Progetto Social Media

Claudio Nuncibello, Matr. 1000015294

Analisi delle Carriere dei Giocatori NBA e Classificazione dello Stile di Gioco

Introduzione

Lo studio dei dati dei giocatori NBA rappresenta un'interessante opportunità per analizzare l'evoluzione delle carriere e la specializzazione degli atleti nel contesto di un campionato altamente competitivo come quello americano.

Questo progetto, articolato in due fasi principali, si prefigge di costruire un dataset etichettato che categorizza i giocatori secondo il loro stile di gioco, denominato (*player_type*). L'analisi mira non solo a descrivere statisticamente le caratteristiche dei giocatori, ma anche a offrire spunti sul cambiamento delle tendenze strategiche e delle specializzazioni individuali nel panorama della pallacanestro moderna.

Obiettivi dello Studio

L'obiettivo principale di questo lavoro è identificare e classificare gli stili di gioco di ciascun giocatore, analizzando una vasta gamma di dati raccolti durante tutte le stagioni della loro carriera.

A tal fine, si è sviluppato un processo strutturato che, a partire dalla raccolta e pulizia dei dati, porta all'elaborazione di statistiche avanzate e alla creazione di un dataset finale contenente etichette rappresentative degli stili di gioco. Un elemento centrale di questo lavoro è l'analisi dell'evoluzione delle franchigie NBA, con particolare attenzione a come lo stile di gioco odierno favorisca un approccio collettivo, pur mantenendo un forte focus sulle stelle di ciascuna squadra, le quali continuano a giocare un ruolo cruciale nello sviluppo delle tattiche di gioco.

Oltre alla creazione del dataset, sarà intrapresa una fase di classificazione mediante due modelli di machine learning: Random Forest e Logistic Regression. Questi modelli, alla fine del progetto, permetteranno di classificare nuovi giocatori NBA, attribuendo loro le categorie di gioco più appropriate in base alle caratteristiche individuate nel dataset.

1 Creazione del dataset

Il primo passo dell'analisi ha riguardato il download e la preparazione dei dati. Utilizzando la libreria `nba_api`, sono stati raccolti dati relativi ai giocatori NBA, inclusi gli ID dei giocatori, i loro nomi e gli anni di attività. Dopo una fase di pulizia dei dati, sono stati selezionati solo i giocatori con carriere iniziate dopo la stagione 1980-81 e concluse prima del 2024.

1.1 Raccolta e Preparazione dei Dati

la costruzione di questo dataset ha lo scopo di raccogliere dati relativi alle carriere dei giocatori NBA per identificare il loro stile di gioco (*player_type*) utilizzando un approccio quantitativo e sistematico. I dati raccolti riguardano statistiche di gioco stagionali e metriche avanzate, che sono state elaborate per costruire un dataset adatto all'addestramento di un classificatore supervisionato. La costruzione si basa su due componenti principali:

- La raccolta e la pulizia dei dati grezzi relativi alle carriere dei giocatori NBA, salvati in un file CSV (`player_nba.csv`).

Player_ID	Player_Name	First_Season	Last_Season
203518	Alex Abrines	2016-17	2017-18
1630173	Precious Achiuwa	2020-21	2023-24

Table 1: Esempio di dati contenuti in `player_nba.csv`

- La generazione e aggregazione di statistiche avanzate per ogni giocatore, salvate in un altro file CSV (`nba_stats.csv`).

Infine, attraverso un'accurata definizione delle etichette, il dataset è stato preparato per la classificazione, fornendo informazioni utili per comprendere le tendenze nei ruoli e nelle specializzazioni dei giocatori NBA.

1.2 Calcolo delle Statistiche Avanzate

Per ogni giocatore, sono state recuperate le statistiche di gioco stagionali tramite l'endpoint `playergamelog` della libreria `nba_api`. Queste statistiche sono state poi manipolate per ottenere statistiche avanzate utili alla classificazione:

- Percentuali avanzate, come la percentuale di tiro da tre punti (*FG3_PCT*), da due punti (*FG2_PCT*) e la percentuale di tiro effettiva (*eFG_PCT*).
- Rapporto assist e palle perse(*(AST_TO_RATIO)*).
- Medie per partita per ciascuna metrica.

I dati aggregati sono stati salvati nel file `nba_stats.csv`, pronto per ulteriori analisi.

