REPORT
Gaming Hours vs. Academic & Work Performance

UNIVERSITÀ DEGLI STUDI
DI SALERNO

Autori:

Alessandro Cigliano 0512119063
Carmine Di Manso 0512119521

Sommario

1. ANALISI SCENARIO

1.1 Contesto

Nel panorama socioeconomico contemporaneo, la digitalizzazione e la diffusione su larga scala dei videogiochi ha radicalmente trasformato le abitudini quotidiane, rendendo l'intrattenimento digitale, e in particolare il settore del *gaming*, una componente strutturale del tempo libero per ampie fasce della popolazione. Se da un lato i videogiochi rappresentano un potente strumento di svago e socializzazione, dall'altro la letteratura scientifica e il dibattito pubblico sollevano costanti interrogativi riguardo le potenziali ripercussioni che un'esposizione prolungata a tali attività può avere sulle performance cognitive e produttive.

L'obiettivo primario consiste nell'analizzare come i videogiochi possano alterare il rendimento di uno studente/lavoratore.

Lo studio si propone di integrare variabili psicofisiche correlate, come la qualità del sonno, i livelli di stress e la capacità di concentrazione, per determinare se il gaming agisca come un fattore di distrazione (detrimento della performance) o una semplice forma di intrattenimento innocua.

1.2 Obiettivo

Lo scenario analizzato è stato formulato come un problema di apprendimento supervisionato. Nello specifico, il progetto si concentra sulla predizione del livello di performance individuale (**Performance_Impact**), sfruttando le feature comportamentali come predittori.

Il task è stato affrontato mediante un singolo approccio: la **Classificazione**, utile per categorizzare l'individuo in classi di rischio (es. impatto negativo, neutro, positivo) e individuare possibili cause di un basso o alto rendimento.

2. IL DATASET

Il dataset preso in analisi è "**Gaming Hours vs Academic & Work Performance**" di Prince Rajak.

(https://www.kaggle.com/datasets/prince7489/gaming-hours-vs-academic-and-work-performance)

2.1 Contenuto

Nome Colonna	**Descrizione**	**Tipo di Dato**
User_ID	Feature che rappresenta l'identificativo univoco per ogni utente nel dataset.	Stringa
Age	Feature che rappresenta l'età dell'utente.	Intero

Gender	Feature che rappresenta il genere dell'utente (es. Male, Female).	Stringa
Occupation	Feature che rappresenta l'occupazione o lo stato lavorativo dell'utente.	Stringa
Game_Type	Feature che rappresenta il genere di videogioco preferito o più giocato.	Stringa
Daily_Gaming_Hours	Feature che rappresenta il numero medio di ore trascorse a giocare ogni giorno.	Float
Weekly_Gaming_Hours	Feature che rappresenta il numero totale di ore trascorse a giocare in una settimana.	Float
Primary_Gaming_Time	Feature che rappresenta la fascia oraria della giornata in cui l'utente gioca principalmente.	Stringa
Sleep_Hours	Feature che rappresenta il numero medio di ore di sonno per notte.	Float
Stress_Level	Feature che rappresenta la valutazione del livello di stress percepito dall'utente.	Intero
Focus_Level	Feature che rappresenta la valutazione della capacità di concentrazione dell'utente.	Intero
Academic_or_Work_Score	Feature che rappresenta il punteggio o valutazione delle prestazioni in ambito accademico o lavorativo.	Intero
Productivity_Level	Feature che rappresenta la valutazione del livello di produttività generale.	Intero
Performance_Impact	Feature che rappresenta l'impatto percepito del gaming sulle prestazioni complessive.	Stringa

Questo set di dati è **sintetico** ed esplora la relazione tra le abitudini di gioco quotidiane e settimanali e il loro impatto sulle prestazioni accademiche o sul posto di lavoro. Nonostante si tratti di dati sintetici, quindi non recuperati direttamente da persone tramite questionari, a detta dell'autore sono stati generati considerando diversi pattern identificati in vari studi e ricerche scientifiche, **risultando quindi quantomeno realistici** e appetibili per la realizzazione di un modello di machine learning. **È composto da 1000 righe e 14 feature** illustrate prima. Cattura fattori comportamentali chiave come la durata del gioco, la fascia oraria di gioco preferita, le ore di sonno, i livelli di stress, i livelli di concentrazione e i punteggi di produttività.

I dati sono progettati per aiutare analisti, studenti e ricercatori a comprendere come diversi modelli di gioco possano influenzare positivamente, negativamente o neutralmente i risultati delle prestazioni. A detta dell'autore è adatto per analisi esplorative dei dati (EDA), studi di correlazione, visualizzazione dei dati e creazione di modelli di machine learning basati su classificazione e regressione.

3. ANALISI E GESTIONE DELLE CRITICITÀ

3.1 Data Cleaning

Al fine di migliorare la qualità dei dati presenti nel dataset è necessario effettuare data cleaning, ovvero un controllo sui dati per identificare **missing values**, **outliers** oppure classi sbilanciate. Tra le feature del dataset si può subito identificare come superflua, ai fini della realizzazione del modello, "**User_ID**", utile soltanto per identificare ciascun lavoratore/studente. Successivamente verrà presa in considerazione la matrice di correlazione delle variabili, utile per individuare altre feature ridondanti.

3.2 Verifica dei missing values

Nell'ambito della fase di pre-processing dei dati, è stata condotta un'analisi preliminare volta a identificare la presenza di valori mancanti (*missing values*), la cui gestione è importante per garantire la robustezza del modello.

L'ispezione del dataset *Gaming Hours vs Academic and Work Performance,* eseguita tramite un programma scritto in **python**, ha rivelato una completa integrità dei dati. Su un totale di 1.000 istanze e 14 feature analizzate non è stata riscontrata alcuna cella vuota o valore nullo. Trattandosi di un dataset sintetico la totale assenza di missing values risulta plausibile ed è verificata.

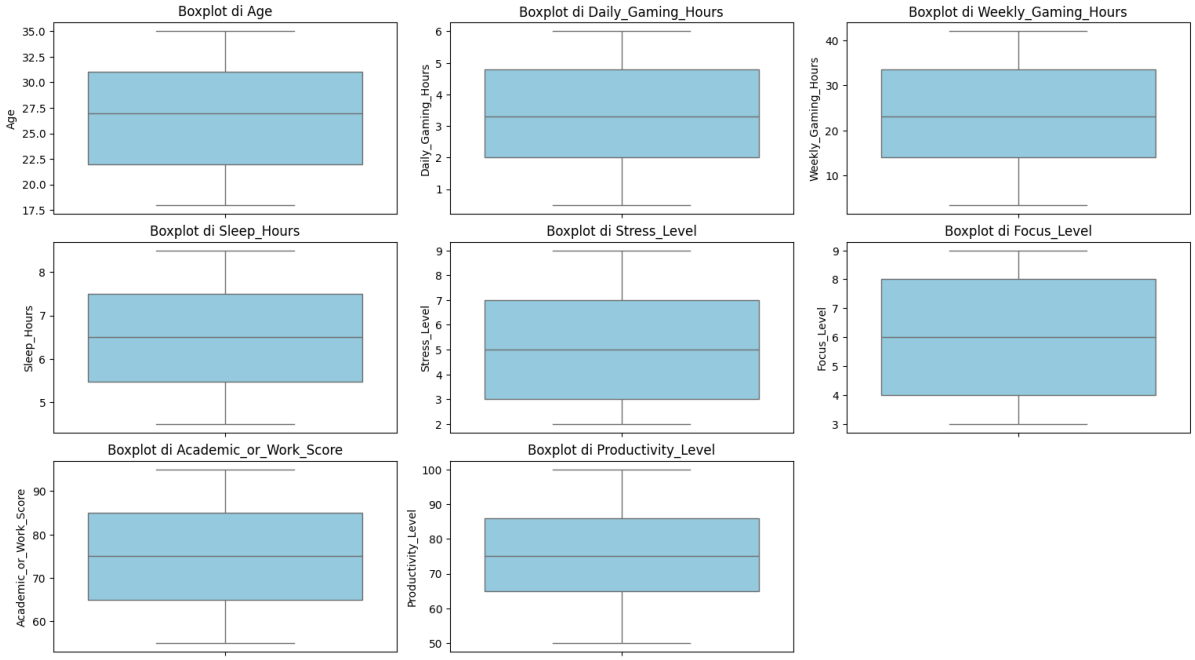
Di conseguenza, non si è reso necessario applicare tecniche di imputazione (come la sostituzione con media/mediana o l'utilizzo di algoritmi predittivi per i dati mancanti) né strategie di eliminazione delle righe (listwise deletion). Questa caratteristica del dataset ha permesso di preservare l'interezza del campione originale per le fasi successive di analisi esplorativa e modellazione.

3.3 Verifica degli outliers

Successivamente alla verifica della completezza dei dati, è stata effettuata un'analisi per identificare eventuali **outliers** (valori anomali) che potessero distorcere le prestazioni del modello o indicare errori di misurazione.

