POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Matematica

Tesi di Laurea Magistrale

Machine Learning e Inferenza Bayesiana Gerarchica per l'analisi dei portieri d'élite.

Relatori:
Prof. Gianluca MASTRANTONIO
Dott. Marco GABRIELLI

Candidato:
Matteo SANTANGELO

Anno accademico 2025-2026

Sommario

In questo elaborato vengono implementati dei modelli di classificazione, volti ad estrarre le caratteristiche che differenziano un portiere di livello élite. Inoltre, viene implementato un modello gerarchico bayesiano, per ottenere la curva prestazionale di ogni portiere e creare un ranking basato sui risultati ottenuti.
Nel capitolo 2 si illustrano i dataset utilizzati e l'analisi preliminare svolta su ognuno di essi.
Il capitolo 3 racconta il processo di addestramento del modello di classificazione e di estrazione delle features utili allo scopo dell'elaborato.
Nel capitolo 4 si costruisce il modello bayesiano gerarchico, completando l'elaborato con l'esposizione dei ranking ottenuti.

Ringraziamenti

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Per decenni, il calcio è stato considerato uno sport refrattario all'analisi numerica. Il calcio è un gioco fluido, a punteggio basso e caratterizzato da un'elevata componente di casualità. Tuttavia, l'ultimo decennio ha segnato una rivoluzione. Quello che una volta era affidato esclusivamente all'occhio esperto dell'osservatore o all'intuito dell'allenatore, oggi viene integrato da una mole di dati senza precedenti. Ogni tocco di palla, ogni posizionamento difensivo e ogni frazione di secondo di una partita vengono oggi tracciati, trasformando il campo da gioco in un immenso laboratorio di generazione dati. Questi dati, integrati con conoscenze statistiche possono essere un ottimo strumento di aiuto alle decisioni. In questo elaboarato se ne esplorano due possibili modi di utilizzo.
La prima parte si concentra su un modello di classificazione, capace di identificare i portieri élite e di estrarne le caratteristiche che li rendono elitari. Questo modello potrebbe tornare utile sia in fase di scouting che in fase di formazione dei ragazzi.
Nella seconda parte, si implementa un modello bayesiano, quindi non basato sulla classica statistica frequentista che meno si addice ad uno sport come il calcio, per provare a tracciare la carriera di ogni portiere negli anni a disposizione. Ottenute queste curve prestazionali, verrà definita una classifica dei migliori portieri secondo i dati.
La scelta di effettuare questa analisi sul ruolo del portiere non è casuale. Il calcio negli ultimi anni, non ha solo introdotto i dati, ma ha lasciato spazio ad un nuovo modo di interpretare lo sport stesso, e questo ha dato vita ad un'evoluzione del ruolo del portiere. Fino a non molti anni fa, un portiere bravo nella distribuzione della palla con i piedi era un'eccezione. Al giorno d'oggi, invece, i portieri devono anche saper riconoscere situazioni di gioco, individuare l'uomo libero e effettuare passaggi precisi anche sotto pressione.

Capitolo 2

Datasets

In questo capitolo, vengono esposte le caratteristiche dei datasets utilizzati. I dati in questione sono stati forniti da uno dei noti provider di statistiche sportive.

2.1 Player Season stats

Il dataset Player Season Stats contiene le statistiche di tutti i calciatori dei top 5 campionati europei (Serie A, Premier League, La Liga, Ligue 1, Bundesligsa), delle rispettive coppe di lega e delle 3 competizioni europee per club (Champions League, Europa Leauge e Conference League). I dati fanno riferimento al periodo che va dalla stagione calcistica 2019/2020 all'inizio di quella corrente e sono aggregati per stagione, quindi per ogni riga si hanno le statistiche medie ottenute per ogni stagione da un singolo calciatore. Dato che l'elaborato si concentra sul ruolo del portiere, sono stati eliminati dal dataset tutti i dati che non facevano riferimento a questo. Eliminati questi dati, si è ottenuto un dataset con circa 4500 osservazioni e 98 statistiche. Statistiche che sono di 3 tipi: Dati di conteggio, rapporti e metriche di valutazione della prestazione. Per completare il dataset e renderlo congruo alla classificazione, è stata aggiunta una variabile binaria, che sarà la variabile target del modello. Questa variabile ha valore 1, se un portiere è considerato di livello élite, 0 altrimenti. Sono stati definiti come elitari, i portieri che hanno partecipato, almeno una volta nelle diverse stagioni presenti nel dataset, agli ottavi di Champions League o che sono stati nominati nella top 10 della classifica del premio Yashin. Tutte le altre statistiche, rappresentano il set di features a disposizone per l'addestramento del modello di classficazione.

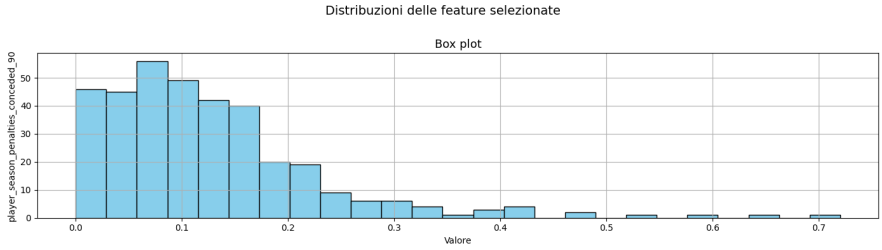

Figura 2.1: Rigori concessi per partita

2.1.1 Analisi preliminare

Analisi delle distribuzioni

Analizzando il dataset in questione e guardando in particolare le distribuzioni dei dati contenuti in esso, si sono riscontrati tre diverse "tipologie" di statistiche. Le diverse tipologie nonostante derivino da una distribuzione a campana classica, presentano alcune differenze. Queste differenze nascono quando si hanno statistiche come i gol concessi o i rigori concessi (Figura 2.1), che sono valori discreti che non hanno un'alta frequenza, quindi si ha una mezza campana con il picco sullo 0. Un altro fattore che influenza le distribuzioni è la grande presenza di valori nulli, in quanto, i portieri che non giocano sono molti di più rispetto a quelli che giocano regolarmente. Questo fattore influenza soprattutto statistiche come i rapporti (Figura 2.2), in cui si ha un'alta frequenza di zeri (Casi in cui il portiere non gioca mai) e di 1 (Casi in cui il portiere gioca poco e ottiene statistiche praticamente perfette). In questi casi si ottengono due picchi esterni con una distribuzione a campana centrale. La terza tipologia di statistica considera i dati con una distribuzione a campana classica o pseudo-tale (Figura 2.3), in quanto in alcuni casi i dati sono discreti e non continui. L'analisi delle distribuzioni, infine, tramite l'utilizzo di box-plot, è servita anche ad individuare la presenza di outliers, dovuti ad errori o a casistiche particolari molto poco riproducibili. Nel caso specifico, ad esempio, guidati dal box plot in figura 2.4, sono stati eliminati tutti i valori maggiori o uguali a 29. Questo procedimento di affinamento delle features è stato svolto per ogni statistiche del dataset.

Analisi della correlazione

Un pilastro centrale della fase preliminare è stato lo studio delle correlazioni intercorrenti tra le variabili del dataset, passaggio imprescindibile per ottimizzare il set di features destinato alla classificazione. Attraverso l'impiego di heatmap (Figure 2.5, 2.6, 2.7), sono state identificate e rimosse le metriche ridondanti, permettendo così uno snellimento del database. Data l'elevata dimensionalità dei dati, che

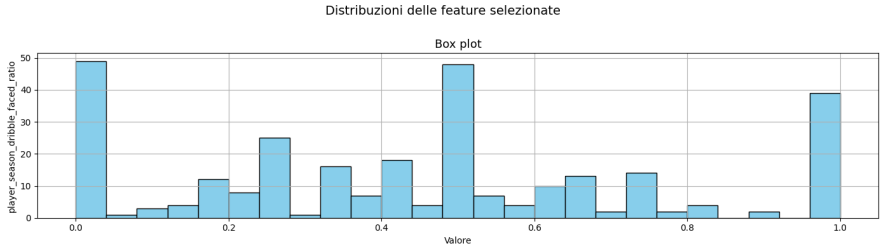

Figura 2.2: Percentuale di dribbling fermati per partita

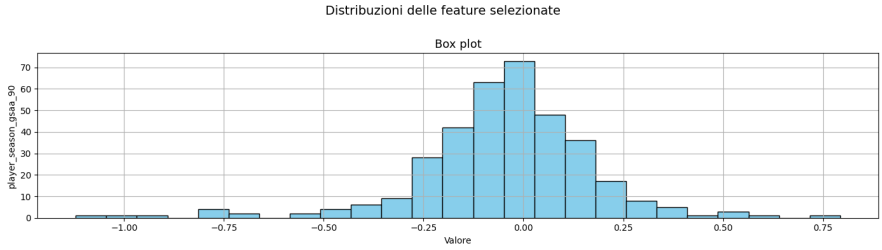

Figura 2.3: Post-shot expected goal subiti per partita

rendeva impraticabile l'uso di un'unica matrice di correlazione, si è proceduto ad una scomposizione del dataset in aree tematiche (distribuzione, parate, azioni difensive, ecc.). Per ciascun ambito è stata generata una heatmap dedicata, isolando e trattando le collinearità specifiche di ogni categoria. Per esempio nella Figura 2.5, si nota come la statistica *aggressive_actions_*90 (azioni aggressive per ogni 90 minuti), sia altamente correlata alle statistiche $defensive\_actions\_90$ e $fouls\_90$. Oppure osservando la figura 2.6, si nota come siano inversamente correlate le statistcihe $forward\_pass\_proportion\_90$ e *sideways_pass_proportion_*90. Nel caso specifico del portiere, questa correlazione inversa è molto più accentuata, poichè chi gioca in porta non effettua passaggi indietro. Di conseguenza tutti i passaggi sono effettuati o in avanti o laterali, e quindi all'aumentare della proporzione dell'uno diminusce la proporzione dell'altro. Altre statistiche molto correlate, e perciò ridonandanti, sono ad esempio il *gsaa_*90 e *gsaa_ratio* (Figura 2.7), queste statistiche indicano quanti gol ha sventato un portiere rispetto ai gol attesi che avrebbe dovuto subire. Queste contengono informazioni molto simili in quanto: la prima, considera quanti gol ha sventato in media ogni novanta minuti; la seconda, la percentuale di gol sventati rispetto a quanti tiri ha subito.