1.3 Classificazione dello Stile di Gioco

Ogni giocatore è stato classificato in una delle seguenti categorie:

1. **Shooter**: giocatore specializzato nei tiri da tre punti.
2. **Attacker**: giocatore abile nell'attaccare il canestro.
3. **Assist Man**: giocatore focalizzato sulla distribuzione del pallone.
4. **Defender Pro**: giocatore con competenze difensive elevate.
5. **Role Player**: giocatore con abilità in almeno due categorie.
6. **NBA Star**: giocatore eccellente in almeno tre categorie.
7. **Rotation Player**: giocatore che non eccelle in nessuna categoria specifica.

Per usare un approccio supervisionato per il nostro classificatore è stata implementata una funzione che assegna un'etichetta (*player_type*) a ciascun giocatore in base alle sue statistiche. Il dataset finale, contenente anche le etichette, è stato salvato in `nba_data.csv`. è doveroso notare che tale funzione si può identificare essa stessa come metodo di classificazione, al contempo questo espediente è necessario per la realizzazione di un modello di classificazione supervisionato.

2 Discussione dei Risultati

La distribuzione dei giocatori nelle categorie mostra un chiaro sbilanciamento, con una predominanza dei Rotation Player (categoria 7, 1275 giocatori), e una riduzione drastica nel numero di specialisti (1, 3) o giocatori eccellenti in più categorie (6). Questa distribuzione riflette dinamiche importanti legate ai ruoli, alle competenze richieste nel basket e all'evoluzione del gioco.

2.0.1 Numeri a confronto

Classe	Frequenza
NBA Star (6)	209
Role Player (5)	464
Rotation Player (7)	1275
Shooter (1)	95
Attacker (2)	227
Assist Man (3)	30
Defender Pro (4)	239

Table 2: Distribuzione delle Classi di Stile di Gioco

Classe	Giocatori Anni 2000+ (%)	Giocatori Anni '80/'90 (%)
Classe 1 (Shooter)	5.87	0.50
Classe 2 (Attacker)	6.33	12.90
Classe 3 (Assist Man)	1.44	0.79
Classe 4 (Defender Pro)	9.79	8.83
Classe 5 (Role Player)	16.12	21.53
Classe 6 (NBA Star)	7.70	9.03
Classe 7 (Rotation Player)	52.74	46.43

Table 3: Distribuzione Percentuale dei Giocatori per Classe

Le Classi Più Popolate

In entrambe le epoche, le classi più popolate sono "Rotation Player" (Classe 7) e "Role Player" (Classe 5), che riflettono un aspetto cruciale del gioco moderno: la profondità dei roster. Tuttavia, la distribuzione tra le due classi presenta differenze interessanti:

Classe 7 (Rotation Player)

- **Anni 2000+**: 52.74%
- **Anni '80/'90**: 46.43%

La percentuale di giocatori classificati come "Rotation Player" è aumentata nel periodo 2000+, indicando una crescente enfasi sulla rotazione dei giocatori e sull'importanza di mantenere alti livelli di intensità durante l'intero arco della partita. Questo cambiamento riflette il passaggio a un gioco più veloce, basato sulla gestione della fatica e sullo sfruttamento delle risorse in profondità.

Classe 5 (Role Player)

- **Anni 2000+**: 16.12%
- **Anni '80/'90**: 21.53%

Sebbene il numero di "Role Player" sia diminuito nei giocatori degli anni 2000+, questo continua a essere un gruppo fondamentale, seppur con una riduzione della specializzazione rispetto al passato. Gli anni '80/'90 vedevano più giocatori concentrati su ruoli specifici, mentre oggi c'è una tendenza a preferire giocatori più versatili, capaci di adattarsi a più situazioni.

Le Classi che Hanno Subito i Maggiori Cambiamenti

Classe 1 (Shooter)

- **Anni 2000+**: 5.87%
- **Anni '80/'90**: 0.50%

Il ruolo degli "Shooter" ha visto un aumento significativo, passando dallo 0.50% degli anni '80/'90 al 5.87% negli anni 2000+. Questo cambiamento è direttamente correlato all'evoluzione del tiro da tre punti come parte integrante del gioco moderno. Giocatori come Stephen Curry hanno dimostrato che il tiro da lunga distanza può dominare l'attacco, influenzando l'intero gioco e ridisegnando la posizione di "Shooter".