Per il rilevamento, è stato utilizzato il metodo dell'intervallo interquartile (IQR - Interquartile Range) e con l'ispezione dei **boxplot**. L'analisi ha evidenziato che la distribuzione dei dati rientra interamente nei limiti statistici attesi:

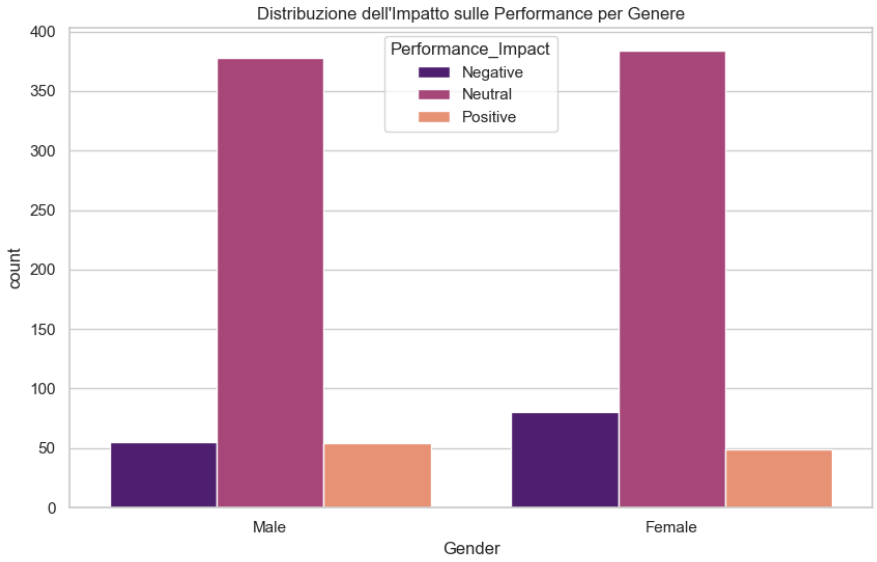
- Le ore di gioco giornaliere (***Daily_Gaming_Hours*)** variano da 0.5 a 6.0, un range plausibile per il contesto dello studio.
- Le ore di sonno (***Sleep_Hours***) sono comprese tra 4.5 e 8.5, senza estremi fisiologicamente impossibili.
- I punteggi di performance e i livelli di stress mostrano una dispersione coerente senza picchi isolati.

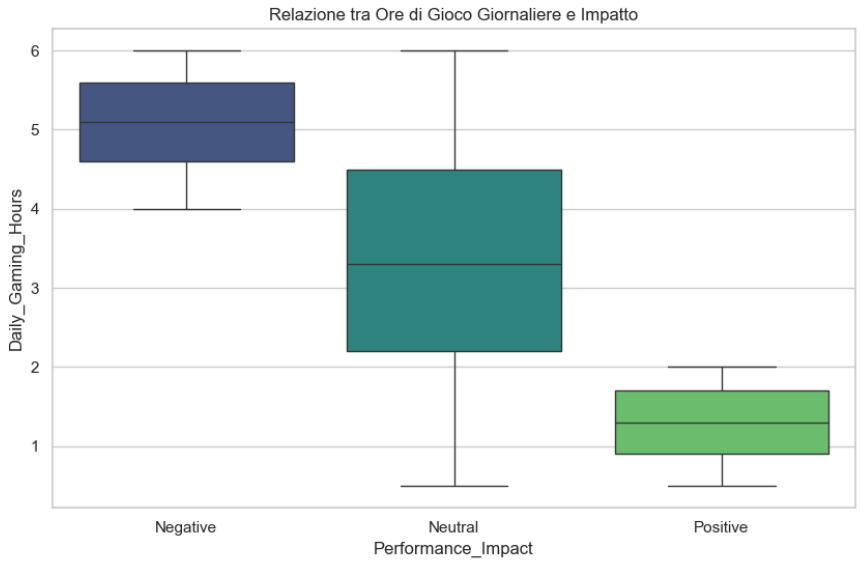


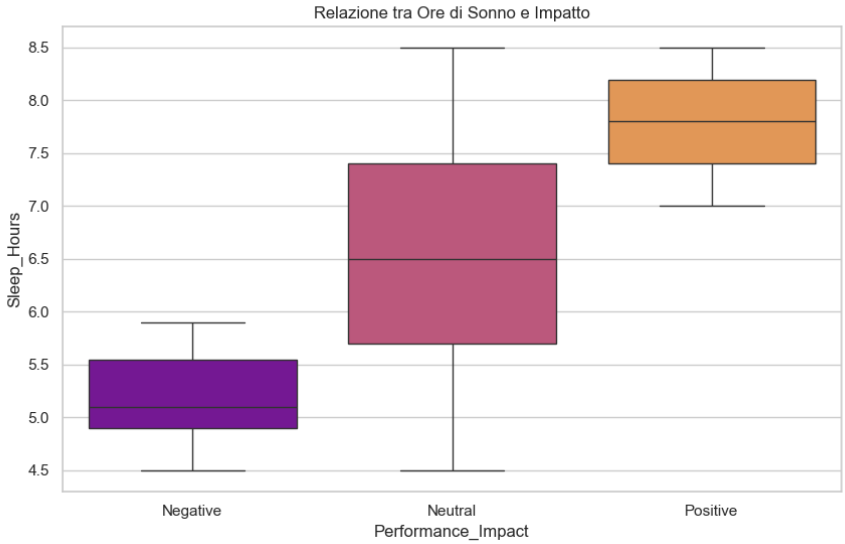
- Colonna 'Age': 0 outlier rilevati.
- Colonna 'Daily_Gaming_Hours': 0 outlier rilevati.
- Colonna 'Weekly_Gaming_Hours': 0 outlier rilevati.
- Colonna 'Sleep_Hours': 0 outlier rilevati.
- Colonna 'Stress_Level': 0 outlier rilevati.
- Colonna 'Focus_Level': 0 outlier rilevati.
- Colonna 'Academic_or_Work_Score': 0 outlier rilevati.
- Colonna 'Productivity_Level': 0 outlier rilevati.

3.4 Analisi della variabile dipendente

Di seguito verranno riportati alcuni grafici rappresentati della distribuzione della variabile dipendente rispetto alcune delle feature presenti nel dataset, in modo tale da meglio comprendere la struttura e la distribuzione dei dati nelle varie classi.

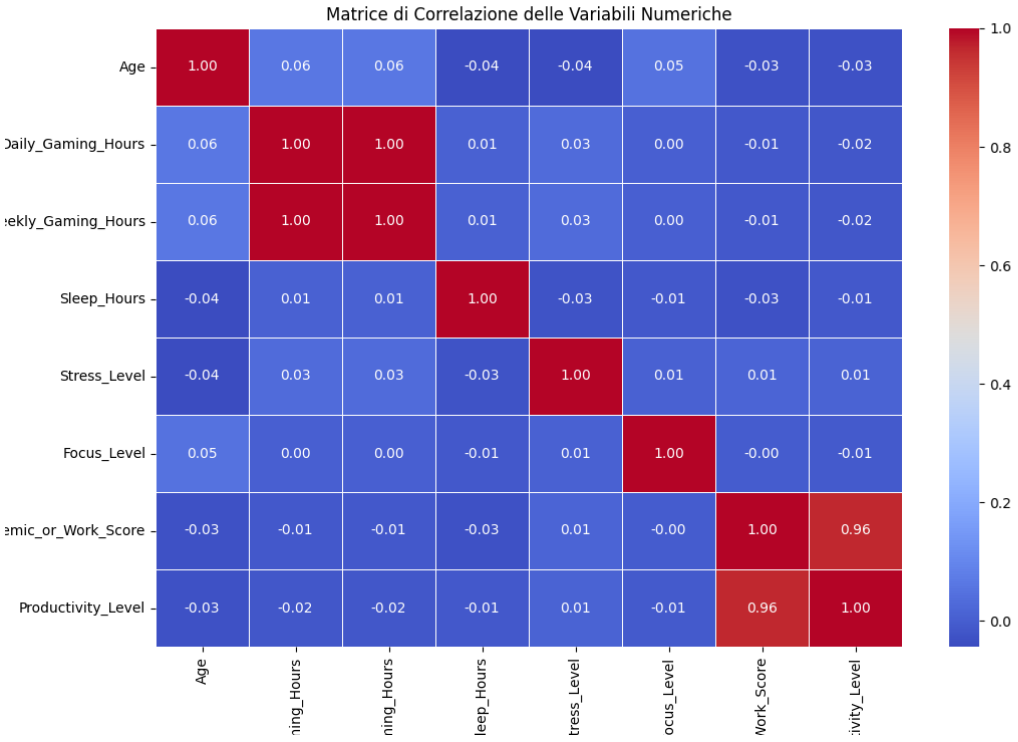






3.5 Matrice di correlazione delle variabili

Di seguito possiamo osservare la **matrice di correlazione delle variabili**, uno strumento statistico utilizzato per misurare le correlazioni tra le variabili di un dataset.



La matrice ci permette di capire e osservare i pattern e le relazioni tra le varie feature, in modo tale da capire e scegliere in maniera efficace le feature migliori per le classificazioni e quali invece danno informazioni ridondanti.

