



SAPIENZA  
UNIVERSITÀ DI ROMA

Sapienza Università di Roma · Facoltà di Lettere e Filosofia

# Riconoscimento di Manovre Complesse in Ambienti Videoludici Competitivi



# ROCKET LEAGUE

Un approccio basato su Random Forest e Parsing Vettoriale

**Corso:** Corso di Laurea in Filosofia e Intelligenza Artificiale – Intelligenza Artificiale 2

**Docente:** Prof.ssa Flavia Monti

**Gruppo di lavoro:** Brando Cappucci (2139484); Matteo Maria Martino Remondini (2136808); Lorenzo Stranieri (2153723)

**Data:** 2 Febbraio 2026



# Sommario / Abstract

- **Obiettivo:** classificare skillshot in Rocket League a partire da dati grezzi di telemetria.
- **Sfida:** serie temporali a lunghezza variabile e sbilanciamento delle classi.
- **Approccio:** parsing vettoriale + feature engineering fisico-comportamentale (quartili, toggle count).
- **Dati:** UCI Rocket League Skillshots (298 sequenze, 7 classi, 6892 frame, durata media 23.13 frame).
- **Modelli:** confronto Random Forest vs SVM con validazione stratificata.
- **Risultati:** accuratezza 86.67% su test, F1-Weighted ~0.86.

**298**

Sequenze

**7**

Classi

**86.67%**

Accuratezza test

**~0.86**

F1-Weighted

Pipeline riproducibile, interpretabile e orientata al coaching competitivo.



# Contesto: Rocket League ed E-Sport

- **E-sport:** grandi volumi di dati utili per coaching e analisi prestazionale.
- **Rocket League:** gameplay basato sulla fisica, ibrido calcio + veicoli a razzo.
- **Skillshot:** manovre aeree complesse (es. Ceiling Shot, Air Dribble) definite da sequenze di input e stati fisici.
- **Opportunità IA:** classificazione automatica delle manovre per analisi e training dei giocatori.



## Calcio

Obiettivo, palla, campo



## Veicoli a razzo

Boost, salto, rotazioni



## Input

Sequenze ad alta frequenza



## Telemetria

Posizione, velocità, rotazioni



## Classificazione IA

Supporto al coaching e analisi

GARAGE



# Domanda di Ricerca

È possibile addestrare un modello di Machine Learning per **classificare correttamente una manovra** osservando solo i **dati grezzi di telemetria**, indipendentemente dalla **durata** dell'esecuzione e dallo **stile del giocatore**?

## Durate variabili delle manovre



**Air Dribble**

~ 50 frame



**Power Shot**

~ 10 frame

Esempi dal dataset mostrano ampia eterogeneità di durata tra skillshot.

## Implicazioni per il modello

- Necessità di **rappresentazioni robuste** a lunghezze variabili.
- **Feature engineering compatto** (es. quartili, toggle count) per condensare serie temporali.
- **Valutazione adeguata** in presenza di sbilanciamento (es. F1-Weighted, CV stratificata).





# Obiettivi del Progetto

---

1

**Acquisizione e pulizia dei dati** in modo riproducibile ed efficiente.

2

**Feature engineering mirato** per superare la lunghezza variabile delle serie temporali.

3

**Confronto tra modelli** (Random Forest vs SVM) con protocollo di validazione adeguato.

4

**Interpretazione dei risultati** per identificare i segnali fisici e comportamentali discriminanti.

# Dataset: UCI Rocket League Skillshots

---

**298**

Sequenze  
Campioni multivariati

**7**

Classi di manovre  
Es. Power Shot, Air Dribble

**6,892**

Frame totali  
Dopo parsing vettoriale

**23.13**

Durata media (frame)  
Eterogeneità tra classi

**80/20**

Split stratificato  
Hold-Out con stratificazione



# Parsing Vettoriale Ottimizzato



## Problema

- File **.data ibrido**: righe etichetta alternate a righe dati.
- Parsing iterativo tradizionale **lento** e soggetto a errori.
- Necessità di ricostruire **Sequence\_ID** e **Class\_Label**.



