



VIDEO GAME SUCCESS

Alessio Fasano

COSA È VIDEO GAME SUCCESS?

La crescente popolarità dei videogiochi come forma di intrattenimento e arte interattiva ha portato a un'attenzione sempre maggiore verso la comprensione di ciò che rende un videogioco di successo.

Questo progetto si propone di utilizzare modelli di machine learning per analizzare dati storici e predire le probabilità di successo di un videogioco, fornendo così uno strumento di supporto decisionale per sviluppatori e publisher.





COSA PERMETTE DI FARE IL SISTEMA?

- **PREVISIONE NUMERICA DEL SUCCESSO CHE UN VIDEOGIOCO AVRÀ.**
- **IDENTIFICAZIONE DELLE CORRELAZIONI TRA I DATI, FORNENDO STATISTICHE UTILI PER COMPRENDERE LE PREVISIONI.**
- **VISUALIZZAZIONE GRAFICA DELLA CONOSCENZA.**



1. SPERIMENTAZIONE

**Apprendimento Supervisionato
con Stacking**



PREDIRE IL SUCCESSO DI UN VIDEOGIOCO

Il sistema utilizza modelli di machine learning per analizzare un insieme di esempi, coppie input + output, per predire l'output per nuovi casi di cui si conosce solo l'input.

Il modello utilizza dati come genere, piattaforma, players e altri fattori chiave per fornire una stima continua e precisa sul punteggio che gli utenti daranno al videogioco.

L'algoritmo di Ensemble Learning utilizzato dal sistema è stato lo Stacking

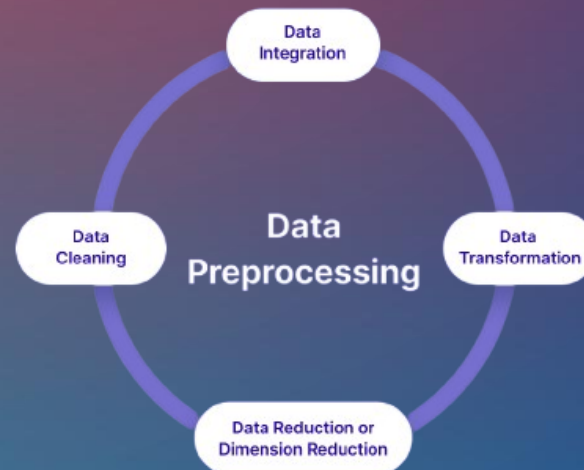


DATASET

Il dataset utilizzato per l'addestramento e il test del regressore è l'concatenazione di due dataset che contengono una raccolta di caratteristiche legate all'entità videogioco .

È stata effettuata un'opportuna procedura di preprocessing sui dati(gestione dei valori mancanti, del tipo di valore continuo o discreto e la normalizzazione dei valori)

Il dataset contiene in tutto 11150 esempi che è stato diviso in set di training e un set di test, precisamente 70% per il training e 30% per il test.



SCELTA DEGLI IPERPARAMETRI E RISULTAT

Per ogni base learners e il meta-learners è stata effettuata una fase di validazione per poter scegliere gli iperparametri.

La validazione è stata effettuata utilizzando la tecnica del k-fold con $k = 10$.

BASE LEARNERS

Sono stati scelti questi base learners perché meglio si adattano al dataset utilizzato

Random Forest Regressor

Gradient Boosting Regressor

Ridge Regressor

Support Vector Regressor

META LEARNER

Sono stati testati questi meta learners in fase di sperimentazione e scelto il migliore per il nostro sistema

Linear Regression

Multy-Layer Perceptron Regressior

BASE LEARNING:

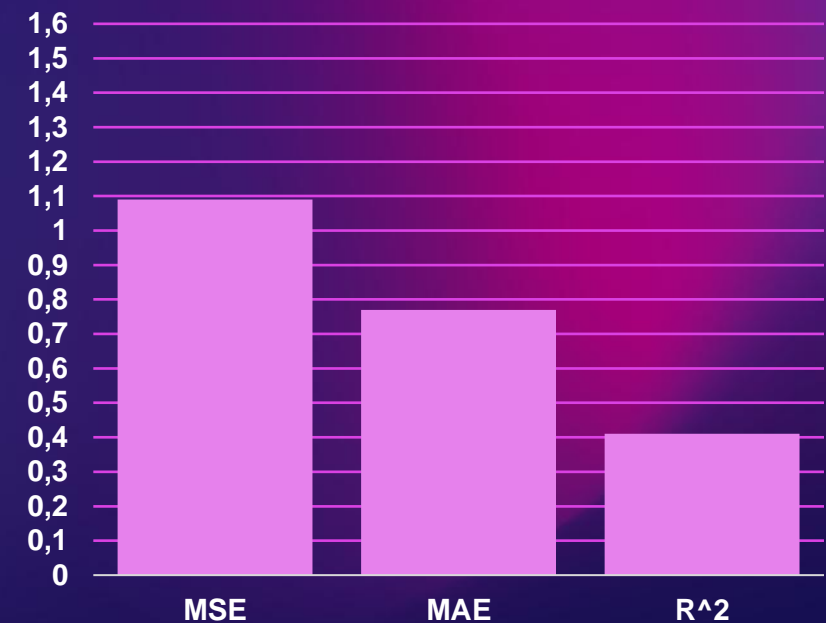
Random Forest Regressor

RFr In base ai dati di training etichettati, l'algoritmo genera una funzione ottimale che predice valori continui per nuovi esempi.

- 'max_depth': None,
- 'n_estimators': 300,
- 'random_state': 0

In fase di validazione, MSE: 1.09 (+/- 0.06)

IN FASE DI VALIDAZIONE



BASE LEARNING:

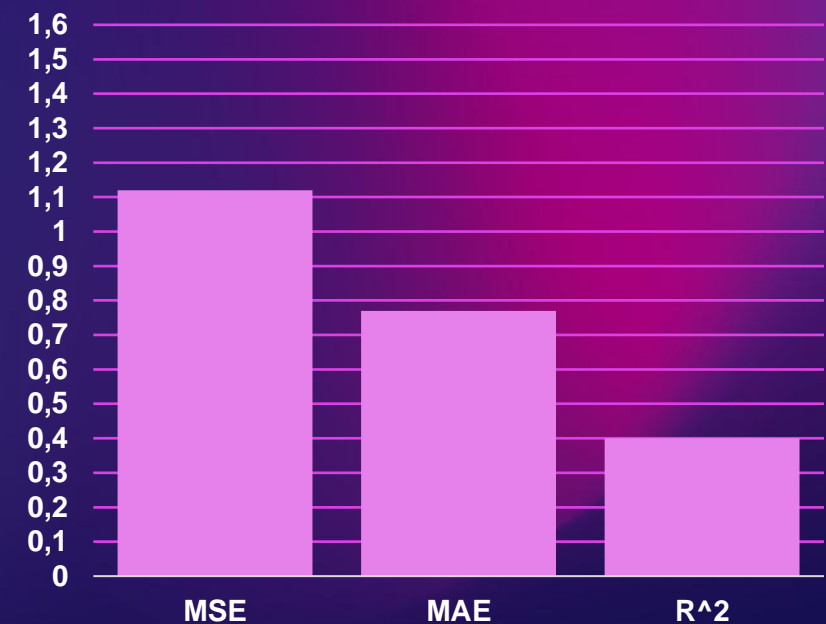
Gradient Boosting Regressor

Gbr In base ai dati di training etichettati, l'algoritmo genera una funzione ottimale che predice valori continui per nuovi esempi.

- 'learning_rate': 0.1,
- 'max_depth': 7,
- 'n_estimators': 100,
- 'random_state': 0

In fase di validazione, MSE: 1.12 (+/- 0.06)

IN FASE DI VALIDAZIONE



BASE LEARNING:

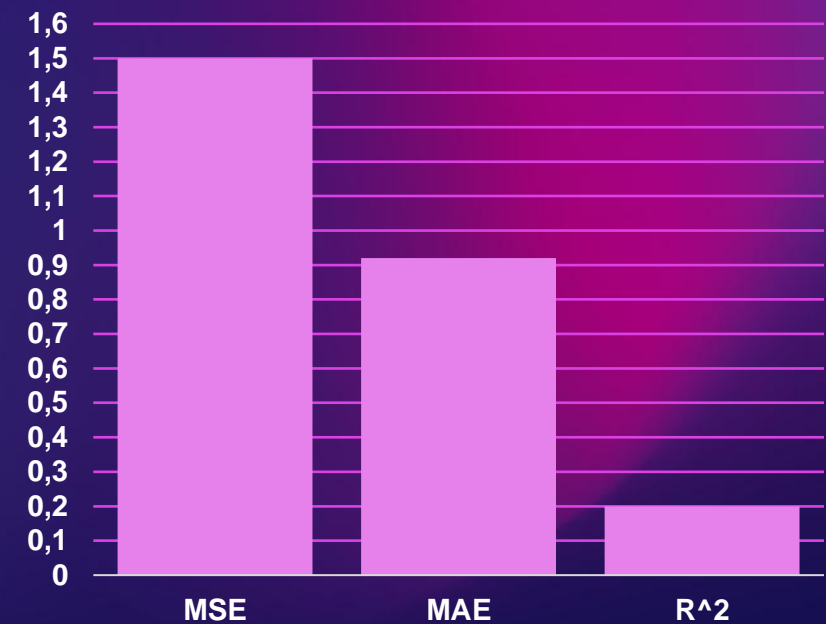
Ridge Regressor

Rr In base ai dati di training etichettati, l'algoritmo genera una funzione ottimale che predice valori continui per nuovi esempi.