- **Correlazione Perfetta (1.00): Tra Daily_Gaming_Hours e Weekly_Gaming_Hours** c'è una correlazione totale. Questo indica una ridondanza, dato che le ore di gioco settimanali sono ricavabili da quelle giornaliere. Per evitare problemi di multicollinearità è necessario rimuovere la feature **Weekly_Gaming_Hours,** perché le **Daily_Gaming_Hours** è una feature considerata più significativa nel determinare l'impatto immediato sui livelli di sonno, stress e produttività quotidiana.
- **Correlazione forte (0.96): Tra Academic_or_Work_Score e Productivity_Level**. Questo indica una ridondanza dato che il valore di **Academic_or_Work_Score** verrà influenzato eccessivamente da quello di **Productivity_Level.** Per evitare multicollinearità è necessario rimuovere la feature **Productivity_Level.**
- **Correlazioni Deboli:** Sorprendentemente, in questo dataset, le ore di gioco (Daily_Gaming_Hours) mostrano una correlazione quasi nulla con lo stress o le ore di sonno (valori vicini allo 0). Questo significa che non esiste un legame *lineare* diretto tra queste variabili, il che potrebbe rendere più complessa la previsione per modelli semplici come la regressione lineare.

3.6 Verifica dei duplicati

A completamento dell'analisi preliminare sulla qualità del dataset, è stata effettuata una verifica volta ad escludere la presenza di record duplicati che potessero introdurre **bias** nelle stime statistiche o **sovra-rappresentare** specifiche istanze durante l'addestramento del modello.

Pur non considerando più la feature **User_ID**, poiché non utile per la realizzazione del modello, non sono presenti duplicati nel Dataset. Pertanto, non è stato necessario applicare tecniche di rimozione, confermando l'integrità dei dati per le fasi successive. Il controllo è stato fatto in codice python utilizzando la **funzione duplicated()** per ogni istanza.

3.7 Stato attuale del dataset

Stato originale (Gaming_Hours_vs_Performance.csv)

User_ID	Age	Gender	Occupation	Game_Type	Daily_Gaming_Hours	Weekly_Gaming_Hours	Primary_Gaming_Time	Sleep_Hours	Stress_Level	Focus_Level	Academic_or_Work_Score	Productivity_Level	Performance_Impact
U0001	21	Male	Working Professional	Action	4.0	28.0	Morning	4.6	6	4	69	66	Negative
U0002	35	Female	Student	Sports	1.0	7.0	Night	5.4	2	7	67	72	Neutral
U0003	26	Male	Student	Puzzle	2.0	14.0	Morning	8.0	4	8	82	82	Positive
U0004	32	Male	Working Professional	Action	1.0	7.0	Night	4.9	7	7	71	66	Neutral
U0005	19	Male	Working Professional	Action	2.1	14.7	Morning	7.0	7	7	67	63	Neutral
U0006	29	Female	Student	Casual	5.3	37.1	Evening	6.0	9	8	78	75	Neutral
U0007	35	Male	Student	Sports	4.4	30.8	Evening	7.2	3	7	95	92	Neutral
U0008	25	Male	Working Professional	Sports	2.0	14.0	Evening	8.2	5	8	75	70	Positive
U0009	32	Male	Working Professional	Strategy	0.9	6.3	Morning	8.2	7	4	86	87	Positive
U0010	25	Male	Working Professional	Action	4.6	32.2	Morning	6.7	8	7	80	80	Neutral
U0011	20	Female	Working Professional	Simulation	5.2	36.4	Evening	5.1	4	9	82	86	Negative
U0012	35	Female	Working Professional	Action	3.4	23.8	Evening	8.4	2	8	62	67	Neutral
U0013	21	Female	Student	Strategy	2.9	20.3	Evening	6.3	6	7	66	69	Neutral
U0014	18	Male	Working Professional	Puzzle	4.7	32.9	Morning	6.7	2	7	75	77	Neutral
U0015	35	Male	Student	Casual	0.8	5.6	Evening	8.0	3	3	86	82	Positive
U0016	27	Female	Working Professional	Sports	5.7	39.9	Night	5.2	8	4	89	87	Negative
U0017	35	Male	Student	Action	3.0	21.0	Morning	5.5	3	5	56	60	Neutral
U0018	34	Male	Working Professional	Casual	4.4	30.8	Evening	4.7	3	3	76	72	Negative
U0019	24	Male	Student	Casual	1.7	11.9	Evening	5.0	9	4	85	86	Neutral
U0020	30	Female	Student	Strategy	2.4	16.8	Morning	6.1	2	8	61	56	Neutral

Stato attuale (Gaming_Hours_vs_Performance versione 1.1.csv)

Age	Gender	Occupation	Game_Typ	Daily_Gaming_Hours	Primary_Gaming_Time	Sleep_Hours	Stress_Level	Focus_Level	Academic_or_Work_Score	Performance_Impact
21	Male	Working Professional	Action	4.0	Morning	4.6	6	4	69	Negative
35	Female	Student	Sports	1.0	Night	5.4	2	7	67	Neutral
26	Male	Student	Puzzle	2.0	Morning	8.0	4	8	82	Positive
32	Male	Working Professional	Action	1.0	Night	4.9	7	7	71	Neutral
19	Male	Working Professional	Action	2.1	Morning	7.0	7	7	67	Neutral
29	Female	Student	Casual	5.3	Evening	6.0	9	8	78	Neutral
35	Male	Student	Sports	4.4	Evening	7.2	3	7	95	Neutral
25	Male	Working Professional	Sports	2.0	Evening	8.2	5	8	75	Positive
32	Male	Working Professional	Strategy	0.9	Morning	8.2	7	4	86	Positive
25	Male	Working Professional	Action	4.6	Morning	6.7	8	7	80	Neutral
20	Female	Working Professional	Simulation	5.2	Evening	5.1	4	9	82	Negative
35	Female	Working Professional	Action	3.4	Evening	8.4	2	8	62	Neutral
21	Female	Student	Strategy	2.9	Evening	6.3	6	7	66	Neutral
18	Male	Working Professional	Puzzle	4.7	Morning	6.7	2	7	75	Neutral
35	Male	Student	Casual	0.8	Evening	8.0	3	3	86	Positive
27	Female	Working Professional	Sports	5.7	Night	5.2	8	4	89	Negative
35	Male	Student	Action	3.0	Morning	5.5	3	5	56	Neutral
34	Male	Working Professional	Casual	4.4	Evening	4.7	3	3	76	Negative
24	Male	Student	Casual	1.7	Evening	5.0	9	4	85	Neutral
30	Female	Student	Strategy	2.4	Morning	6.1	2	8	61	Neutral

Rispetto al dataset originale sono state rimosse le feature:

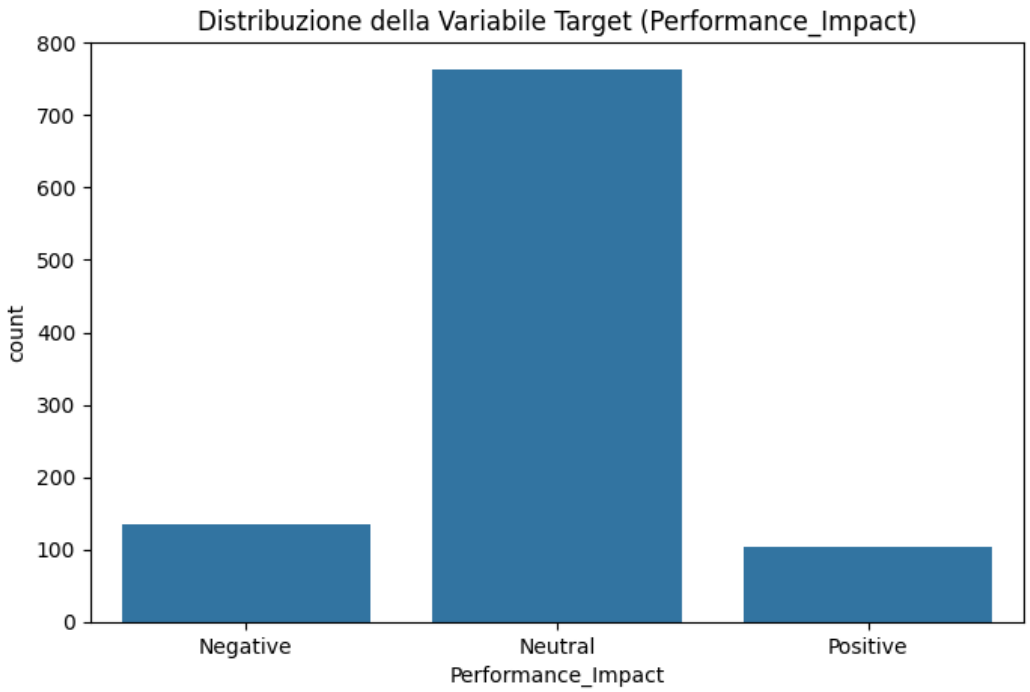
- **User_ID**
- **Weekly_Gaming_Hours**
- **Productivity_Level**