## Soluzione con Pandas

- **Maschere booleane** per individuare le righe etichetta.
- **cumsum()** per generare il **Sequence\_ID**.
- **ffill()** per propagare **Class\_Label** sulle righe dati.

```
# Estratto semplificato
mask = split_data[1].isna()
df['Sequence_ID'] = mask.cumsum()
df['Class_Label'] = df['temp'].ffill()
final_df = split_data[~mask].copy()
```



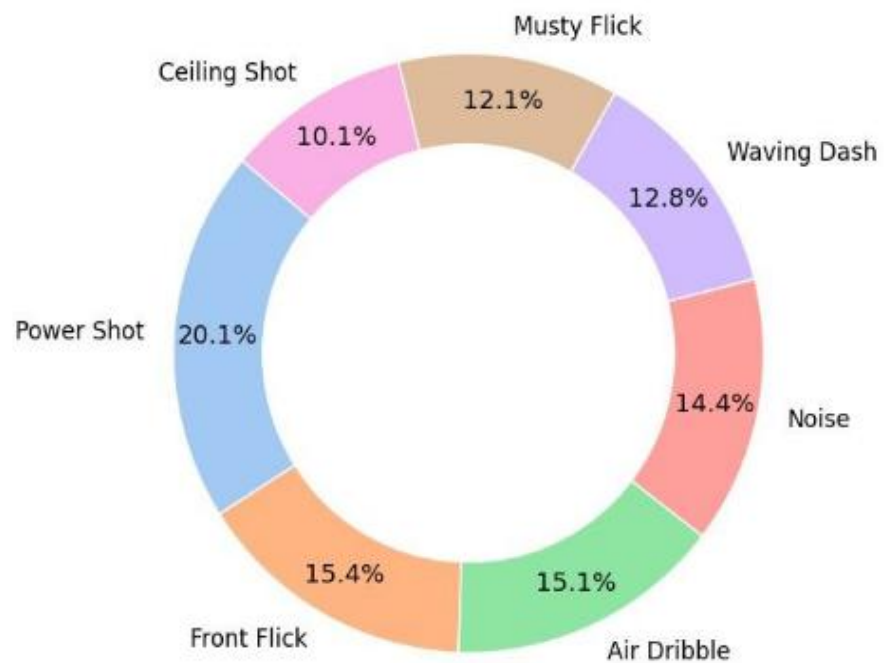
## Vantaggi

- **Efficienza**: parsing vettoriale senza loop lenti.
- **Pulizia**: dataset immediatamente **analizzabile**.
- Risultato: **6,892 frame** consolidati con metadati coerenti.

# EDA: Distribuzione delle Classi

Il dataset presenta uno sbilanciamento tra le classi di manovre.