Classe 2 (Attacker)

- **Anni 2000+**: 6.33%
- **Anni '80/'90**: 12.90%

In contrasto con l'aumento degli "Shooter", la percentuale degli "Attacker" è diminuita. Nel periodo '80/'90, l'attacco fisico e il gioco uno-contro-uno erano prevalenti, ma oggi è emersa una maggiore enfasi sul gioco di squadra, sulle transizioni veloci e sulla condivisione del pallone. Il calo di questa classe è una riflessione del cambiamento nelle modalità di attacco, passando da un gioco di isolamento a un gioco più collettivo.

Le Classi che Hanno Visto Pochi Cambiamenti

Classe 4 (Defender Pro)

- **Anni 2000+**: 9.79%
- **Anni '80/'90**: 8.83%

Il ruolo del "Defender Pro" ha visto un lieve aumento, ma fondamentalmente la sua importanza è rimasta costante. Sebbene l'evoluzione del gioco moderno richieda difensori più versatili e in grado di difendere su più posizioni, la centralità del difensore come specialista nella protezione del pitturato non è cambiata radicalmente. I difensori continuano a giocare un ruolo cruciale, sebbene in contesti diversi rispetto al passato.

Classe 6 (NBA Star)

- **Anni 2000+**: 7.70%
- **Anni '80/'90**: 9.03%

La classe "NBA Star", che include i giocatori più talentuosi e influenti, ha visto una lieve diminuzione. Passando dal 9.03% degli anni '80/'90 al 7.70% degli anni 2000+, questo suggerisce un cambiamento nel modo in cui le squadre costruiscono i loro roster. Negli anni '80/'90, le squadre erano più inclini a costruire attorno a una o due superstar, mentre oggi la maggior parte delle squadre si concentra su un gioco collettivo, utilizzando una varietà di giocatori con ruoli specifici per sfruttare il massimo potenziale del roster. Tuttavia, i "NBA Stars" continuano ad avere un impatto significativo sul gioco, sebbene non siano più l'unica fonte di successo per le squadre.

Considerazioni

In generale, l'evoluzione del gioco NBA negli anni 2000+ ha portato a un cambiamento nelle classi di giocatori, con un'enfasi maggiore sulla versatilità e sulla profondità dei roster. Le classi "Rotation Player" e "Role Player" sono aumentate in termini di numero e percentuale, mentre le classi specialistiche come "Shooter" e "Attacker" hanno visto significativi spostamenti. L'importanza della figura dell'"NBA Star" ha leggermente diminuito, in linea con l'evoluzione verso un gioco più collettivo e distribuito. L'introduzione del tiro da tre punti come strategia dominante ha favorito l'ascesa della classe "Shooter", mentre il focus sul gioco di squadra ha ridotto la presenza degli "Attacker".

L'Utilizzo di un Solo Dataset per la Classificazione

Visto l'analisi delle differenze tra i giocatori degli anni '80/'90 e quelli degli anni 2000+, ci si può chiedere se abbia senso creare un classificatore che racchiuda tutti gli anni senza separare le due epoche.

L'uso di un solo dataset per classificare i giocatori NBA ha un forte senso, soprattutto quando si considera che, nonostante l'evoluzione nello stile di gioco, i giocatori sono stati sempre valutati con le stesse metriche fondamentali. Le statistiche tradizionali come punti, rimbalzi, assist e minuti giocati, insieme a parametri più moderni come il tiro da tre punti e il plus-minus, sono applicabili in modo coerente attraverso le epoche. Sebbene la distribuzione delle classi possa essere cambiata a causa dell'evoluzione del gioco, le caratteristiche che definiscono ogni classe di giocatori (ad esempio, un "shooter" o un "NBA star") non sono completamente mutati; ciò che è cambiato è la frequenza di ciascun tipo di giocatore, non la loro definizione di base.

Infine, un classificatore basato su un unico dataset è più efficiente e pratico da mantenere. Gestire e aggiornare periodicamente più modelli separati per diverse epoche sarebbe più oneroso e complesso, mentre un unico modello fornisce una visione unificata, facilitando l'analisi e l'interpretazione dei dati. In questo modo, si possono tracciare più facilmente tendenze e proiezioni future senza dover gestire diverse architetture di modello per periodi distinti.