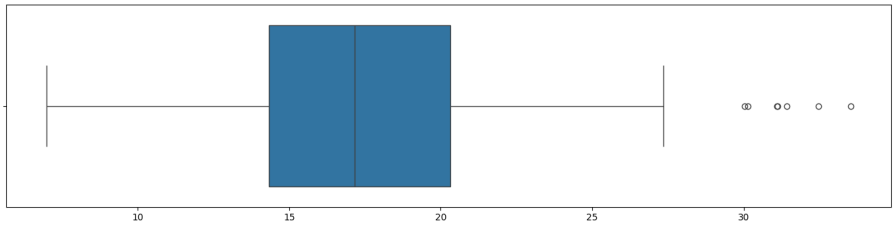

Figura 2.4: Box plot conduzioni palla al piede ogni 90 minuti

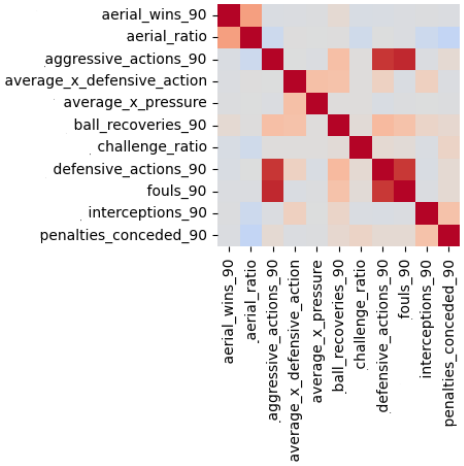

Figura 2.5: Heatmap statistiche sulle azioni difensive

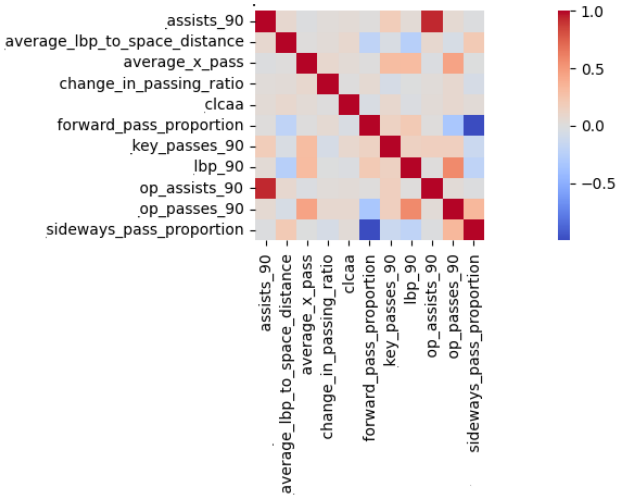

Figura 2.6: Heatmap statistiche di passaggio

2.2 Player Match stats

Il dataset Player Match stats si basa sullo stesso bacino di calciatori del primo, ma i dati sono aggregati diversamente. Infatti, in questo caso, in ogni riga del dataset si hanno i valori delle statistiche ottenute da un singolo portiere in una singola partita. Come il precedente, parte dalla stagione 2019/2020 e arriva alla stagione attualmente in corso; in egual modo è stato pulito da tutte le righe appartenenti a calciatori di movimento, ottenendo un dataset di poco più di 30000 osservazioni con un numero di 80 statistiche. Tra queste statistiche verranno scelte le covariate(X del modello bayesiano), in base alle performance ottenute dal modello con il set di covariate selezionato. Statistiche che si dividono tra: dati di conteggio, rapporti tra questi e metriche di valutazione. Inoltre, per realizzare il ranking, è stata aggiunta una variabile che rappresenta la valutazione della singola prestazione, ottenuta tramite una media ponderata di tutte le statistiche registrate nella singola partita. Questa variabile sarà la Y_i del modello Bayesiano. In aggiunta alla variabile prestazionale, sono stati definiti anche due indici numerici necessari per la corretta

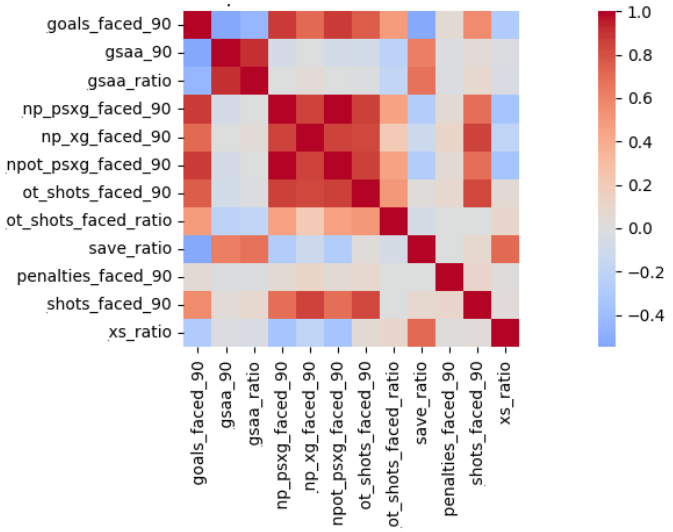

Figura 2.7: Heatmap statistiche di parata

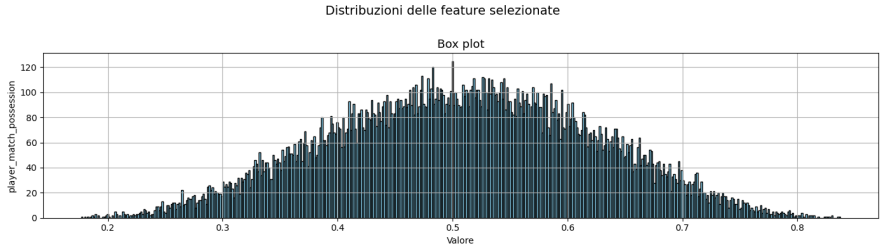

Figura 2.8: Distribuzione variabile possessi

implementazione del modello autoregressivo. Il primo, assegna un numero specifico ad ogni atleta (indice j del modello bayesiano). Il secondo, funge da contatore delle partite nella singola stagione, che viene azzerato ad ogni inizio di stagione.

2.2.1 Analisi preliminare

Guardando le statistiche presenti in questi dataset, si nota che in alcuni campi i valori nulli sono tanti a causa del fatto che sono molti di più i portieri che giocano poco o niente, rispetto a quelli che giocano sempre. Nell'analizzare le distribuzioni dei dati si sono riscontrate delle distribuzioni simili tra loro e di seguito si riportano quelle più rappresentative. In particolare, si hanno distribuzioni a campana, o simil tali (Figura 2.8). Nel caso del passing ratio (Figura 2.9), ad esempio, si nota, un andamento a campana, sporcato da tutti quei casi in cui il portiere esegue soltanto passaggi semplici o un numero basso di passaggi, ottenendo così un valore del 100%. Nel caso della statistica dei gol concessi (Figura 2.10), invece, si ha un picco sullo 0,

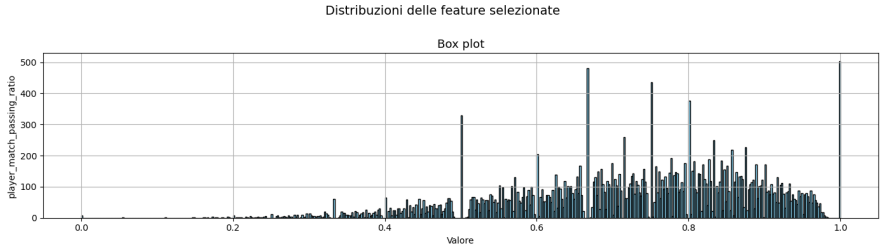

Figura 2.9: Distribuzione variabile passing ratio

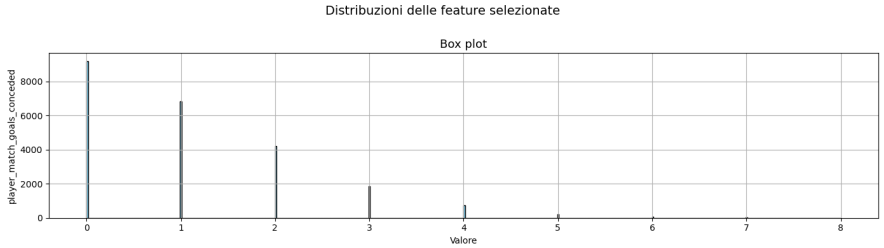

Figura 2.10: Distribuzione variabile goals conceded

che ragionevolmente si abbassa, rispecchiando pienamente la natura di sport a basso punteggio del calcio. Anche in questo caso è stata fatta l'analisi sulla correlazione delle variabili. Le statistiche di questo dataset, sono il livello più "grezzo" del dataset precedente, in quanto non sono raggrupate in valori medi stagionali. Per esempio, al posto di $forward\_pass\_proportion\_90$ e $sideways\_pass\_proportion\_90$ (proporzione di passaggi media per 90 minuti registrate nella singola stagione) si avrà $forward\_pass\_proportion$ e $sideways\_pass\_proportion$ (proporzione di passaggi effettuati nella singola partita). Per questo motivo, si ha che le statistiche altamente correlate nel dataset precedente, risultano esserlo anche in questo. Di conseguenza, si fa riferimento alle stesse heatmap (Figure 2.5, 2.6, 2.7).

Capitolo 3

Classificazione

In questa sezione viene affrontato il problema della classificazione. Questa verrà effettuata utilizzando 3 diversi algoritmi: Random Forest, Gradient Boosting e Regressione Logistica. Gli algoritmi in questione sono stati irrobustiti utilizzando delle tecniche di bilanciamento dei dati come l'oversampling e l'undersampling. Questo aggiustamento è necessario, in quanto, nel caso specifico, per definizione di elitarietà, si ottiene una variabile target molto sbilanciata. Ogni algoritmo, infine, è stato valutato tramite diverse metriche.