(le tabelle sono una porzione del dataset originale riportate al solo scopo di mostrare i cambiamenti nelle feature non rappresentano la totalità del dataset)

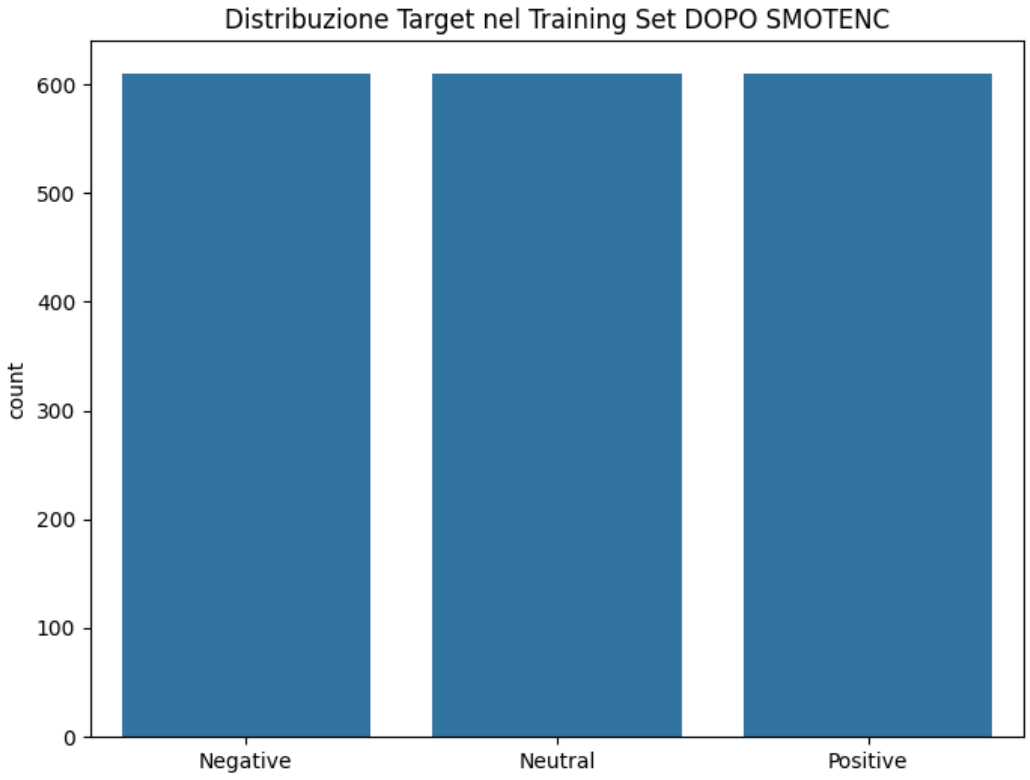
3.8 Analisi e gestione della variabile target

La variabile target individuata per lo studio è **Performance_Impact**, una variabile categorica che descrive l'effetto delle abitudini di gioco sulle prestazioni (scolastiche o lavorative) classificandolo in tre livelli: *Positive*, *Neutral* e *Negative*.

Dall'analisi esplorativa della distribuzione delle classi, è emerso che il dataset presenta un evidente **sbilanciamento delle classi**. La classe *"Neutral"* **(76,2 %)** risulta essere la maggioritaria, con una frequenza significativamente più alta rispetto alle classi *"Positive" **(10,3 %)*** e *"Negative" **(13,5 %)***. Questo porterebbe il modello a sviluppare un bias per cui verrebbe maggiormente predetto il valore neutral, sfavorendo la predizione delle classi minoritarie “Positive” e “Negative”.



Tra le possibili strategie per fornire dei dati di training equilibrati è stata scartata la tecnica dell'**undersampling** (riduzione della classe maggioritaria) poiché, dato il numero limitato di istanze complessive del dataset (1000 righe), tale approccio avrebbe comportato una drastica perdita di informazioni utili, riducendo eccessivamente la base dati disponibile per l'apprendimento, causando così **underfitting**. Si è optato pertanto per una tecnica di **Oversampling**, nello specifico **SMOTENC** (Synthetic Minority Over-sampling Technique Nominal Continous). Questa scelta ha permesso di generare sinteticamente nuove istanze per le classi minoritarie, arricchendo lo spazio delle feature e consentendo al modello di apprendere confini decisionali più robusti senza sacrificare i dati reali della classe dominante.



Un aspetto critico dell'implementazione ha riguardato la prevenzione del ***Data Leakage***: la tecnica **SMOTENC** è stata applicata esclusivamente al Training Set (dopo lo split dei dati), lasciando inalterato il Test Set**.** Questo accorgimento garantisce che la generazione di dati sintetici, non influenzi la fase di valutazione, assicurando che le metriche finali riflettano le prestazioni del modello su dati realistici e non completamente artefatti.

Dato che il dataset è composto da solamente 1000 istanze è stato ritenuto opportuno impiegare l'**80 %** di esso (alterato con SMOTENC) per i dati di training e il **20 %** rimanente per i dati di test, così da avere un modello ben addestrato.

3.9 Normalizzazione del dataset

Il problema principale del dataset era la presenza di molte variabili categoriche, per cui è stato deciso di codificare le variabili categoriche utilizzando la codifica label-encoding e l'utilizzo delle variabili "dummy", perché i modelli di Machine Learning lavorano con valori numerici. La variabile target è stata codificata nel seguente modo:

(es. Negative=0, Neutral=1, Positive=2)

Per quanto riguarda invece le feature numeriche è stata applicata una trasformazione di **scaling** utilizzando il **MinMaxScaler**, portando tutte le feature numeriche nel range **[0, 1**]. Questa scelta progettuale è stata fondamentale per un motivo in particolare:

L'algoritmo di sovracampionamento sintetico (SMOTENC), utilizzato per bilanciare le classi, si basa sul calcolo della distanza Euclidea tra i campioni. Senza normalizzazione, le feature con ordini di grandezza maggiori (es. *Academic_Score*) avrebbero dominato il calcolo della distanza rispetto a quelle con valori piccoli (es. *Daily_Gaming_Hours)* generando bias dannosi per il modello.

Esempio Dataset normalizzato (Gaming_Hours_Normalized.csv)

Age	Daily_Gam	Sleep_Hou	Stress_Lev	Focus_Lev	Academic_	Gender_M	Occupatio	Game_Typ	Game_Typ	Game_Typ	Game_Typ	Game_Typ	Primary_G	Primary_Gam	Performance_Impact
0.1764705	0.6363636	0.0249999	0.5714285	0.1666666	0.3500000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0
0.9999999	0.0909090	0.2250000	0.0	0.6666666	0.3000000	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1
0.4705882	0.2727272	0.875	0.2857142	0.8333333	0.6750000	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	2
0.8235294	0.0909090	0.1000000	0.7142857	0.6666666	0.4000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1
0.0588235	0.2909090	0.625	0.7142857	0.6666666	0.3000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1
0.6470588	0.8727272	0.375	0.9999999	0.8333333	0.5750000	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1
0.9999999	0.7090909	0.675	0.1428571	0.6666666	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1
0.4117647	0.2727272	0.9249999	0.4285714	0.8333333	0.5	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2
0.8235294	0.0727272	0.9249999	0.7142857	0.1666666	0.7749999	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	2
0.4117647	0.7454545	0.55	0.8571428	0.6666666	0.625	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1
0.1176470	0.8545454	0.1499999	0.2857142	1.0	0.6750000	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0
0.9999999	0.5272727	0.9750000	0.0	0.8333333	0.1750000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
0.1764705	0.4363636	0.4499999	0.5714285	0.6666666	0.2750000	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1
0.0	0.7636363	0.55	0.0	0.6666666	0.5	1.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1
0.9999999	0.0545454	0.875	0.1428571	0.0	0.7749999	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.5294117	0.9454545	0.1750000	0.8571428	0.1666666	0.8500000	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0
0.9999999	0.4545454	0.25	0.1428571	0.3333333	0.0250000	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1
0.9411764	0.7090909	0.0500000	0.1428571	0.0	0.5250000	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0
0.3529411	0.2181818	0.125	0.9999999	0.1666666	0.75	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1
0.7058823	0.3454545	0.3999999	0.0	0.8333333	0.1500000	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	1
0.4117647	0.1818181	0.4499999	0.8571428	0.1666666	0.4250000	1.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1
0.7647058	0.0545454	1.0	0.0	0.0	0.375	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	2
0.5294117	0.9090909	0.1750000	0.0	0.5	0.4000000	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0
0.0588235	0.2909090	0.9750000	0.0	0.8333333	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1
0.1764705	0.0727272	0.1750000	0.1428571	0.8333333	0.375	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1
0.5882352	0.4181818	0.575	0.7142857	0.3333333	0.3250000	1.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1
0.8235294	0.6727272	0.3000000	0.7142857	1.0	0.1000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0
0.5294117	0.5090909	0.125	0.7142857	0.0	0.375	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1
0.1764705	0.3090909	0.9750000	0.0	0.8333333	0.875	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1

4. REALIZZAZIONE DEL MODELLO

4.1 Scelta del modello

CLASSIFICATORE

Abbiamo deciso di sviluppare un modello di predizione **Random forest,** che è generalmente più robusto di un singolo **Decision Tree** perché riduce il rischio di overfitting mediando i risultati di molti alberi diversi.