3 Classificazione

3.1 Analisi e implementazione

Nel contesto di questo progetto, la classificazione dei giocatori NBA è stata realizzata utilizzando un approccio di machine learning, in cui si è cercato di predire il tipo di giocatore (*player_type*) a partire dalle caratteristiche individuali dei giocatori. In questo capitolo, vengono discusse le scelte metodologiche e il codice utilizzato per costruire, ottimizzare e valutare i modelli di classificazione.

3.2 Caricamento e Pre-processing

Il dataset, caricato da un file CSV, è stato pre-processato per rimuovere colonne non necessarie come , *PLAYER_ID, PLAYER_NAME e SEASON_YEAR* che non contenevano informazioni utili per la predizione. Successivamente, sono stati eliminati i valori mancanti, garantendo un dataset completo per l'analisi.

A questo punto, i dati sono stati separati in variabili indipendenti (features) e variabile dipendente (target), quest'ultima rappresentata dalla colonna *player_type*, che indica il tipo di giocatore.

3.3 Bilanciamento delle Classi

Il dataset originale presenta uno sbilanciamento significativo tra le diverse classi del target. Alcune classi sono molto più rappresentate rispetto ad altre, il che potrebbe compromettere la capacità del modello di generalizzare correttamente. Tuttavia, è importante sottolineare che, pur essendo presente uno sbilanciamento, questo riflette un aspetto fondamentale dello studio e delle dinamiche reali delle categorie di giocatori. Per questa ragione, l'obiettivo non era quello di bilanciare completamente le classi, ma piuttosto di evitare che il modello fosse eccessivamente influenzato dalla prevalenza delle classi maggioritarie.

In tal senso, è stato utilizzato SMOTE (Synthetic Minority Over-sampling Technique) per generare nuovi esempi sintetici per le classi minoritarie, ma senza alterare in modo significativo la distribuzione complessiva del dataset. Questo approccio ha consentito di migliorare le performance del modello nelle classi meno rappresentate, mantenendo al contempo l'importanza dello sbilanciamento originale, che rispecchia la realtà del fenomeno studiato. Pertanto, il bilanciamento ottenuto tramite SMOTE non ha modificato la struttura fondamentale del dataset, ma ha solo cercato di migliorare la capacità del modello di riconoscere pattern nelle classi minoritarie, senza snaturare il contesto complessivo delle classi sbilanciate.

3.4 Creazione dei Modelli e Addestramento

Per la creazione e l'addestramento dei modelli, sono stati utilizzati due algoritmi di machine learning: Random Forest e Logistic Regression. Entrambi i modelli sono stati ottimizzati attraverso la ricerca dei migliori iperparametri per massimizzare le loro performance.

Per il modello di Random Forest, il miglior risultato è stato ottenuto utilizzando i seguenti parametri: `max_depth = 20` e `n_estimators = 100`. I risultati del modello sono molto promettenti, con una precisione globale dell'89% e una *accuracy* complessiva pari al 92%. Tuttavia, il modello ha mostrato performance variabili tra le diverse classi. Le classi con un numero maggiore di campioni, come la classe 7, hanno avuto una precisione e un recall molto elevati (entrambi superiori al 98%), mentre le classi meno rappresentate, come la classe 3, hanno registrato risultati più bassi, con una precisione di 0.60 e un recall di 0.67. In generale, Random Forest ha mostrato una buona capacità di distinguere tra le classi maggioritarie e minoritarie, mantenendo un buon equilibrio tra precisione e recall, ma con performance non ottimali nelle classi più rare.

Per quanto riguarda la Logistic Regression, il miglior modello è stato ottenuto con `C = 10` e `solver = 'liblinear'`. Sebbene la precisione globale del modello sia inferiore rispetto a Random Forest, con una *accuracy* di 72%, la Logistic Regression ha dimostrato una maggiore capacità di riconoscere le classi minoritarie. Ad esempio, per la classe 3, che ha una bassa rappresentanza, il recall è stato molto alto (0.89), ma la precisione è risultata bassa (0.30). Le classi maggioritarie, come la classe 7, hanno visto un buon equilibrio tra precisione (0.96) e recall (0.82), con un f1-score complessivo di 0.88.

Nel complesso, la Random Forest ha mostrato una maggiore efficacia in termini di

accuracy complessiva e una capacità migliore di generalizzare sui dati maggioritari. Tuttavia, la Logistic Regression ha evidenziato una maggiore capacità di identificare correttamente le classi minoritarie, con un recall più elevato nelle classi meno rappresentate. La scelta tra i due modelli dipende quindi dal tipo di priorità: se l'obiettivo è ottimizzare l'*accuracy* complessiva, Random Forest risulta preferibile, mentre se l'interesse è focalizzato sull'identificazione accurata delle classi minoritarie, la Logistic Regression potrebbe offrire risultati migliori.