Distribuzione delle Classi nel Dataset



## Osservazioni e implicazioni

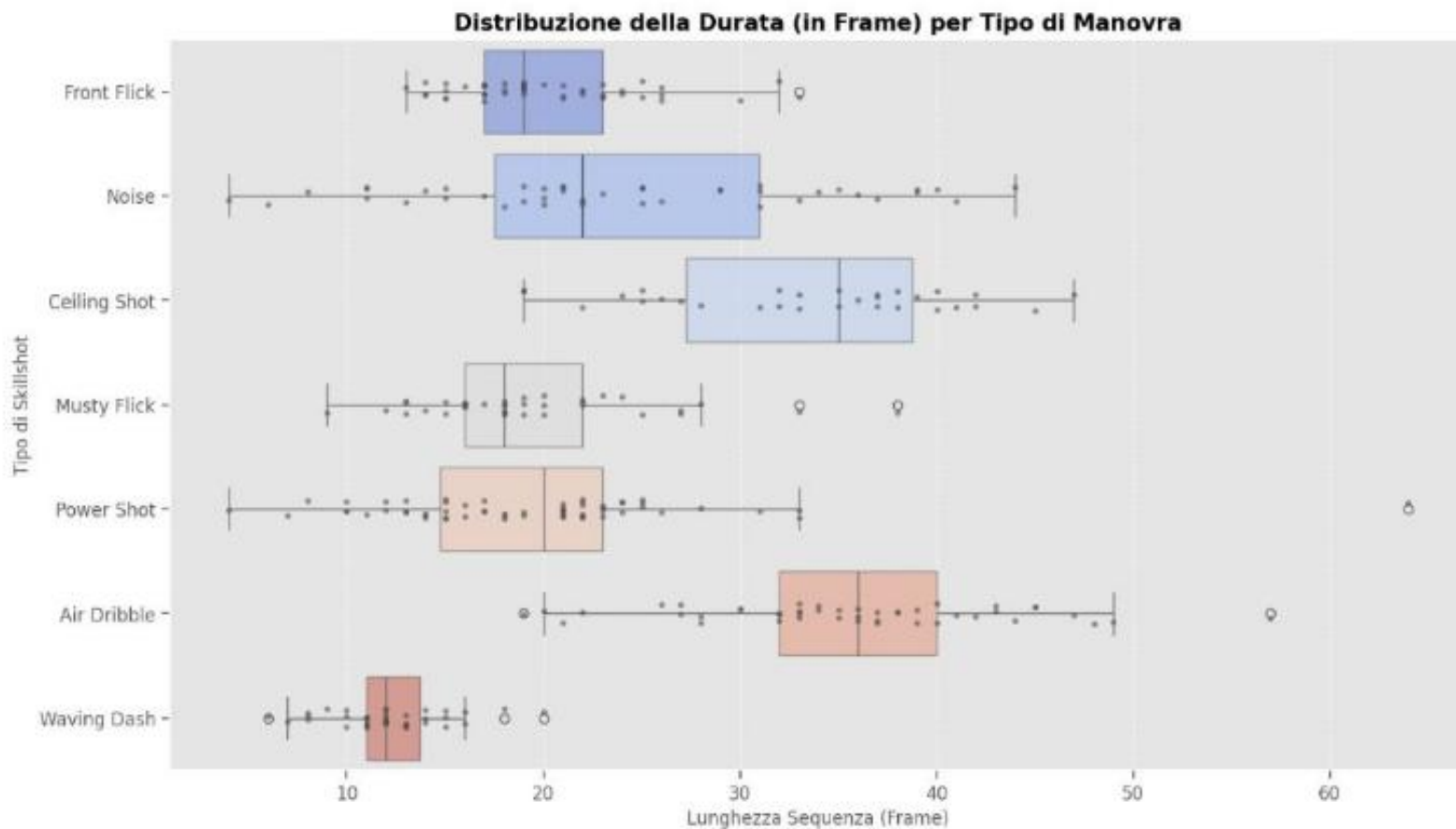
- **Power Shot** sovra-rappresentata.
- **Ceiling Shot** relativamente rara.
- Usare **Stratified CV** in validazione.
- Valutare con **F1-Weighted** oltre all'accuracy.

Impatto: modelli e soglie devono considerare lo squilibrio per evitare bias sulle classi frequenti.



# EDA: Variabilità Temporale (durata per classe)

Box-plot delle durate in frame per ogni manovra: evidenziate Air Dribble (più lunga) e i Flick (più brevi).



## Osservazioni chiave

- **Air Dribble** mostra durate maggiori e variabilità più ampia.
- **Flick** (Front, Musty) risultano **più brevi** e concentrati.
- La diversa scala temporale richiede **feature compatte** (es. quartili) robuste alla lunghezza.

Implicazione: la rappresentazione aggregata per sequenza facilita il confronto tra manovre eterogenee.