3.1 Algoritmi di classificazione

3.1.1 Regressione Logistica

La regressione logistica (Botev et al. 2019) è un modello lineare generalizzato che assume, nel caso di una classificazione binaria, che la risposta Y segua una distribuzione di Bernoulli di parametro $h(\mathbf{x}^T\boldsymbol{\beta})$. Dove $h(u) = 1/(1 + e^{-u})$, e il parametro β viene stimato dal dataset di training massimizzando la verosimiglianza delle risposte di training. Questo modello ha l'obiettivo di modellare la probabilità che un dato input x appartenga ad una determinata classe. La probabilità è definita come:

$$g(y = 1|\boldsymbol{\beta}, \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}^\top \beta}} \tag{3.1}$$

In particolare si ha che il logaritmo del rapporto tra le probabilità (log-odds ratio) è una funzione lineare dei parametri:

$$\ln \frac{g(y = 1|\boldsymbol{\beta}, \mathbf{x})}{g(y = 0|\boldsymbol{\beta}, \mathbf{x})} = \mathbf{x}^\top \boldsymbol{\beta} \tag{3.2}$$

Di conseguenza, il confine tra le classi $\{x : g(y = 0|\boldsymbol{\beta}, \mathbf{x}) = g(y = 1|\boldsymbol{\beta}, \mathbf{x})\}$ è l'iperpiano definito dall'equazione $\mathbf{x}^T\boldsymbol{\beta} = 0$.

3.1.2 Alberi decisionali e Random Forest

Alberi decisionali

Un albero decisionale binario è un albero in cui ogni nodo ν corrisponde ad un sottospazio $\mathcal{R}_\nu$ dello spazio delle feature $\mathcal{X}$. Il nodo radice corrisponde allo spazio delle feature stesso. Ogni nodo interno ν contiene una condizione logica che divide $\mathcal{R}_\nu$ in due sottospazi disgiunti. I nodi foglia (nodi terminali dell'albero) non sono suddivisi e formano una partizione di $\mathcal{X}$. Più in generale un albero binario $\mathbb{T}$ partizona uno spazio delle feature $\mathcal{X}$ in tante regioni quanti sono i nodi foglia. Denotando il set dei nodi foglia con $\mathcal{W}$. La funzione di predizione g che corrisponde ad un albero può essere scritta come:

$$g(\mathbf{x}) = \sum_{w \in \mathcal{W}} g^w(\mathbf{x}) \mathbb{1}\{\mathbf{x} \in \mathcal{R}_w\} \tag{3.3}$$

.

Random Forests

Uno dei limiti principali degli alberi decisionali, è che un singolo albero di decisione è facile da interpretare ma spesso fragile: tende a "imparare a memoria" i dati di addestramento (overfitting), perdendo precisione quando vede dati nuovi. Il Random Forest risolve questo problema unendo centinaia o migliaia di alberi per ottenere una previsione più robusta e accurata. Invece di addestrare ogni albero utilizzando l'intero set di dati, l'algoritmo crea dei campioni più piccoli del dataset originale e addestra ogni albero su un set di features diverso. L'idea alla base del Random Forests è dunque quella di decorrelare gli alberi forzandoli ad essere diversi (Botev et al. 2019). In particolare, durante la costruzione di ogni albero, per ogni split, non si scelgono tutte le p variabili disponibili, ma solo un sottoinsieme m (solitamente $m \approx \sqrt{p}$). Così facendo anche i predittori più forti non saranno sempre disponibili, forzando l'algoritmo a scegliere altre variabili riducendo la correlazione tra alberi. La previsione finale è il voto di maggioranza di tutti gli alberi prodotti. Le Random Forest a causa di questo tipo di costruzione perdono l'interpretabilità immediata del singolo albero, ma permettono di calcolare l'importanza delle variabili. L'importanza di una variabile x_j è definita come la riduzione di valore di una funzione di perdita indotta dai nodi in cui quella variabile è stata utilizzata, questo valore viene poi mediato su tutti i B alberi della foresta:

$$I_{RF}(x_j) = \frac{1}{B} \sum_{b=1}^{B} I_{T_b}(x_j) \tag{3.4}$$

dove

$$I_T(x_j) = \sum_{v\text{interno} \in T} \Delta_{Loss}(v) \mathbb{1}\{x_j \text{ è associata con } v\}$$

3.1.3 Gradient Boosting

A differenza del random forest, dove gli alberi sono indipendenti, il boosting costruisce i modelli in modo sequenziale (Botev et al. 2019). Si parte da un modello molto semplice (weak-learner) g_0 applicato ai dati $\tau = \{(x_i, y_i)\}_{i=1}^n$, per poi migliorare o potenziare ("boost") tale apprenditore ottenendo $g_1 := g_0 + m_1$. In questo contesto, la funzione m_1 viene individuata minimizzando la perdita di addestramento (training loss) per $g_0 + m_1$ all'interno di una determinata classe di funzioni $\mathcal{M}$. Ad esempio, $\mathcal{M}$ potrebbe rappresentare l'insieme delle funzioni di predizione ottenibili tramite un albero decisionale con profondità massima pari a 2. Data una funzione di perdita (Loss), m_1 si ottiene quindi come soluzione al problema di ottimizzazione:

$$m_1 = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \operatorname{Loss}\left(y_i, g_0(\mathbf{x}_i) + m(\mathbf{x}_i)\right) \tag{3.5}$$

Questo processo può essere ripetuto per g_1 al fine di ottenere $g_2 = g_1 + m_2$, e così via, producendo la funzione di predizione finale:

$$g_B(x) = g_0(x) + \sum_{b=1}^{B} m_b(x) \tag{3.6}$$

Invece di adottare il passo di aggiornamento $g_b = g_{b-1} + m_b$, si preferisce solitamente utilizzare un aggiornamento graduale nella forma $g_b = g_{b-1} + \gamma m_b$, dove γ rappresenta un parametro di step-size (o learning rate) opportunamente scelto. Nel gradient boosting il parametro γ gioca un ruolo cruciale: esso determina l'entità dello spostamento effettuato nella direzione del gradiente negativo della funzione di perdita. Nel caso specifico della funzione di perdita quadratica (squared-error loss), il gradiente negativo coincide con il residuo $y_i - g_{b-1}(x_i)$. Formalmente, si può osservare che:

$$-\left[\frac{\partial \operatorname{Loss}(y_i, z)}{\partial z}\right]_{z=g_{b-1}(x_i)} = -\left[\frac{\partial (y_i - z)^2}{\partial z}\right]_{z=g_{b-1}(x_i)} = 2(y_i - g_{b-1}(x_i)) \tag{3.7}$$

Questa uguaglianza dimostra che, ad ogni iterazione, l'algoritmo non fa altro che addestrare un nuovo weak learner per prevedere gli errori residui del modello corrente. Il contributo di ogni nuovo modello m_b viene scalato dal fattore γ, che agisce come regolarizzatore: un valore di γ ridotto (tipicamente compreso tra 0.01 e 0.1) impedisce al modello di adattarsi eccessivamente al rumore dei dati di addestramento, riducendo drasticamente il rischio di overfitting e migliorando la capacità di generalizzazione del sistema finale.

3.1.4 Oversampling e Undersampling

Tutti gli algoritmi illustrati fin ora sono algoritmi "classici" orientati a lavorare con dati bilanciati. Poichè in questo elaborato l'obiettivo è estrarre caratteristiche dei portieri di élite, per definizione, si ottengono dei dati fortemente sbilanciati, in cui si è interessati a classificare la classe minoritaria. Di conseguenza, il costo di un falso negativo (i.e. l'errore di identificare la classe minoritaria) deve essere maggiore di un falso positivo (i.e. l'errore di identificare la classe maggioritaria). Per bilanciare i dataset sbilanciati si utilizzano due metodi:

- **Oversampling**: Si aumentano i dati della classe minoritaria;
- **Undersampling**: Si diminuiscono i dati della classe maggioritaria.

I due metodi possono essere anche combinati, in particolare, in questo elaborato i metodi di undersampling sono sempre stati utilizzati in combinazione con metodi di oversampling, in quanto, il dataset aveva già dimensioni limitate, ed eliminare ulteriori dati avrebbe rappresentato una perdita di informazione notevole (Piyadasa and Gunawardana 2023).

SMOTE (Synthetic Minority Oversampling Technique)

L'idea principale dell'algoritmo di oversampling SMOTE è quella di generare dati sintetici della classe minoritaria. Per ottenere questo risultato, seleziona in modo casuale un'istanza della classe minoritaria e poi individua i suoi k vicini più prossimi della classe minoritaria. Quindi genera nuove istanze sintetiche mediante interpolazione tra l'istanza minoritaria originale e i suoi k vicini più prossimi. In questo modo non copia gli esempi già presenti, ma li sfrutta per crearne di nuovi. Uno degli svantaggi di questo algoritmo è che potrebbe inserire rumore nel dataset.

ADASYN (Adaptive Synthetic Sampling)

L'ADASYN è una tecnica di oversampling che mira a creare dati sintetici in regioni dello spazio più vicine al confine decisionale. L'obiettivo è quello di sviluppare più campioni sintetici per i campioni della classe minoritaria più difficili da identificare. Utilizza come modelli per i dati sintetici i campioni della classe minoritaria se alcuni dei loro vicini più prossimi appartengono alla classe opposta. Maggiore è il numero di vicini appartenenti alla classe opposta, maggiore è la probabilità che venga utilizzato come modello. Dopo aver selezionato i modelli, genera i nuovi dati mediante interpolazione tra il modello e i vicini più prossimi della stessa classe.

Tomek Links

La tecnica Tomek Links è una tecnica di undersampling che si concentra sulla pulizia dei dati vicino al confine decisionale. Se due campioni sono vicini e appartengono a classi diverse, formano un Tomek Link. L'algoritmo dunque, elimina il campione del Tomek Link appartenente alla classe maggioritaria. Il presupposto di base è che i campioni che sono vicini più prossimi ma appartengono a classi diverse contribuiscono al rumore dei dati di addestramento.

ENN (Edited Nearest Neighbors)

L'ENN è una tecnica di undersampling che rimuove le osservazioni dalla classe maggioritaria se la maggior parte dei vicini del campione appartiene alla classe opposta. Si tratta di dati che potrebbero essere classificati erroneamente da un classificatore. Questa tecnica può ridurre il rumore nei dati eliminando i campioni sovrapposti presenti al confine decisionale tra le classi, contribuendo a creare un confine tra le classi più pulito.