DATASET

Il **Random forest** è stato addestrato in una prima versione con il dataset bilanciato tramite SMOTENC ed una seconda versione con il dataset non bilanciata.

4.2 Valutazione del modello

Per la valutazione del modello usiamo: **l'Accuracy, la Precision, il Recall, il F1-Score** e verrà mostrata la **matrice di confusione**.

4.3 Scelte implementative

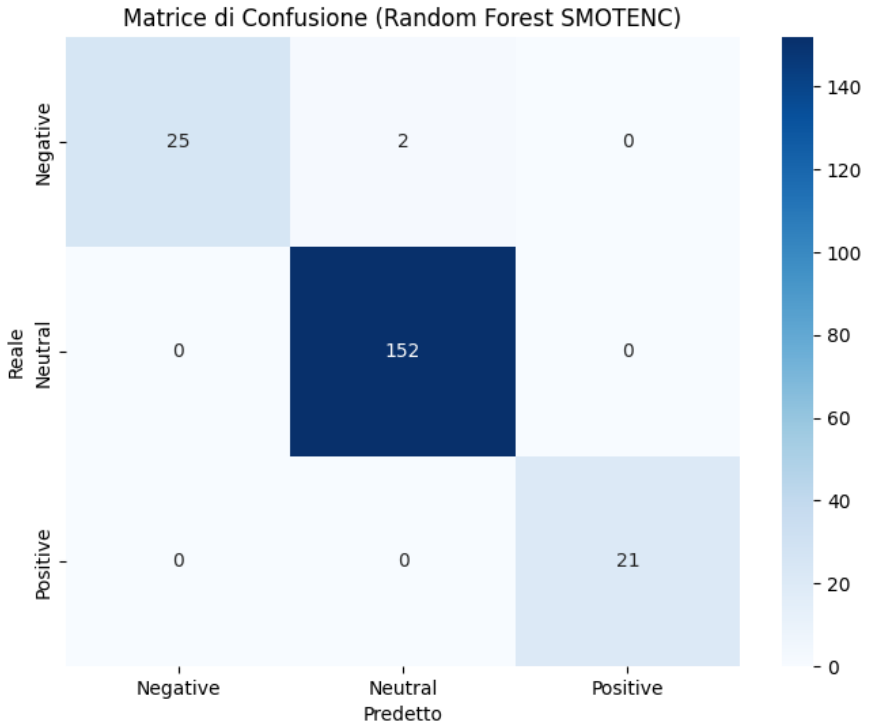
Per l'implementazione del sistema, delle metriche di valutazione e per il caricamento dei dati sono state utilizzate le principali librerie utili al Machine Learning, tra cui:

- **sklearn:** per la creazione dei modelli e il preprocessing dei dati.
- **pandas**: per il caricamento del dataset, per la creazione e la gestione dei dataframe.
- **imblearn**: per il bilanciamento del dataset.
- **numpy**: per la gestione dell'imputazione sulle variabili continue.
- **matplot** e **seaborn**: per la creazione dei grafici.

5. TESTING ED ANALISI DELLE PRESTAZIONI

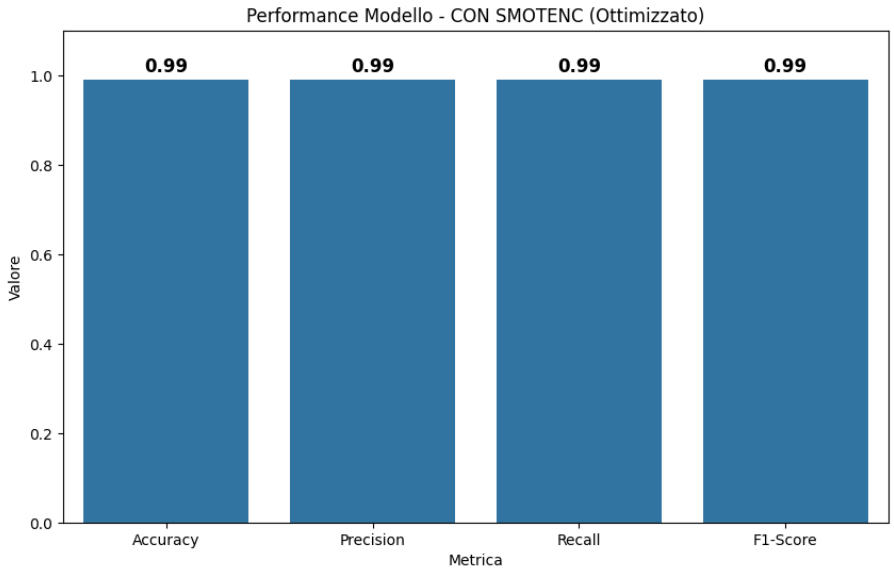
CON DATASET DI TRAINING BILANCIATO SMOTENC

Confusion matrix



Dalla **confusion matrix** si può notare che il modello ha commesso solo due predizioni errate

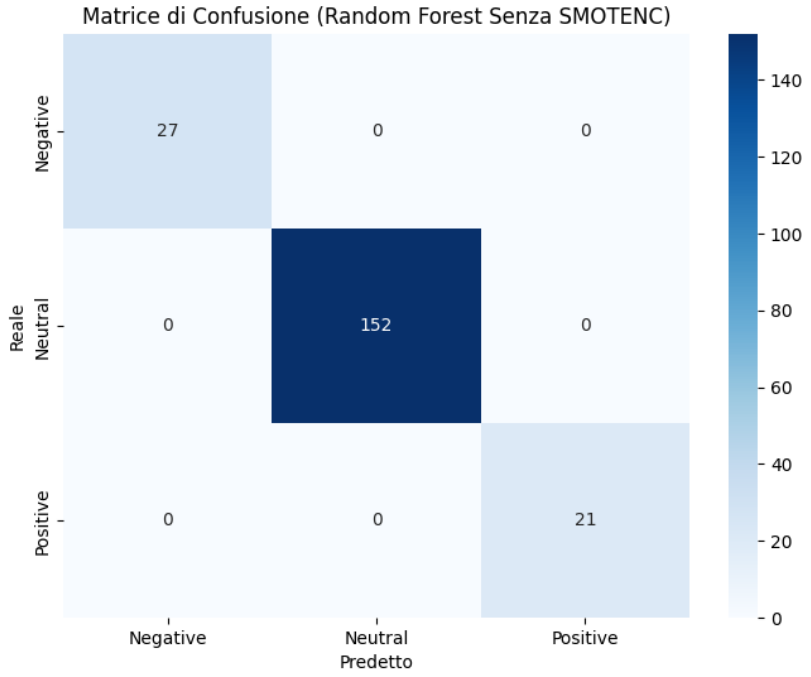
Performance



Dai risultati delle metriche Accuracy, Precision, Recall e F1-Score, che risultano essere quasi perfette, si può dedurre che i dati sono di origine sintetica.