# Feature Engineering fisico-comportamentale

- **Quartili (q25, q75):** sostituiscono Min/Max per le variabili fisiche, riducendo l'influenza degli **outlier**.
- **Toggle Count:** per ogni tasto del controller conta i cambi di stato **0 $\leftrightarrow$ 1** per misurare frenesia/spamming.
- **Obiettivo:** catturare sia la **dinamica fisica** sia lo **stile dell'input** (pressione costante vs tapping rapido).
- **Output:** ogni sequenza temporale  $\rightarrow$  **vettore compatto** di aggregati robusti, **invariante alla lunghezza**.



## Quartili per robustezza

Usiamo **q25** e **q75** (oltre alla media) per descrivere variabili fisiche evitando valori estremi.



## Toggle Count (input)

Conta i cambi di stato di un segnale binario.

**Esempio:** 0 1 0 1 1 0  $\rightarrow$  4 toggle

$$\text{count\_toggles}(x) = \sum |\Delta x| > 0$$



## Vettore per sequenza

X\_physics: mean, q25, q75

X\_inputs: mean, toggle\_count

Rappresentazione compatta e interpretabile per la classificazione.



# Modellazione e Validazione

Confronto tra modelli, strategia di validazione e criteri di valutazione



Random Forest



SVM (RBF)



**Protocollo di validazione:** Hold-Out **80/20** (stratificato). Sul training: **Stratified 5-Fold CV**.



**Metrica principale:** **F1-Weighted** per gestire lo sbilanciamento delle classi.



**Ottimizzazione:** **Grid Search** su Random Forest per selezionare gli iperparametri migliori.





**Razionale:** RF è robusto a feature miste e meno sensibile allo scaling; SVM RBF funge da confronto non lineare.



# Cross-Validation: risultati e tuning

Confronto F1-Score medio ( $\pm$  std) su 5-Fold CV stratificata e configurazione ottimale per Random Forest.

Modello	F1-Score medio ( $\pm$ std)	Note
 Random Forest	0.8764 $\pm$ 0.0433	Vincitore
 SVM (RBF)	0.7816 $\pm$ 0.0462	Baseline non lineare di confronto

Validazione su training set con **5-Fold CV stratificata**. Metrica principale: **F1-Weighted**.

## Miglior configurazione RF

 **n\_estimators:** 200

 **criterion:** gini

 **max\_depth:** None

Selezionata tramite **GridSearchCV** sul training set.