3.1.5 Metriche di valutazione dei modelli di classificazione

Per l'interpretazione dei risultati come in ogni modello del genere si è partiti dalla matrice di confusione.
La matrice di confusione è una tabella che confronta le predizioni effettuate dal modello con le etichette reali dei dati. Per un problema di classificazione binaria (come quello trattato in questo elaborato), la matrice si presenta come una tabella 2×2, con i seguenti componenti:

- **Vero Positivo (True Positive, TP)**: I casi in cui il modello ha correttamente identificato la classe positiva.
- **Vero Negativo (True Negative, TN)**: I casi in cui il modello ha correttamente identificato la classe negativa.
- **Falso Positivo (False Positive, FP)**: I casi in cui il modello ha erroneamente identificato una classe positiva quando la vera classe è negativa.
- **Falso Negativo (False Negative, FN)**: I casi in cui il modello ha erroneamente identificato una classe negativa quando la vera classe è positiva.

Questa struttura consente di analizzare in dettaglio il comportamento del modello in differenti scenari, mettendo in luce non solo la percentuale di predizioni corrette, ma anche la natura degli errori commessi. Dalla matrice di confusione si possono estrarre diverse metriche chiave che forniscono una visione più dettagliata delle prestazioni del modello:

Accuratezza (Accuracy)

Indica la proporzione di predizioni corrette sul totale dei casi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.8}$$

Questa metrica molto semplice, nei casi in cui il dataset è sbilanciato può essere fuorviante, in quanto, pur predicendo soltanto la classe maggioritaria si otterebbe un'accuratezza molto alta. Quindi, nel caso in esame, bisogna fare attenzione poichè un valore di accuratezza molto elevato non rispecchia, automaticamente, la capacità del modello di riconoscere entrambe le classi.

F1-Score

L'F1-Score è la media armonica tra Precision e Recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.9}$$

dove, la Precision indica la proporzione di veri positivi sul totale delle predizioni positive:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{3.10}$$

e la Recall indica la capacità del modello di identificare correttamente le istanze positive:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3.11}$$

La metrica F1 è utile perchè penalizza i valori estremi. Se un modello ha una Recall perfetta (trova tutti i positivi) ma una Precision pessima (spara a caso su tutto), l'F1-Score sarà basso. Per questo motivo è una metrica ottima in casi in cui il dataset è molto sbilanciato.

ROC-AUC (Receiver Operating Characteristic and Area Under the Curve)

La metrica ROC-AUC consiste nel calcolare l'area sottostante la curva ROC. La curva ROC è un grafico che mostra il trade-off tra il tasso di Veri Positivi (Sensitivity o Recall) e il tasso di Falsi Positivi (1-Specificity) al variare della soglia decisionale del modello, dove:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.12}$$

Nella curva ROC, l'asse X rappresenta il tasso di falsi positivi (FPR) e l'asse Y rappresenta il tasso di veri positivi (TPR).

L'AUC rappresenta l'area sottostante la curva appena descritta. Questo valore ci aiuta ad identificare la capacità del modello di separare le due classi, indipendentemente dalla soglia specifica che scegliamo per decidere se un campione appartiene a una classe o ad un'altra:

- AUC = 1.0: Modello perfetto;
- AUC = 0.5: Il modello equivale ad un lancio di moneta (è casuale).

In poche parole, più la curva ROC è vicina all'angolo superiore sinistro del grafico, migliore è la capacità del modello di distinguere tra casi positivi e negativi. Al contrario, una curva più vicina alla linea diagonale rappresenta un modello che non fornisce previsioni migliori di quelle casuali.

3.2 Pre-processing e implementazione del modello

Per la fase di classificazione è stato utilizzato il dataset Player Season Stats, sottoposto a una pulizia rigorosa dopo l'analisi preliminare. Tale processo ha comportato la rimozione degli outliers per ogni statistica, l'eliminazione delle feature eccessivamente correlate e l'esclusione dei dati relativi alla stagione calcistica corrente (2025/2026). Inoltre, sono stati eliminati i profili dei portieri con un impiego inferiore ai 450 minuti totali nell'arco dei cinque anni considerati.
Una volta ottenuta la versione finale del dataset, i dati sono stati uniformati tramite la tecnica del MinMaxScaling implementata tramite la libreria Scikit-learn (Pedregosa et al. 2011), che trasforma ogni valore secondo l'equazione:

$$x_{scaled} = \frac{(x - x_{min})}{(x_{max} - x_{min})}, \tag{3.13}$$

dove x_{min} e x_{max} rappresentano rispettivamente il valore minimo e massimo della feature nel dataset.
Il set ottimale di variabili è stato successivamente isolato attraverso l'algoritmo di Recursive Feature Elimination (RFE) (Darst et al. 2018). Questo metodo opera per sottrazione, partendo dal set completo delle statistiche e rimuovendo iterativamente la variabile meno significativa in base all'importanza assegnata dal modello, fino al raggiungimento del sottoinsieme ottimale di features. Questo procedimento è stato implementato su tutti e 3 i modelli di classificazione utilizzati, ottenendo così il set di features migliore per ogni modello. Inoltre, la RFE è utile per eliminare il rumore informativo, che potrebbe influire negativamente sulle performance del modello.
Data la natura sbilanciata del dataset, in cui i portieri d'élite rappresentano una

minoranza, è stata applicata una strategia combinata di bilanciamento delle classi. Nello specifico, l'utilizzo congiunto di SMOTE per l'oversampling sintetico della classe minoritaria e dei Tomek Links per la rimozione dei punti di confusione ai confini decisionali, ha permesso di rendere più netta la distinzione tra le categorie. Queste tecniche sono state implementate tramite le libreria imbalanced-learn di python (Lemaître et al. 2017)
Per garantire la massima robustezza, la classificazione è stata affidata a tre diversi algoritmi: Regressione Logistica, Random Forest e Gradient Boosting. Ognuno di questi è stato implementato tramite la libreria python scikit-learn (Pedregosa et al. 2011), ed è stato sottoposto ad una procedura di Hyperparameter Tuning. Per ogni algoritmo selezionato, è stato definito uno spazio di ricerca (grid) basato sulle proprietà intrinseche del modello e sulla letteratura di riferimento. Iniziando dalla Regressione Logistica, la configurazione si è focalizzata principalmente sulla gestione della regolarizzazione per contrastare l'overfitting. In particolare, è stato testato il parametro C attraverso i valori $0.1, 1$ e 10, dove valori inferiori imprimono una regolarizzazione più severa penalizzando i coefficienti elevati per semplificare l'architettura del modello. A supporto del processo di convergenza, la tolleranza per i criteri di arresto dell'algoritmo, indicata dal parametro tol, è stata esplorata nei valori 0.01 e 0.001, mentre come solutore è stato adottato liblinear, selezionato per la sua nota efficacia su dataset di dimensioni medio-piccole e per la flessibilità nel supportare diverse penalità. Per quanto concerne il Random Forest, l'attenzione si è spostata sul controllo della complessità dei singoli alberi. Il numero di stimatori (n_estimators) è stato valutato tra $200, 400$ e 600 unità, partendo dal presupposto che un numero elevato di alberi tenda a stabilizzare le previsioni a fronte di un maggiore carico computazionale. La profondità massima degli alberi, definita da max_depth, è stata limitata a 8 o 10 per evitare che il modello catturasse il rumore presente nei dati di addestramento. In parallelo, la crescita della struttura è stata regolata tramite i parametri min_samples_split (testato su 2 e 4) e min_samples_leaf (testato su 1 e 2), i quali agiscono come regolarizzatori naturali impedendo la creazione di nodi troppo specifici. Infine, l'analisi del Gradient Boosting ha richiesto un approccio differente a causa della sua struttura sequenziale. In questo caso, i parametri sono stati scelti per ottimizzare la convergenza del processo di boosting. Il learning rate è stato impostato sui valori 0.1 e 0.2, considerando che un tasso di apprendimento più contenuto richiede generalmente un numero superiore di stimatori ma favorisce una generalizzazione superiore. Coerentemente con tale logica, il parametro n_estimators è stato esplorato nei valori $300, 500$ e 700. Nonostante il Gradient Boosting utilizzi tipicamente "weak learners" molto semplici, la profondità massima degli alberi (max_depth) è stata estesa in un range di $4, 6$ e 8 per permettere al modello di intercettare interazioni tra le variabili caratterizzate da una maggiore complessità strutturale. La valutazione, inoltre, è stata condotta tramite una 5-fold cross-validation stratificata, che assicura la

medesima proporzione di portieri d'élite in ogni iterazione, riducendo il rischio di overfitting e garantendo che i modelli apprendano da partizioni di dati sempre differenti. Questo procedimento consiste nel dividere il dataset in 5 parti e utilizzare, a rotazione, 4 parti per l'addestramento e una per il test. Per affinare la gestione dello sbilanciamento delle classi, inoltre, la soglia decisionale (convenzionalmente settata a 0.5) è stata ottimizzata per ogni modello massimizzando l'F1-Score attraverso l'analisi della curva Precision-Recall, garantendone la miglior performance possibile. Infine, l'estrazione delle feature più influenti è stata eseguita analizzando i pesi mediati sui 5 fold per garantirne la stabilità. Per i modelli basati su alberi (Random Forest e Gradient Boosting), si è operato quantificando il contributo di ciascuna variabile nella riduzione dell'impurità all'interno dei nodi degli alberi. In particolare, ogni volta che una feature viene selezionata per effettuare una divisione dei dati, l'algoritmo calcola quanto tale suddivisione riesca a rendere i nodi risultanti più omogenei rispetto alla variabile target. Il valore finale è rappresentato dalla somma di tutte le riduzioni di impurità ottenute grazie a quella specifica feature, mediata su tutti gli alberi della foresta e normalizzata in modo che il totale sia pari a 1. Pertanto, una feature con un punteggio elevato indica una variabile che ha giocato un ruolo determinante e frequente nel separare correttamente i portieri "Élite" da quelli "Standard" lungo l'intera struttura del modello. Per quanto riguarda la Regressione Logistica, invece, sono stati estratti i coefficienti assoluti, resi confrontabili grazie alla precedente normalizzazione tramite MinMaxScaler.