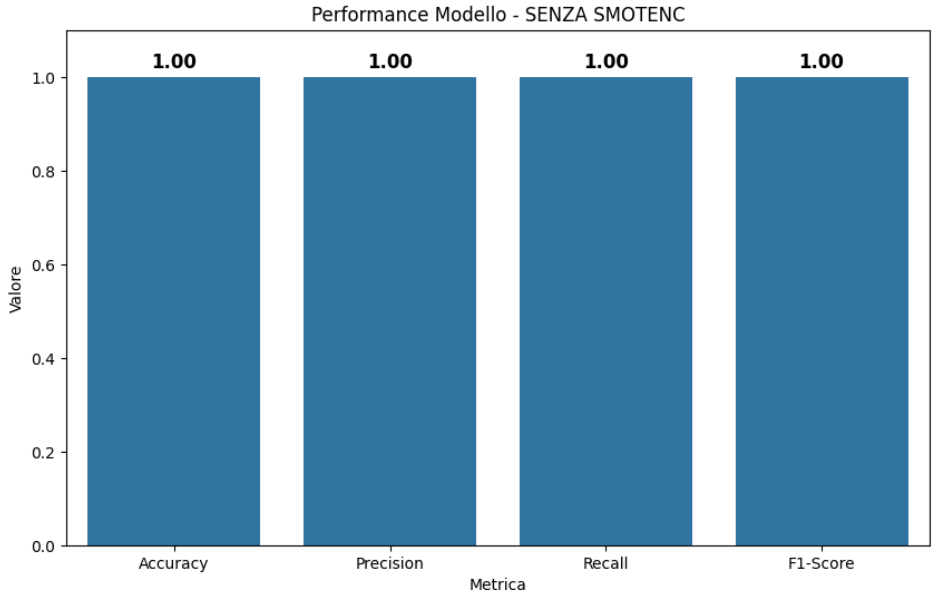
CON DATASET DI TRAINING NON BILANCIATO

Confusion matrix



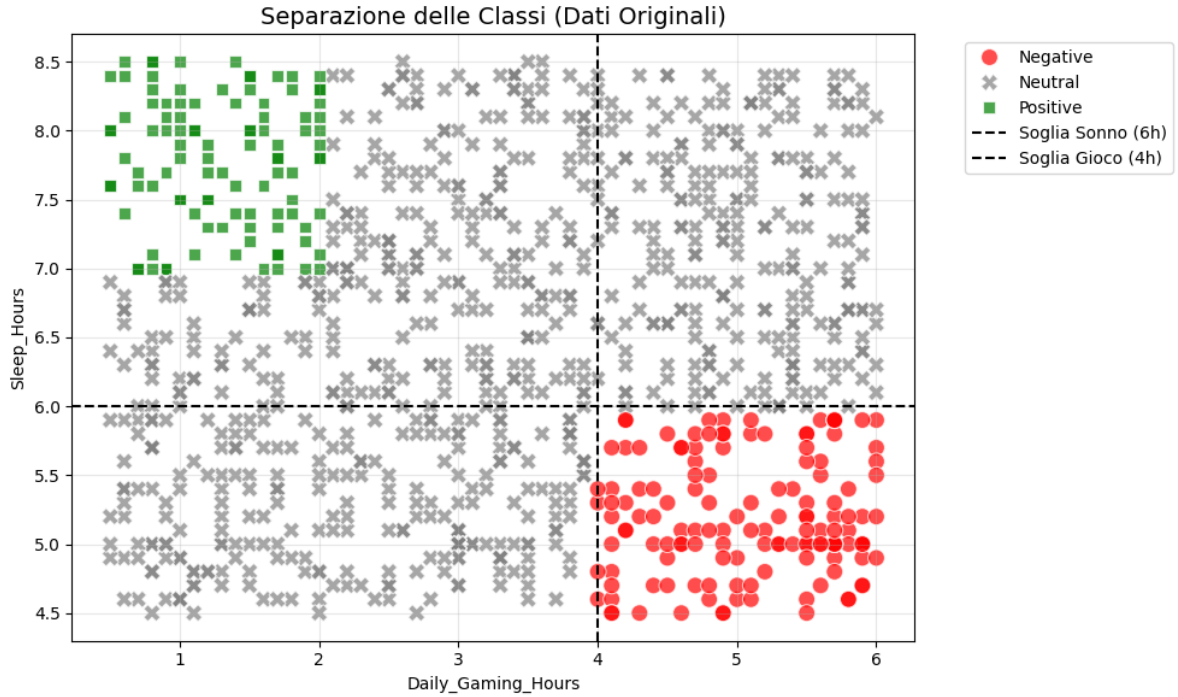
Addestrando il modello senza bilanciare i dati di training con SMOTENC paradossalmente si ottengono risultati perfetti. Una possibile causa di ciò potrebbe essere la distinzione netta tra le categorie positive, neutral e negative. Ciò verrà approfondito successivamente, analizzando l'importanza di ogni singola feature per la predizione.

Performance



Le performance confermano ciò che è già stato osservato dalla confusion matrix.

SEPARAZIONE DELLE CLASSI E FEATURE IMPORTANCE



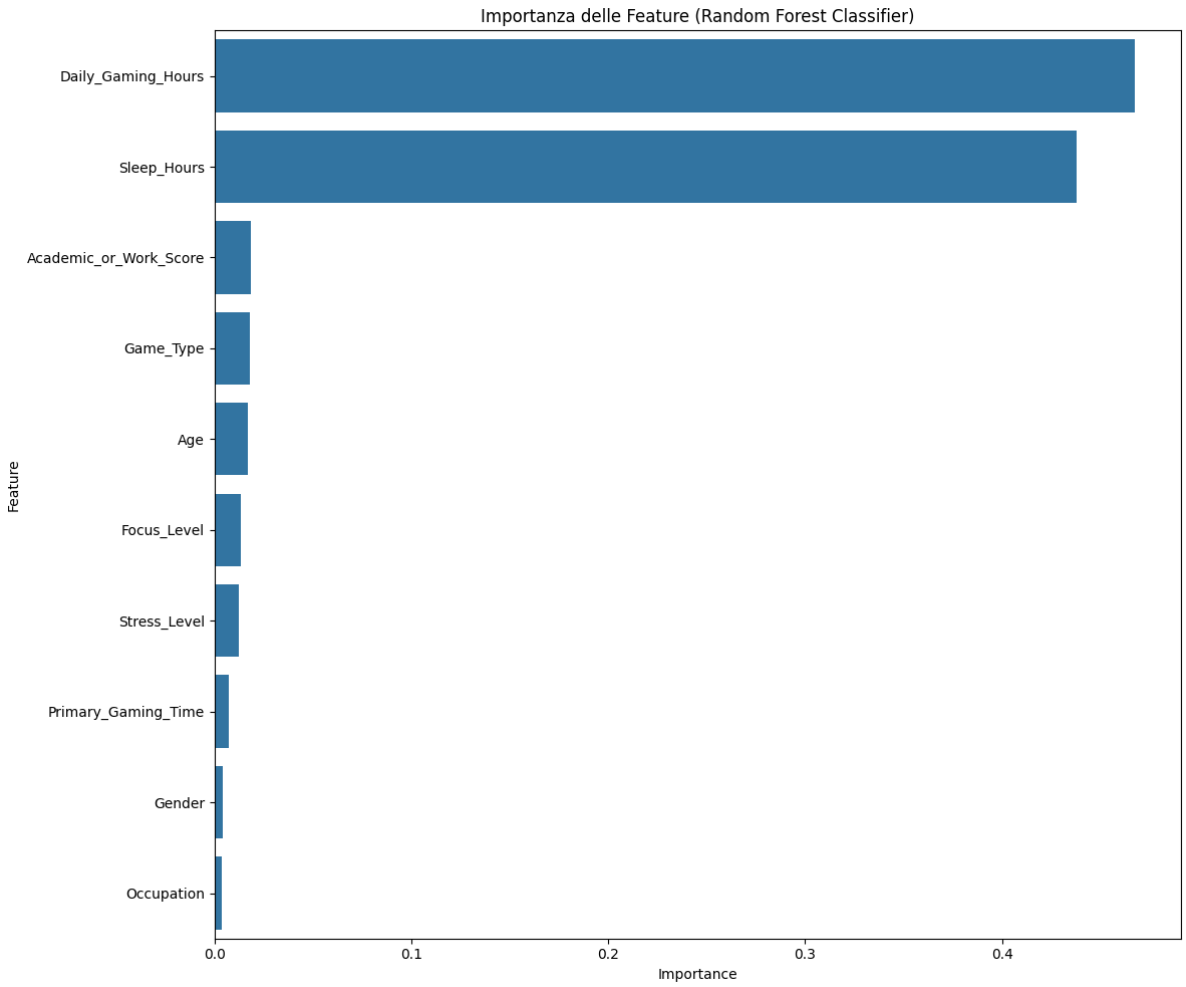
Questo grafico contribuisce a spiegare il perché il modello ha performance così ottimali (praticamente perfette). Da quanto si può osservare c'è una netta distinzione tra le categorie positive, neutral e negative:

Tutti gli studenti/lavoratori che dormono più di 7 ore e giocano fino a un massimo di 2 ore rientrano nella categoria positive.

Quelli che giocano per 4 ore o più e dormono meno di 6 ore rientrano nella categoria negative.

Il resto rientra nella categoria neutral.

Questo spiega anche il perché la maggior parte degli studenti/lavoratori rientra nella categoria neutral, che ha una fascia più ampia.



Il grafico mostra l'importanza di ogni singola feature per la predizione su Performance_Impact, calcolata utilizzando la **MDI** (Mean Decrease In Impurity) basata sull'indice di Gini. Come già anticipato nel grafico precedente le feature più decisive risultano essere le ore di sonno, e le ore di gioco di ogni studente/lavoratore.

6. EXTRA. MODELLO SCARTATO: RANDOM FOREST REGRESSOR

6.1 Introduzione

È stata preso in considerazione anche lo sviluppo di un secondo modello sullo stesso dataset, dato che, a detta dell'autore, è utilizzabile anche per lo sviluppo di modelli di machine learning basati sulla regressione. Per quanto riguarda la parte di data cleaning vengono compiute le stesse operazioni fatte per il modello illustrato precedentemente. L'unica modifica aggiuntiva è la rimozione di Performance_Impact per effettuare la predizione.

Per quanto riguarda le variabili categoriche in questo caso è stato utilizzato il **One-Hot encoding**, e non trattandosi di un task di classificazione non è stato necessario utilizzare SMOTENC sui dati di training, dato che le predizioni non vengono fatte su delle classi ma su valori numerici. La divisione tra dati di training e dati di test è la medesima utilizzata per il modello di classificazione. Per garantire che le feature abbiano lo stesso peso durante l'addestramento è stato utilizzato il **MinMaxScaler**.

Il classificatore utilizzato è il **random forest regressor,** e la variabile target è **Academic_or_work_score**. L'obiettivo è quindi dedurre il punteggio accademico o lavorativo di uno studente/lavoratore sulla base delle proprie abitudini.

