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 aspetto chiave del lavoro consiste nell'evidenziare come i giocatori NBA moderni siano diventati sempre meno specializzati in singoli ruoli e più versatili nel loro contributo al gioco, una tendenza chiaramente riscontrabile nella distribuzione delle classi etichettate.

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_train.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.

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

2.1 La predominanza della categoria 7 *(Rotation Player)*

Con 1275 giocatori, la categoria 7 rappresenta oltre la metà del totale. Questo indica che la maggior parte dei giocatori non eccelle in alcuna area specifica ma possiede competenze sufficienti per essere utile alla squadra. Le ragioni di questa prevalenza possono essere:

1. **Economia della squadra**: I giocatoi in questa categoria sono giocatori marginali, che trovano il loro posto in squadra prediligendo il gioco di squadra, favorendo la realizzazione tramite schemi e affidandosi ai loro compagni per portare a termine un azione.

2. **Richiesta di giocatori flessibili**: La strategia moderna consiste nel costruire la squadra attorno alle proprie stelle; per questo motivo, i giocatori identificati come *Rotation Player* sono utilizzati specialmente per completare i roster.

3. **Durata delle carriere**: I *Rotation Player* tendono ad avere carriere più lunghe rispetto a specialisti o stelle. Inoltre, i roster delle squadre sono ampi e non tutti i giocatori risultano indispensabili alle dinamiche di gioco, ma possono essere sfruttati per aumentare il ritmo di allenamento o per garantire copertura in caso di infortuni. Questo aumenta il numero totale di giocatori che hanno militato in questa categoria dagli anni '80 a oggi.

2.2 Il ruolo delle stelle: NBA Star (6)

Con solo 209 giocatori nella categoria 6, è evidente che i giocatori eccellenti in almeno tre categorie siano una minoranza esclusiva:

1. **Rarità del talento**: Essere una stella NBA richiede un livello eccezionale di competenze atletiche, tecniche e tattiche, che solo pochi giocatori possono raggiungere.

2. **Pressione e aspettative**: Le stelle NBA, nella maggior parte dei casi, non riescono a mantenere la stessa intensità di gioco per tutta la carriera, a causa dell'usura fisica e delle elevate aspettative. Questo fattore influisce sulle loro statistiche complessive, riducendo il numero totale di giocatori che rientrano in questa categoria.

2.3 Le categorie intermedie

1. **Role Player**: I giocatori che rientrano in questa categoria potrebbero essere considerati i "collanti" della squadra. Analizzando questi atleti, si possono osservare caratteristiche simili a quelle delle stelle NBA, ma con una differenza fondamentale: il loro gioco è più verticale, poiché non sono i protagonisti principali. Probabilmente, i giocatori di questa classe rappresentano delle future stelle o, in alternativa, sono atleti il cui ruolo tattico li porta a eccellere solo in determinate aree del gioco.

2. **Defender Pro**: I giocatori specializzati nella difesa richiedono un alto livello di intensità e competenza, che spesso non è valorizzata a sufficienza rispetto alle abilità offensive, spiegando il numero inferiore.

3. **Attacker**: La categoria 2 rappresenta un numero moderato di atleti che si specializzano nell'attacco al canestro. Questo è coerente con la natura del gioco, che richiede a molti giocatori di contribuire in questa area, ma non al punto di raggiungere livelli di eccellenza.

2.4 Categorie meno rappresentate

1. **Numero limitato di Shooter (1) e Assist Man (3)**: La categoria 1 (95 giocatori) e la categoria 3 (30 giocatori) rappresentano il numero più basso, riflettendo una minoranza di specialisti puri. Questo si spiega con:

   (a) **Difficoltà nell'eccellere**: Per essere classificati come Shooter o Assist Man, i giocatori devono eccellere in modo straordinario in un ambito specifico. Nel gioco moderno, come discusso in precedenza, l'importanza del gioco è affidata principalmente alle stelle, che spesso dominano queste statistiche elevate. Tuttavia, quando eccellono in tutte le aree del gioco, vengono identificati come STAR.

   (b) **Influenza del sistema di gioco**: L'NBA moderna è caratterizzata da attacchi più fluidi e da movimenti di palla più dinamici, che premiano, in alcuni casi, la tattica di squadra e, in altri, l'isolamento di giocatori chiave. Questo riduce la necessità di specialisti estremi.

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.