3.3 Risultati

Nella sezione finale di questo capitolo si analizzano i risultati ottenuti utilizzando i processi precedentemente descritti. Sono state effettuate due diverse classificazioni. Nel primo caso, si considera la singola stagione del singolo portiere, quindi, si valutano le caratteristiche di una stagione d'élite. Nel secondo caso, si considera l'intera carriera, quindi, si estraggono le caratteristiche di una carriera (nei limiti dei dati disponibili) d'élite. In entrambi i casi i modelli hanno raggiunto la migliore performance con la stessa configurazione di parametri. La Regressione Logistica, con un numero massimo di iterazioni fissato a 10000, ha performato meglio con valori di C pari a 10 e tolleranza pari a 0.01. Il gradient Boosting, ha ottenuto le migliori performance con il numero di alberi settato a 700, la profondità massima dell'albero a 4 e un learning rate uguale a 0.2. Infine, il Random Forest, ha fatto registrare i migliori risultati con un numero di alberi settato a 600, la profondità massima di questi a 8, il numero minimo di campioni necessari per scindere un nodo interno e il numero minimo di campioni richiesti per trovarsi in un nodo foglia pari a 2.

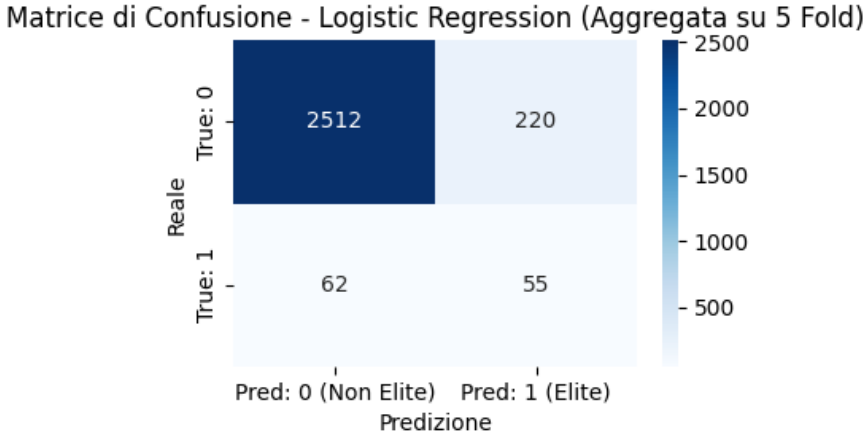

Figura 3.1: Matrice di confusione Regressione Logistica

3.3.1 Classificazione portiere-stagione

Nella prima classificazione, sono state considerate le singole stagioni, quindi si è scelto di definire elitarie le stagioni dei portieri presenti agli ottavi di Champions League e dei portieri presenti nella top 10 del premio Yashin. I dati sono stati aggregati per portiere e stagione, quindi uno stesso portiere nell'arco dei 5 anni potrebbe avere stagioni d'élite o meno. L'idea alla base di questa soluzione era quella di mantenere il parametro elitario molto limitato, evitando di definire come elitarie le carriere di portieri che hanno partecipato magari una sola volta agli ottavi di Champions League nei 5 anni.
In questo contesto, come si evince dalle matrici di confusione (Figure 3.1, 3.2, 3.3), in tutti e 3 i modelli si riscontrano valori di accuracy e ROC-AUC molto elevate. Come discusso in precedenza (sezione 3.1.5), questo non indica una buona adattabilità del modello, in quanto avendo un dataset altamente sbilanciato è inevitabile che l'accuracy lo sia. In questo caso avere un valore di ROC-AUC alto, visti i valori bassi dell'F1-score (Figura 4.9), indica che il modello "capisce" la differenza tra le due classi, ma non sa ancora dove "tracciare la linea" per decidere chi è chi. In dataset sbilanciati come questo, la metrica regina è l'F1-score e in questo caso risulta essere bassa, indicando una troppa generosità del modello (identifica tanti portieri normali come élite). Queste valutazioni valgono per tutti e 3 i modelli, infatti, nonostante il migliore sia il Random Forest, questo non ottiente performance tanto migliori degli altri due. Questa carenza di risultati ha portato a modificare l'aggregazione dei dati per provare a migliorare le performance.

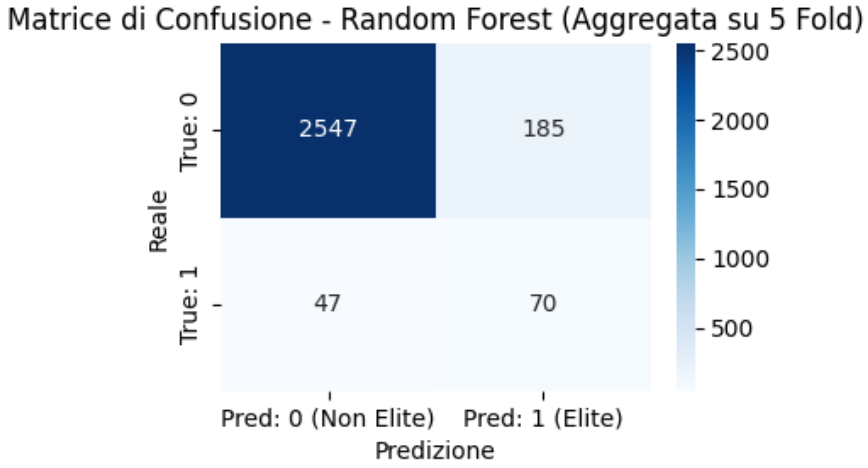

Figura 3.2: Matrice di confusione Random Forest

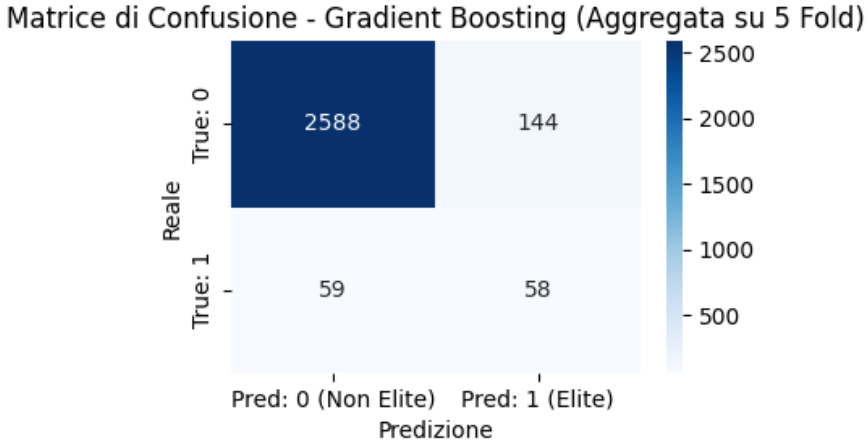

Figura 3.3: Matrice di confusione Gradient Boosting

3.3.2 Classificazione per portiere

Visti i risultati ottenuti nella prima classificazione, si è deciso di aggregare ulteriormente i dati, passando ad un'aggregazione per portiere e per stagione, a una soltanto per portiere. In questo modo si è ottenuto un dataset contenente le statistiche legate alle carriere dei singoli portieri nei 5 anni di cui si dispongono i dati. Sono stati considerati come portieri elitari, in questo caso, tutti i portieri che, almeno una volta nei 5 anni, hanno partecipato agli ottavi di Champions League o sono comparsi nella top 10 del premio Yashin. Questa scelta ha rimpicciolito ulteriormente il dataset, il che solitamente non è vantaggioso, ma nel caso specifico

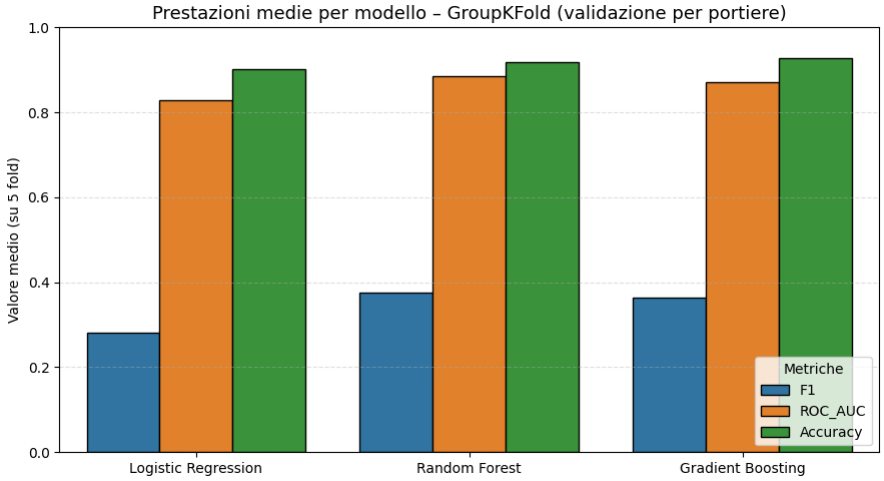

Figura 3.4: Metriche di valutazione

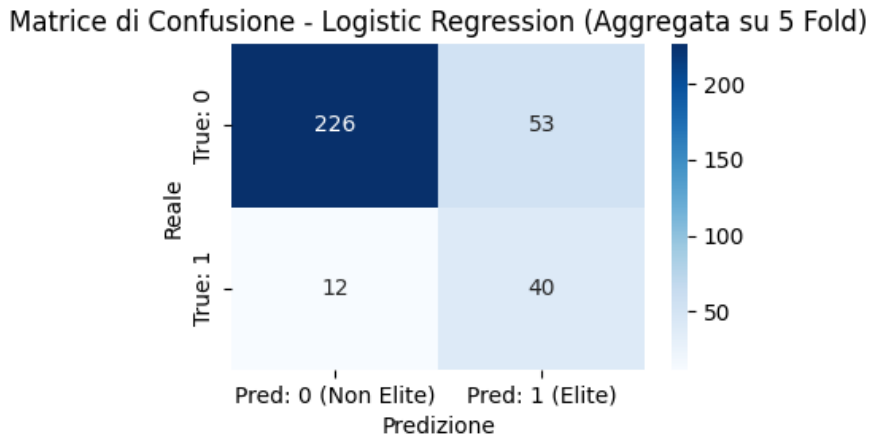

Figura 3.5: Matrice di confusione Regressione Logistica

ha ridotto lo sbilanciamento, consentendo ai modelli di performare meglio.
Dalle matrici di confusione (Figure 3.5, 3.6, 3.7). si evince che il modello che performa meglio è il Random Forest, ma il Gradient Boosting non ottiene performance così differenti. Anche in questo caso il peggior modello è la Regressione Logistica, ma i miglioramenti rispetto alla prima classificazione sono notevoli. Infatti, come nel primo caso, si ottengono valori di accuracy e di ROC-AUC molto alta. Ma, a differenza della classificazione precedente, è molto più alto il valore dell'F1-score (Figura 3.8), indicando un miglioramento nella capacità del modello di riconoscere le caratteristiche desiderate.

