UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Corso di Laurea in Informatica

Analisi di un modello di Machine Learning per la previsione della categoria dei giochi da tavolo attraverso i valori di Shapley

Relatore: Riccardo Martoglia

Candidato: Giovannoni Luca

Anno accademico 2020/2021

Sommario

INTRODUZIONE

Il progetto "Learning Board Games Mechanics for Computational Thinking (even) in a Social Distancing Context" mira a determinare quali giochi o meccaniche di gioco sono più appropriate per favorire il Computational Thinking (CT), rivolto agli studenti delle scuole superiori e possibilmente laureandi, anche con particolare riguardo all'attuale contesto di Social Distancing (SD). Il metodo di analisi è duplice e innovativo in questo campo:

- Approccio "guidato dagli esperti", ovvero prendendo alcuni giochi da tavolo popolari e mappando (grazie all'aiuto degli esperti) il loro design di gioco per trovare elementi di CT, anche considerando SD.
- Approccio "data-driver" complementare, ovvero applicheremo una data analytics e machine learning e la sua interpretazione per estendere la scala di analisi. L'obiettivo è quello di identificare e scoprire le meccaniche di gioco o le caratteristiche dei giochi più adatte al CT in un contesto SD.

La CT è una competenza molto utile per studiare un problema complesso, comprenderlo e sviluppare soluzioni che sia un essere umano, un computer o entrambi possano comprendere, è considerato come la competenza di base necessaria per diventare un programmatore o informatico di successo.

Le CT comprendono la conoscenza di scomposizione dei problemi, pensiero algoritmico, astrazione e valutazione di possibili soluzioni.

Per iniziare si è indagato su alcuni giochi da tavoli più popolari, mappando il loro design di gioco per costrutti di pensiero computazionale per identificare le meccaniche di gioco più adatte per formare una CT. Gli esperti hanno arricchito anche le mappature con specifiche annotazioni riguardanti la fattibilità o efficacia di ogni meccanica in un contesto SD.

Alcuni membri del gruppo hanno proposto questa decomposizione di CT in sei costrutti:

- CT 1 Logical Reasoning (LOG): coinvolge principalmente ragionamento deduttivo, ma include anche ragionamento induttivo e ragionamento abduttivo
- CT 2 Algorithmic thinking (ALG): la capacità di organizzare il proprio pensiero in una serie di passi logici ripetibili

- CT 3 Generalization (GEN): dividere gli elementi utili da quelli secondari o inutili
- CT 4 Evaluation (EVL): la capacità di testare, valutare il proprio lavoro al fine di migliorare a volta successiva
- CT 5 Patterns (PAT): identificare e classificare i modelli comportamentali, sociali e grafici collegati all'intelligenza visiva.
- CT 6 Decomposition (DEC): capacità di dividere un problema in uno più piccolo che può essere risolto separatamente

Il primo problema che abbiamo dovuto affrontare è stato quello di recuperare una quantità sufficiente di dati da permetterci di fare un'analisi di quest'ultimi e successivamente anche di applicare il machine learning, per fare questo ci siamo affidati al sito BoardGameGeek che racchiude e cataloga milioni di giochi da tavolo con tutte le loro caratteristiche.

In secondo luogo, si è redatta un'analisi dettagliata dei dati raccolti, mostrando le caratteristiche dei giochi raccolti, mostrando insieme di giochi con caratteristiche comuni e un prospetto generale dei dati raccolti.

Fatto questo, grazie al lavoro di Matteo Pontiroli, possiamo mostrare i risultati dell'applicazione di alcuni algoritmi di machine learning con l'obiettivo di addestrare un classificatore per riconoscere le categorie dei giochi da tavolo.

In ultimo utilizzeremo una particolare libreria che permette di analizzare i risultati dei modelli di machine learning creati, mostrando quali caratteristiche hanno spinto un determinato modello a classificare un determinato gioco in una specifica classificazione.

Parte I

Tecnologie utilizzate

1 CONTESTO

I giochi da tavolo stanno diventando sempre più dominanti nella vita di tutti i giorni, questi infatti sono utilizzato non solo per intrattenimento ma anche per educazione, favorire la creatività e molti altri ambiti. Anche nell'ambito tecnologico questi giochi stanno piede, infatti ultimamente si sente sempre più spesso parlare di intelligenze artificiali come giocatori o progettazione di nuovi giochi da tavolo con l'auto anche di computer.

Questa tesi è parte di un progetto più ampio che ha l'obiettivo di impiegare l'analisi dei dati e tecniche di machine learning per comprendere meglio i giochi da tavolo e riuscire a sfruttare questa conoscenza per utilizzare questi giochi nell'ambito sociale.

L'obiettivo di questa parte del progetto è quello di analizzare un modello di machine learning creato per predire le categorie di un gioco da tavolo attraverso la descrizione del gioco stesso, per capire quali scelte ha intrapreso il modello nel classificare un gioco.

Per fare questo è stato necessario l'utilizzo di diverse tecnologie che ci hanno permesso di raggiungere l'obiettivo. A partire dalla libreria Pandas che è stata fondamentale per raccogliere i dati e per fare prime analisi di questi. Agli algoritmi di machine learning che potessero restituire il risultato migliore per i nostri scopi. In ultimo la libreria SHAP che ci ha permesso di analizzare nel dettaglio il modello di machine learning.

2 LIBRERIA PANDAS

Pandas è una libreria Python che permette l'analisi di dati, attraverso strutture dati veloci, flessibili ed espressive per dati "relazionali" o "etichettati".

Inoltre, ha l'obiettivo più ampio di diventare il più potente e flessibile strumento open source di analisi e manipolazione dei dati, disponibile in qualsiasi lingua. Questa libreria è adatta per diversi tipi di dati:

- Dati tabulari con colonne eterogenee, come in una tabella SQL o in un foglio di calcolo Excel
- Dati di serie temporali ordinate e non ordinate (non necessariamente di frequenza fissa)
- Dati a matrice arbitraria (omogenee o eterogenee)
- Qualsiasi altra forma di insiemi di dati osservabili o statistici.

I dati non devono per forza essere etichettati per essere inseriti in una struttura di dati su Pandas.

Le due strutture di dati primari di Pandas sono: serie (una dimensione) o dataframe (due dimensioni), queste permettono di gestire la stragrande maggioranza di casi tipici di utilizzo: finanza, statistica, scienze sociali e molti settori dell'ingegneria.

Altri punti di forza di questa libreria sono:

- Facile gestione dei dati mancanti (rappresentati da NaN)
- Dimensione mutabile: le colonne possono essere inserite e cancellate dal dataframe o oggetti a più di due dimensioni
- Allineamento automatico ed esplicito dei dati: gli oggetti possono essere esplicitamente allineati a un insieme di etichette, oppure l'utente può semplicemente ignorarle e lasciare serie, dataframe, etc.
- Una potente e flessibile funzione di raggruppamento (group by) per eseguire operazioni "split-apply-combine" sui set di dati, sia per l'aggregazione che per la trasformazione di questi
- Semplifica la conversione di dati frammentati e diversamente indicizzati in altre strutture dati Python o NumPy in un oggetto del dataframe
- Slicing, fancy indexing e subsetting intelligente basato su etichette per grossi set di dati
- Marging e joining intuitivo

- Rimodellazione e pivoting flessibile
- Etichettatura gerarchica degli assi (possibilità di avere più etichette per tick)
- Robusti strumenti per il caricamento dei dati da flat file (CVS e delimited), Excel, database, e salvataggio e caricamento dei dati dal formato ultraveloce HDF5
- Funzionalità specifiche per le serie temporali: generazione e conversione dell'intervalli di date e della frequenza, statistiche su finestre mobili, spostamento della data e lagging
- Velocità, diversi algoritmi di basso livello sono stati ampiamente ottimizzati grazie al Cython. Tuttavia, come con qualsiasi altra generalizzazione di solito si sacrifica le prestazioni. Quindi, se sì ci concentra su una caratteristica per la propria applicazione si può essere in grado di creare uno strumento specializzato più veloce

Molti di questi principi servono per affrontare mancanze in altri linguaggi o per ambiente scientifici e di ricerca.

Il modo migliore di pensare alle strutture dati di Pandas è come un contenitore flessibile per i dati dimensionali. Per esempio, il dataframe è un contenitore di series, e una series è un contenitore di dati scalari. Vorremmo poter inserire e rimuovere questi oggetti di questi contenitori in modo simile ad un dizionario Python.

Grazie a questa libreria siamo riusciti a creare molto velocemente una struttura dati che potesse contenere i dati recuperati dal sito BoardGameGeek, dati estremamente eterogenei per i quali non c'è stato possibile creare una struttura dati apposita prima del recupero di questi in quanto non avremmo potuto sapere a priori quali dati ci avrebbe restituito il sito. Infatti, Pandas ci ha permesso di caricare in un dataframe i dati grezzi, per essere analizzati e filtrati. In più, tramite i metodi di plotting messi a disposizione dalla libreria abbiamo potuto fare una prima exploratory data analysis nella quale abbiamo analizzato quali e quanti dati il sito mettesse a disposizione per ogni gioco.

3 MODELLO MACHINE LEARNING

Per la stesura di questo capitolo i ringraziamenti vanno a Matteo Pontiroli che ha analizzato diversi modelli di machine learning con lo scopo di prevedere la categoria o meccanica di un gioco. L'obiettivo è quello di addestrare un modello che, partendo dalla descrizione di un gioco da tavolo, possa determinare le sue categorie. Ogni gioco da tavolo può appartenere a più categorie, per questo motivo la scelta del modello di machine learning è ricaduta su un modello multi-output.

In questa prima parte analizzeremo quali classi si è scelto di utilizzare della libreria sklearn, il pre-processing e la codifica della descrizione dei giochi e i risultati ottenuti dai modelli.

L'analisi dei motivi di queste scelte sono mostrati nella seconda parte.

3.1 MULTILABEL CLASSIFICATION

La multilabel classification è un problema di classificazione dove si vuole classificare ogni campione come appartenente a 'm' classi differenti, selezionandole da 'n' possibili classi, dove 'm' può variare da 0 a 'n' classi. I classificatori multilabel potrebbero trattare le classi simultaneamente, tenendo contro della correlazione fra esse. La libreria utilizzata è la scikit-learn, questa fornisce due classi applicabili ad ogni tipologia di classificatore per fornire supporto alla classificazione multilabel: MultiOutputClassifier e ClassifierChain.

Lo scopo della classe MultiOutputClassifier è di estendere i classificatori in modo tale da renderli in grado di stimare una serie di funzioni target che sono addestrate su una singola matrice di dati e che producono rispettivamente una serie di risposte. Le risposte sono completamente indipendenti le une dalle altre. Le ClassifierChain invece sono un modo di combinare un determinato numero di classificatori binari in un singolo modello multilabel che è capace di sfruttare anche correlazione tra le classi.

Per aumentare la qualità delle feature che verranno fornite al classificatore in fase di training (e relative fasi di test), si può procedere con la text pre-processing delle descrizioni dei boardgames. La struttura dati utilizzata per contenere i dati necessari è un Dataframe di Pandas.

La text processing pensata per questa applicazione consiste nei seguenti step:

- Rimozione dei caratteri speciali / non alfabetici
- Rimozione dei caratteri singoli
- Sostituzione degli spazi multipli con spazi singoli
- Conversione in lowercase
- Rimozione delle stopwords (tokenizzazione e successiva selezione)
- Lemmatizzazione dei token rimanenti
- Ricostituzione di un testo pretendo dai token
- Sovrascrittura delle descrizioni nel Dataframe

3.2 CODIFICA DELLE CATEGORIE DI GIOCO E DELLE DESCRIZIONI

La codifica scelta è la One-Hot Encoding attraverso la classe MultiLabelBinarizer che codificherà in vettori di zero e uni dove, ogni riga, avrà un vettore lungo quante le diverse categorie di gioco nel dataset e nel quale si avrà un uno in corrispondenza di un gioco appartenente a tale categoria, zero altrimenti.

```
OUTPUT
Rows (boardgames): 4916
Columns (features): 185
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

Figura A: un esempio di codifica delle categorie di un gioco

Perché un modello di learning possa funzionare anche per input testuali come la descrizione di un gioco da tavolo, è necessario codificare il testo in maniera tale da trasformare ogni descrizione in una rappresentazione numerica/vettoriale (vettore di feature). Questo è possibile attraverso la classe CountVectorizer e il relativo componente che permette di trasformare le term frequency (TF) in misure TF-IDF calcolando l'Inverted Document Frequency e moltiplicandola alla TF. Tale componente prende il nome di TfidfVectorizen.

Il risultato è un vettore riga per ogni boardgame formato da tremila feature. Ogni feature si riferisce ad un termine presente nel vocabolario costituito dall'insieme delle descrizioni dei giochi. Tale vocabolario è ridotto a tremila parole e per ognuna di esse è calcolata la corrispondente TF-IDF per ogni boardgame.