3.5 Analisi delle Curve ROC

La curva ROC (Receiver Operating Characteristic) è un grafico che rappresenta la performance di un classificatore binario o multiclasse, valutando il compromesso tra il tasso di veri positivi (True Positive Rate - TPR) e il tasso di falsi positivi (False Positive Rate - FPR). L'area sotto la curva (AUC - Area Under the Curve) fornisce un'indicazione sintetica della qualità del modello: più l'AUC è vicino a 1, migliore è il modello.

3.5.1 1. Modello Random Forest

- Le curve ROC per tutte le classi sono molto vicine all'angolo in alto a sinistra, indicando un'eccellente capacità predittiva.
- I valori di AUC sono quasi perfetti, con molte classi che raggiungono AUC = 1.00. La classe 6 ha un AUC leggermente inferiore ma comunque molto alto (AUC = 0.99).
- Questo dimostra che il modello Random Forest è estremamente efficace nel distinguere le diverse classi, con prestazioni quasi ottimali.

3.5.2 2. Modello Logistic Regression

- Le curve ROC sono meno aderenti all'angolo in alto a sinistra rispetto al modello Random Forest, con alcune classi che mostrano un compromesso maggiore tra TPR e FPR.
- Gli AUC variano tra 0.82 (classe 5) e 0.98 (classe 3), indicando che alcune classi sono gestite molto bene (ad esempio, la classe 3), mentre altre (ad esempio, la classe 5) mostrano margini di miglioramento.
- Nonostante ciò, il modello Logistic Regression mantiene un buon livello di accuratezza generale, ma inferiore rispetto al Random Forest.

3.5.3 Conclusioni

- Il Random Forest è chiaramente superiore in termini di capacità predittiva rispetto alla Logistic Regression, con AUC quasi perfetti e una separabilità delle classi molto alta.
- La Logistic Regression, pur mostrando buoni risultati, evidenzia una minore efficacia, in particolare per alcune classi, come la classe 5 (AUC = 0.82).