# Risultati su Test Set

**86.67%**

Accuratezza  
Prestazioni sul test set

**13.33%**

Baseline casuale  
Confronto di riferimento

**0.86**

F1-Weighted  
Gestisce lo sbilanciamento

**60**

Campioni (test)  
Split Hold-Out 80/20

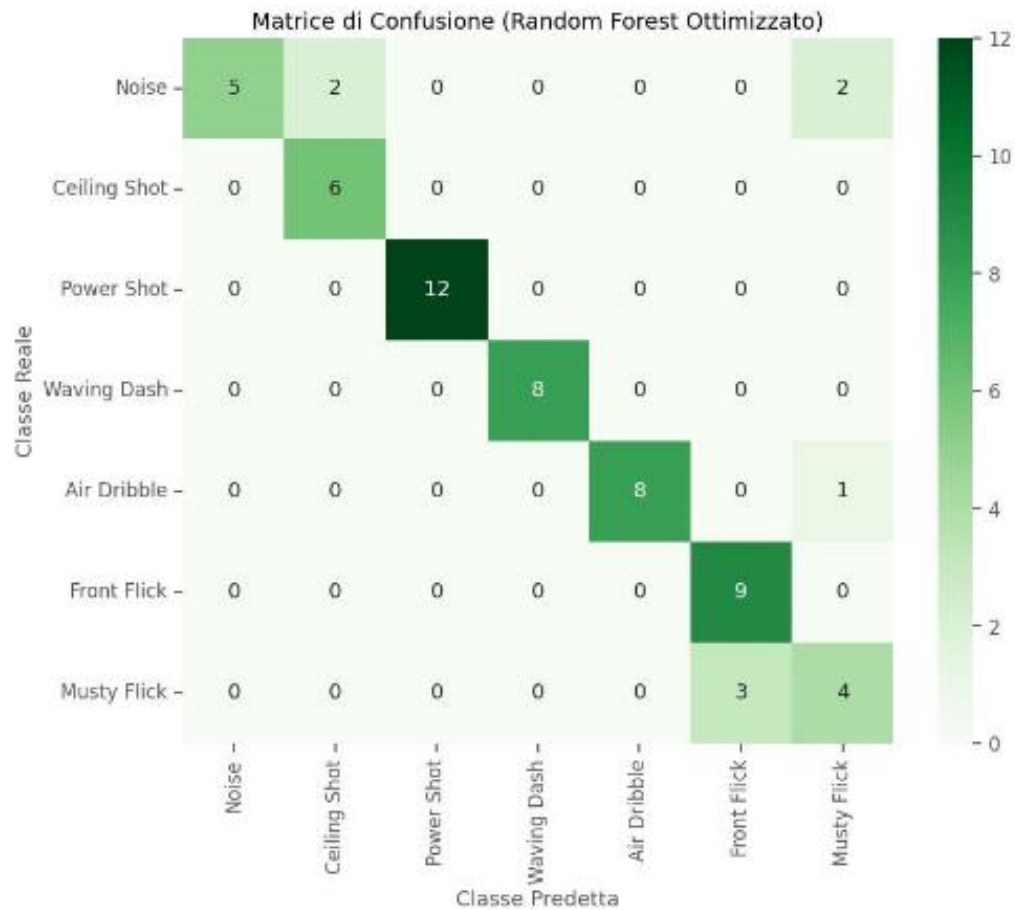
**7**

Classi  
Manovre da distinguere

Evidenze qualitative: **Power Shot** e **Waving Dash** riconosciute con alta precisione; lievi confusioni tra **Musty Flick** e **Front Flick** per similarità cinematica.

# Matrice di Confusione (Test set)

Prestazioni di classificazione per le 7 classi di manovre. Righe = Reale · Colonne = Predetto