```
OUTPUT
0.0
0.1627588551946906
0.0
0.10575208435727144
0.0
0.0
0.07312027953053658
...
0.0
0.07958909990683884
0.0
```

Figura B: un esempio di output della codifica delle feature di un gioco da tavolo

3.3 ADDESTRAMENTO E TEST DELLE PERFORMANCE DEL MODELLO RANDOM FOREST

Dopo la divisione del dataset in training set e test set si è proceduti con il lancio dell'algoritmo attraverso la classe RondomForestClassifier, il quale, su un set di 50.000 giochi, ha ottenuto come risultato una predizione corretta nel 16.8% delle volte. In valore assoluto non è un buon risultato ma, se consideriamo il dataset limitato, soddisfacente.

4 LIBRERIA SHAP

Si vuole analizzare il modello di machine learning per comprendere l'impatto delle feature sulla predizione della categoria. L'idea è di considerare ogni feature come un giocatore di una squadra e il set di dati come la squadra stessa. Ogni giocatore (feature) dà il proprio contributo al risultato della squadra, e la somma di questi contributi dà il valore della variabile obiettivo. (curiosità: SHAP è basato sul gioco Shapley Values)

Per produrre questa analisi si è deciso di utilizzare la libreria SHAP (SHapley Additive exPlanations), questa libreria permette di analizzare i principali modelli di machine learning, utilizza un metodo per rappresentare le previsioni individuali di un modello. Gli autori di SHAP hanno proposto due approcci di stima il primo denominato KernelSHAP, ovvero un approccio di stima alternativo basato su kernel per i valori di "Shapley" ispirato ai modelli di surrogati locali. L'altro approccio è lo TreeSHAP, ottimizzato per i modelli basati sugli alberi. Inoltre, la libreria viene fornita di molti metodi di interpretazione globali basati su aggregazione di valori Shapley. Prima di addentrarci nel dettaglio della libreria bisogna spiegare un punto cardine sulla quale poggia la libreria, i valori Shapley.

4.1 VALORI DI SHAPLEY

Supponiamo il seguente scenario: si è addestrato un modello di apprendimento automatico per prevedere i prezzi degli appartamenti. Per un certo appartamento si prevede 300.000€ e dovete interpretare questa previsione. L'appartamento è al secondo piano, ha una superficie di 50 m^2 e si trova al secondo piano, ha un parco nelle vicinanze e sono vietati i gatti.

Sapendo che la previsione media per tutti gli appartamenti è di 310.000€, quanto ha contribuito ogni valore delle caratteristiche alla predizione rispetto alla predizione media?

La risposta è semplice per i modelli di regressione lineare. L'effetto di ogni caratteristica è il peso della caratteristica per il valore della stessa. Questo funziona solo a causa della linearità del modello, per i modelli più complessi abbiamo bisogno di una soluzione diversa. Per esempio, la libreria LIME suggerisce modelli locali per stimare gli effetti. Un'altra soluzione viene dalla teoria dei giochi cooperativi: il valore di Shapley p è un modello per assegnare i pagamenti ai giocatori a seconda del loro contributo al pagamento totale. I giocatori cooperano in una coalizione e ricevono un certo profitto da questa cooperazione.

Il "gioco" è il compito di predizione per una singola istanza del set di dati, il "guadagno" è la previsione effettiva per questa istanza meno la predizione media di tutte le altre istanze, i "giocatori" sono i valori delle caratteristiche dell'istanza che collaborano per ricevere il "guadagno" (ovvero predire un certo valore). Nel nostro esempio i valori delle caratteristiche sono: vicinanza con un parco, sono vietati i gatti, è di 50 m^2, è al secondo piano. Queste hanno collaborato per ottenere la predizione di 300.000€. Il nostro obiettivo è quello di spiegare la differenza tra la media delle previsioni (310.000€) e la previsione effettiva (300.000€): una differenza di -10.000€. Una risposta potrebbe essere: la vicinanza con il parco ha contribuito con 30.000€, la dimensione di 50 m^2 ha contribuito con 10.000€, è al secondo piano ha

contribuito con 0€, i gatti sono vietati con -50.000€. Sommando i contributi si ottiene -10.000€, ovvero la previsione finale meno la previsione media.

4.2 Come si calcola il valore di Shapley per ogni caratteristica

Il valore di Shapley è il contributo marginale medio del valore di una caratteristica su tutte le possibili coalizioni. Nell'esempio seguente valutiamo il contributo del valore della caratteristica "i gatti sono vietati" quando viene aggiunta alla coalizione "vicinanza con un parco" e "dimensione di 50 m^2". Supponiamo che "vicinanza con un parco", "dimensione di 50 m^2" e "i gatti sono vietati" siano in una coalizione, estraendo a caso un altro appartamento dai dati e usando il suo valore per la caratteristica "è al x piano". Il valore della caratteristica "è al secondo piano" è stata sostituita con il valore "è al primo piano" estratta a caso, poi prevediamo il valore dell'appartamento con questo nuovo valore della caratteristica. Il secondo passo è rimuovere "i gatti sono vietati" dalla coalizione e sostituiamo il suo valore con un valore casuale della caratteristica di un appartamento estratto a caso, per esempio "i gatti sono ammessi". Prevediamo ora il prezzo dell'appartamento per la coalizione "vicinanza con il parco" e "dimensione di 50 m^2": 320.000€. Il contributo della caratteristica "il gatto è vietato" è di 310.000€ - 320.000€ = -10.000€. Questa stima dipende dai valori dell'appartamento estratto a caso che è servito da "donatore" per il gatto e dai valori della caratteristica "dimensione". Otterremmo stime migliori se ripetiamo questa fase di campionamento e facessimo la media dei contributi. Ripetiamo questo calcolo per tutte le possibili coalizioni. Il valore di Shapley è a media di tutti i contributi marginali di ogni possibile coalizione. Il tempo di calcolo aumenta esponenzialmente con il numero di caratteristiche. Una soluzione per mantenere il tempo di calcolo gestibile è di calcolare i contributi solo per alcuni campioni delle possibili coalizioni.

Nel nostro caso tutte le possibili coalizioni sono:

- Nessun valore delle caratteristiche
- "vicinanza con un parco"
- "50 $m2$ "
- "si trova al secondo piano"
- "vicinanza con un parco" + "50 $m2$ "
- "vicinanza con un parco" + "si trova al secondo piano"
- "50 $m2$ " + "si trova al secondo piano"
- "vicinanza con un parco" + "50 $m2$ " + "si trova al secondo piano"

Se stimiamo i valori di Shapley per tutti i valori delle caratteristiche, otteniamo la distribuzione completa della previsione (meno la media) tra i valori delle caratteristiche.

4.3 IL VALORE DI SHAPLEY IN DETTAGLIO

Siamo interessanti a come ogni caratteristica influenza la predizione di un'istanza di dati. In un modello lineare è facile calcolare i singoli effetti, in un modello lineare la predizione per un'istanza di dati è la seguente:

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

Dove x è l'istanza per la quale vogliamo calcolare i contributi. Ogni x_j è il valore della caratteristica con $j = 1, \ldots, p$. β_j è il peso corrispondente della caratteristica j.

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

Il contributo ϕ_j della j-esima caratteristica della predizione $\hat{f}(x)$ è:

Dove $E(\beta_j X_j)$ è la stima dell'effetto medio della caratteristica j. Il contributo è la differenza tra l'effetto della caratteristica meno l'effetto medio, se sommiamo tutti i contributi delle caratteristiche di un'istanza il risultato è il seguente:

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(\hat{f}) &= \sum_{j=1}^{p} (\beta_j x_j - E(\beta_j X_j)) \\ &= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} E(\beta_j X_j)) \\ &= \hat{f}(x) - E(\hat{f}(X))\end{aligned}$$

Questo è il valore previsto per il punto di dati X meno il valore medio previsto. I contributi delle caratteristiche possono essere negativi. Il valore di Shapley è una soluzione per calcolare i contributi delle caratteristiche per le singole previsioni per un qualsiasi modello di apprendimento automatico.

4.4 CARATTERISTICHE

Il valore di Shapley è definito da una funzione di valore val dei giocatori S. Questo valore è il contributo di una caratteristica ponderato sommando tutte le possibili combinazioni di valori delle caratteristiche:

$$\phi_j(val) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(val\left(S \cup \{j\}\right) - val(S)\right)$$

dove S è un sottoinsieme delle caratteristiche utilizzate nel modello, x è il vettore dei valori delle caratteristiche dell'istanza da spiegare e p il numero delle caratteristiche. $valx(S)$ è la predizione per i valori delle caratteristiche dell'insieme S che sono marginalizzati sulle caratteristiche che non sono incluse nell'insieme S:

$$val_x(S) = \int \hat{f}\left(x_1, \ldots, x_p\right) d\mathbb{P}_{x \notin S} - E_X(\hat{f}\left(X\right))$$

In realtà si eseguono più integrazioni per ogni caratteristica che non è contenuta in S. Un esempio concreto: il modello di apprendimento automatico lavora con 4 caratteristiche $x1$ $x2$ $x3$ $x4$ e valutiamo la predizione per la coalizione S composta dai valori delle caratteristiche $x1$ $x2$:

$$val_x(S) = val_x(\{1,3\}) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}\left(x_1, X_2, x_3, X_4\right) d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}\left(X\right))$$

Anche se sembra simile ai contributi delle caratteristiche del modello lineare il valore di Shapley è molto diverso. Il valore della caratteristica è il valore numerico o categorico di una caratteristica e di un'istanza; il valore di Shapley è il contributo della caratteristica alla predizione; la funzione di valore è la funzione di payout per le coalizioni id giocatori (valori della caratteristica).

Il valore di Shapley è l'unico metodo di attribuzione che soddisfa le proprietà di: efficienza, simmetria, dummy e additività che insieme possono essere considerate una definizione di payout equo. Dimostriamo che le caratteristiche elencate sopra siano effettivamente intrinseche nel valore di Shapley:

- Efficienza: i contributi delle caratteristiche devono sommarsi alla differenza di previsione per x e la media

$$\sum_{j=1}^{p} \phi_j = \hat{f}\left(x\right) - E_X(\hat{f}\left(X\right))$$

- Simmetria: i contributi di due valori di due caratteristiche j e k dovrebbero essere gli stessi se contribuiscono equamente a tutte le possibili coalizioni.

$$val(S \cup \{j\}) = val(S \cup \{k\})$$

Per ogni $S \subseteq \{1, \ldots, p\} \setminus \{j, k\}$ allora $\phi_j = \phi_k$

- Dummy: una caratteristica j che non cambia il valore previsto (indipendentemente da quale coalizione di valori di caratteristica viene aggiunta) dovrebbe avere un valore di Shapley di 0.

$$val(S \cup \{j\}) = val(S)$$

Per ogni $S \subseteq \{1, \ldots, p\}$ allora $\phi_j = 0$

- Additività: per un gioco con vincite combinate $val+val^+$ i rispettivi valori di Shapley sono i seguenti: $\phi_j+\phi_j^+$. Supponiamo di aver addestrato un modello Random Forest, significa che la predizione è una media di molti alberi decisionali. La proprietà dell'additività garantisce che per un valore di caratteristica è possibile calcolare il valore di Shapley per ogni albero individualmente, poi fare la media e ottenere il valore di Shapley per la caratteristica in tutta la foresta (quindi ogni albero).

4.5 STIMA DEL VALORE DI SHAPLEY

Tutte le possibili coalizioni (insiemi) di valori di caratteristiche devono essere valutate con e senza la caratteristica j-esima per calcolare il valore esatto di Shapley. Quando si hanno molte caratteristiche trovare la soluzione esatta potrebbe diventare problematico, poiché il numero di coalizioni possibili aumenta esponenzialmente con l'aggiunta di caratteristiche. Un possibile soluzione è l'approssimazione con il campionamento Monte-Carlo:

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}\left(x_{+j}^m\right) - \hat{f}\left(x_{-j}^m\right) \right)$$

dove $\hat{f}\left(x_{+j}^m\right)$ è la predizione per x, ma con il numero casuale di valori delle caratteristiche sostituiti da valori casuali in un certo punto z, eccetto il rispettivo valore della caratteristica j. Il vettore x_{-j}^m è quasi identico al vettore x_{+j}^m, ma i valori di x_j^m sono anche presi dal campione z. Ognuna di queste M nuove istanze sono assemblate da due istanze. Si noti come nel seguente algoritmo, l'ordine delle caratteristiche rimane nella stessa posizione nel vettore quando viene passato alla funzione di predizione. L'ordine all'interno del vettore è usato come un "trucco", infatti dando alle caratteristiche un nuovo ordine otteniamo un meccanismo casuale che ci aiuta a mettere insieme le istanze.

Per le caratteristiche che appaiono a sinistra della caratteristica x_j prendiamo i valori delle osservazioni originali, e per le caratteristiche a destra prendiamo i valori casuali da un'istanza casuale.

Approssimazione della stima del valore di Shapley per il valore di una singola caratteristica

- Output: valore di Shapley per la caratteristica j-esima
- Parametri: numero delle iterazioni M, istanza di interesse x, indice della caratteristica presa in osservazione j, matrice dei dati X, modello di apprendimento automatico f
- $\forall\, \mathbf{m} = \mathbf{1}, \dots, \mathbf{M}$
- Prendi un'istanza casuale z nella matrice X
- Ordina l'istanza x: $x_0 = (x_1, \dots, x_j, \dots, x_p)$
- Ordina l'istanza z: $z_0 = (z_1, \dots, z_j, \dots, z_p)$
- Costruire due nuove istanze:
    - Istanza con j: $x_{+j} = (x_1, \dots, x_{j-1}, x_j, z_{j+1}, \dots, z_p)$
    - Istanza senza j: $x_{-j} = (x_1, \dots, x_{j-1}, z_j, \dots, z_p)$
    - Calcoliamo il contributo marginale: $\phi_j(x) = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$
- Calcolo il valore di Shapley come la media: $\phi_j(x) = 1/m \sum_{m=1}^{M} (\phi_j^m)$

In primo luogo, si selezione un'istanza di interesse x, una caratteristica j e il numero di iterazioni M. Per ogni iterazione di selezione un'istanza casuale z dai dati e si genera un ordine casuale delle caratteristiche. Vengono create due nuove istanze combinando i valori dell'istanza di interesse x e del campione z dai dati e si genera un ordine casuale delle caratteristiche. Le due nuove istanze vengono create combinando i valori dell'istanza di interesse x e del campione z. l'istanza x_{+j} l'istanza di interesse, ma tutti i valori dopo la caratteristica j sono sostituiti dai valori delle caratteristiche del campione z. l'istanza x_{-j} è la stessa della precedente ma in più ha la caratteristica j sostituita dal valore della caratteristica j nel campione z. La differenza nella predizione viene calcolata $\phi_j(x) = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$, viene fatta poi la media di tutte le differenze $\phi_j(x) = 1/m \sum_{m=1}^{M} (\phi_j^m)$.

La media pesa implicitamente i campioni in base alla distribuzione di probabilità di X. Questa procedura va poi ripetuta per ciascuna delle caratteristiche per ottenere tutti i valori di Shapley.

4.6 VANTAGGI

La differenza tra la predizione e la predizione media è che la predizione media è equamente distribuita tra i valori delle caratteristiche dell'istanza (proprietà di efficienza). Questa proprietà distingue i valori di Shapley da altri metodi come LIME. LIME, infatti, non garantisce che la predizione sia equamente distribuita tra le caratteristiche. Il valore di Shapley permette spiegazioni contrastive. Invece di confrontare una previsione con la previsione media all'interno del set di dati, si può confrontare con un sottoinsieme o anche con un singolo punto di dati.

4.7 SVANTAGGI

Il valore di Shapley richiede molto tempo di calcolo. Nel 99.9% dei problemi del mondo reale è fattibile solo la soluzione approssimativa. Un calcolo esatto del valore di Shapley è computazionalmente costoso perché ci sono molteplici possibili coalizioni dei valori delle caratteristiche e l'assenza di una caratteristica deve essere simulata inventando istanze casuali, questo aumenta la varianza della stima dei valori di Shapley. Questo problema può essere attenuato campionando le coalizioni e limitando il numero delle iterazioni M. Diminuendo M si diminuisce il tempo di calcolo ma aumenta la varianza del valore di Shapley. M dovrebbe essere abbastanza grande da stimare accuratamente i valori di Shapley, ma anche abbastanza piccolo da completare il calcolo in un tempo ragionevole. Come molti altri metodi di interpretazione basati sulla permutazione, il metodo del valore di Shapley soffre dell'inclusione di istanze di dati non realistiche quando queste sono correlate. Per simulare che il valore di una caratteristica sia mancante da una coalizione, marginalizziamo quella caratteristica. Questo si ottiene campionando i valori della distribuzione marginale della caratteristica. Questo funziona se le caratteristiche sono indipendenti.

Ma se le caratteristiche fossero dipendenti allora potremmo campionare valori di caratteristica che non hanno senso per questo caso. Una soluzione potrebbe essere quella di permuta delle caratteristiche correlate insieme e ottenere un valore di Shapley reciproco per loro. Un altro adattamento è il campionamento condizionato: cioè le caratteristiche sono campionate condizionatamente alle caratteristiche che sono già nel gruppo. Mentre il campionamento condizionale risolve il problema degli irrealistici, viene introdotto un nuovo problema: i valori risultanti non sono più valori di Shapley poiché violano l'assioma della simmetria.

4.8 SHAP

L'obiettivo di SHAP è spiegare la previsione di un’istanza X calcolando il contributo di ciascuna caratteristica alla previsione. Il metodo SHAP calcola i valori di Shapley della teoria dei giochi della coalizione, i valori delle caratteristiche di un'istanza di dati agiscono come attori in una coalizione. I valori di Shapley ci dicono come distribuire equamente il "payout" (ovvero la previsione) tra le caratteristiche. Un giocatore può essere un valore di caratteristica individuale o un gruppo di valori caratteristico (per esempio per spiegare un’immagine si può raggruppare più pixel in un "super pixel" e la previsione distribuita tra di loro). Un'innovazione che SHAP porta è che il valore di Shapley è rappresentato come un metodo di attribuzione di funzionalità additiva, cioè un modello lineare.