- 'alpha': 0.1,
- 'random_state': 0

In fase di validazione, MSE: 1.50 (+/- 0.08)

IN FASE DI VALIDAZIONE



BASE LEARNING:

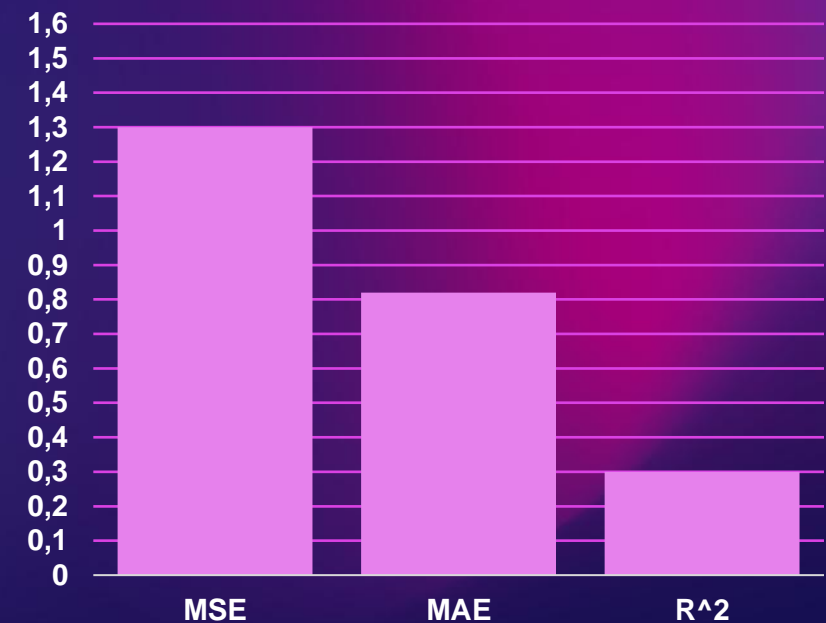
Support Vector Regressor

SVr In base ai dati di training etichettati, l'algoritmo genera una funzione ottimale che predice valori continui per nuovi esempi.

- 'C': 10,
- 'degree': 2,
- 'gamma': 'scale',
- 'kernel': 'rbf'

In fase di validazione, MSE: 1.30 (+/- 0.08)

IN FASE DI VALIDAZIONE



STACKING

Si utilizzano un gruppi di classificatori eterogenei

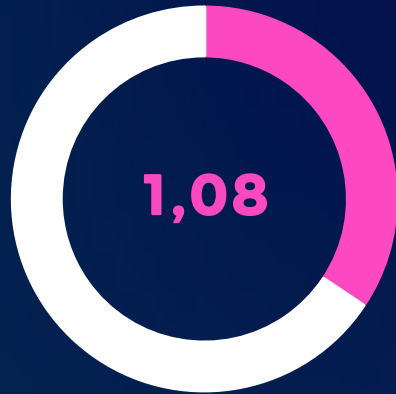
Ci sono due tipi di classificatori:

- **Base Learner:** Vengono addestrati sul dataset e le loro predizioni vengono utilizzate per generare un nuovo dataset, in cui le meta-feature sono le predizione di ogni base learner e la feature target è il valore reale dell'input.
- **Meta-Learner:** viene addestrato sul nuovo dataset ed effettua la predizione finale.



RISULTATI VALIDAZIONE STACKING

Mean Squared Error



STACKING

Meta learn:

Linear Regression

MAE: 0.75 (+/- 0.02)

R2: 0.42 (+/- 0.04)



STACKING

Meta learn:

Multy-Layer Perceptron

MAE: 0.77 (+/- 0.03)

R2: 0.41 (+/- 0.05)

RISULTATI OTTENUTI NELLA FASE DI TEST

Mean Squared Error



STACKING

Meta learn:
Random Forest
MAE: 0.70
R2: 0.45



STACKING

Meta learn:
Multy-Layer Perceptron
MAE: 0.70
R2: 0.45

MODELLO MIGLIORE

**Random
Forest
Regressor**

Linear Regression

**Gradient
Boosting
Regressor**

**Ridge
Regressor**

**Support
Vector
Regression**



2. BASE DI CONOSCENZA



BASE DI CONOSCENZA

Per permettere l'identificazione delle correlazioni tra le caratteristiche di un videogioco è stata utilizzata una base di conoscenza.

Una base di conoscenza è costituita da un insieme di formule espresse mediante un linguaggio di rappresentazione della conoscenza di un dominio e ammette come formule determinate proposizioni.