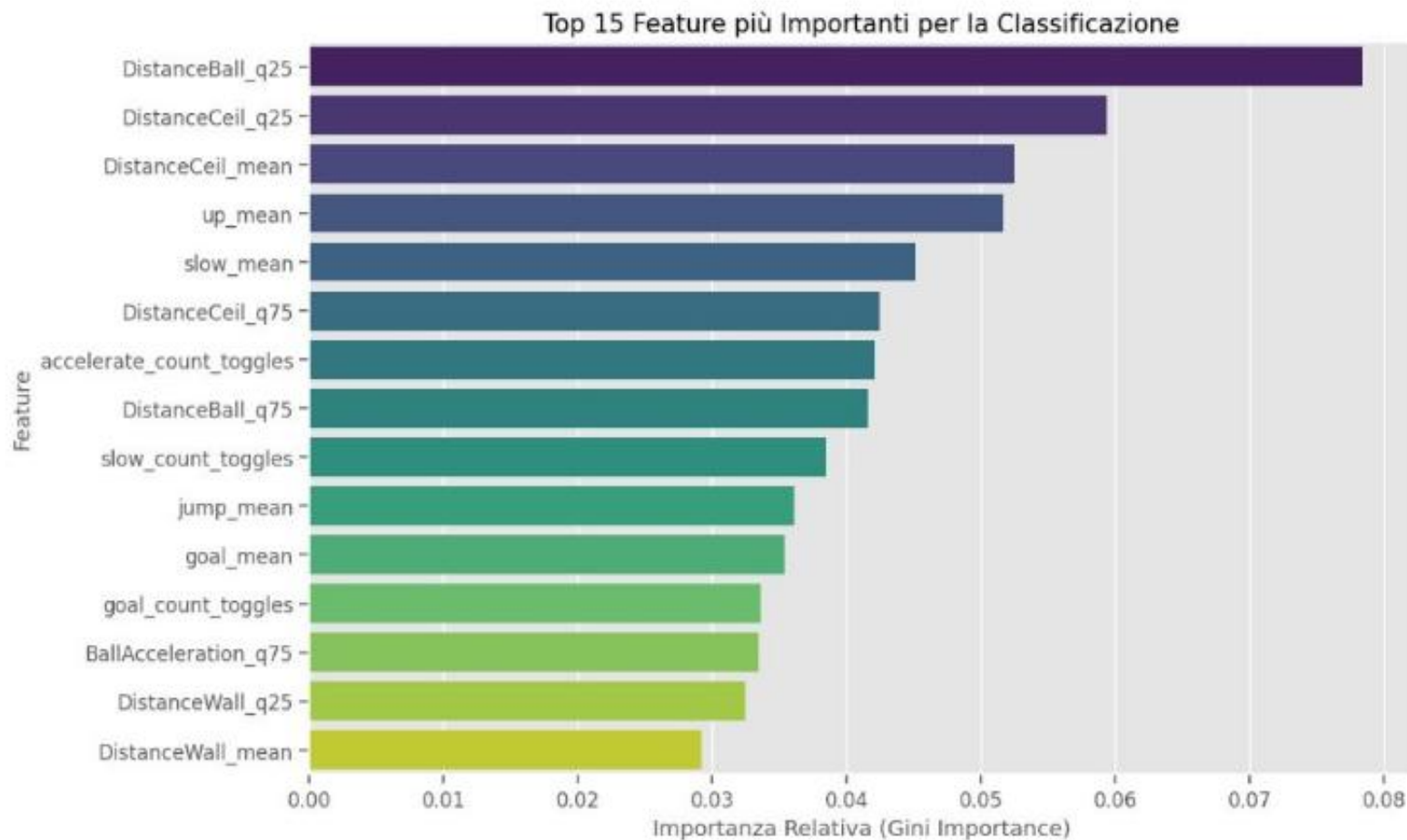
## Evidenze principali

- **Diagonale forte** complessiva.
- **Power Shot** e **Waving Dash** classificati correttamente con alta precisione.
- Lieve scambio tra **Musty Flick** e **Front Flick**.

Letture: massimi sulla diagonale indicano previsioni corrette; i riquadri evidenziati segnalano confusioni specifiche.



# Feature Importance (Gini) · Random Forest



## Lettura e interpretazione

- **Quartili (q25, q75)**: catturano dinamica fisica robusta agli outlier.
- **count\_toggles**: misura la frenesia degli input (spamming vs pressione costante).
- Conferma della **strategia di feature engineering** fisico-comportamentale.

Impatto: le feature evidenziate guidano le decisioni del modello migliorando interpretabilità e performance.

# Conclusioni



## Efficacia del supervised learning

- Classificazione delle **skillshot** da telemetria grezza confermata.
- Prestazioni su test: **Accuratezza 86.67% · F1-Weighted ~ 0.86.**

## Contributi chiave

- **Parser vettoriale** per ingestione rapida e riproducibile.
- **Feature comportamentali**: quartili **q25/q75** e **count\_toggles**.

**Random Forest** superiore a **SVM (RBF)** in cross-validation, confermata la robustezza anche sul test set.

## Note Aggiuntive

- Ampliare e **bilanciare** il dataset.
- Esplorare **modelli sequenziali** (TCN, Transformers).
- Arricchire le feature con ulteriori **segnali di controllo** e fisici.
- Integrare in **strumenti di coaching** e analisi prestazionale.

**F1 ~ 0.86**

**RF > SVM**



# Riferimenti

---

Fonti bibliografiche e dataset utilizzati nel progetto.



# ROCKET LEAGUE

Romain Mathonat, Jean-François Boulicaut, Mehdi Kaytoue-Uberall (2020). Rocket League Skillshots Data Set. UCI Machine Learning Repository.

<https://archive.ics.uci.edu/dataset/858/rocket+league+skillshots>