}} [21]$$

- MSE: Mean Squared Error, stesso tipo di errore del RMSE ma senza la deviazione standard, semplicemente:

$$MSE = \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}[13]$$

- R^2: R-Squared, o anche noto come coefficiente di determinazione è definito:

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_{Res}}{SS_T}$$

  SS_T è una misurazione della variabilità in y senza considerare l'effetto della variabile regressa x e SS_{Res} è la variabilità restante in y dopo che x è considerata. R^2 è spesso menzionato come "proporzione della variabilità spiegata del regressore x. Siccome $0 \leq SS_{Res} \leq SS_T$, segue che $0 \leq R^2 \leq 1$. Più i valori di R^2 tendono ad avvicinarsi a 1, maggiore l'implicazione che la variabilità di y sia spiegata dal modello di regressione.[20]

Un'altra importante misurazione che non è presa in considerazione in questa ricerca è il tempo di esecuzione dei vari algoritmi. Successivamente nella valutazione dei modelli vi sarà comunque una spiegazione sulla scelta in quanto strettamente correlata ai risultati ottenuti.

3.2 Lasso

Le due tecniche standard per migliorare una regressione lineare sono la subset selection e la Ridge regression, ma entrambe hanno degli svantaggi. Nella subset selection, si selezionano solo le variabili indipendenti con un buon coefficiente di correlazione ed abbiamo un processo discreto di selezione in

quanto le variabili indipendente vengono selezionate o scartate; ne consegue che il modello sarà meno stabile in quanto più dipendente dalle poche variabili in analisi e quindi meno stabili. Diversamente, la Ridge regression è un processo continuo che rimpicciolisce i coefficienti ma senza portarli a 0, quindi dando un modello più robusto al costo dell'interpretabilità del modello. Least Absolute Shrinkage and Selection Operator, lasso, rimpicciolisce i coefficienti fino allo 0, quindi seleziona le altre variabili indipendenti con coefficienti maggiori allo 0 cercando di ottenere le qualità di entrambe la subset selection e la Ridge regression.[30]