Testing e analisi

METRICHE UTILIZZATE PER LA REGRESSIONE

In questo paragrafo verrà spiegato cosa sono e a cosa servono le metriche utilizzate per misurare la qualità del modello basato sulla regressione**.** Il **MAE** (acronimo di *Mean Absolute Error*, in italiano **Errore Medio Assoluto**) è una delle metriche più comuni utilizzate per valutare le prestazioni di un modello di **regressione**.

In termini semplici, il MAE misura la **media della grandezza degli errori** in un insieme di previsioni, senza considerare la loro direzione (ovvero, se la previsione è superiore o inferiore al valore reale). È la media delle distanze verticali assolute tra ogni punto dati reale e la linea di regressione (o iperpiano) prevista.

Formalmente, il MAE è definito dalla seguente equazione:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right|$$

Dove:

- n è il numero totale di osservazioni (campioni) nel dataset.

- Yi è il valore reale.
- Ŷi è il valore predetto.

Per comprendere appieno il MAE, è necessario analizzare le sue proprietà distintive:

1. Unità di misura intuitiva A differenza dell'MSE (Mean Squared Error), il MAE mantiene la stessa unità di misura della variabile target.

Esempio: Se stai prevedendo il prezzo delle case in euro (€), un MAE di 10.000 significa che, in media, il tuo modello sbaglia il prezzo (in eccesso o in difetto) di 10.000 €. Questo lo rende estremamente facile da interpretare per gli stakeholder non tecnici.

2. Trattamento degli Outlier (Robustezza) Il MAE è considerato più robusto agli outlier rispetto all'MSE o all'RMSE (Root Mean Squared Error).

- Poiché il MAE calcola la differenza assoluta (lineare), un errore molto grande non viene "elevato al quadrato".
- Di conseguenza, un singolo dato anomalo (outlier) non influenzerà drasticamente la media dell'errore complessivo, contrariamente a quanto accade con l'MSE che penalizza enormemente gli errori grandi.

3. Penalità Lineare Il MAE assegna un peso uguale a tutti gli errori. Un errore di 10 unità è esattamente il doppio peggiore di un errore di 5 unità. Non "punisce" gli errori grandi tanto severamente quanto le metriche quadratiche.

L'**RMSE** (acronimo di Root Mean Squared Error, in italiano Radice dell'Errore Quadratico Medio) è la metrica standard più diffusa per valutare un modello di regressione.

Rappresenta la radice quadrata della media degli errori al quadrato. In termini statistici, indica la deviazione standard dei residui (errori di previsione). Misura quanto sono "sparsi" questi residui attorno alla linea di regressione ottimale: più l'RMSE è basso, più i punti dati sono concentrati vicino alla linea di previsione.

L'RMSE si calcola estraendo la radice quadrata dell'MSE (Mean Squared Error):

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

- n è il numero totale di osservazioni (campioni) nel dataset.
- Yi è il valore reale.
- Ŷi è il valore predetto.
- (Ŷi -Yi) è il quadrato della differenza per ogni punto (errore quadratico).

Per capire la natura dell'RMSE, bisogna analizzare come manipola matematicamente l'errore:

1. Unità di misura coerente

Come il MAE, l'RMSE è espresso nella stessa unità di misura della variabile target (y).

Esempio: Se prevedi il prezzo delle case in euro (€), l'MSE ti darebbe un risultato in "Euro al quadrato" (difficile da interpretare), mentre l'RMSE ti riporta il valore in euro (€), rendendolo immediatamente comprensibile.

2. Sensibilità agli Outlier (**Penalità Quadratica)**

Questa è la differenza cruciale rispetto al MAE. Prima di fare la media, gli errori vengono elevati al quadrato.

- Questo processo amplifica enormemente gli errori grandi.
- Un errore di 10 non vale il doppio di un errore di 5, ma vale **quattro volte tanto**.
- Di conseguenza, l'RMSE "punisce" severamente il modello se fa anche solo poche previsioni molto sbagliate (outlier).

3. Proprietà Matematiche (Differenziabilità)

A differenza del MAE (che usa il valore assoluto e ha un "angolo" non derivabile nello zero), la funzione quadratica alla base dell'RMSE è liscia e differenziabile ovunque. Questo la rende computazionalmente più efficiente per molti algoritmi di ottimizzazione (come il Gradient Descent).

L'**R^2** è una metrica statistica che rappresenta la **proporzione della varianza** della variabile dipendente (il target y) che è prevedibile o "spiegata" dalle variabili indipendenti (le feature X) del modello.

A differenza di MAE e RMSE, che misurano l'errore assoluto (in unità di misura del problema, es. euro, metri), l'R^2è un **punteggio adimensionale** (solitamente tra 0 e 1) che indica la "**bontà di adattamento**" (*goodness of fit*) del modello rispetto ai dati.

La Formula Matematica

Per comprendere l'R^2, dobbiamo definire due quantità:

1. **TSS (Total Sum of Squares):** La varianza totale intrinseca nei dati. È la somma degli errori se usassimo semplicemente la **media** come predizione per tutti.
2. **RSS (Residual Sum of Squares):** La somma degli errori quadratici residui del **tuo modello**.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = 1 (Perfetto): Il modello spiega perfettamente tutta la variabilità dei dati. Le previsioni corrispondono esattamente ai valori reali (RSS = 0).

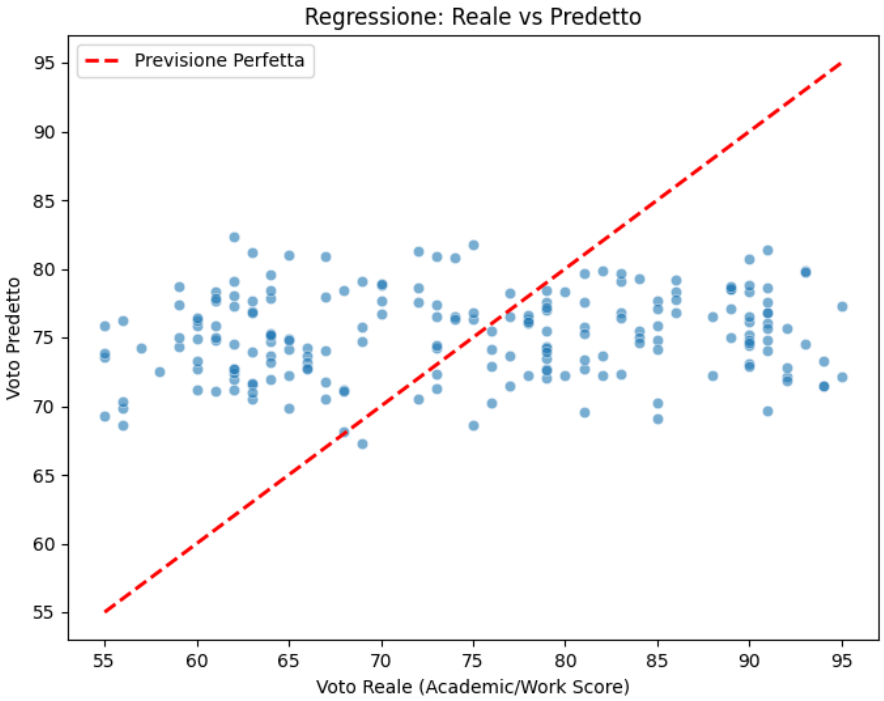
R^2 = 0(Modello Base): Il modello non spiega nulla in più rispetto alla semplice media dei dati. È come tirare a indovinare usando sempre il valore medio.

0 < R^2 < 1: Indica la percentuale di varianza spiegata.

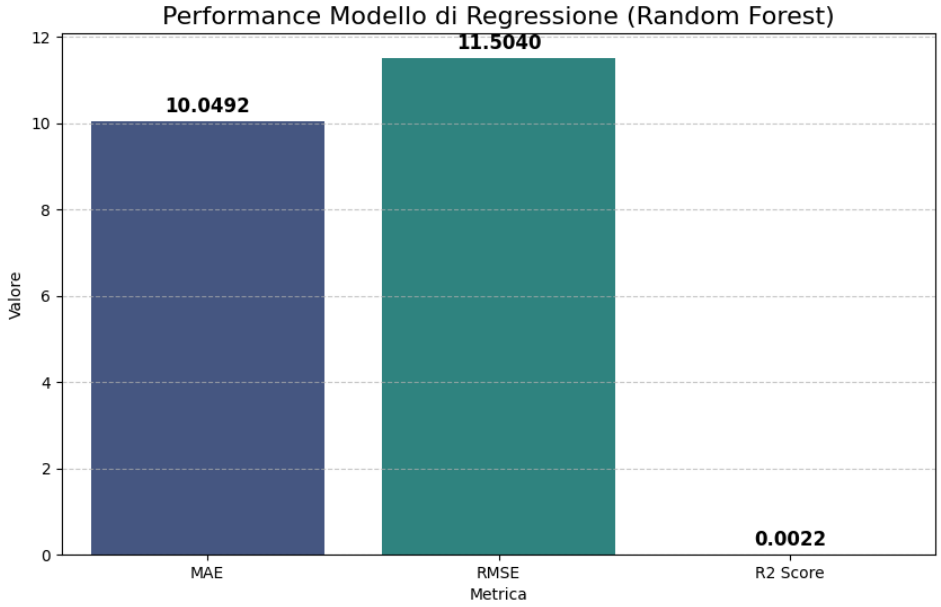
Esempio: Un R^2 di **0.80** significa che l'80% delle variazioni nel target è spiegato dalle feature del tuo modello, mentre il restante 20% è rumore o dovuto a fattori non considerati.