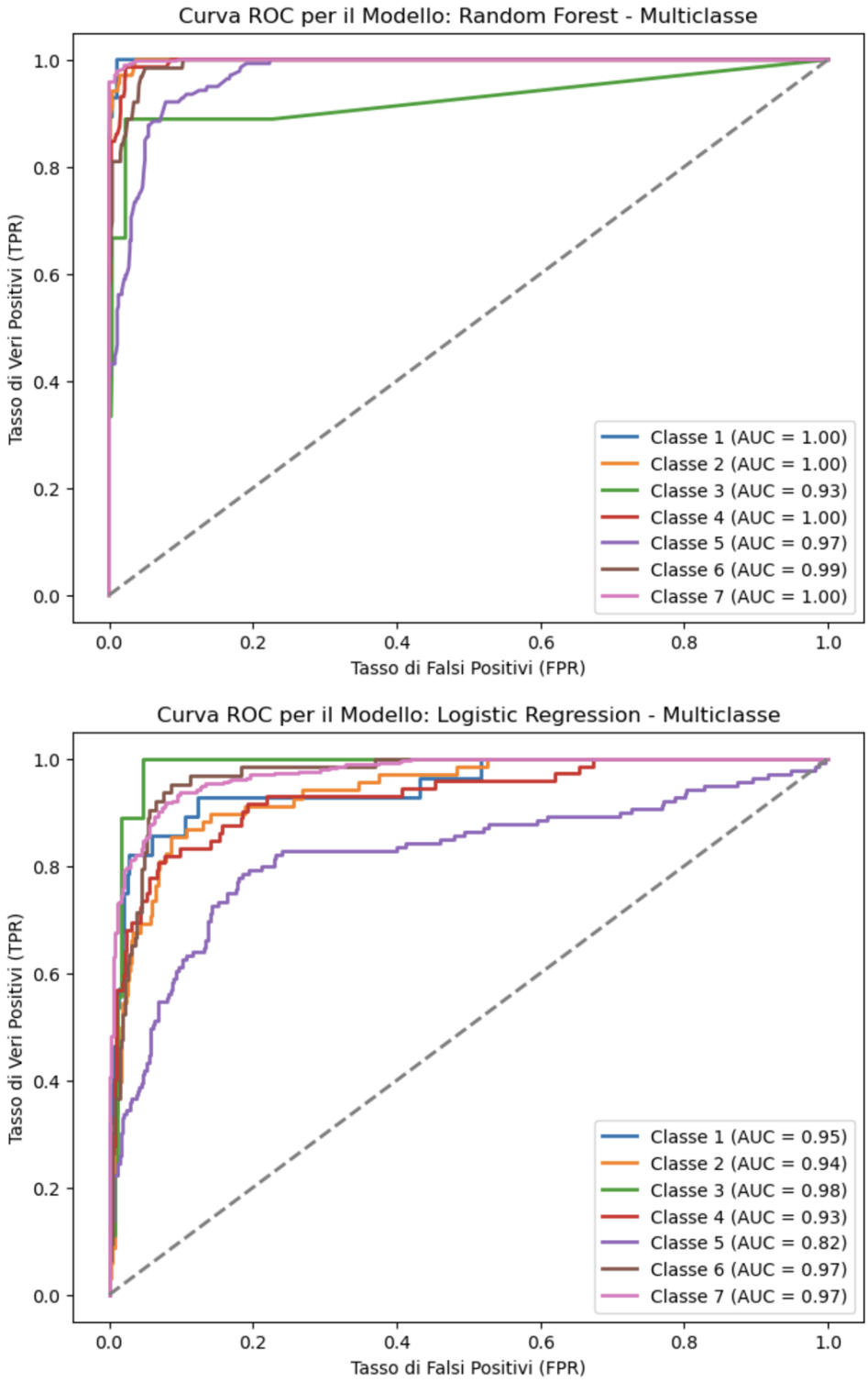

Figure 1: Curve ROC

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).

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.

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.