3.3 Risultati

Figure 3.1: Modelli per MSE, $RMSE$ e MAE

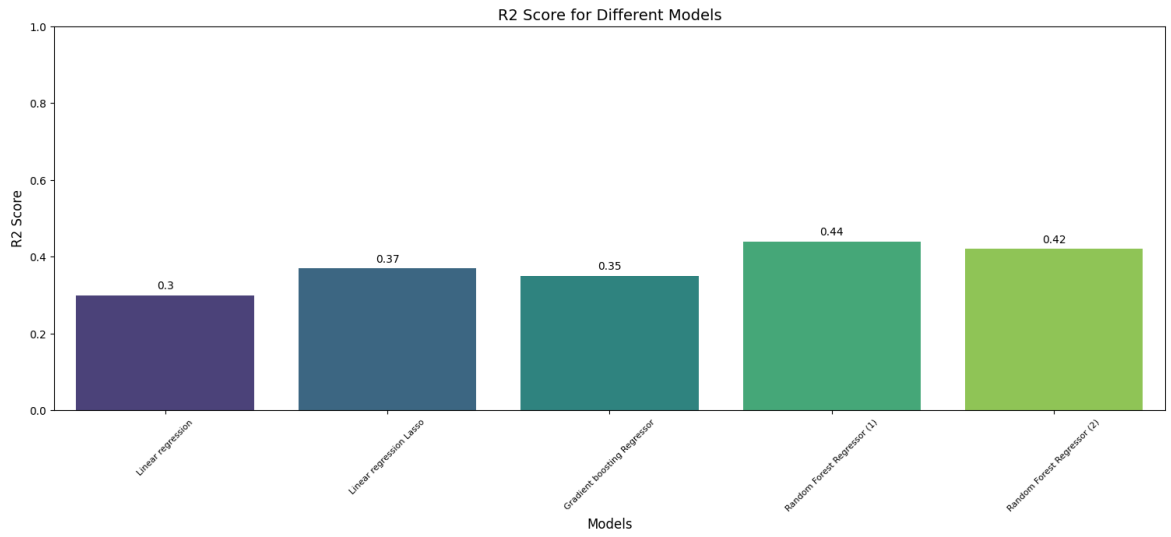

Figure 3.2: Modelli per coefficiente di determinazione R^2

3.3.1 Risultati della regressione lineare

La regressione lineare è la più veloce per quanto riguarda la computazione, che ricordiamo non essere presente nel grafico, ma risulta la peggiore in quanto performance sia per i vari errori, sia per il coefficiente di determinazione R^2. Questo può essere interpretato come una mancanza di forti correlazioni che abbiamo precedentemente visto nel capitolo 2, si vuol precisare che per ottenere il miglior risultato il dataset è stato sottoposto a una subset selection dove sono stati presi in considerazione solo le variabili indipendenti aventi come coefficienti di correlazione con la target variable un valore maggiore di 0.015 e minore di -0.015. I risultati ottenuti migliorano di molto e raggiungono le performance del Gradient Boosting Regressor quando sottoponiamo il dataset alla tecnica di lasso. A beneficiarne di più da questa tecnica è il coefficiente di determinazione che supera quello calcolato con il Gradient Boosting Regressor,e che migliora di 0.07 punti rispetto

alla semplice regressione lineare. L'ultima osservazione, è rappresentata dal seguente grafico della distribuzione dei residui:

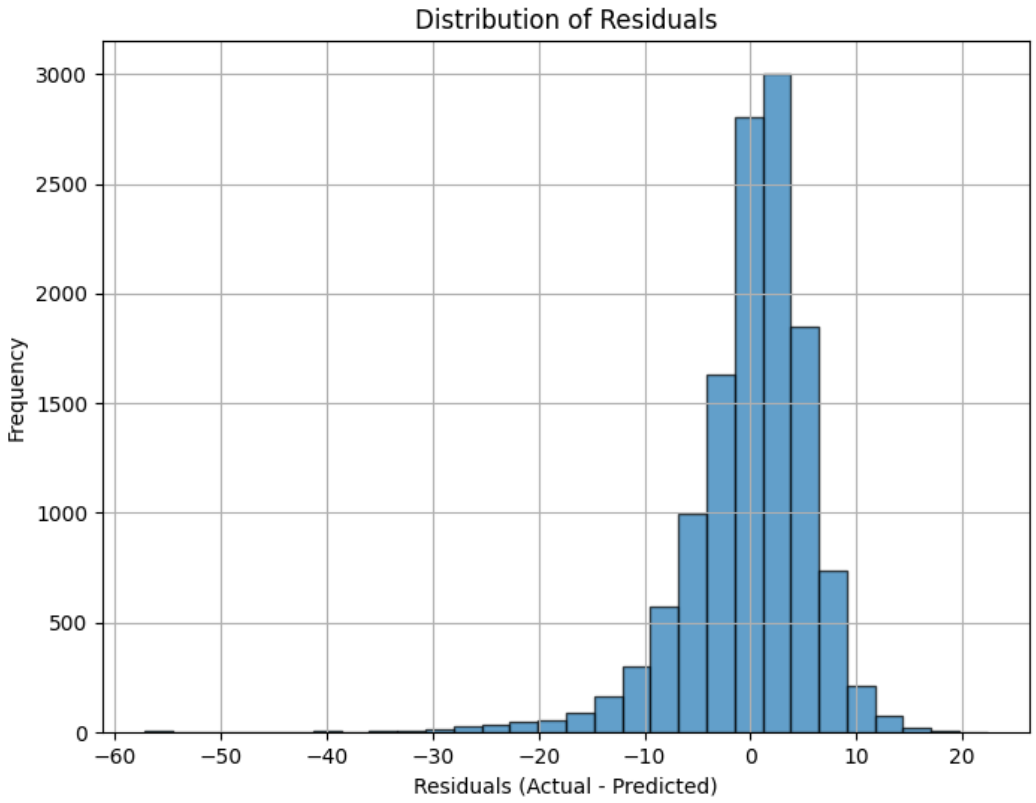

Figure 3.3: Residui della regressione lineare

Un'altra evidenza che la regressione lineare non sta dando il massimo dei risultati; una buona distribuzione dei residui è generalmente caratterizzata dalla Bell curve accentrata allo 0.