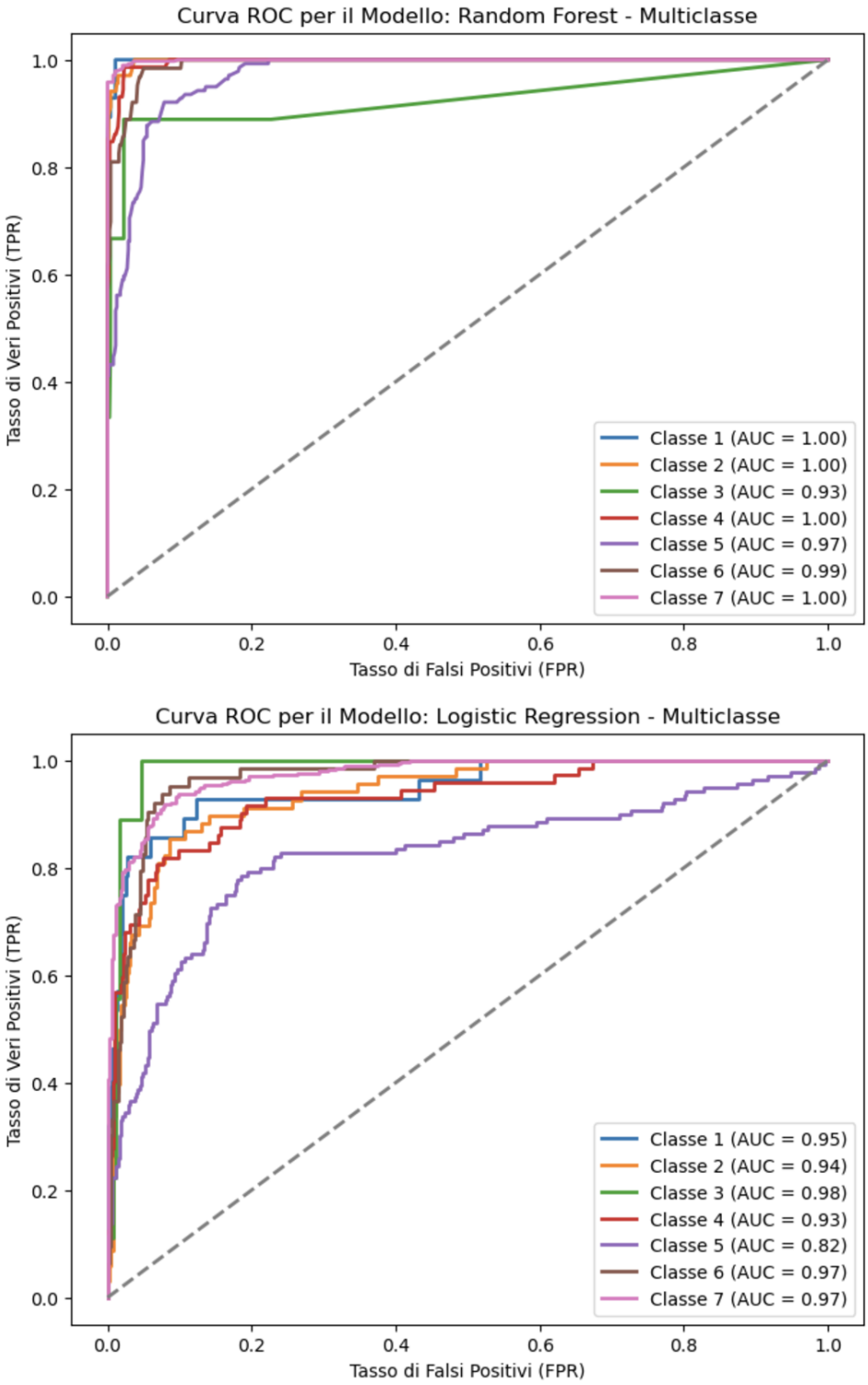

Figure 1: Curve ROC

In sintesi, il modello Random Forest è preferibile per questo problema di classificazione multiclasse, grazie alla sua capacità di modellare relazioni complesse nei dati e garantire una distinzione accurata tra le classi.

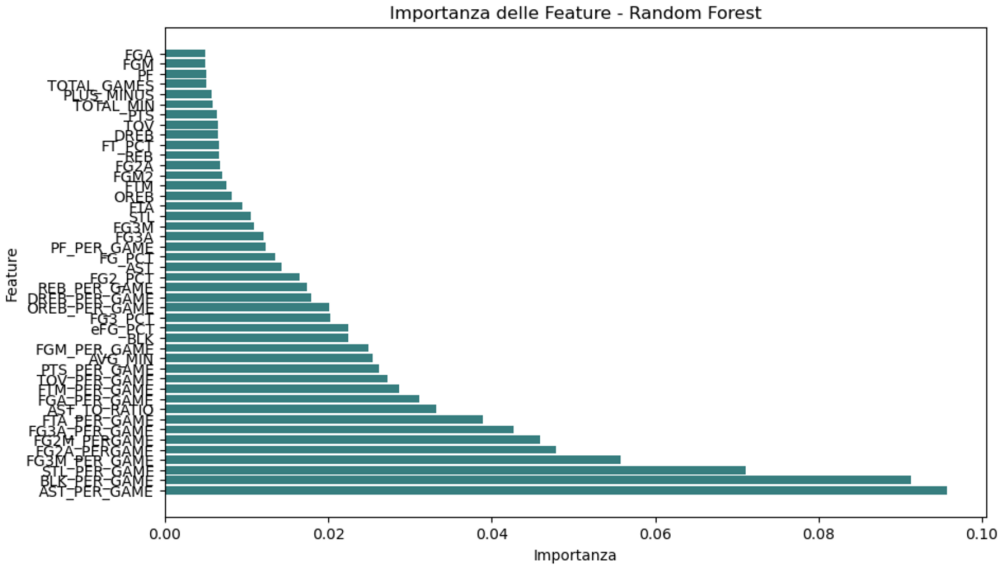

Figure 2: Importanza delle Feature

3.6 Analisi dell'Importanza delle Feature

Il modello Random Forest assegna maggiore peso alle variabili che spiegano meglio la variabilità del target, come `PTS_PER_GAME`, `AST_PER_GAME` e `BLK_PER_GAME`. Queste statistiche, legate alla performance diretta e al contributo complessivo di un giocatore in partita, risultano fondamentali. D'altra parte, le variabili con valori aggregati (come il totale delle partite giocate) o meno rappresentative hanno un impatto minimo.