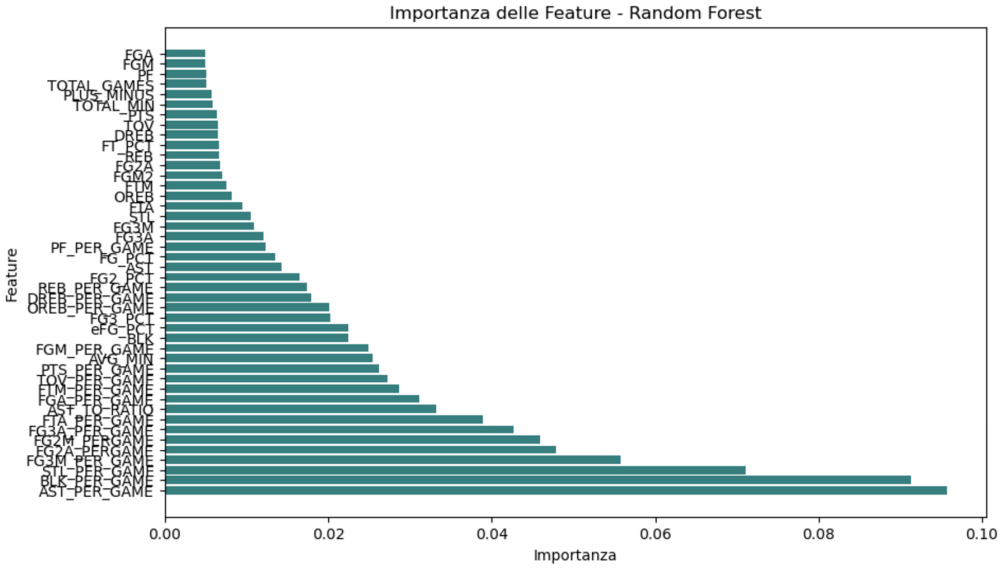

Figure 2: Importanza delle Feature

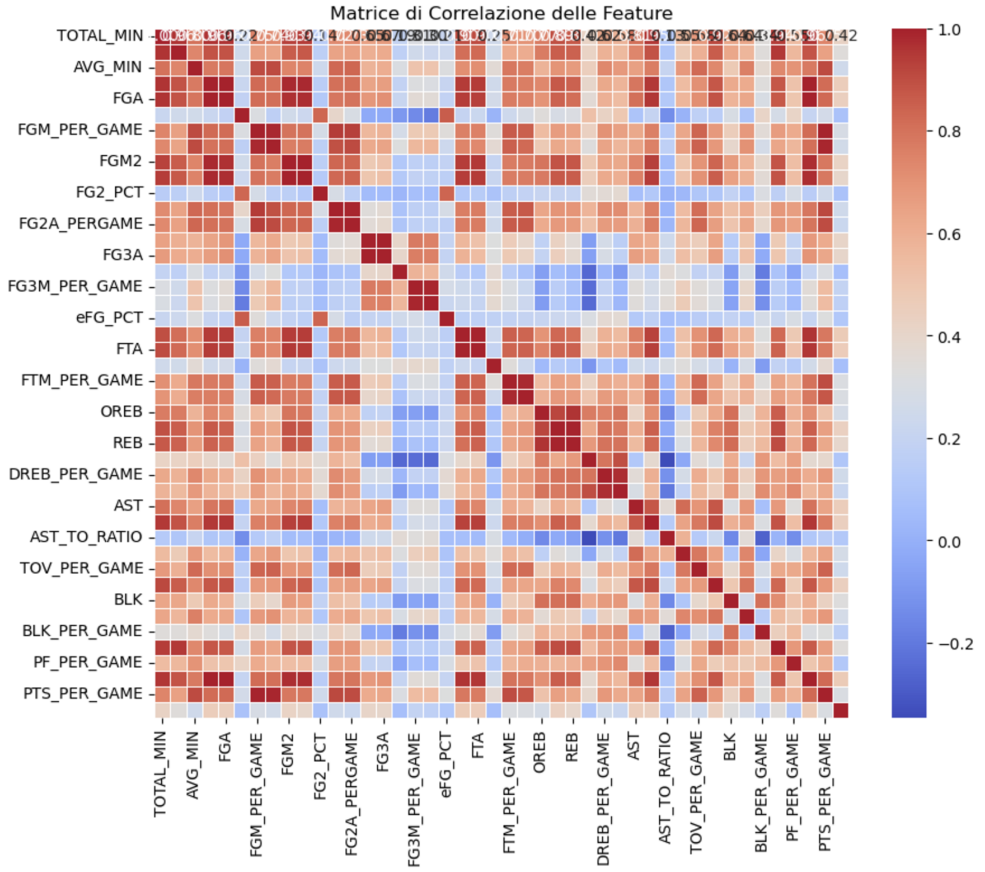

Figure 3: Heatmap della Correlazione

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 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.