3.3.2 Risultati del Gradient Boosting Regressor

I risultati del Gradient Boosting Regressor (GBR) sono molto simili ai risultati della rigressione lineare con lasso e questa similitudine potrebbe essere spiegata considerando la natura dell'algoritmo in questione con la tecnica di

regolarizzazione. Prendendo come riferimento ciò che è stato detto nel capitolo 2.7 riguardo il GBR e il capitolo 3.2 per lasso, parliamo di un modello e una tecnica che come obiettivo hanno quello di pesare tutti i coefficienti; con il primo cerchiamo di scalare con dei coefficienti ogni learner debole (solitamente un albero, come anche in questo caso), con il secondo la medesima cosa ma sulle singole variabili indipendenti.

3.3.3 Risultati del Random Forest Regressor

Nei grafici di riferimento 3.1 e 3.2, si differenziano due tipi di Random Forest Regressor. Il primo, quello leggermente più preciso e con un coefficiente di determinazione maggiore, è stato ricavato utilizzando tutte le variabili a nostra disposizione meno le variabili dipendenti causa di multicollinearità come menzionato nel capitolo 2.5, mentre per il secondo solamente le variabili con i coefficienti di correlazione ottenuti dalla subset selection. I risultati sono molto simili tra i due e l'aumento di dati e miglioria di precisione del dataset (1) rispetto al dataset (2) ci suggerisce che gli alberi di decisione hanno dei pattern che probabilmente non sono ancora stati individuati tramite il feature engineering. Si tornerà su questo punto successivamente, mentre per visualizzare l'aumento del coefficiente di determinazione ottenuto rispetto agli altri modelli si rappresenta una distribuzione degli errori, che con il Random Forest Regressor assume una forma più a "campana" e meglio distribuita rispetto al grafico dei residui ottenuto per la regressione lineare (fig. 3.3).