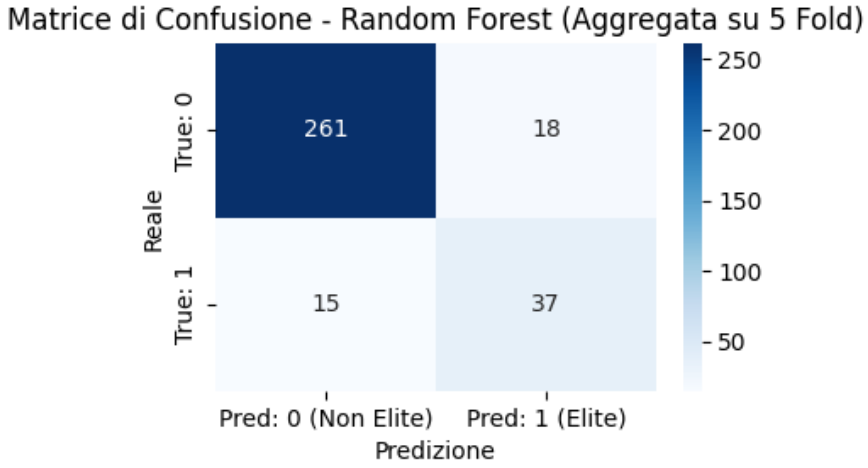

Figura 3.6: Matrice di confusione Random Forest

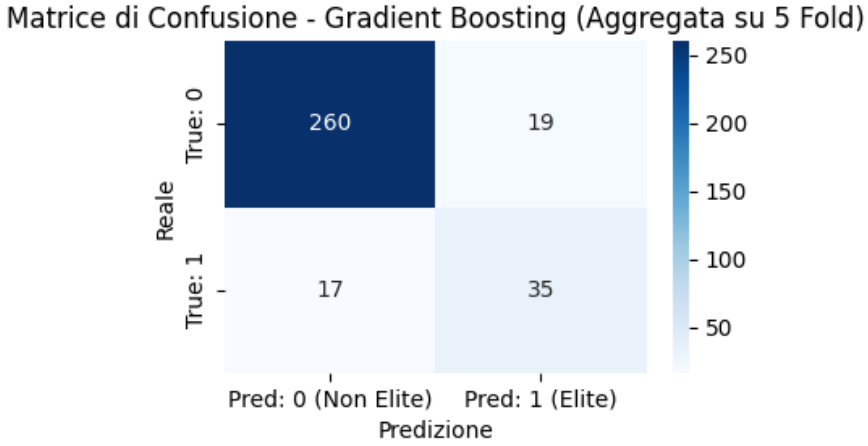

Figura 3.7: Matrice di confusione Gradient Boosting

Valutazione dei limiti della classificazione

Ogni modello presenta un limite dovuto alla natura dei dati e alle procedure utilizzate. Per verificare il raggiungimento di questo limite è stato utilizzato un sistema di ensemble voting, combinato con la valutazione della correlazione dei modelli. In questo caso che, come si evince nell'heatmap in figura 3.9, i modelli non sono eccessivamente correlati l'ensemble voting dovrebbe essere molto efficace, in quanto unisce i punti di vista dei tre modelli. Il sistema di votazione consiste nell'estrarre le probabilità predette da ogni modello su ogni dato e successivamente calcolare la media aritmetica tra queste. In questo caso come metrica di valutazione si tiene

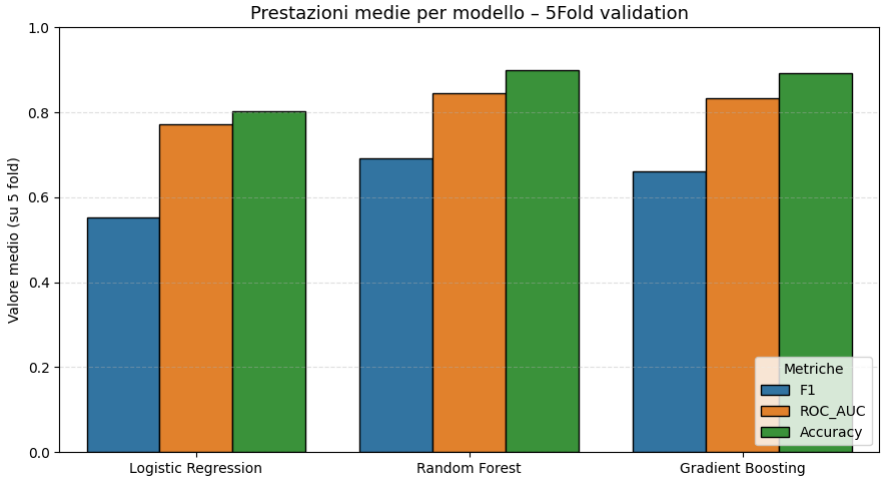

Figura 3.8: Metriche di valutazione

conto della metrica ROC-AUC (Tabella 3.1) , in quanto questa è indipendente dalla soglia. Questa indipendenza è necessaria dal momento che il modello ensemble lavora sulla media della soglia. Come si evince dalla tabella 3.1 il modello ensemble migliora le performance rispetto ai modelli singoli, ma il miglioramento rispetto al modello migliore non è poi così elevato. Ciò indica che il modello è molto vicino al limite informativo dei dati e che i risultati ottenuti sono i migliori possibili con questo dataset e questa procedura. Un altro elemento che ci porta a questa conclusione è la separazione delle classi che si osserva nel grafico in figura 3.10. In particolare, si nota un'alta separabilità nei casi in cui il portiere gioca molto poco o mai, o nei casi in cui il portiere è ad un alto livello secondo il modello. Nel mezzo invece non è semplice separare le due classi, e ciò è dovuto alla natura dei dati. Poichè alcuni portieri con performance di livello elitario, possono militare in squadre che non hanno mai raggiunto gli ottavi di Champions League, o viceversa, esistono portieri non di primissimo livello che hanno raggiunto quel traguardo.

Modello	**ROC-AUC**
Random Forest	0.839
Gradient Boosting	0.817
Logistic Regression	0.786
Ensemble	0.848

Tabella 3.1: Confronto modelli singoli e modello ensemble

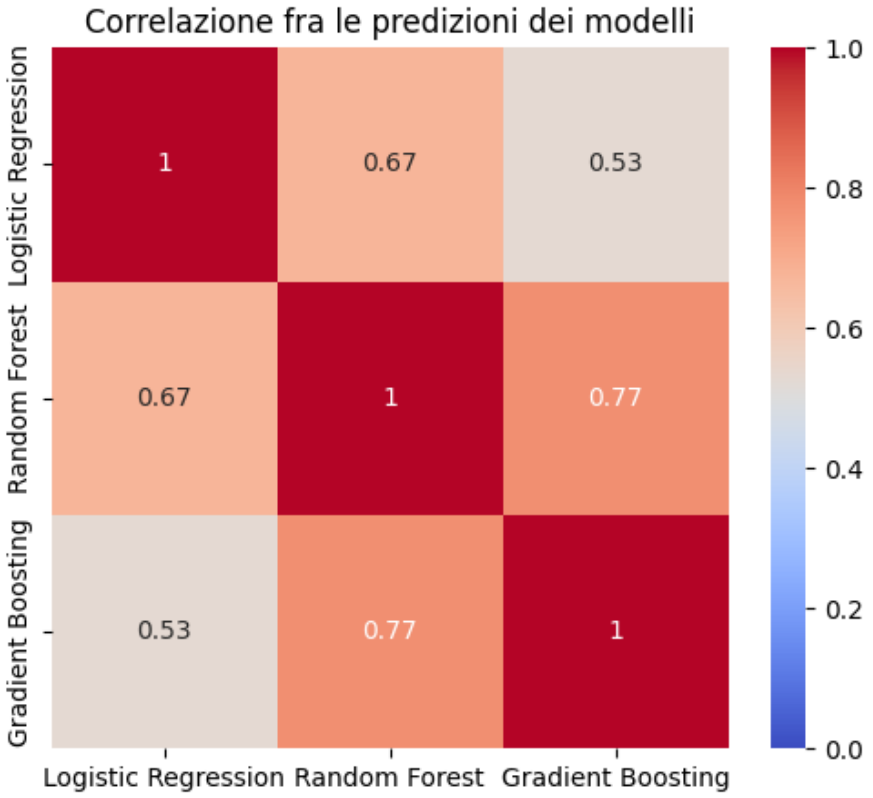

Figura 3.9: Heatmap della correlazione delle predizioni

Estrazione caratteristiche

Una volta addestrato il modello a riconoscere i portieri d'élite si possono estrarre le statistiche che influiscono di più nel riconoscimento di questi ultimi. In particolare nel grafico 3.11 si hanno le 20 statistiche più influenti nel modello Random Forest. Per quanto riguarda il modello migliore si ha che tra le 20 migliori statistiche la maggior parte sono statistiche di distribuzione della palla e di capacità di effettuare una parata. Statistiche come il rapporto di passaggi riusciti sottopressione (pressured passing ratio) e il numero di passaggi che superano una linea avversaria (through balls), sono la prova dell'evoluzione del ruolo del portiere. Al giorno d'oggi un portiere non deve solo saper parare, e quindi evitare gol che hanno statistiche di successo elevato, ma deve anche saper giocare con i piedi sottopressione, riconoscere gli spazi ed individuare l'uomo libero. Per cercare di individuare un valore assoluto del peso di ciascuna area specifica si è deciso di dividere le statistiche in tre gruppi fondamentali. Il primo gruppo (Distribuzione palla), contiene tutte le statistiche di passaggio e le metriche di impatto offensivo. Il secondo (Capacità di parata), raggruppa tutte le statistiche che rappresentano la capacità di intervento tra i