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

Dove g è il modello di spiegazione, z' il vettore di coalizione, M è la massima dimensione della coalizione, ϕ appartenente ai numeri reali è l'attribuzione di una caratteristica alla caratteristica j, ovvero il valore di Shapley. Quello che noi chiamiamo "vettore di coalizione" è chiamato "caratteristica semplificata". In questo vettore di coalizione una caratteristica è rappresentata da 1 se è presente, 0 altrimenti. Per calcolare i valori di Shapley simuliamo che solo alcuni valori di caratteristica sono presenti e alcuni non lo sono. La rappresentazione come modello lineare di coalizioni è un trucco per il calcolo delle ϕ. Presa un'istanza di interesse X il vettore di coalizione per X in cui sono a 1 tutte le caratteristiche presenti. In questo modo la formula si semplifica a:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

I valori di Shapley sono l'unica soluzione che soddisfa le proprietà di: efficienza, simmetria, dummy e additività. SHAP soddisfa anche questi requisiti poiché calcola i valori di Shapley. Nel paper di SHAP si possono trovare discrepanze tra le proprietà di Shapley e quelle di SHAP; infatti, quest'ultimo descrive le tre seguenti proprietà:

- Local Accuracy
- Missingness: ovvero che le caratteristiche mancanti ottengono un'attribuzione di 0. Questa proprietà non è tra le proprietà "normali" di Shapley, è stata aggiunta poiché una caratteristica mancante potrebbe un valore di Shapley arbitrario sena danneggiare la proprietà locale di precisione. (in pratica questo è rilevante solo per le funzionalità che sono costanti
- Consistency: questa proprietà indica che se un modello cambia, quindi anche il contributo di una caratteristica aumenta o rimane lo stesso, allora il valore di Shapley aumenterà o rimarrà lo stesso

4.9 TREESHAP

SHAP mette a disposizione anche una variante per modelli di apprendimento automatico basato su alberi come: decision tree, random forest e gradient boosted trees. TreeSHAP è stato introdotto come un'alternativa veloce e specifica per il modello KernelSHAP, ma si è scoperto che può produrre attribuzioni di caratteristiche poco intuitive. Il TreeSHAP definisce la funzione di valore usando l'aspettativa condizionata $E_{X_s|X_c}\left(\hat{f}(x|x_s)\right)$ invece che l'aspettativa marginale. Il problema è che le caratteristiche che non hanno alcuna influenza sulla funzione di predizione f possono comunque ottenere una stima TreeSHAP diversa da 0, questa stima diversa da 0 può esserci quando la caratteristica è correlata con un'altra caratteristica che ha effettivamente un'influenza sulla predizione.

Il TreeSHAP usa l'aspettativa condizionale $E_{X_s|X_c}\left(\hat{f}(x|x_s)\right)$ per stimare gli effetti, vediamo con un esempio come possiamo calcolare la predizione attesa per un singolo albero, un'istanza x e un sottoinsieme di caratteristiche S. Se S fosse l'insieme di tutte le caratteristiche allora la predizione del nodo cadrebbe sull'istanza x, quindi sarebbe la predizione attesa. Se S fosse un insieme vuoto useremo la media ponderata delle previsioni di tutti i nodi terminali. Se S contenesse alcune ma non tutte le caratteristiche ignoreremmo le predizioni dei nodi non

raggiungibili. Con irraggiungibile intendiamo che il percorso decisionale che porta a questo nodo contraddice i valori x_s. Dei restanti nodi terminali facciamo la media delle previsioni ponderate per le dimensioni del nodo (cioè il numero di campioni di allenamento in quel nodo). La media dei nodo). La media dei nodi terminali restanti, ponderata per il numero di istanze per nodo, è la previsione attesa per x dato S. Il problema è che dovremmo applicare questa procedura per ogni possibile sottoinsieme di S dei valori delle caratteristiche. TreeSHAP calcola in tempo polinomiale invece che esponenziale. L'idea alla base è di spingere tutti i possibili sottoinsiemi S giù per l'albero allo stesso tempo. Per ogni nodo decisionale dobbiamo tenere traccia del numero di sottoinsiemi. Questo dipende dai sottoinsiemi nel nodo padre e dalla caratteristica di divisione.

Per esempio, quando la prima divisione di un albero è sulla caratteristica $x3$, allora tutti i sottoinsiemi che contengono quella caratteristica andranno in un nodo. I sottoinsiemi che non contengono la caratteristica $x3$ vanno entrambi in un nodo con peso ridotto. Sfortunatamente sottoinsiemi di dimensioni diverse hanno pesi diversi. L'algoritmo deve tener traccia del peso complessivo dei sottoinsiemi in ogni nodo, questo complica di molto l'algoritmo. Il calcolo può essere esteso a più alberi grazie alla proprietà di additività dei valori di Shapley, questi valori sono di un insieme di alberi sono la media (pesata) dei valori di Shapley dei singoli alberi.

Parte 2

Data exploratory analysis e analisi del modello di machine learning

5 CREAZIONE DEL DATASET

Per la costruzione di un modello di machine learning è necessaria una grossa mole di dati. Nel nostro caso i dati sono stati recuperati dal sito BoardGameGeek (BGG), il quale raccoglie e cataloga milioni di giochi da tavolo suddividendoli in categorie, meccaniche di gioco, votazioni, etc. Per il recupero di questi dati è stato necessario implementare un algoritmo di scarping. Poiché per utilizzare la API è necessario l'ID, l'algoritmo recupera dalla pagine HTML l'ID di un gioco per poi inserirlo nella richiesta della API. La API restituisce un file XML con i dati del gioco. L'algoritmo è stato implementato utilizzando la libreria Beautiful Soup.

5.1 SCRAPING E BEAUTIFUL SOUP

La discriminante nella scelta dei giochi da inserire nel nostro dataset è stato l'attributo "game rank", un voto dato dagli stessi creatori del sito che valuta un gioco. La scelta di questo attributo discriminante per rientrare nel nostro dataset non è stata fatta casualmente; infatti, i giochi con una votazione più bassa (alcune volte addirittura assente) sono spesso giochi poco conosciuti o molto vecchi, per i quali si hanno poche informazioni e a volte errate. Per il nostro algoritmo di machine learning è estremamente importante che ogni gioco abbia lo stesso numero di informazioni in modo da avere il dataset il più "pulito" possibile.

Per recuperare gli ID dei giochi con il geek rank più alto è stato necessario utilizzare un algoritmo di scraping. L'algoritmo naviga una tabella di giochi ordinata per game rank, da questa, grazie alla libreria Beautiful Soup, è stato possibile cercare all'interno del codice HTML gli ID dei vari giohi. L'ID è stato recuperato dal tag anchor (<a>) che contiene il link alla scheda di un gioco, il tag è stato identificato grazie alla classe CSS utilizzata per dargli lo stile. Recuperato il link del anchor è stato sufficiente fare un parsing della stringa per recuperare l'ID da inserire nell'API.

Un problema presentatosi durante questa fase è l'errore http 429 (Too many request) che il server restituiva quando si facevano più di una richiesta ogni 5 secondi, di conseguenza è stato necessario l'aggiunta di un timer che scandisse le richieste, con il conseguente rallentamento del recupero dei dati. A questo punto si è creato un file XML per ogni gioco, essendo il formato più comodo da utilizzare in futuro attraverso la libreria Pandas.

5.2 DATI RECUPERATI

I dati recuperati per ogni gioco sono il fulcro per capire come analizzare un gioco; infatti, per ogni gioco BGG mette a disposizione un vasto numero di dati.

I dati recuperati per ogni gioco sono i seguenti:

- **Type**: tipo di gioco, i tipi di gioco possono essere:
    - Boardgame
    - Boardgame expansion
    - Boardgame accessory

    - Videogame
    - Rpg item
    - Rpg issue (for periodicals)
- **Name**: nome del gioco, in alcuni casi un gioco può avere anche più di un nome, in questo caso il type “primary” o “secondary” indicano le diverse versioni del nome del gioco
- **Description**: descrizione del gioco, questo attributo diventerà fondamentale per l’allenamento del machine learning
- **Yearpublished**: anno di pubblicazione
- **Minplayer**: numero minimo di giocatori
- **Maxplayer**: numero massimo di giocatori
- **Suggested_numplayer**: numero di giocatori consigliato dagli utenti, questo attributo si sviluppa in sotto-attributi che indicano quali voti hanno espresso gli utenti per il numero di giocatori suggerito
- **Playingtime**: tempo di gioco medio
- **Minaplaytime**: tempo di gioco minimo
- **Maxplaytime**: tempo di gioco massimo
- **Minage**: età minima per giocare
- **Suggested_playerage**: come per il numero suggerito di giocatori anche in questo caso questo attributo è composto da sotto-attributi che indicano i suggerimenti degli utenti sull’età minima.
- **Language_dependence**: indica quanto il gioco è direttamente collegato alla capacità linguistica di un giocatore