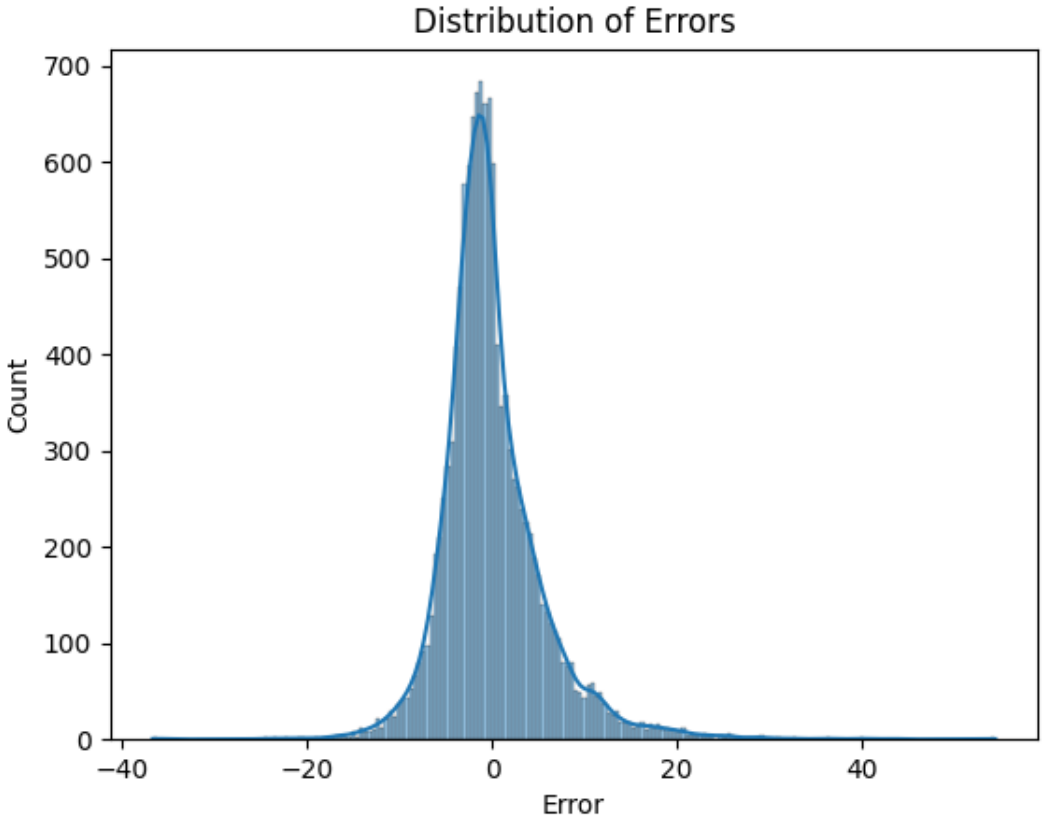

Figure 3.4: Distribuzione errore del Random Forest Regressor

Un ultimo grafico da accompagnare al modello è l'albero decisionale ottimale; in quanto troppo grande da visualizzare per esteso, mostriamo le prime 10 che formano l'albero per osservare:

- Presenza di due variabili ricavate con il feature engineering: 'Category score' e 'Negative score';
- La maggioranza di queste variabili che formano l'albero sono variabili con coefficienti di correlazione non tra i più alti osservati precedentemente nella figura 2.3.

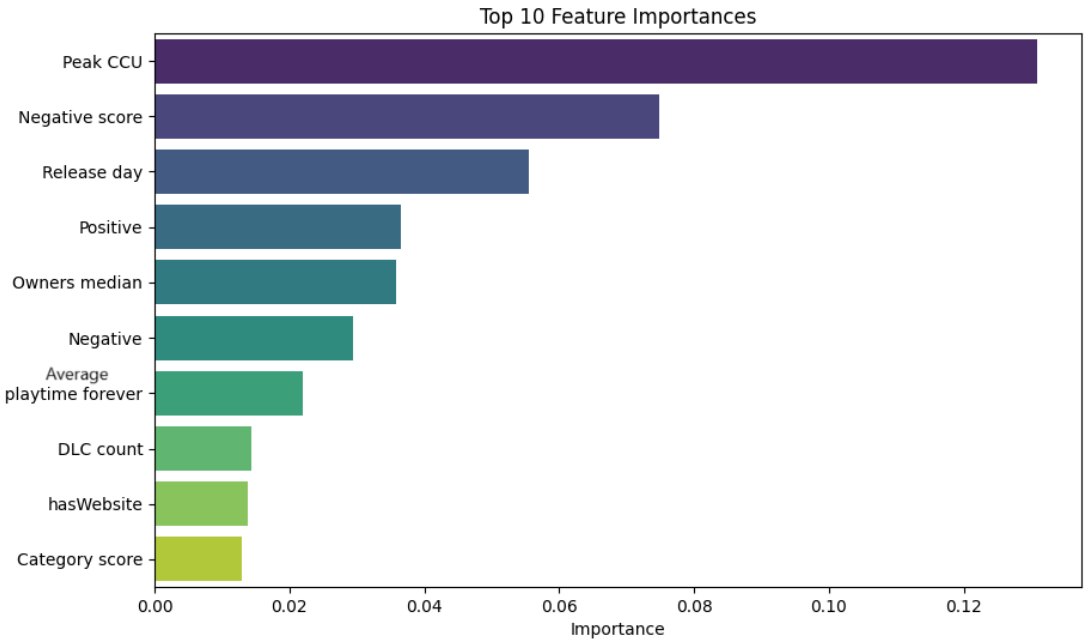

Figure 3.5: Migliori 10 features dell'albero decisionale

3.4 Conclusioni

La complessità della ricerca è prevalentemente concentrata nel feature engineering. Svolta tutta la fase di pre-processing, i risultati ottenuti sono molto imprecisi: sia per i vari tipi di errore, ma soprattutto per i coefficienti di determinazione sui vari modelli. Il valore del coefficiente di determinazione R^2 è stato aumentato e migliorato prevalentemente dall'aggiunta delle diverse variabili ricavate combinate con criterio per trovare correlazioni oltre le semplici variabili 'lavorate' o 'numerate' che già si avevano a disposizione. Il modello migliore, considerando ogni tipo di misurazione per la precisione quali errori e coefficiente di determinazione, risulta essere il random forest regressor. Non sorprendono i risultati in questo, in quanto il modello è incentrato prevalentemente nel trovare e identificare dei pattern meno lampanti e