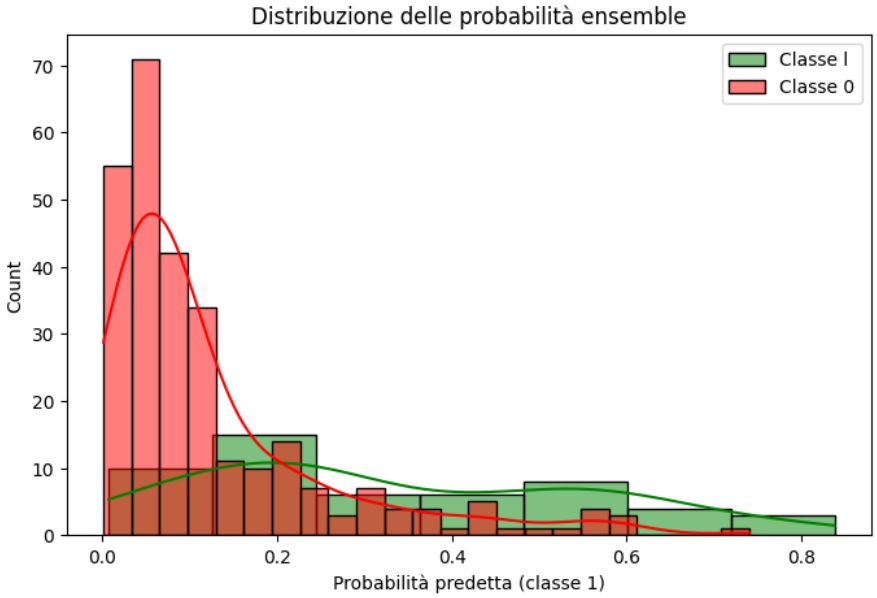

Figura 3.10: Distribuzione delle classi

pali. Infine, il terzo (Azioni difensive e posizionamento), ha al suo interno tutte le statistiche che riguardano le uscite e gli errori di posizionamento. Definiti i gruppi, se ne è calcolato il peso di influenza sommando i pesi delle statistiche presenti in ognuno di questi. Dalla tabella 3.2 viene fuori ancor di più quanto affermato in precedenza. Nel mondo professionistico contemporaneo di alto livello, la differenza tra un portiere di livello standard e un portiere di livello élite è data sempre di più dalla capacità dello stesso di distribuire la palla. Questo dato potrebbe far storcere il naso, ma non vuol dire che i portieri moderni non hanno bisogno di saper interpretare il ruolo in maniera "tradizionale". Infatti le caratteristiche classiche di un portiere hanno ancora un peso importante nel raggiungere determinati livelli. Sommando l'area delle parate e delle azioni difensive si arriva ad un peso del 42,9% del peso totale.

Gruppo	Peso
Distribuzione palla	0.571
Capacità di parata	0.271
Azioni difensive e posizionamento	0.158

Tabella 3.2: Pesi di influenza aree tecniche

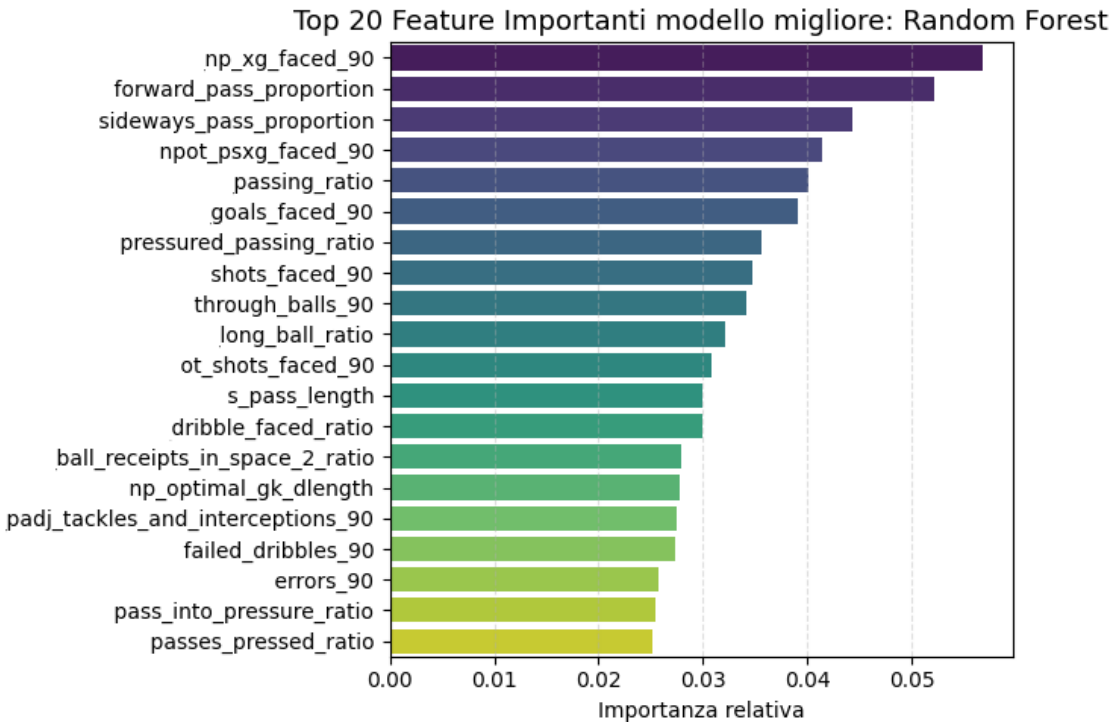

Figura 3.11: Top 20 features modello migliore

Capitolo 4

Ranking

In questa sezione si affronta il problema delle definizione di un ranking tra i migliori portieri al mondo, basato sui dati a disposizione. L'idea di base è quella di utilizzare la statistica bayesiana per calcolare la curva di performance di ogni portiere, negli anni disponibili, e confrontare il valore dell'area sottostante la curva.

4.1 Statistica Bayesiana e Modello Autoregressivo

4.1.1 Statistica Bayesiana

La statistica Bayesiana si fonda sul teorema di Bayes: date le osservazioni y e il parametro di interesse $\boldsymbol{\theta}$

$$f(\boldsymbol{\theta}|y) = \frac{f(y|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(y)}. \tag{4.1}$$

Nell'approccio Bayesiano, al contrario di quello frequentista, in cui i parametri di interesse sono considerati fissi, i parametri sono variabili aleatorie la cui distribuzione può rappresentare, sia la legge di probabilità che genera il parametro, sia l'incertezza sul valore del parametro stesso. Quindi, supponendo che $\boldsymbol{\theta}$ sia il parametro di interesse, si ha che:

- $f(\boldsymbol{\theta}|y)$ è la distribuzione a posteriori;
- $f(y|\boldsymbol{\theta})$ è la congiunta delle osservazioni (che può essere vista come la verosimiglianza);
- $f(\boldsymbol{\theta})$ è la distribuzione a priori (la conoscenza che si ha del parametro prima di osservare i dati);
- $f(y)$ è la costante di normalizzazione.

L'obiettivo di questo approccio è aggiornare l'informazione sul parametro mediante l'osservazione dei dati. Questo aggiornamento viene tipicamente fatto tramite algoritmi MCMC (Markov Chain Monte Carlo) la cui teoria va oltre l'intento di questo elaborato. Uno dei vantaggi della statistica Bayesiana è che, a differenza della statistica classica, non richiede grandi quantità di dati per essere affidabile, il metodo bayesiano può produrre risultati sensati anche con pochi dati.

4.1.2 Modello autoregressivo (AR)

Il modello autoregressivo di ordine 1 assume che:

$$x_t = \alpha x_{t-1} + w_t \tag{4.2}$$

dove w_t è un white noise, ovvero un modello in cui le variabili sono semplicemente indipendenti. Questo modello, dunque, fa dipendendere ogni punto da quello precedente, aggiungendo una componente casuale. Dalla statistica sappiamo che il processo in questione è gaussiano ($E(X_t) = 0$), in quanto combinazione lineare di processi gaussiani. Inoltre, sappiamo anche che il processo ha senso solo se $|\alpha| < 1$.

Metrica di valutazione del modello: Watanabe-Akaike Information Criterion (WAIC)

Per valutare la bontà del modello, si usa un indice capace di tenere in considerazione sia la verosimiglianza del modello, sia la sua complessità. Nel caso particolare, visto che si utilizza un processo bayesiano, il criterio di valutazione scelto è il WAIC:

$$\text{WAIC} = -2\sum_{i=1}^{n} \log\left(\frac{1}{B}\sum_{b=1}^{B} f(y_i|\boldsymbol{\theta}^b)\right) + 2\sum_{i=1}^{n} \text{Var}_{\text{posterior}}\left(\log f(y_i|\boldsymbol{\theta})\right) \tag{4.3}$$

dove con $\text{Var}_{\text{posterior}}\left(\log f(y_i|\boldsymbol{\theta})\right)$ si intende una stima della varianza effettuata utilizzando i campioni della distribuzione a posteriori. Questo indice è appropriato solo se i dati risultano indipendenti. Quando si confrontano più modelli bayesiani, si preferisce quello con il valore WAIC più basso.

4.2 Pre-processing

Per la parte di ranking è stato utilizzato il dataset Player Match Stats. Come nel precedente caso, è stato pulito dopo l'analisi preliminare. Tale processo ha portato all'eliminazione delle features eccessivamente correlate e degli outliers per ogni statistica. Inoltre, sono stati eliminati i dati dei portieri che hanno registrato meno di 40 presenze negli anni in esame. Infine, come nella classificazione, i dati sono