R^2 < 0 (Negativo): Sì, è possibile (spesso crea confusione). Succede se il modello è **peggiore** della semplice media. Indica un modello completamente sbagliato o mal calibrato (es. usare una regressione lineare su dati altamente non lineari senza trasformazioni)

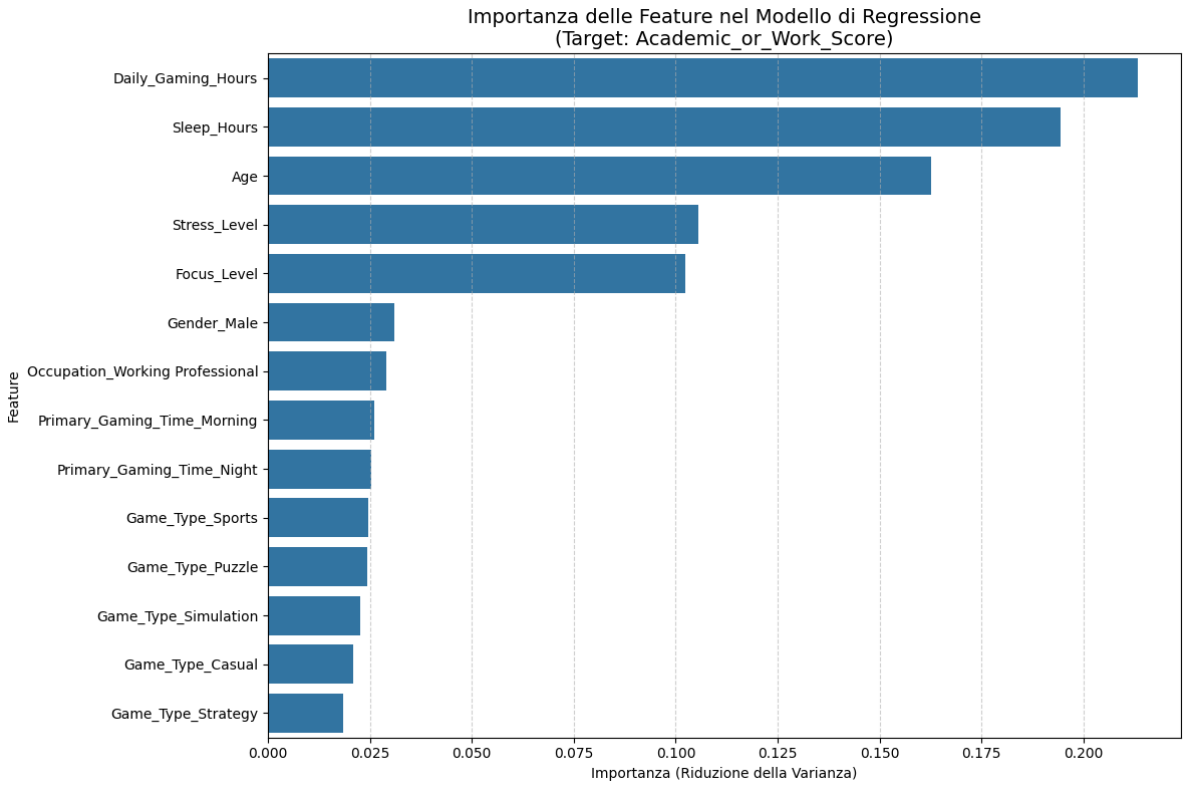
RISULTATI



Dal grafico si evince che il modello ha una precisione bassissima sulla variabile **Academic_Or_Work_Score**, dato che i risultati delle predizioni sono paragonabili a una scelta casuale. Questo è dipeso dalla scarsa correlazione causale tra le feature.



Il grafico delle performance riconferma ciò che è stato detto in precedenza, sottolineando quanto sia poco performante.



Il grafico presentato illustra l'importanza di ogni singola feature per la predizione dell'Academic_Or_Work_Score, valutata in base alla **riduzione dell'impurezza (varianza)**. In questo contesto di regressione, tale metrica quantifica il contributo di ciascuna feature alla diminuzione dell'**errore quadratico medio (MSE)** all'interno degli alberi decisionali: più alto è il valore, maggiore è la capacità della variabile di spiegare la variabilità del target e ridurre l'incertezza della predizione.

Le tre feature più decisive sono **Daily_Gaming_Hours**, **Sleep_Hours** e **Age**. In particolare, le ore di gioco giornaliere risultano essere la feature presa più in considerazione dal modello per la predizione (con un valore superiore a 0.20), confermando che la *quantità* di tempo speso ha un impatto molto più significativo sul rendimento rispetto alla tipologia di attività.

Variabili come **Stress_Level** e **Focus_Level** mostrano un'importanza minore (circa 0.10), indicando che il benessere mentale è in parte un correttivo importante per raffinare la stima del voto.

L'ultima parte del grafico è composta da **variabili binarie** (es. Game Type, Primary_Gaming_Time, Gender). I loro bassi valori di Gini Importance suggeriscono che, agli occhi del modello, *a cosa* si gioca, *quando* si gioca è trascurabile rispetto al volume totale di ore investite per il gaming e al riposo perso.

6.2 Considerazioni sul modello scartato

Si può affermare che le dichiarazioni fatte dal creatore del dataset, che afferma il possibile utilizzo di quest'ultimo per la creazione di modelli basati sulla regressione, sono in parte

false. Questo perché il dataset di base non fornisce delle feature che possano guidare il modello a stimare correttamente il voto accademico o la performance lavorativa del soggetto.

7. CONCLUSIONI

A seguito dell'analisi del dataset e sviluppo di due modelli differenti su di esso, possiamo affermare che parte di ciò che viene descritto dal suo autore va in conflitto con i risultati ottenuti. Con il modello principale del progetto, basato sulla classificazione, abbiamo ottenuto risultati fin perfetti. Questo dimostra che il dataset contiene dati artefatti, e non dati sintetici che provengono da pattern reali, o quantomeno realistici, derivati da studi, come invece viene affermato dall'autore. Il primo segnale di allarme è sicuramente la totale assenza di outlier, poiché in uno scenario quantomeno realistico, ci si potrebbe aspettare la presenza di persone con delle ore di gioco eccessive e scarse ore di sonno, che comunque ottengono buone performance a lavoro o a scuola, con un impatto generalmente positivo. Altro aspetto negativo del dataset è lo scarso rapporto causale tra le feature (ovviamente escludendo quelle ridondanti), come visto dalla matrice di correlazione. Questo aspetto rende estremamente imprecise le predizioni su feature che non siano la **Performance_Impact**, come visto per il modello basato sulla regressione sulla variabile **Academic_Or_Work_Score**.

SVILUPPI FUTURI

Un futuro sviluppo cruciale potrebbe riguardare la raccolta di dati reali tramite questionari somministrati a studenti universitari e lavoratori. Questo permetterebbe di verificare se le correlazioni nette osservate reggono anche di fronte a dati reali e, di conseguenza, più complessi da predire, dove la relazione tra gioco e rendimento è spesso più sfumata e non lineare.

Un altro possibile sviluppo potrebbe essere cercare feature più specifiche, utili e impattanti (feature engineering) per migliorare la capacità predittiva in contesti reali. Per esempio si potrebbe considerare:

- **Tipologia di sessione:** Distinguere tra gioco 'competitivo' (spesso stressante) e 'cooperativo/rilassante' (che potrebbe invece ridurre lo stress).
- **Dispositivo di gioco**: Questo parametro potrebbe essere molto utile per capire gli effetti del gaming sulla salute e performance di studenti/lavoratori; per esempio, si potrebbe verificare se hanno performance migliori i soggetti che giocano da computer o console, che di solito giocano con schermi abbastanza grandi, oppure chi gioca da mobile, di solito con uno schermo molto più piccolo.
- **Postazione di gioco**: Tale feature potrebbe indicare il grado di comodità della postazione, che contribuirebbe a capire se soggetti che giocano in luoghi confortevoli ne risentono meno rispetto a chi gioca in postazioni scomode.
- **Tipo di lavoro/studio** e **Ore di lavoro/studio**: Queste due feature sarebbero molto utili per capire quali lavori o corsi di studio in particolare, e con quante ore,

contribuiscono a un livello di stress maggiore e necessità di impiegare più tempo per il sonno. Per esempio:

Prendiamo in considerazione due soggetti differenti: un cuoco e un impiegato d'ufficio part-time. Se entrambi giocano sei ore al giorno, teoricamente il Performance_Impact del cuoco sarà peggiore, o quantomeno uguale, a quello dell'impiegato, che ha un lavoro ipoteticamente meno stressante e che comprende meno ore della giornata.