chiari rispetto alla semplice misura di coefficienti di correlazione; ma a discapito della velocità. Il modello Random Forest Regressor è molto più costoso in termini di costi computazionali e questo difetto potrebbe trascinarsi per un'eventuale distribuzione del modello che potrebbe raccogliere e valutare i dati in real-time. Il Gradient Boosting Regressor è una via di mezzo tra gli altri: i risultati si trovano a metà seppur più vicino alla regressione lineare che al Random Forest Regressor, ma con costi computazionali leggermente aumentati rispetto alla regressione lineare. La regressione lineare al contrario del Random Forest Regressor in termini di velocità è drasticamente migliore in quanto i risultati vengono rilasciati in pochi secondi contro i diversi minuti del Random Forest Regressor; le migliorie portate dal feature engineering hanno dato risultati numerici migliori rispetto alle medesime operazioni sugli altri modelli. Qualche punto chiave per concludere su come procedere per un'eventuale potenziamento e aumento di precisione:

- Feature engineering: al primo posto per importanza, si osserva un dataset estremamente difficile da lavorare per via della mancanza di correlazione tra le variabili. Riuscire ad estrarre l'essenza di ognuna di queste, combinare e valutare dove le multitudini di variabili insieme formano un dato più importante rispetto al singolo è la chiave che ha portato ad aumentare drasticamente i risultati. Per questa ricerca, siamo riusciti ad ottenere più di 700 colonne dalle 39 di partenza. Alcune di esse sono state osservate, valutate e modellate di conseguenza ma non tutte per via del grande numero. Ricercare la correlazione per tutte le variabili decodificate e la loro rappresentazione è una manzione che richiede molto tempo e dedizione con grafici di supporto da analizzare e numeri da valutare.

- Esplorare altri modelli: provare un approccio diverso con altri iperparametri dei modelli già visti ma anche con altri modelli non esplorati in questa ricerca è una possibile strada da percorrere. Il Random Forest Regressor ad esempio ci ha fornito una visuale diversa sulle variabili, partendo non tra le migliori osservate utilizzando i coefficienti di correlazione ma bensì osservando i pattern che alcune variabili assumono tra loro. Non solo per una miglioria del risultato in sé, ma l'osservazione e l'analisi dei vari modelli fornisce automaticamente un paragone e può evidenziare falle o possibili migliorie da applicare anche per gli altri modelli.

- Live-data testing: testare i modelli con dati raccolti in tempo reale, seppur non rientra nello scopo di questa ricerca, è comunque un passo importante per l'analisi: instaurare delle pipeline in grado di raccogliere e lavorare i dati è altrettanto importante per la valutazione di un modello in quanto potrebbe fornire un'ottica più concreta di come questi dati rappresentino la nostra target variable, considerando le valutazioni fatte precedentemente sulla multicollinearità e su quanto risolvere un dataset statico sia differente dall'esprimere concretezza su dati prodotti all'ordine del giorno.

Bibliography

[1] (2024). *Python documentation*. Python Software Foundation.

[2] Alshdaifat, E., Alshdaifat, D., Alsarhan, A., Hussein, F., and El-Salhi, S. M. F. S. (2021). The effect of preprocessing techniques, applied to numeric features, on classification algorithms' performance. *Data*, 6(2).

[3] Bhatia, M. K. (2016). *Data Analysis and its Importance*. IRJAES - International Research Journal of Advanced Engineering and Science.

[4] Brown, S. (2021). *Machine Learning, explained*. MIT Sloan.

[5] Bustos, M. (2023). *Steam Games Dataset*. Kaggle.com.

[6] Clement, J. (2023a). *Number of peak concurrent Steam users worldwide from 2015 to 2023*. Statista.

[7] Clement, J. (2023b). *Video gaming worldwide - Statistics Facts*. Statista.

[8] Clement, J. (2024). *Steam gaming platform - Statistics Facts*. Statista.

[9] Fernando, J. (2024). Fiscal quarters (q1, q2, q3, q4) explained. *Investopedia*.