- **Version**: versione del gioco, per esempio le versioni del gioco nelle varie lingue oppure versioni successive o rivisitate
- **Stats**: ranking e rating del gioco, come:
    - **Rating**:
        - **User rated**
        - **Average**
        - **Bayesian average (media bayesiana)**
    - **Ranks:**
        - **Stddev (scarto quadratico medio)**
        - **Median**

- **Owned**
- **Trading**
- **Wanting**
- **Wishing**
- **Num comment**
- **Num weights**
- **Avarange weight**

- **Historical:** storico del gioco
- **Marketplace:** dati del marketplace, quindi tutte le possibilità di acquisto, le informazioni riguardanti i prezzi, la data in cui è stato inserito, il prezzo e le condizioni
- **Comments:** commenti degli utenti sul gioco
- **Rating comments:** rating dei commenti degli utenti
- **Boardgamecategory:** per categorie si intendono alcuni elementi che ricorrono spesso nel gioco, intesi come civiltà, elementi storici real- mente esistiti, fatti realmente accaduti e molti altri che non vengono elencati. Alcuni esempi possono essere Aviation / Flight, Miniatures, Wargame, World War II, Action / Dexterity, Fantasy
- **Boardgamemechanic:** sono le meccaniche presenti nel gioco. Un gioco può possedere più meccaniche come drafting, set collection, sudden death ending**.**

Mostriamo un esempio di dati raccolti e visualizzati attraverso Pandas.

	type	id	name	description	yearpublished	minplayers	maxplayers	playingtime	minplaytime	maxplaytime	minage
0	boardgame	833	For the People	(from GMT website:)For the People is a grand s...	1998	2	2	360	360	360	12

boardgamecategory	boardgamemechanic	boardgamefamily	boardgamedesigner	boardgameartist	boardgamepublisher
American Civil War,Wargame	Campaign / Battle Card Driven,Dice Rolling,Poi...	Country: USA,Players: Two Player Only Games	Mark Herman	Rodger B. MacGowan,Kurt Miller,Mark Simonitch	The Avalon Hill Game Co,Devir,GMT Games

boardgameexpansion	No necessary in-game text	Some necessary text - easily memorized or small crib sheet	Moderate in-game text - needs crib sheet or paste ups	Extensive use of text - massive conversion needed to be playable	Unplayable in another language
No expansions avaiable	0.0	1.0	3.0	7.0	1.0

6 EXPLORATORY DATA ANALYSIS

Exploratory data analisys è un approccio all'analisi di un dataset di dati per riassumere le loro caratteristiche principali. Nel nostro caso si è analizzato in particolare la descrizione dei giochi, essendo l'unico campo utilizzato per prevedere la categoria di un gioco.

Questa prima analisi è funzionale a capire se il dataset raccolto contenga informazioni sufficienti e aggiornate. Considerando che il sito BGG contiene milioni di giochi non è raro che molti di questi non abbiamo tutti i dati necessari al nostro modello come una descrizione poco curata o a volte assente per quelli più vecchi.

Per la creazione di questa analisi è stata fondamentale la libreria Pandas descritta precedentemente.

6.1 ANALISI PRELIMINARE DEL DATASET

Per avere una prima impressione dei dati è importante capire in quale lasso temporale sono stati pubblicati i giochi del nostro dataset. Infatti (come detto precedentemente) su BGG sono presenti un'enorme quantità di giochi, ma molti sono molto vecchi o poco conosciuti.

Mostriamo ora il grafico della distribuzione dell'anno di pubblicazione dei giochi appartenenti al nostro dataset.

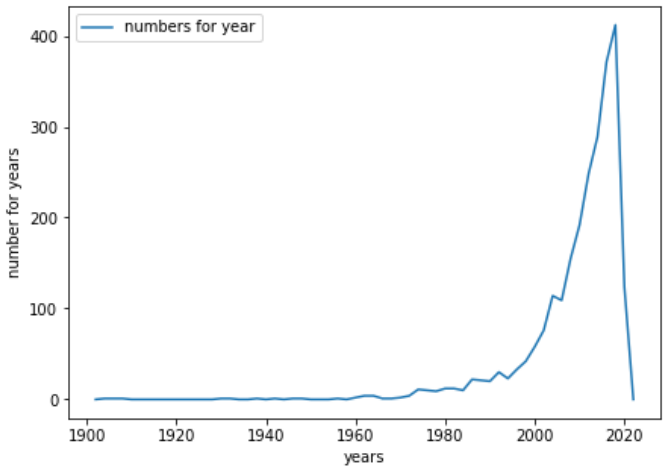

Figura C: grafico della distribuzione della pubblicazione dei giochi negli anni

Il grafico mostra la distribuzione dei giochi del nostro dataset negli anni. Notiamo come la maggior parte si assesta tra gli anni '80 fino a giorni nostri. Questo è dovuto dal fatto che negli anni '80 c'è stato il boom dei giochi da tavolo. Da quegli anni in poi il numero di giochi da tavolo prodotto è aumentato vertiginosamente, e di conseguenza anche il voto dei giochi essendo il mercato più prospero.

Mostriamo ora il grafico della distribuzione dei voti dei giochi.

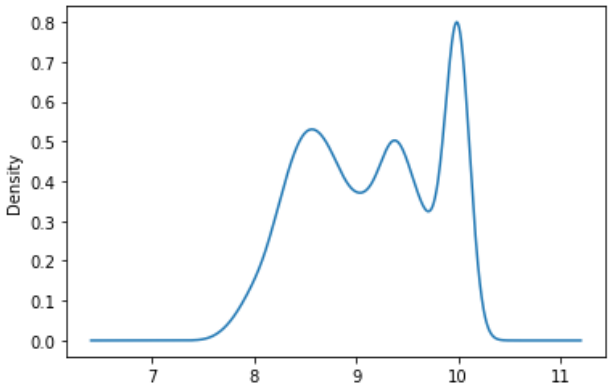

Figura D: distribuzione dei voti dei giochi

Come il grafico mostra, i giochi del nostro dataset hanno una media di voti tra 8 e 10, con una grossa percentuali di voti estremamente alti. Questo porta sia ad un vantaggio che ad uno svantaggio: il vantaggio è di avere la certezza che le informazioni sui giochi siano decisamente pulite, poiché i giochi in questione saranno sicuramente ancora supportati ed utilizzati. Lo svantaggio è la mancanza di giochi meno utilizzati ma che comunque avrebbero potuto dare un contributo alla nostra ricerca.

In ultimo mettiamo in correlazione gli anni di pubblicazione dei giochi con i loro voti.

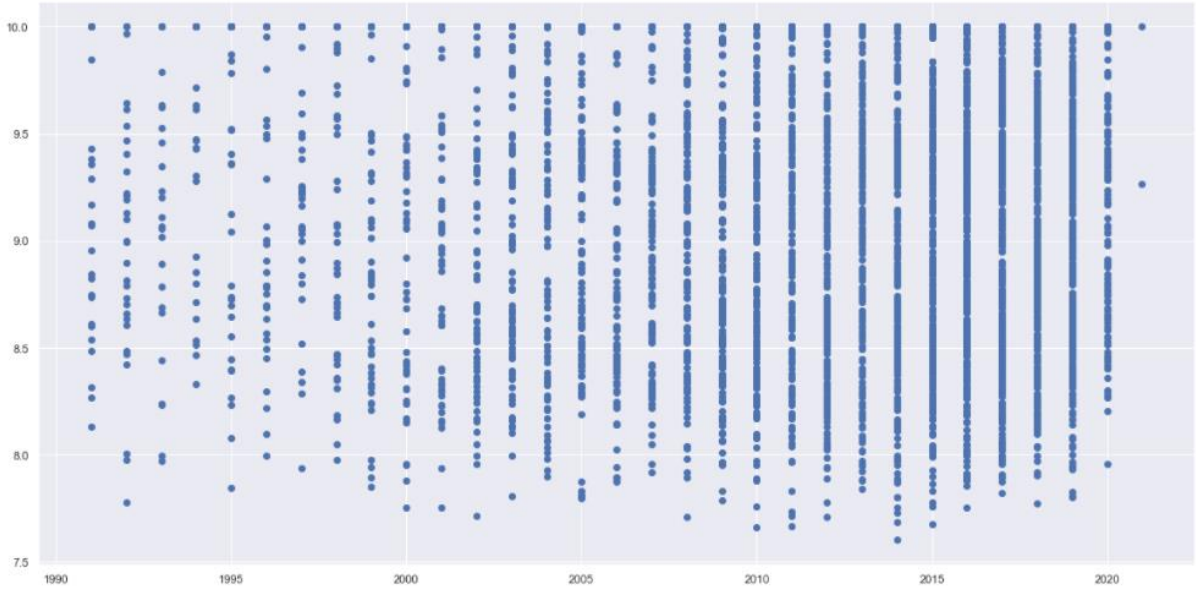

Figura E: grafico della distribuzione dei voti per anno

Il grafico conferma la nostra supposizione, infatti con il passare degli anni i produttori di giochi da tavolo hanno migliorato la qualità di questi, grazie anche a un mercato fertile che richiedeva giochi più raffinati.

Grazie a questo breve sguardo del dataset possiamo capire che i giochi a nostra disposizione siano recenti e ad oggi ancora giocati e supportati. Quindi possiamo ritenere le informazioni raccolte buone per la creazione del modello.

6.2 ANALISI DELLE CATEGORIE E DESCRIZIONI

Essendo la previsione della categoria alla base del modello di machine learning attraverso la descrizione dei giochi è importante analizzare questi due parametri nel dettaglio.

Partiamo mostrando le 20 categorie più preponderanti nel nostro dataset.

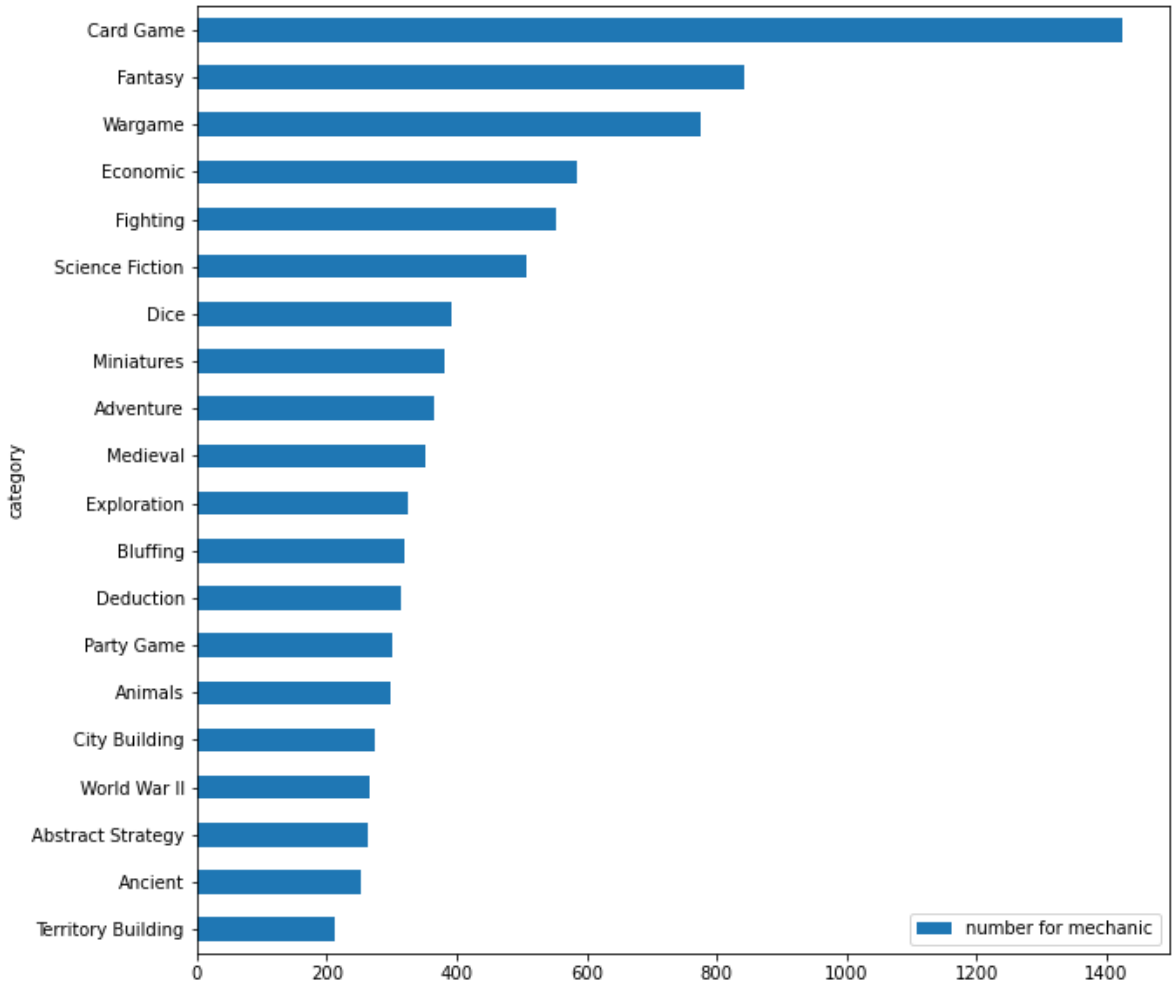

Figura F: 20 categorie più presenti nel dataset

L'istogramma di frequenza ci mostra purtroppo un dataset estremamente sbilanciato. La categoria "Card Game" è la più diffusa, presente in circa 1400 giochi su 5000. Anche "Fantasy"

e “Wargame” sono molto diffuse, assestandosi tra gli 800 e 100 giochi nella quale sono presenti. Dalla settima in posizione “Dice” in poi si assestano tutte tra 200 e 400 giochi nella quale sono presenti. Quindi se dovessi scegliere delle categorie da prendere come esempi prenderei sicuramente le prime tre, infatti essendo le più presenti saranno sicuramente anche quelle con una previsione del modello migliore.

Il campo Boardgamecategory è composto da più categorie per gioco, questo è stato il motivo per la quale ci siamo spostati da un problema di previsione a singolo output ad un problema di previsione multioutput. Questo accade per il fatto che ogni boardgame può appartenere ad una o più categorie di gioco; pertanto, l’output desiderato sarebbe una lista di categorie da assegnare al gioco.

Per quanto riguarda la descrizione, questa si rivela molto sporca; infatti, sono presenti diversi caratteri speciali come virgolette, punteggiatura varia, parentesi, trattini, etc… Questi caratteri ai fini della text classification andranno rimossi. Sono presenti inoltre più spazi consecutivi, anche quest’ultimi andranno filtrati. Un problema notato durante la fase di analisi è capire se la descrizione abbia le informazioni necessarie per categorizzare i giochi, queste infatti non sono particolarmente lunghe ma abbastanza dettagliate almeno quanta basta per riuscire a costruire un modello di machine learning. In qualsiasi caso il problema è passato in secondo piano, poiché solo l’applicazione dell’algoritmo ti permette di capire se la descrizione è sufficiente per categorizzare i giochi.

7 ANALISI DEL MODELLO DI MACHINE LEARNING

In questo capitolo analizzeremo il modello di machine learning attraverso la libreria Python SHAP. Il modello scelto per l'analisi è il Random Forest, essendo quello che ha restituito i risultati migliori. A causa del limite imposto dalla mia strumentazione per poter analizzare il modello è stato necessario ridurre alcuni parametri: la profondità massima degli alberi decisionali è stata ridotta a 5, il numero di feature massime per gioco ridotto a 1000 (ovvero 1000 termini della descrizione per gioco, in particolare sono state scelte in base al TF-IDF).

7.1 DIFFERENZE NEL CALCOLO DELL'IMPORTANZA DI UNA FEATURE

Il modello di machine learning mette a disposizione la possibilità di mostrare l'impatto delle feature in tutte le previsioni.

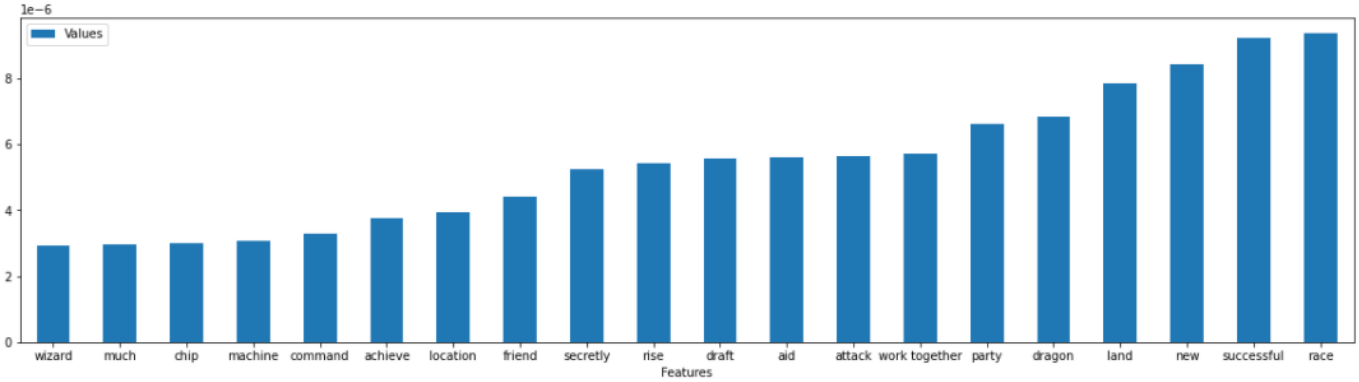

Figura E: feature più impattanti sulle previsioni

L'importanza delle feature è fornita dall'attributo feature_importances_ del modello di machine learning e sono calcolate come la media e la deviazione standard dell'accumulo della diminuzione dell'impurità all'interno di ciascun albero. Questo grafico ci dà un primo impatto su quali feature sono state decisive per la classificazione dei giochi ma non ci permettono di analizzare nel dettaglio i motivi per la quale queste sono importanti.

Ora ci viene in aiuto la libreria SHAP, in particolare l'oggetto TreeExplaner che, come spiegato nella parte 1, ci permette di analizzare nel dettaglio il modello. La classe mette a disposizione un metodo rapido per stimare i valori di SHAP per i modelli ad albero o insiemi di alberi, in base a diverse ipotesi sulla dipendenza delle caratteristiche.