Il dominio rappresentato è il dominio dei videogiochi, che include aspetti quali generi, platform, players e altre caratteristiche che influenzano il successo di un titolo.

La sintassi utilizzata è quella definita da Prolog, un linguaggio logico autonomo progettato per la programmazione dichiarativa e utilizzato direttamente all'interno di un ambiente dedicato o tramite interpreti Prolog.

FATTI

Gli individui nella base di conoscenza in Prolog sono rappresentati e strutturati nel seguente modo:

Game(game_name, developer, user_score, metacritic, critics, playtime, achievements_count, released_year, released_month, released_day, is_in_top_100_publisher, is_in_top_100_developer genre, platform, publisher).



REGOLE

Le regole in Prolog sono dichiarazioni logiche che permettono di dedurre nuovi fatti o rispondere a query basate su relazioni definite dai fatti esistenti. Si costruiscono usando clausole che combinano condizioni con operazioni logiche.

REGOLA PER TROVARE LA PIATTAFORMA CON PIÙ GIOCHI POPOLARI

```
platform_with_most_games(MaxPlatforms) :-  
    findall(Platform,  
        (game(_, _, UserScore, _, _, _, _, _, _, _, _, _, _, _, _,  
            Platform, _),  
            UserScore > 8.5),  
        PlatformsList),  
    msort(PlatformsList, SortedPlatforms),  
    clumped(SortedPlatforms, CountedPlatforms),  
    max_member(_-MaxCount, CountedPlatforms),  
    include(=(_-MaxCount), CountedPlatforms,  
        MaxPlatforms).
```

REGOLA PER TROVARE I GENERI PIÙ POPOLARI

```
most_popular_genres(MaxGenres) :-  
    findall(Genre,  
        (game(_, _, UserScore, _, _, _, _, _, _, _, _, _, _, _,  
            _),  
            UserScore > 8.5),  
        GenresList),  
    msort(GenresList, SortedGenres),  
    clumped(SortedGenres, CountedGenres),  
    max_member(_-MaxCount, CountedGenres),  
    include(=(_-MaxCount), CountedGenres, MaxGenres).
```

REGOLA PER TROVARE I GIOCHI CON IL PUNTEGGIO PIÙ ALTO DEGLI UTENTI

```
topRatedGames(TopGames) :-  
    findall(Game-UserScore,  
        (game(Game, _, UserScore, _, _, _, _, _, _, _, _, _, _,  
            _, _, _),  
            UserScore > 8.5),  
        GamesList),  
    keysort(GamesList, SortedGames),  
    reverse(SortedGames, ReversedGames),  
    random_select(5, ReversedGames, TopGames).
```


QUERY CON VARIABILI

Per interrogare il sistema sulle caratteristiche di un gioco o sul significato di determinati attributi, si possono effettuare query con variabili sulla base di conoscenza.

Ad esempio, se l'utente desidera conoscere lo sviluppatore con più giochi di successo, può formulare una domanda attraverso la query: `developer_with_most_games(MaxDevelopers)`.

INTERFACCIA UTENTE

**L'utente interagisce
con l'interfaccia che
traduce le interazioni
alla base di
conoscenza
visualizzando le
risposte**

Video Game Success Predictor

Seleziona la modalità:

- ☒ Prevedi il successo!
☐ Video Game Knowledge Base

Previsione del Successo dei Videogiochi

Piattaforma

Classic Macintosh

Genere

Alternative

Numero di Achievements

20

Anno di Uscita

2030

Mese di Uscita

8

Giorno di Uscita

1

Prevedi Successo

Punteggio previsto: 5.46

INTERFACCIA UTENTE

Video Game Success Predictor

Seleziona la modalità:

- ☐ Prevedi il successo!
- ☒ Video Game Knowledge Base

Video Game: Knowledge Base

Seleziona la modalità per interagire con la knowledge base:

- ☐ Interroga la Knowledge Base
- ☒ Utilizza le Relazioni nella KB
- ☐ Grafo interattivo

Relazione tra i Dati nella Knowledge Base

Trova la piattaforma con più giochi popolari tra gli Utenti

Trova lo sviluppatore con più giochi popolari tra gli Utenti

Trova i generi più popolari tra gli Utenti

Trova i giochi con il punteggio più alto degli utenti

Trova i giochi con il punteggio più alto dei critici

Video Game: Knowledge Base

Seleziona la modalità per interagire con la knowledge base:

- ☒ Interroga la Knowledge Base
- ☐ Utilizza le Relazioni nella KB
- ☐ Grafo interattivo

Puoi applicare i seguenti filtri alla tua ricerca:

Seleziona i filtri che vuoi applicare:

Choose an option



Cerca nella Knowledge Base



GRAZIE

Alessio Fasano

Mat. 738662

a.fasano32@studenti.uniba.it