[10] Gupta, S. (2023). One-hot encoding: A comprehensive guide with python code and examples for effective categorical data representation. *Medium*.

[11] Han J, Pei J, K. M. (2011). *Data Mining: concepts and techniques*. Elsevier.

[12] Herd, J. (2023). *The Global Surge Of Independent Games Development Studios*. Forbes.

[13] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

[14] Komorowski, M., Marshall, D. C., Salciccioli, J. D., and Crutain, Y. (2016). *Exploratory Data Analysis*. Springer International Publishing, Cham.

[15] Le, L. (2023). *A Complete History of Valve: Founding, Hits, and Failures*. History-computer.

[16] Leslie Pack Kaelbling, Michael L. Littman, A. W. M. (1996). *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research.

[17] Li, J. (2019). *A combined approach for predicting sparse variables such as tips ratio and daily precipitation*. UCLA.

[18] Mohammed, M., Khan, M., and Bashier, E. (2016). *Machine Learning: Algorithms and Applications*.

[19] Montelli, C. (2021). What is dlc? understanding downloadable content, a feature of nearly every new game. *Business Insider*.

[20] Montgomery, D. C., Peck, E. A., and Vining, G. G. (2012). *Introduction to Linear Regression Analysis.* Wiley.

[21] Otchere, D. A., Ganat, T. O. A., Ojero, J. O., Tackie-Otoo, B. N., and Taki, M. Y. (2022). Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions. *Journal of Petroleum Science and Engineering*, 208:109244.

[22] Patel, H. (2024). Feature engineering explained. *Built in.*

[23] Sarker, I.H., K.-A. B. S. e. a. (2020). *Cybersecurity data science: an overview from machine learning perspective.* Journal of Big data.

[24] Scarlett, R. (2023). Why python keeps growing, explained. *GitHub blog.*

[25] Schneider A, Hommel G, B. M. (2010). *Linear regression analysis: part 14 of a series on evaluation of scientific publications.* Dtsch Arztebl Int.

[26] Segal, M. R. (2004). Machine learning benchmarks and random forest regression. *UCSF: Center for Bioinformatics and Molecular Biostatistics.*

[27] Shewale, R. (2023). *Steam Statistics For 2024 (Users, Popular Games Market).* Demand Sage.

[28] Sydorenko, I. (2023). *What Is a Dataset in Machine Learning: Sources, Features, Analysis.* Label Your Data.

[29] Talevski, D. (2024). *How Much Is the Gaming Industry Worth in 2024?* Techjury.

[30] Tibshirani, R. (1996). *Regression Shrinkage and Selection via the Lasso.* Wiley for the Royal Statistical Society.

[31] Toto, D. S. (2020). *Price breakdown of a $70 video game: digital vs physical.* KantanGames.

[32] Yoo W, Mayberry R, B. S. S. K. P. H. Q. L. J. J. (2014). *A study of Effects of MultiCollinearity in the Multivariable Analysis.* Int J Appl Sci Technol.

Ringraziamenti

Vorrei ringraziare di cuore tutte le persone che hanno partecipato in questo percorso, a cominciare dalla professoressa Elena Loli Piccolomini per avermi dato la disponibilità di condurre questa tesi e per avermi aiutato durante tutto il percorso di ricerca e stesura della tesi. Un ringraziamento va anche a tutti i docenti del mio corso di laurea con cui ho avuto il piacere di frequentare lezioni, svolgere esami e ricevimenti al fine di migliorare culturalmente come persona e apprendere molto da ognuno di loro. Ringrazio i miei genitori per avermi dato la possibilità di studiare in questa bellissima città, un grazie anche a tutto il resto della famiglia, per il supporto e l'affetto che non è mai mancato. Voglio ringraziare anche i miei colleghi universitari per avermi aiutato non solo nello studio ma anche durante le giornate di lezione, per aver reso la mia esperienza universitaria semplice e migliore. Un ringraziamento speciale anche a tutti i miei amici per aver condiviso momenti bellissimi durante questo percorso nei momenti buoni e avermi aiutato nelle difficoltà nei momenti cattivi.