L’idea dietro l’importanza delle caratteristiche di SHAP è semplice: le caratteristiche con un valore di Shapley assoluto maggiore sono più importanti. Quindi volendo mostrare l’importanza globale, facciamo la media dei valori assoluti di Shapley per ogni caratteristica:

$$\frac{1}{n} I_j = \sum_{i=1}^{n} |\phi_j^{(i)}|$$

Successivamente, ordiniamo le caratteristiche per importanza decrescente e mostriamole in un grafico:

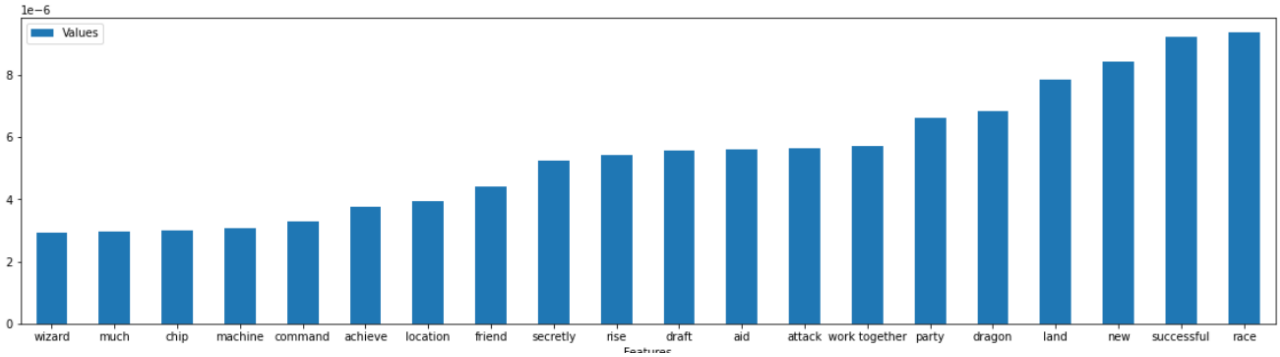

Figura F: grafico delle caratteristiche ordinate per media del valore di Shapley.

L’importanza di una caratteristica è misurata come il valore medio dei valori assoluti di Shapley. Per esempio, la caratteristica “Successful” (valore di Shapley 9.6371717829772e-06) e “Race” (valore di Shapley 9.574247329171205e-06) modificano la probabilità assoluta di una determinata predizione rispettivamente del ~9.6e-03 e ~9.5e-03 punti percentuale.

La media globale dei valori di Shapley è un’alternativa al valore delle caratteristiche date dal modello. L'importante differenza tra le due misurazioni risiede nella natura di queste. Infatti, il valore di Shapley si basa sull'importanza degli attributi di una caratteristica, mentre il valore delle caratteristiche date dal modello (ovvero il valore dato dalla funzione di permutazioni) è basato sulla diminuzione delle prestazioni del modello.

7.1 ANALISI GLOBALE

L'obiettivo di questa analisi è di trovare le regole che hanno portato alla classificazione delle categorie. Cercheremo di capire quali feature hanno impattato e tramite l'analisi delle categorie più presenti (quindi quelle per la quale il modello ha avuto più informazioni per la previsione) estenderemo i ragionamenti a tutte le altre categorie.

Come detto prima le categorie prese in considerazione per l'analisi sono: Card Game, Wargame e Fantasy.

Applichiamo ora il summary plot visto in precedenza ad un caso più specifico, in particolare analizziamo la categoria Fantasy:

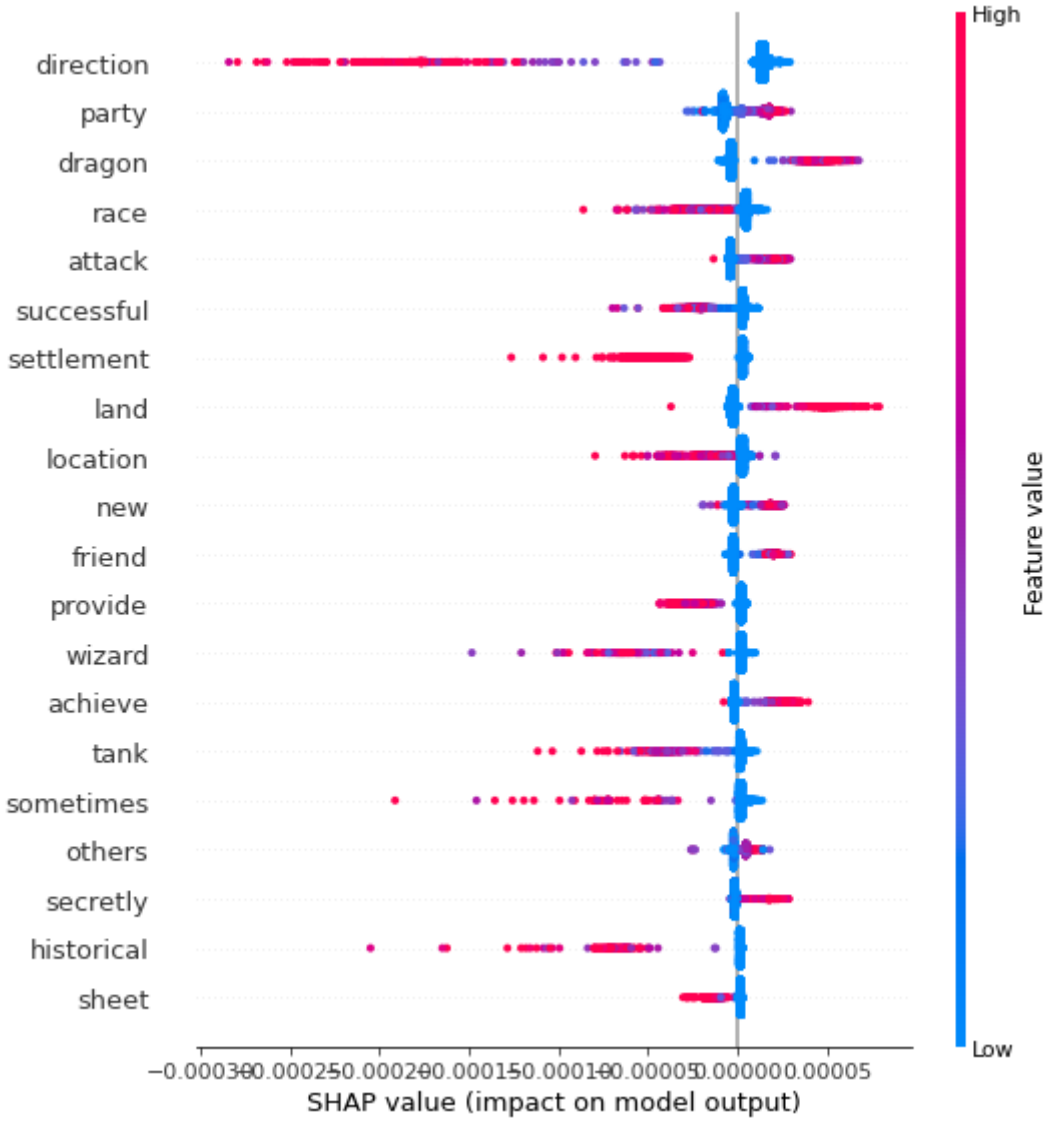

Figura H: Summary plot delle feature della categoria Fantasy

Il grafico mostra sull'asse Y le varie feature più impattanti, sulla X i valori di Shapley mentre il colore rappresenta il valore di TF-IDF. Per esempio, la feature "direction" ha un valore TF-IDF alto in corrispondenza di un valore di Shapley negativo. Ciò significa che la presenza della feature nella descrizione del gioco impatta negativamente sulla previsione, spostandola lontana dalla categoria Fantasy.

Possiamo notare come sono presenti vocaboli significativi per la categoria, come: Dragon, Wizard, attack. Analizzando, per esempio la feature "dragon", notiamo come all'aumentare del TF-IDF aumenta anche il valore di Shapley (quindi l'impatto positivo della parola nella previsione). Mentre se analizziamo la feature "wizard" vediamo come la proporzione è inversa, ovvero all'aumentare del TF-IDF vediamo il valore di Shapley diminuisce diventando negativo, questa feature allontana la previsione dalla categoria fantasy; quindi, la feature wizard si troverà sicuramente in diverse categorie e non è determinante per la categoria Fantasy. Effettivamente categorie come: Mediaval o Mythology è molto probabile che contengano riferimenti a Wizard.

Ragionando sulla feature Dragon, SHAP ci mette a disposizione un grafico chiamato Dependence Plot che permette di vedere la relaizone tra il valore di una feature (nel nostro caso il TF-IDF) e il valore di SHAP nel caso specifico della categoria Fantasy

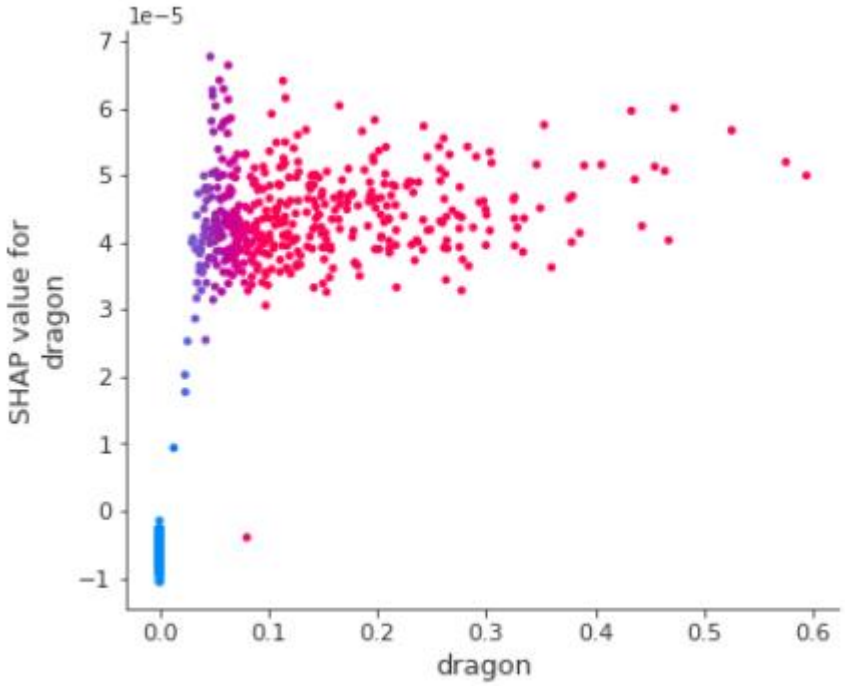

Figura I: dependence plot, mostra la relazione tra il TF-IDF e i valori di Shapley

Ogni punto di questo grafico è una previsione di un gioco della categoria Fantasy. Sull'asse X si trova il valore TF-IDF mentre sull'asse Y il valore di Shapley. Analizzando il grafico possiamo vedere che quando il valore di TF-IDF è sullo 0, l'impatto della feature Dragon sulla previsione della categoria Fantasy è negativa (quindi spinge la previsione lontano dalla

categoria Fantasy). Mentre ad un valore di TF-IDF alto l'impatto della feature aumenta drasticamente, sintomo che la feature porta la predizione verso la categoria Fantasy.

Analizziamo ora la categoria Wargame mostrando il suo summary plot.

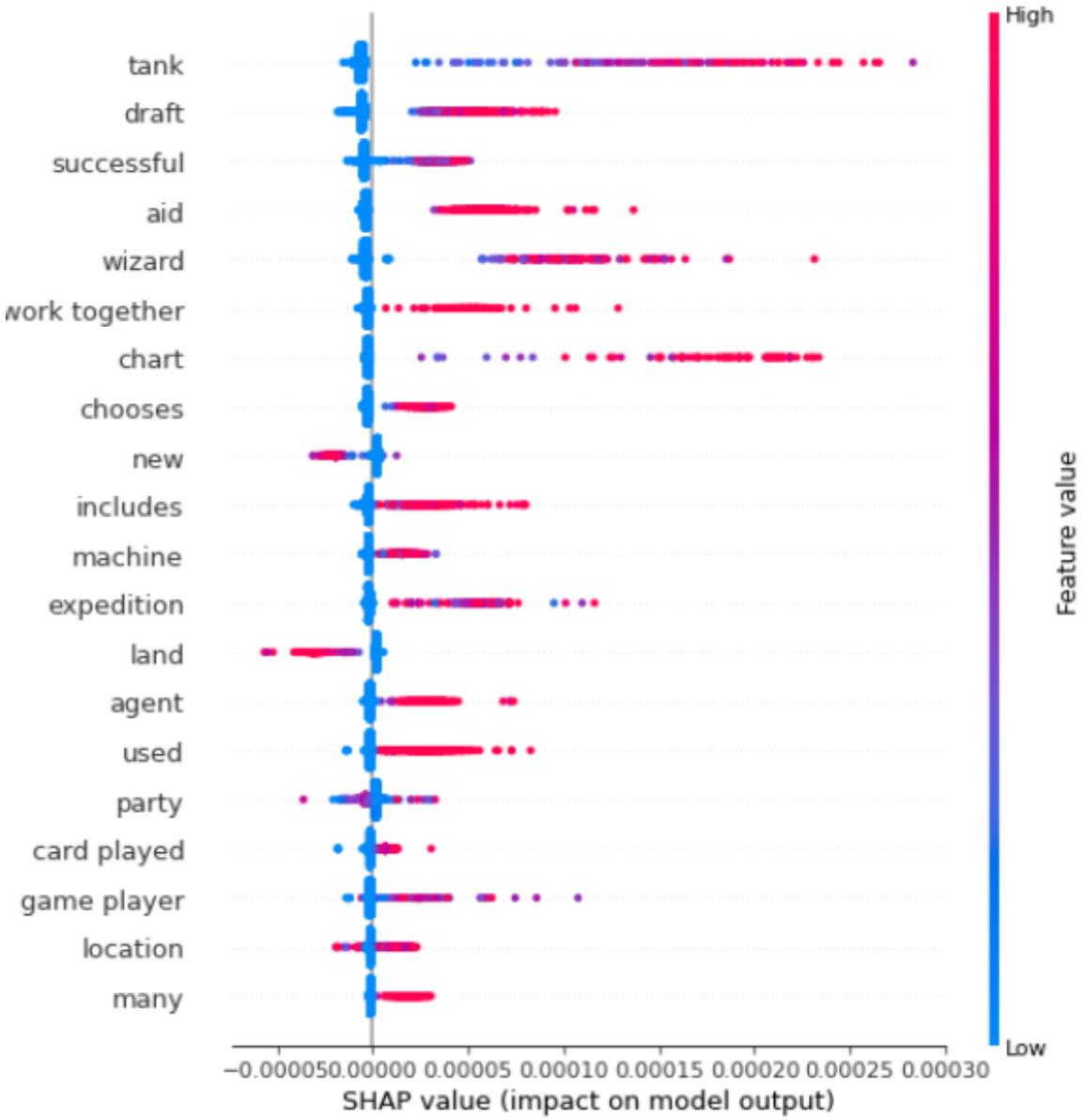

Figura L: summary plot della categoria Wargame

Anche nel caso della categoria Wargame notiamo come le feature rispecchino il tipo di categoria, infatti la feature Tank è predominante nella scala delle feature. Si può notare come all'aumentare del valore di TD-IDF (quindi la parola tank è molto presente in questa categoria ma molto poco nelle altre) aumenti il valore di SHAP calcolato. Inoltre, ci sono altre feature che rispecchiano la categoria come: Machine, agent. Anche queste due, come la feature tank, hanno un elevato TD-IDF e di conseguenza un elevato valore di SHAP. Possiamo notare inoltre una feature in comune con la categoria Fantasy ovvero la feature Wizard che, al contrario del caso precedente, qui all'aumentare del valore di TD-IDF aumenta anche il valore di SHAP.

Questo è dovuto dal fatto che la categoria Wargame è spesso in coppia con la categoria Fantasy (effettivamente guardando il nostro dataset di giochi possiamo notare come i giochi della categoria Wargame molto spesso si trovano anche nella categoria Fantasy).

Allo stesso modo della categoria precedente proviamo ad analizzare nel dettaglio l'impatto della feature tank nella previsione della categoria.

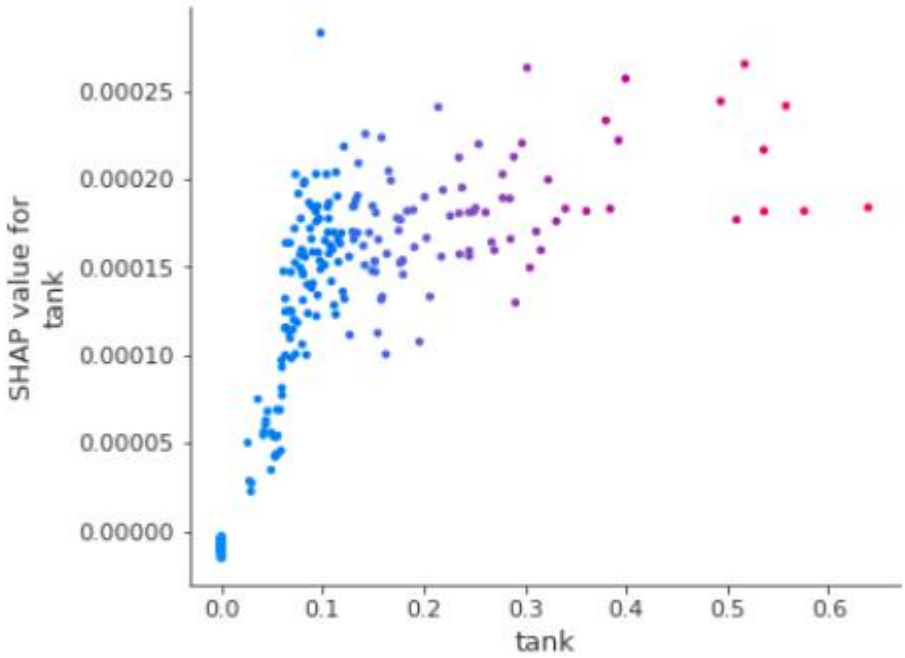

Figura M: dependence plot per la feature Tank nella categoria Wargame.

Anche in questo caso, come in quello precedente, la feature Tank ha impattato molto sulla previsione della categoria Wargame. Infatti, se il valore di TF-IDF è 0 il valore di SHAP (e quindi l'impatto della feature) è nullo. Mentre al crescere del TF-IDF anche il corrispettivo valore di SHAP aumenta. Possiamo anche notare che la maggior parte dei punti si ritrova tra i valori di TF-IDF più bassi, questo è dovuto dal fatto che questa feature sicuramente è presente su almeno altre 2 categorie molto simili a questa: World War II e World War I. SHAP mette a disposizione anche la possibilità di vedere la relazione tra due feature, per esempio abbiamo teorizzato dal summary plot che la feature Agent e la feature Tank siano essenziali per determinare la categoria Wargame, mettiamole a confronto.

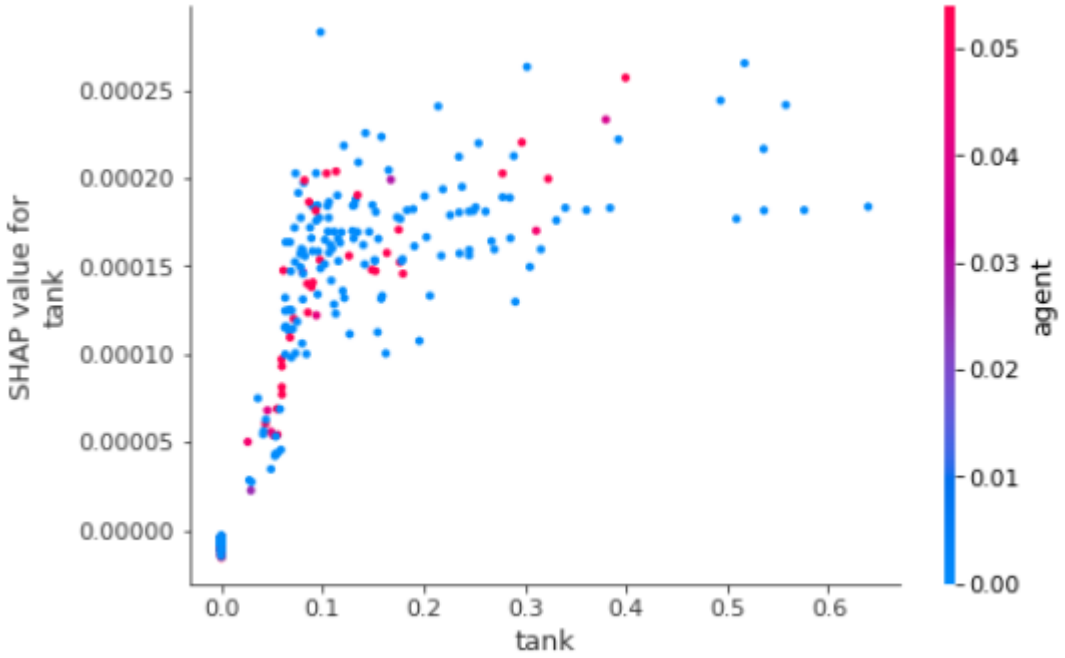

Figura N: dependence plot tra le feature Tank e Agent della categoria Wargame

Il colore dei puntini rappresenta il valore di SHAP di agent collegata al valore di Tank. Possiamo vedere come più o meno si rispecchino, infatti ad un elevato valore di SHAP di tank corrisponde un valore medio / alto della feature Agent, sintomo del fatto che queste due feature spostino in modo simile la previsione verso la categoria Wargame. Proviamo ora, solo per curiosità a mettere a confronto la feature Tank (che sposta la previsione verso la categoria Wargame) con la feature New (che possiamo evincere dal grafico prima spinge contro la previsione della categoria wargame).

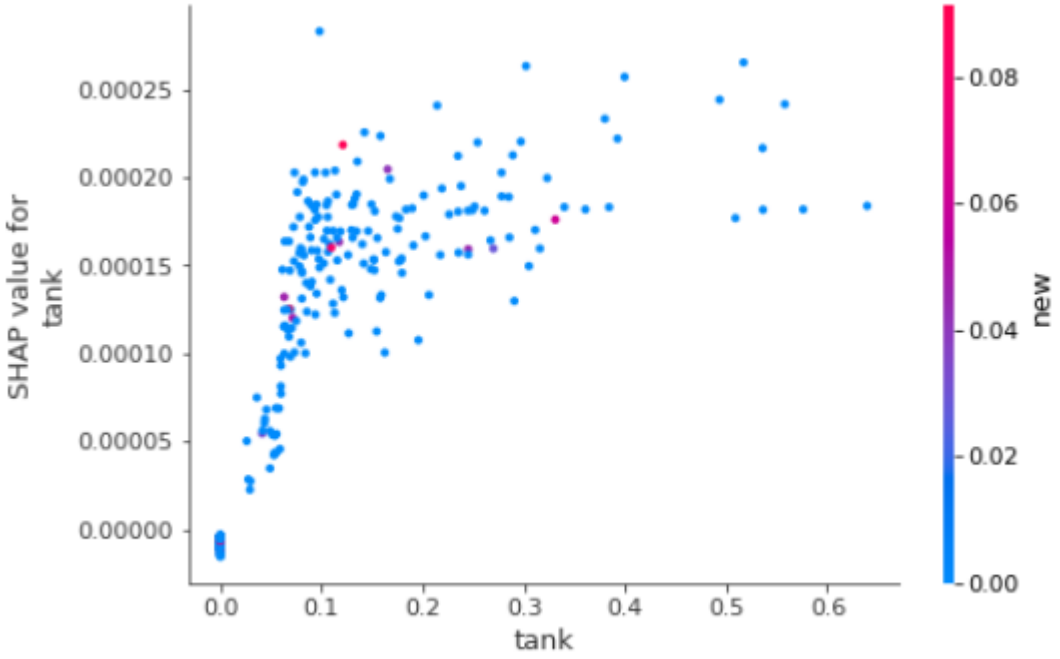

Figura O: dependence plot tra le feature Tank e New della categoria Wargame

In effetti è ben visibile come la feature New ha valori di SHAP inversi rispetto a Tank; infatti, possiamo vedere come nei punti con un valore di SHAP più alto per Tank il valore di SHAP di new sia molto basso (colore azzurro), mentre (anche se non è visibile per una sovrapposizione di punti) quando il valore di SHAP di Tank tende a 0, il valore di SHAP di New sia alto. Quindi la feature New si "oppone" alla previsione della categoria Wargame.

Mostriamo ora il summary plot per la categoria Card Game

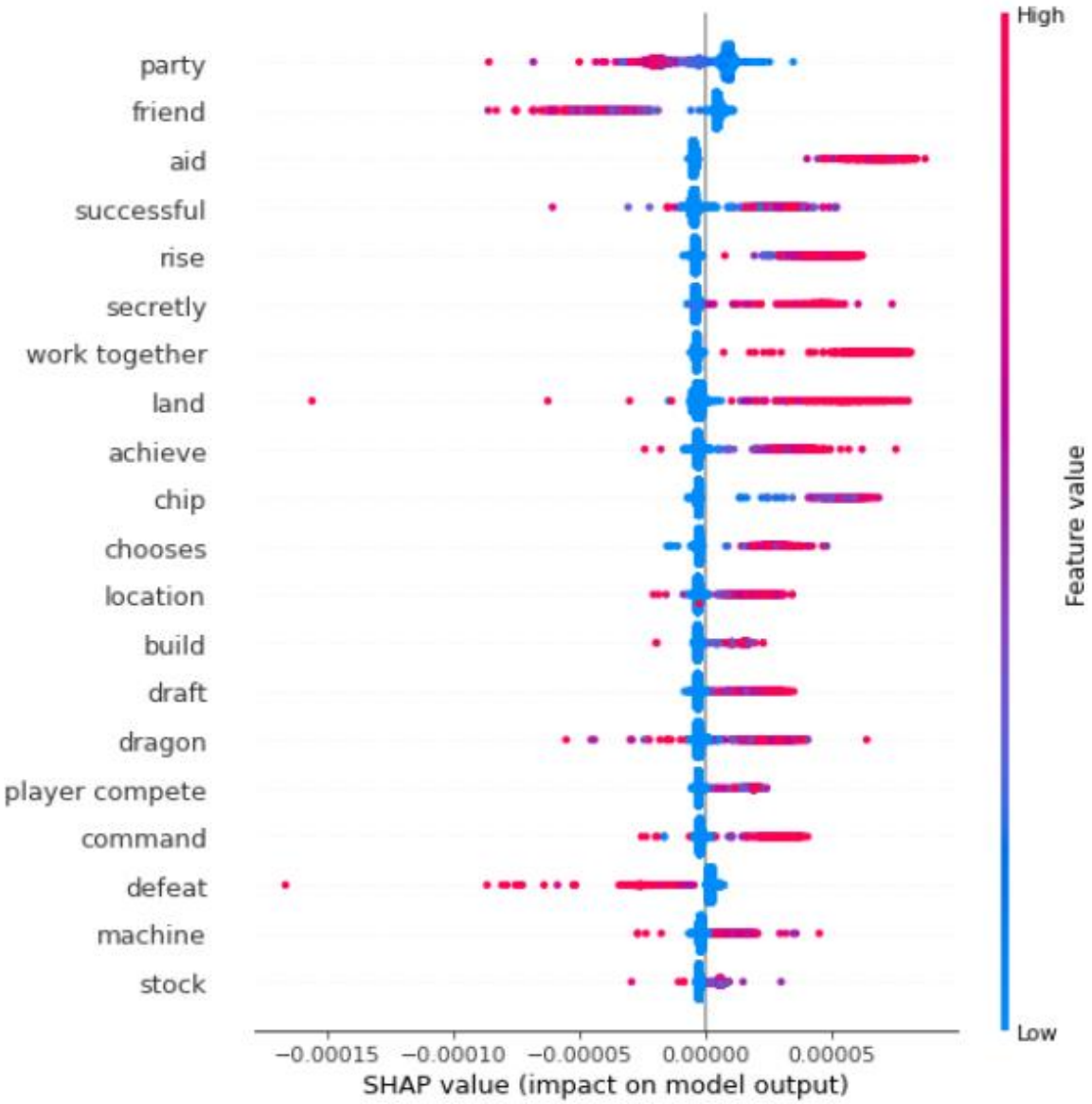

Figura P: summary plot della categoria Card Game

Analizziamo ora il grafico della categoria Card Game, possiamo notare come una delle feature che ha impattato di più è la feature "Party" ma lo ha fatto "negativamente", infatti più la TD-IDF della feature aumenta più diminuisce il valore di SHAP. Questo è dovuto dal fatto che la feature "party" è estremamente presente nella maggior parte delle categorie, così come la feature "Friend". Analizziamo ora la feature "Secretly" notiamo come all'aumentare del suo valore TD-IDF aumenta anche il valore di SHAP, effettivamente nei giochi di carte (in generale) è molto importante nascondere quali carte si possiedono agli avversari. Una feature non banale è "Work Together", questa infatti ha un TF-IDF direttamente proporzionale al valore di SHAP, questo credo che sia una coincidenza. Infatti, ci sono molti giochi di carte che richiedono collaborazione tra i giocatori, ma anche molti altri nella quale i giocatori si sfidano.

Probabilmente per una coincidenza il nostro dataset ha più giochi di carte collaborativi piuttosto che competitivi.

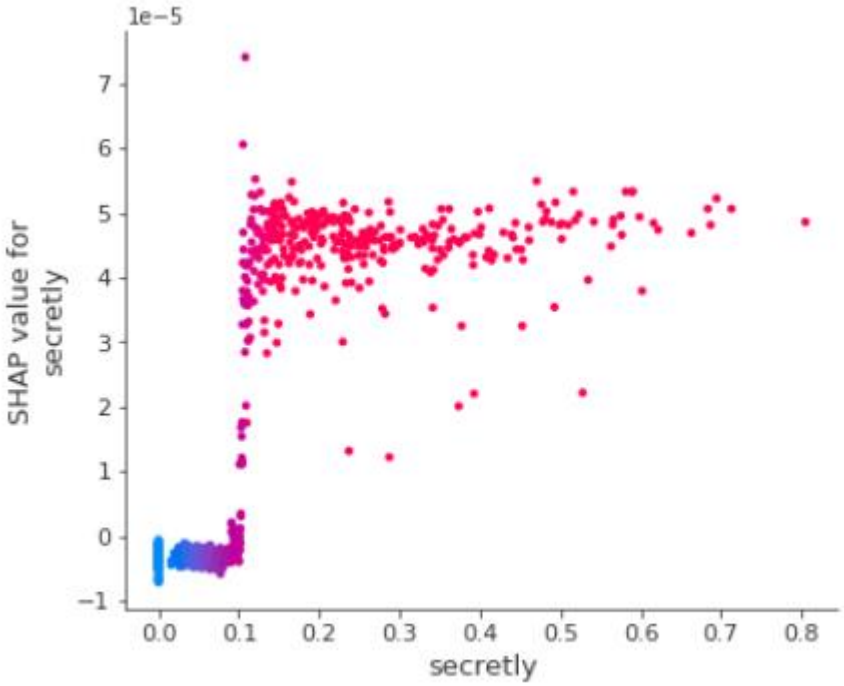

Figura P: dependence plot della feature secretly nella categoria card game