Questo tipo di analisi è utile per:

- Identificare quali feature sono più importanti per il modello.
- Comprendere quali aspetti dei dati influenzano maggiormente le predizioni, fornendo insight utili per applicazioni pratiche.

3.7 Analisi della Heatmap della Correlazione

La heatmap rappresenta la matrice di correlazione delle feature, mostrando le relazioni lineari tra le variabili tramite una scala cromatica. I valori di correlazione possono variare tra -1 (correlazione negativa perfetta) e 1 (correlazione positiva perfetta), mentre valori vicini a 0 indicano assenza di correlazione.

La matrice evidenzia che alcune variabili sono altamente correlate tra loro, come `FGM_PER_GAME` con `PTS_PER_GAME` e `REB` con `DREB_PER_GAME`.

Questa analisi aiuta a comprendere le relazioni tra le variabili e fornisce spunti per migliorare il pre-processing dei dati e l'efficienza del modello.

4 Conclusioni

In sintesi, il codice sviluppato ha fornito un'implementazione completa di una pipeline di classificazione, che include la preparazione dei dati, il bilanciamento delle classi, la

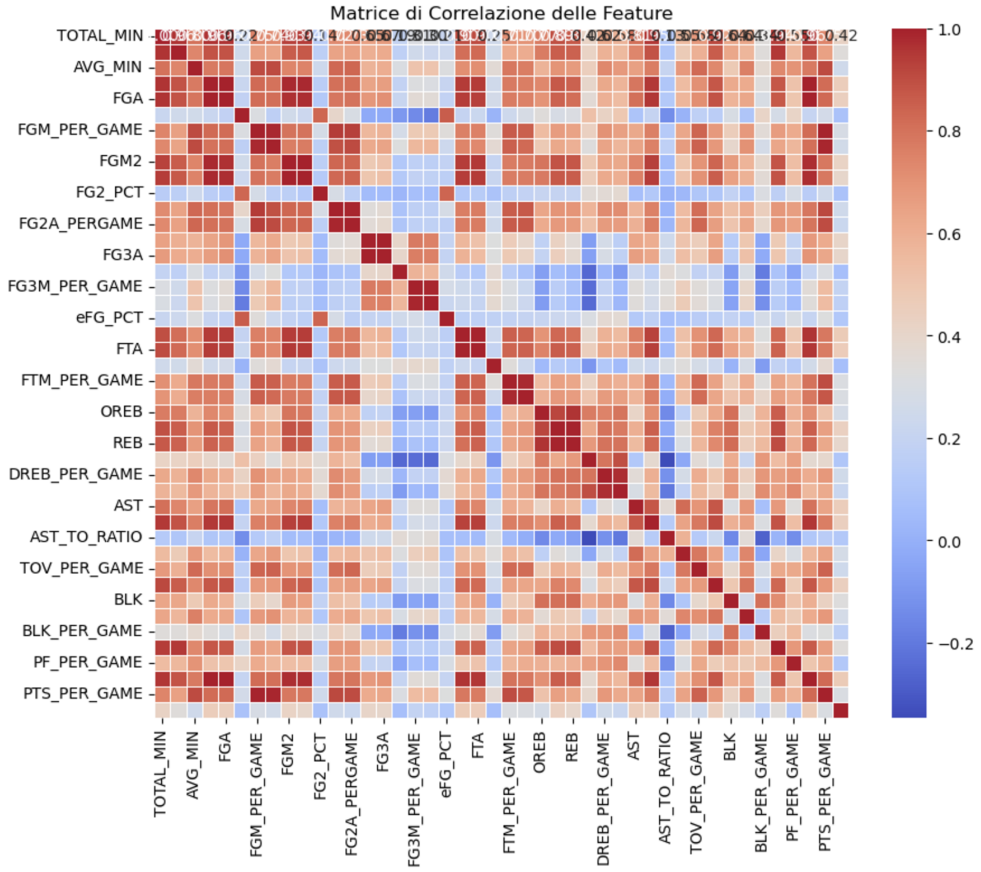

Figure 3: Heatmap della Correlazione

costruzione e l'ottimizzazione dei modelli, nonché una serie di analisi visive per interpretare e migliorare le performance. L'uso di SMOTE per il bilanciamento, combinato con la valutazione accurata delle metriche e delle curve ROC, ha permesso di ottenere modelli pronti per classificare nuovi giocatori.