Anche in questo caso secretly probabilmente è una feature molto significativa per la categoria Card Game, infatti possiamo vedere come ad un valore di TF-IDF di 0 o dintorni il valore di SHAP è 0 se non più basso. Una curiosità di questa feature è che quando il valore di TF-IDF diventa significativo toccando il 0.1, il valore di SHAP aumenta vertiginosamente, assestandosi tra i 6 e i 3 quando il TF-IDF varia tra i 0.1 e 0.8, sintono del fatto che questa feature è estremamente presente nei giochi di carte e quasi inesistente nelle descrizioni degli altri giochi

Dall'analisi globale si evincono le regole utilizzate dal modello per predire le categorie:

- Le feature, per ogni categoria, possono influenzare la previsione in due modi: positivamente (quindi la feature spinge la previsione verso un determinato insieme di categorie) o negativamente (quindi la feature allontana la feature allontana la previsione da un determinato insieme di categorie)
- C'è un forte collegamento tra il TF-IDF e il valore di Shapley; infatti, quando questi sono direttamente proporzionali la feature influenza positivamente la previsione, mentre quando questi sono inversamente proporzionali la feature influenza negativamente la previsione
- Le feature meno comuni sono quelle che influenzano maggiormente la previsione. Come possiamo vedere dall'esempio della feature Wizard, le troppo comuni a più di una categoria tendono a diventare meno influenti sulla previsione; infatti, nel caso della feature Wizard, l'impatto di questa feature diventa negativo per la categoria Fantasy dal momento in cui si scopre essere molto presente in diverse categorie. Mentre la feature Tank essendo logicamente molto importante per la categoria Wargame ed essendo

molto rara nelle altre categorie diventa un elemento decisivo per la previsione della categoria Wargame.

7.2 Analisi locale

L'analisi locale è fondamentale per capire se le regole che abbiamo appreso a livello globale effettivamente vengano applicate ad ogni singola previsione. Questo tipo di analisi non è raro che restituisca risultati contrastanti con l'analisi globale, quest'ultima infatti prende in considerazione decine di migliaia di previsioni e ci da una utilissima visione d'insieme, ma a volte non permette di scovare scelte fatte dal modello in alcuni casi più specifici di alcuni giochi.

Prendiamo ora 2 giochi per ogni categoria studiata e mostriamone l'analisi locale tramite il force plot messo a disposizione da SHAP.

Analizziamo il gioco "Terra Mistica", un gioco Fantasy strategico.

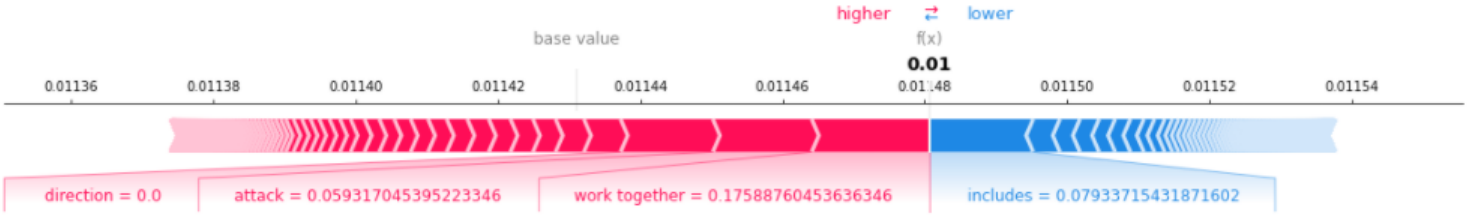

Figura Q: Force plot del gioco Terra Mistica per la categoria Fantasy

- Il valore f(x) (anche detto **valore di output**) è la predizione del gioco analizzato. Ovvero la media dei valori di TF-IDF di tutte le feature del gioco.
- Il "**base value**" è la media dei valori di output i tutti i giochi della categoria "Fantasy", il paper di shap descrive questo dato come "il valore che sarebbe previsto se non si conoscesse nessuna feature dell'output corrente"
- Le feature rosse e blu spingono la previsione più "in alto" o "in basso" rispetto al base value
- La feature **Includes** abbassa la previsione rispetto al base value poichè il suo valore di SHAP in questo gioco (0.079) è minore rispetto alla media dei valori di SHAP della feature su tutte le osservazioni (0.093).
- La feature **attack** viceversa alza il valore di output poichè il suo valore di SHAP in questa previsione (0.059) è maggiore della media dei valori di SHAP per la stessa feature (0.009)

Nel caso del gioco osservato notiamo come le feature si comportino allo stesso modo dell'analisi globale, ovvero la feature come Attack spingono la previsione verso la categoria Fantasy, mentre la feature Include allontana la previsione a livello locale come a livello globale. Una novità è la feature Work Together che non è mostrata nell'analisi globale. Proviamo ora a analizzare questa nuova feature e mettiamola a confronto con una feature conosciuta a livello globale: Attack.

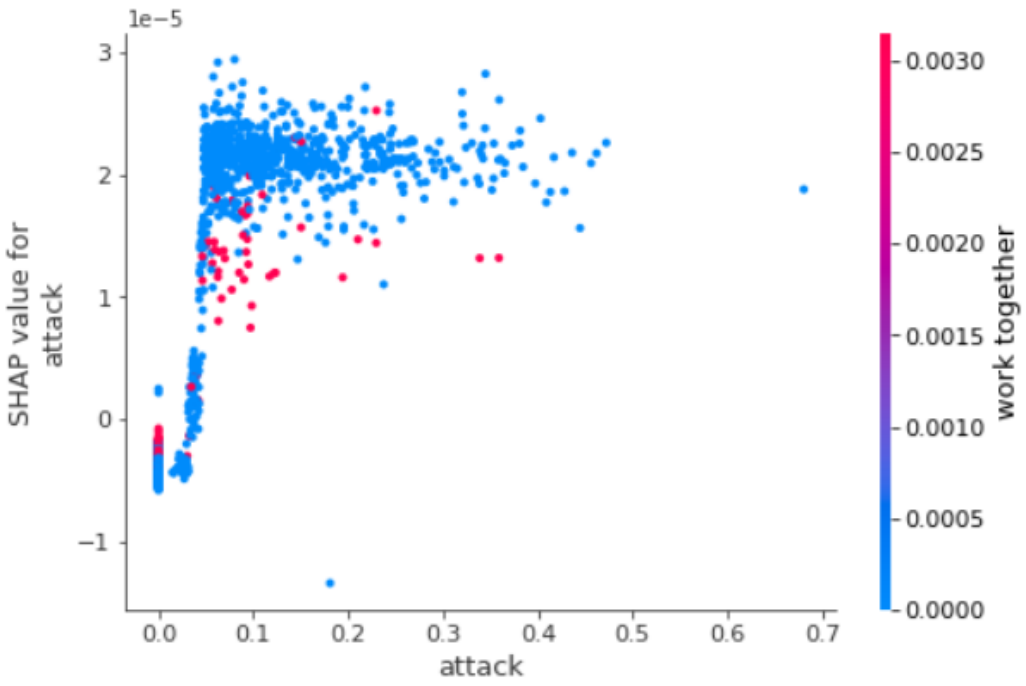

Figura R: dependence plot tra le feature attack e work together per la categoria Fantasy

Il grafico delle dipendenze tra le due feature dà ragione all'analisi globale. Possiamo vedere come i valori di SHAP di Attack siano inversamente proporzionali a quelli di Work Together; infatti, presa la fascia di punti tra un valore di SHAP di Attack 1(le-5) e 3(le-5) i valori di Shapley della feature Work Together tendono a 0. Si può però notare una piccola porzione di punti (precisamente tra i valori 1e-5 e 2e-5) nella quale c'è una proporzione diretta tra le due feature, probabilmente il gioco scelto è proprio uno di questi punti ma possono essere considerati casi isolati.

Mostriamo ora il force plot del gioco Spirit Island, un gioco fantasy d'avventura.

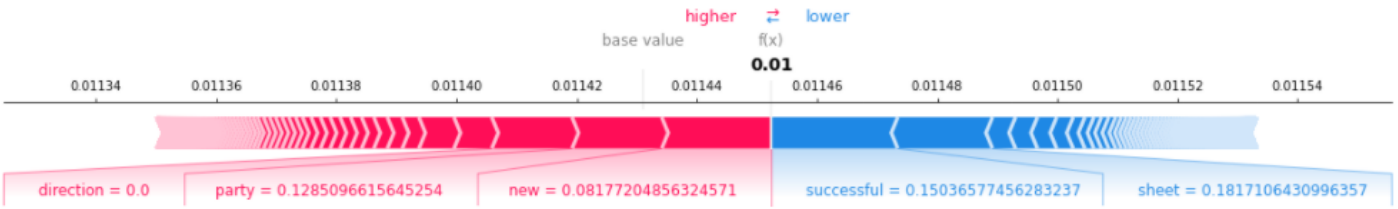

Figura S: force plot per il gioco Spirit Island.

Anche in questo caso c'è una forte correlazione tra l'analisi locale e quella globale. Per esempio, la feature New spinge positivamente verso la categoria Fantasy, come visto nel grafico globale. Altrettanto la feature Sheet spinge negativamente sulla previsione, come ci aspettavamo. Una curiosità è la feature Direction, questa infatti spinge la previsione "positivamente", a differenza di ciò che vediamo nell'analisi globale. La motivazione di questa differenza tra le due analisi risiede nel valore di Shapley di questa feature ovvero 0. La spinta positiva o negativa di una

feature dipende dal suo valore in relazione al valore medio di quella feature in una determinata categoria (in questo caso la categoria Fantasy). Considerando che dalla analisi globale possiamo vedere come i valori di Shapley per la feature Direction sono pressoché tutti negativi (quindi la presenza di questa feature nella descrizione sposta la previsione dalla categoria Fantasy) la sua media sarà sicuramente negativa. Quindi il valore 0 indica che questa feature sarà estremamente rara nella descrizione di questo gioco, questo da una spinta positiva alla previsione verso la categoria Fantasy.

Analizziamo qualche gioco della categoria Wargame, partendo dal gioco Twilight Imperium, un gioco strategico nella quale si intrecciano politica, guerra e economia.

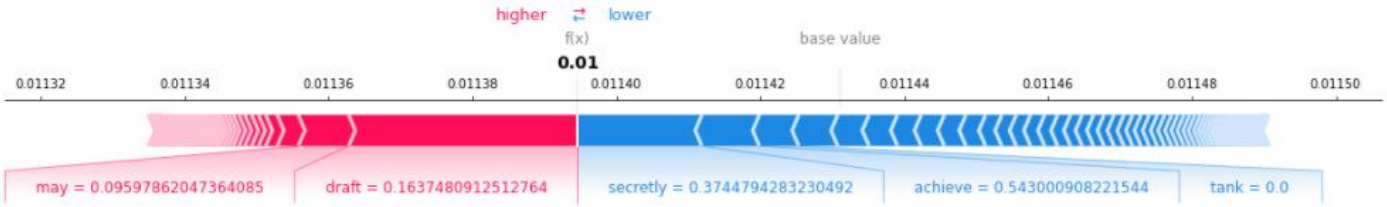

Figura T: force plot per il gioco Twilight Imperium

Le due feature predominanti Includes e Work Together rispecchiano l'analisi globale; infatti, sono molto importanti per la previsione della categoria Wargame. Ma ci sono due curiosità: la prima risiede nella feature Tank, questa infatti seppur abbia valore 0 SHAP considera la "spinta" negativa. Questo perchè se si analizza il grafico globale notiamo che è la feature più importante per la previsione della categoria Wargame, infatti necessita di un punteggio TF-IDF alto per avere un valore SHAP alto, quando invece ha un TD-IDF basso il valore di SHAP tende a 0. Nel caso specifico di questo gioco la feature Tank avrà sicuramente un TF-IDF basso (probabilmente dovuto dal fatto che il gioco in questione è ambientato nel futuro nello spazio, quindi probabilmente nella descrizione a parola tank sarà molto rara o addirittura non presente), quindi un valore di SHAP uguale a 0 nello specifico caso della feature tank avrà un impatto "negativo" che spinge la categorizzazione del gioco lontano dalla categoria Wargame. Mentre la seconda curiosità è sulla feature Attack, infatti quest'ultima non è mostrata sul grafico globale; quindi, la mostreremo sul grafico delle dipendenze per analizzarla nel caso della categoria Wargame

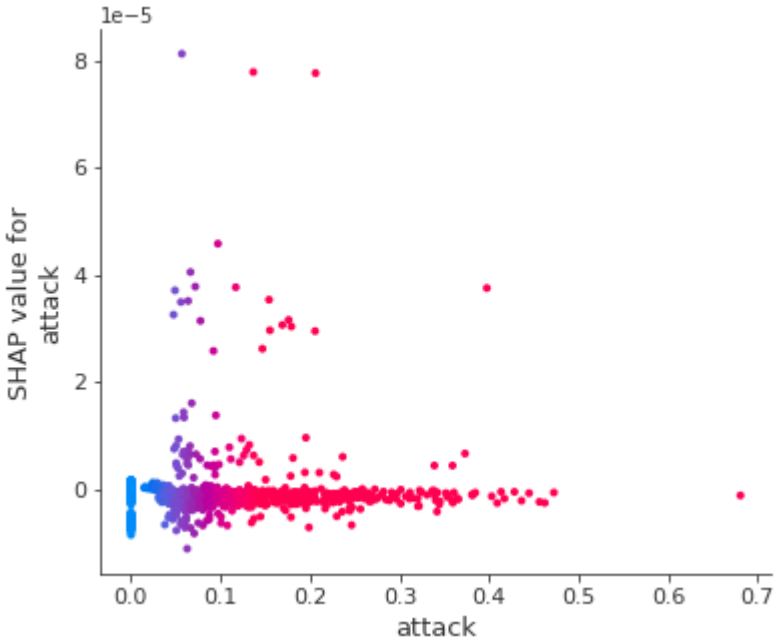

Figura U: dependence plot per la feature Attack nella categoria Wargame

In effetti il grafico delle dipendenze per la feature Attack mostra come all'aumentare del TF-IDF il valore di Shapley non aumenta ma in modo significativo, se non per alcuni casi isolati. Indice del fatto che la feature Attack a livello globale non impatta fortemente sulla previsione della categoria Wargame. Quindi probabilmente il gioco preso d'esempio è uno dei pochi casi nella quale invece ha avuto un forte impatto.

Mostriamo ora il gioco War of the Ring, un gioco di guerra fantasy ambientato nel medioevo.

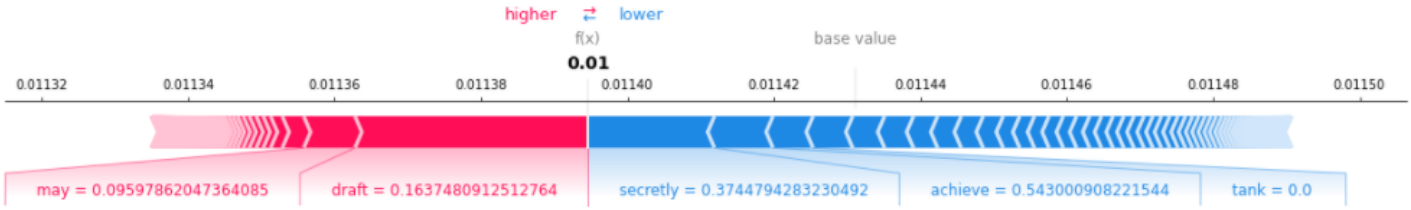

Figura V: force plot del gioco War of the Ring

Partiamo con le ovvietà come le feature May, Draft e Tank. Queste rispecchiano l'andamento globale, infatti se mettiamo a confronto i due grafici notiamo che sono molto simili: May e Draft hanno un valore di Shapley alto quando hanno un valore di TF-IDF alto, mentre tank con valore 0 lo abbiamo già analizzato in precedenza. Analizziamo invece la feature Secretly nuova nel panorama delle feature mostrate dall'analisi globale per la categoria Wargame.

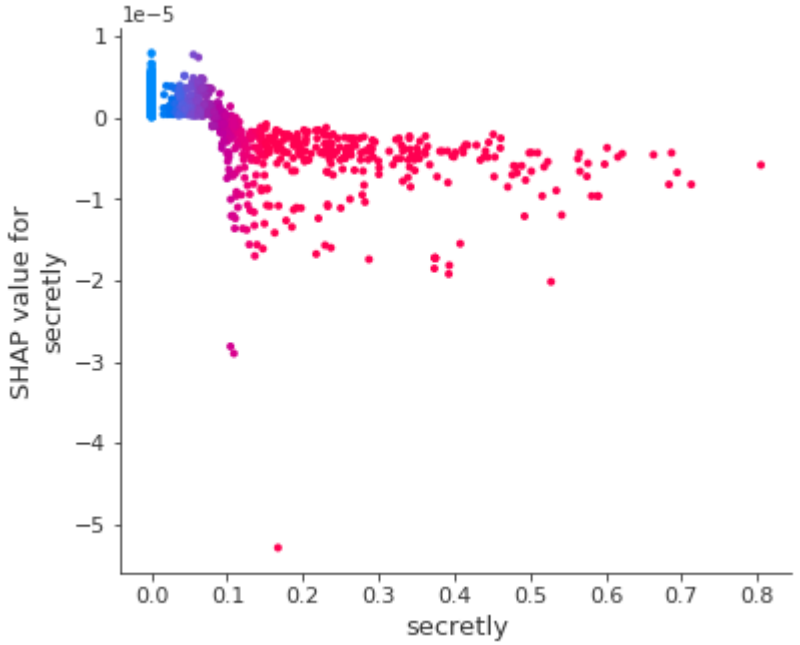

Figura W: dependence plot per la feature Secretly nella categoria Wargame

Il grafico delle dipendenze conferma il grafico del gioco. Infatti, la feature Secretly tende ad allontanare la previsione dalla categoria Wargame, se si analizza il grafico si nota come ci sia una proporzionalità inversa tra il valore di TF-IDF e il valore di SHAP. Infatti, il valore massimo di SHAP lo abbiamo quando il TF-IDF tende a 0.

Analizziamo ora la categoria Card Game prendendo il gioco It's a Wonderful World. Un gioco di carte nella quale il giocatore deve disegnare la propria distopia "perfetta, costruendo edifici e producendo risorse.

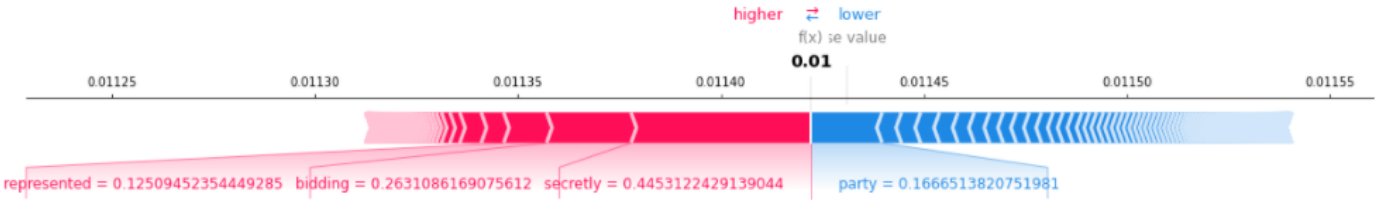

Figura X: force plot del gioco It's a Wonderful World

Riconosciamo immediatamente alcune feature già viste nella analisi globale: la feature Secretly infatti è in linea con l'analisi globale, questa infatti ha una proporzione diretta tra il valore di SHAP e il TF-IDF sintomo del fatto che la feature impatta "positivamente" sulla categoria Card Game. Un'altra feature nota è Party, anche questa in linea con l'analisi globale. Infatti, a valori di TF-IDF alti il valore di SHAP diventa negativo, come nel caso preso di esempio. Analizziamo ora due feature nuove: bidding e represented.

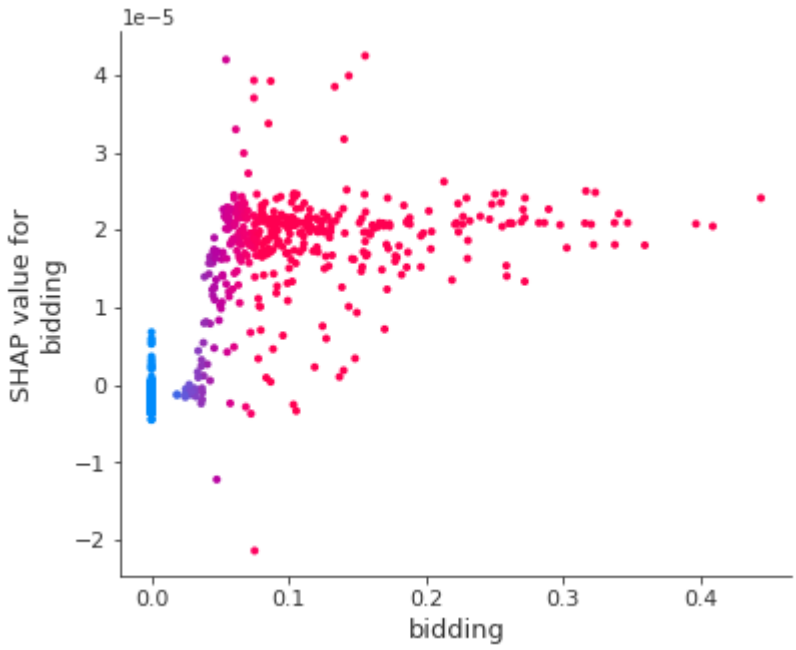

Figura Y: dependence plot per la feature Bidding

Analizziamo il grafico delle dipendenze, quando il valore TF-IDF è 0 i valori di SHAP si assestano nei dintorni di 0. Mentre all'aumentare del valore di TF-IDF il valore di SHAP aumenta sintomo del fatto che la feature anche a livello globale impatta positivamente sulla previsione della categoria. Un altro dettaglio da notare è che i valori di SHAP si assestano tra il 2e-5 e 3e5, e solo in rari casi superano il 3e-5, questo potrebbe essere il motivo per cui la feature non è mostrata tra le 20 più impattanti nel grafico globale. Mostriamo ora la feature Represented.

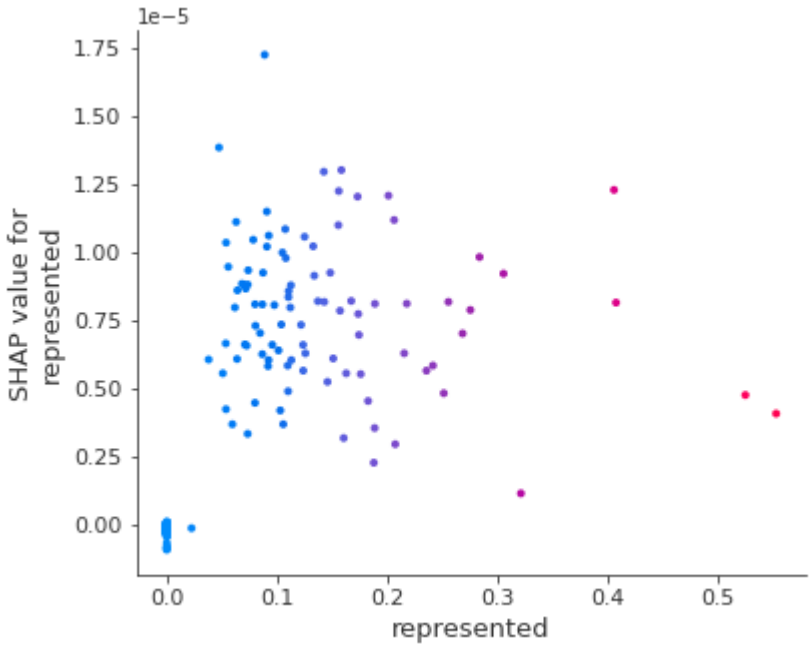

Figura Z: dependece plot per la feature Rappresented

Questa feature è poco rilevante nella categoria Card Game, infatti vediamo come il suo valore di SHAP non supera 1.75e-5, sembra esservi una correlazione debole tra il valore di SHAP e il TF-IDF, oltre al fatto che il massimo valore di SHAP lo si raggiunge quando il valore TF-IDF è molto basso. Probabilmente il gioco osservato è uno dei pochi nella quale questa feature aveva importanza.

Analizziamo ora il gioco Aeon's End, un gioco di ruolo con le carte ambientato in un mondo fantasy.

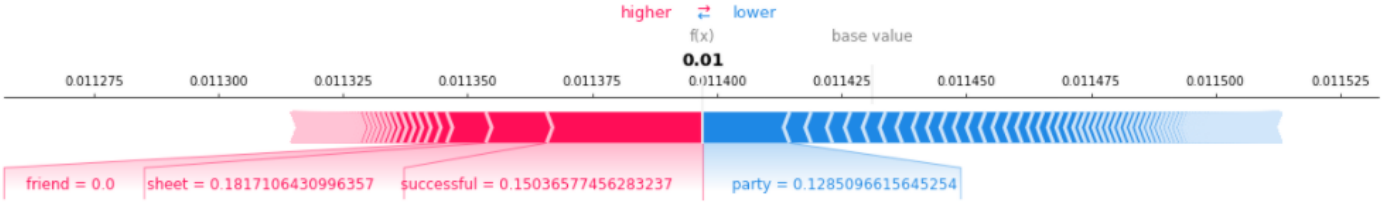

Figura AA: force plot per il gioco Aeon's End

Alcune feature sono già note a livello globale, la feature Party rispecchia la visione globale, infatti spinge "negativamente" la previsione lontano dalla categoria Card Game. Anche la feature Successful è in linea con lo schema globale, infatti la feature spinge "positivamente" la predizione verso la categoria in oggetto. Anche se risulta strano anche la feature Friend è in linea con il grafico globale. Infatti, guardandolo bene vediamo come un valore in genere più è alto il TF-IDF di questa feature più il suo valore di SHAP spinge "negativamente" la previsione lontano dalla categoria Card Game, quindi un valore di SHAP a 0 indica un valore basso di TF-IDF e di conseguenza un impatto "positivo" sulla previsione verso la categoria studiata. Proviamo invece a studiare una feature Sheet nuova nel panorama delle feature di questa categoria.

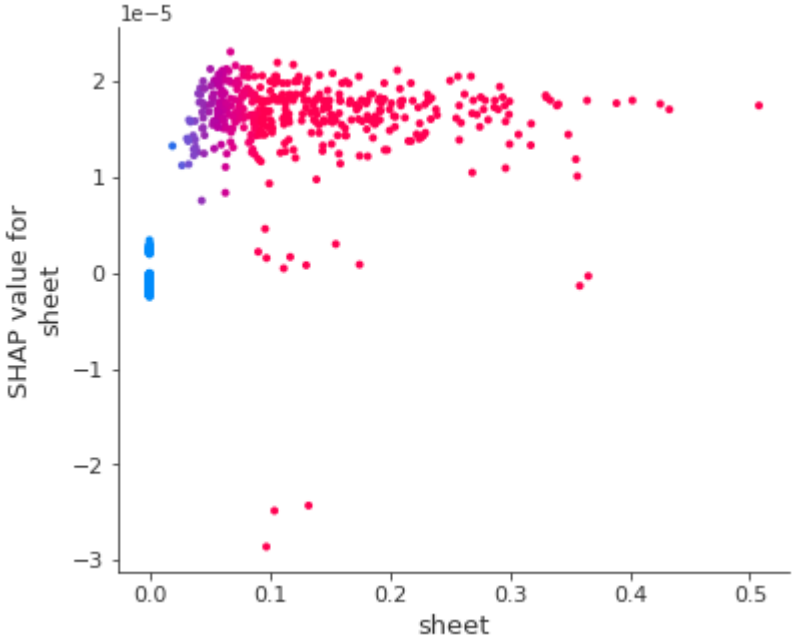

Figura AB: dependence plot per la feature Sheet

Anche se non mostrata nel grafico globale la feature Sheet ha un grosso impatto nella previsione della categoria Card Game, infatti c'è una proporzionalità diretta tra i valori di SHAP e TF-IDF, infatti quando il valore di SHAP su aggira intorno allo 0 anche il valore di TF-IDF si aggira intorno allo 0 mentre al crescere del TF-IDF anche il valore di SHAP aumenta. C'è però da sottolineare un particolare, il valore di SHAP non supera mai il valore di 3e-5, motivo per il quale questa feature non rientra nelle 20 più rilevanti della categoria. Comunque, nel caso del gioco preso in esame l'analisi globale corrisponde all'analisi locale.

L'analisi locale ci ha permesso di aggiungere una nuova regola a quelle già dedotte dall'analisi globale: anche l'assenza di una feature è importante tanto quanto la presenza di quest'ultima. Infatti, in alcuni casi, l'assenza di una feature fondamentale per la previsione di una categoria influenza la previsione in maniera opposta rispetto alla sua presenza. Per esempio, la feature Friend nel caso della categoria Card Game a livello globale spinge negativamente la previsione. Ma nel caso del gioco Aeon's End la totale assenza della feature nella descrizione del gioco ha portato la previsione verso la categoria Card Game. Questo aspetto è molto importante nella libreria di SHAP poiché mostra come questa analizzi non solo le parole presenti ma anche quelle assenti.

CONCLUSIONE

SHAP è un'ottima libreria per l'analisi di un modello di machine learning. In particolare, i valori di Shapley danno tante possibilità di interpretazione delle previsioni rendendo comprensibile un procedimento che generalmente è definito "black box", ovvero è difficile comprendere come il modello di machine learning abbia ragionato.

Nel nostro caso le regole che abbiamo appreso sono le seguenti:

- Le feature, per ogni categoria, possono influenzare la previsione in due modi: positivamente (quindi la feature spinge la previsione verso un determinato insieme di categorie) o negativamente (quindi la feature allontana la feature allontana la previsione da un determinato insieme di categorie)
- C'è un forte collegamento tra il TF-IDF e il valore di Shapley; infatti, quando questi sono direttamente proporzionali la feature influenza positivamente la previsione, mentre quando questi sono inversamente proporzionali la feature influenza negativamente la previsione
- Le feature meno comuni sono quelle che influenzano maggiormente la previsione. Come possiamo vedere dall'esempio della feature Wizard, le troppo comuni a più di una categoria tendono a diventare meno influenti sulla previsione; infatti, nel caso della feature Wizard, l'impatto di questa feature diventa negativo per la categoria Fantasy dal momento in cui si scopre essere molto presente in diverse categorie. Mentre la feature Tank essendo logicamente molto importante per la categoria Wargame ed essendo molto rara nelle altre categorie diventa un elemento decisivo per la previsione della categoria Wargame.
- Anche l'assenza di una feature è importante tanto quanto la presenza di quest'ultima. Infatti, in alcuni casi, l'assenza di una feature fondamentale per la previsione di una categoria influenza la previsione in maniera opposta rispetto alla sua presenza.

A seguito di queste regole possiamo pensare a quali miglioramenti apportare al modello per renderlo più preciso. Ovviamente un maggior numero di casi nel dataset è sempre apprezzato per un modello di macchine learning, non è un segreto che più vasto è il dataset più precise saranno le previsioni del modello.

Ma oltre a questo c'è altro che si può fare: eliminare le feature che non impattano significativamente su nessuna categoria (quindi quelle che hanno un valore di Shapley vicino allo 0 in tutte le categorie). Questo perché il modello si sforza di trovare relazioni tra queste feature e le categorie inutilmente dal momento in cui non hanno nessun impatto significativo per nessuna categoria.

Un ulteriore miglioramento è quello di raggruppare le categorie in macrocategorie, questo perché molte categorie si assomigliano abbastanza da poterle ragionare come macrocategorie e migliorare le previsioni restituite raggruppando le feature. In effetti categorie come: World War 1 e World War 2 possono essere tranquillamente raggruppate in una macrocategoria come Wargame in modo che le feature più impattanti per le categorie WW1 e WW2 possano diventare importanti per una macrocategoria che le raggruppa.

8 BIBLIOGRAFIA

[1] “Pandas”. [Online – 05/04/2022]. Available: https://pandas.pydata.org/

[2] ”SHAP”. [Online – 01/04/2022]. Available: https://shap.readthedocs.io/en/latest/index.html

[3] “SHAP values”. [Online – 02/02/2022]. Available: https://www.kaggle.com/code/dansbecker/shap-values/tutorial

[4] “Interpretable machine learning”. [Online – 15/03/2022]. Available: https://books.google.it/books?id=jBm3DwAAQBAJ&lpg=PP1&ots=EgvYWpGBUZ&dq=machine%20learning%20&lr&hl=it&pg=PP1#v=onepage&q=machine%20learning&f=false

[5] “Cos’è il machine learning?”. [Online – 01/12/2021]. Available: https://www.oracle.com/it/data-science/machine-learning/what-is-machine-learning/

[6] “SHAP (SHapley Additive exPlanations)”. [Online – 10/10/2021]. Available: https://christophm.github.io/interpretable-ml-book/shap.html

[7] Prasad Patil “What is Exploratory Data Analysis?”. [Online – 23/10/2021]. Available: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

[8] Pranshu Sharma “Mastering Exploratory Data Analysis(EDA) For Data Science Enthusiasts”. [Online – 21/10/2021]. Available: https://www.analyticsvidhya.com/blog/2021/04/mastering-exploratory-data-analysiseda-for-data-science-enthusiasts/