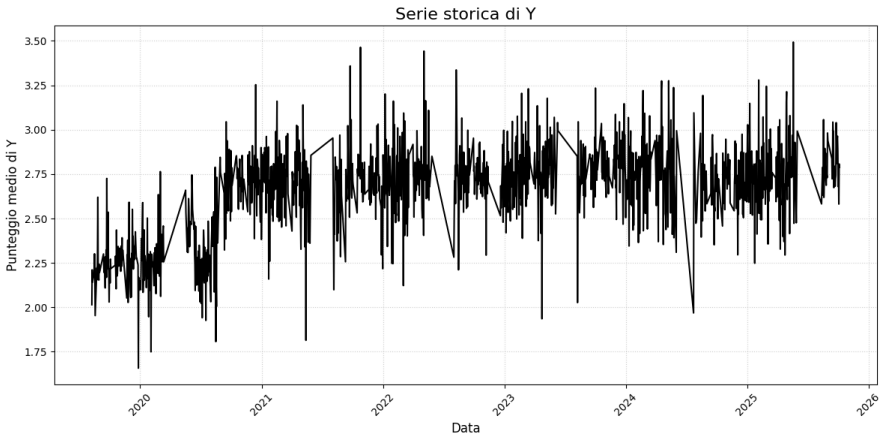

Figura 4.1: Serie storica della variabie Y

stati standardizzati utilizzando il MinMaxScaling (Pedregosa et al. 2011).
Per ottenere la variabile necessaria per il ranking, è stata elaborata una metrica di valutazione della singola prestazione. Tale indicatore deriva dai punteggi ottenuti in ogni match per ciascuna area tecnica individuata. La selezione delle aree e dei relativi pesi è il risultato di un'analisi statistica integrata dal contributo professionale dei preparatori dei portieri di uno dei settori giovanili d'élite del calcio italiano. Nello specifico, la variabile dipendente (Y_i) del modello bayesiano è costituita dalla media ponderata delle seguenti tre aree: Azioni Difensive (35%), Distribuzione palla (45%) e Capacità di parata (20%) (Figura 4.1). Inoltre, al fine di strutturare correttamente i dati per il modello autoregressivo, sono stati introdotti due indici identificativi fondamentali. Il primo, denominato gk_id (che sarà l'indice j del modello), rappresenta un identificativo numerico univoco assegnato a ogni portiere nel dataset. La creazione di questo indice è necessaria per permettere al modello statistico (implementato in Stan) di isolare i parametri specifici di ogni atleta e tracciarne l'evoluzione individuale rispetto alla media globale. Il secondo, definito pos_idx, funge da contatore sequenziale delle partite giocate da ciascun portiere all'interno di una singola stagione. Questo indice permette di ordinare cronologicamente le prestazioni, garantendo che l'influenza del passato sulla prestazione presente sia calcolata seguendo l'effettiva progressione temporale delle partite disputate, indipendentemente dalla data solare assoluta. Questo indice sarà azzerato alla fine di ogni stagione in modo da non mettere in correlazione le partite finali di una stagione con quelle iniziali della successiva.

4.3 Costruzione modello

Per ogni osservazione $i \in \{1, \ldots, N\}$, la performance misurata y_i è modellata come:

$$y_i \sim \text{Normal}(\mu_i, \sigma) \tag{4.4}$$

Dove la media μ_i è composta da:

$$\mu_i = \beta_j + \sum_{k=1}^{K} X_{i,k}\gamma_k + f_i \tag{4.5}$$

Dove:

- β_j: Intercetta casuale (Random Effect) specifica per il portiere j.
- $X_{i,k}\gamma_k$: Effetti fissi delle covariate (es. statistiche di squadra). γ è il vettore dei coefficienti.
- f_i: Processo latente dinamico. Rappresenta la variazione di forma del portiere nel tempo.
- σ: Rumore di osservazione. Cattura l'errore di misura e la varianza non spiegata dai predittori.

Per rappresentare l'evoluzione del termine f_i, che rappresenta le prestazioni del portiere al variare del tempo, si implementa un processo autoregressivo di ordine 1. Processo che riparte da zero ad ogni nuovo inizio di stagione. In questo modo si evita di avere correlazioni tra la fine di una stagione e l'inizio della successiva. Di conseguenza, per definizione di modello autoregressivo di ordine 1, se è la prima partita della stagione si ha:

$$f_i \sim \text{Normal}\left(0, \frac{\sigma_w}{\sqrt{1-\alpha^2}}\right) \tag{4.6}$$

Per le partite successive:

$$f_i = \alpha \cdot f_{i-1} + \epsilon_n, \quad \epsilon_i \sim \text{Normal}(0, \sigma_w) \tag{4.7}$$

Dove: α rappresenta il coefficiente autoregressivo, e, σ_w il rumore del processo, ovvero, la variazione di performance tra una partita e l'altra. Se $\alpha \to 0$, la forma è puramente casuale; se $|\alpha| \to 1$, la forma è altamente correlata al passato.

Per completare il modello bayesiano gerarchico vanno definite le prior che indicano la conoscenza che si ha dei dati prima della loro osservazione. Per le deviazioni standard è stata scelta una distribuzione esponenziale di parametro 1 ($\sigma \sim$ Exponential(1), $\sigma_w \sim$ Exponential(1)) poichè quest'ultima impone la

positività ($\sigma > 0$) e scoraggia varianze eccessivamente alte, che potrebbero portare a overfitting. Definite le prior per le deviazioni standard, si passa alla prior per il coefficiente autoregressivo; per il quale è stata scelta una distribuzione Beta, che allontana il coefficiente dai valori estremi del dominio su cui è definito [-1,1]. Ciò è utile, poichè per $|\alpha| \approx 1$ il modello diventa instabile. Inoltre, viene effettuata una trasformazione che mappa il dominio da [-1,1] in [0,1], necessaria per la compatibilità con la distribuzione Beta, definita solo per valori positivi ($\frac{\alpha+1}{2} \sim \text{Beta}(2,2)$). Per le intercette e le covariate sono state scelte delle normali poco informative (per dati normalizzati una deviazione standard di 5 è considerata ampia). Questa formulazione permette ai coefficienti di muoversi liberamente, ma agisce come un freno contro l'overfitting ($\beta_j \sim \text{Normal}(0,5) \quad \forall j \in \{1,\ldots,J\}$, $\gamma_k \sim \text{Normal}(0,5) \quad \forall k \in \{1,\ldots,K\}$). Infine, per completare il quadro delle prior è stata definita la prior della variabile f. In questo caso, per evitare il problema del Neal's Funnel, è stata utilizzata una parametrizzazione che non definisce f come dipendente direttamente da σ_w. Poichè, se la definissimo in questo modo, per valori di σ_w molto piccoli, lo spazio dei parametri diventerebbe estremamente stretto e "appuntito", causando errori nel campionamento (Betancourt and Girolami 2015). Quindi definiamo:

$$f_{raw} \sim \text{Normal}(0,1). \tag{4.8}$$

Definita f_{raw} la si riscala ($f = f_{raw} * \sigma_w$), distendendo lo spazio geometrico. Questo permette all'algoritmo di muoversi molto più velocemente e senza intoppi, garantendo affidabilità nei risultati.

4.3.1 Creazione ranking e rappresentazione curve prestazionali

Il modello definito permette di rappresentare graficamente la curva rappresentativa della carriera di ogni singolo portiere all'interno del dataset. La curva è composta dalla somme della componente $\boldsymbol{\beta}$: Che contiene il "valore intrinseco" dei portieri; e dalla componente f: che rappresenta la performance di ogni portiere nel tempo estratta dai dati grezzi. Infine per creare il ranking è stato sfruttato il calcolo dell'area sotto la curva (AUC) ottenuta per ogni portiere. Sono stati definiti due tipi di ranking: il primo, basato sull'AUC medio. Il secondo, sull'AUC totale.

4.4 Risultati

4.4.1 Convalida del modello

Definito il modello nella sua completezza, per far si che i risultati ottenuti abbiano una validità statistica, c'è bisogno che il processo bayesiano alla base del modello

arrivi a convergenza. Il modello è stato testato con diverse combinazioni di covariate, e, si è visto che all'aumentare di queste il modello perdeva la capacità di estrarre campioni indipendenti. Di conseguenza, l'indagine si è concentrata su combinazioni con un basso numero di covariate. Le covariate, nel caso specifico, sono tutte le statistiche legate alla performance del portiere. Tra tutti i tentativi (Tabella 4.1) è risultato emergere, con un WAIC pari a 146273 (rispetto a un WAIC del modello base di circa 19200), il modello con le seguenti due covariate:

- npot psxg faced: Valore di gol attesi post-shot affrontati da un portiere (non considerando i rigori).
- forward passes: Passaggi fatti in avanti nella singola partita.

Questo vuol dire che un portiere con alti valori di queste statistiche aiuta il modello a spiegare perché la sua performance è più stabile nel tempo.
Oltre ad avere un ottimo valore del WAIC, questo modello, come si evince dai grafici delle distribuzioni a posteriori dei parametri (Figure 4.3, 4.4, 4.5, 4.6, 4.7), arriva a convergenza. Infatti, tutti i grafici ottenuti hanno la classica forma di quando il campionamento è andato a buon fine. Inoltre, dal fatto che, il valore di σ_w è molto piccolo rispetto a quello di σ, si deduce un'evoluzione della valutazione del portiere molto fluida e prevedibile. Osservando, infine, il grafico di autocorrelazione (Figura 4.2), si nota come il modello riesca ad ottenere campioni indipendenti facilmente.

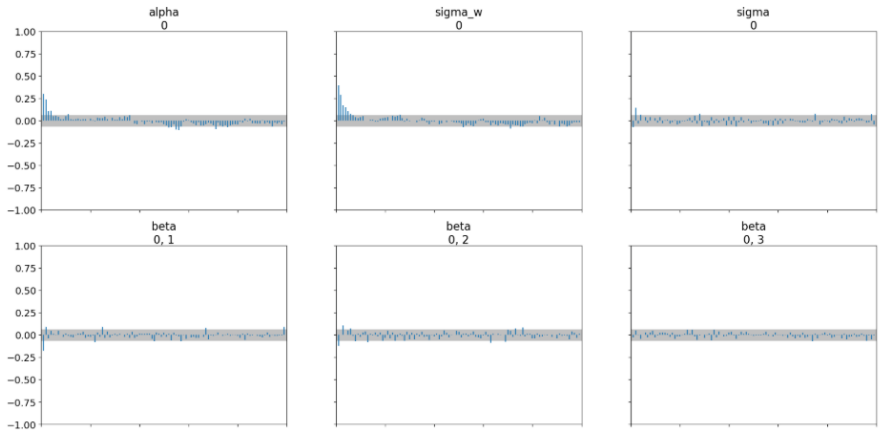

Figura 4.2: Autocorrelation plot.

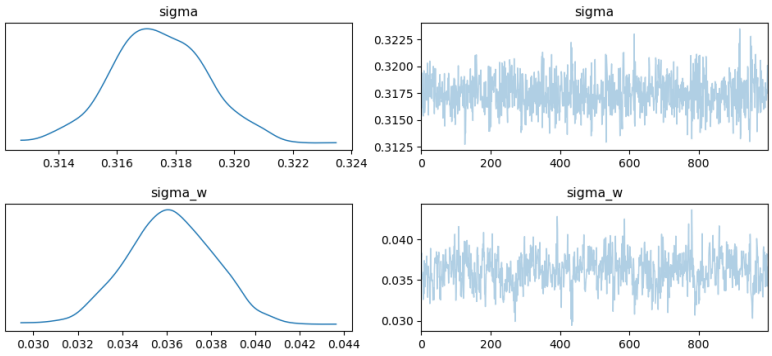

Figura 4.3: Trace plot σ e σ_w.

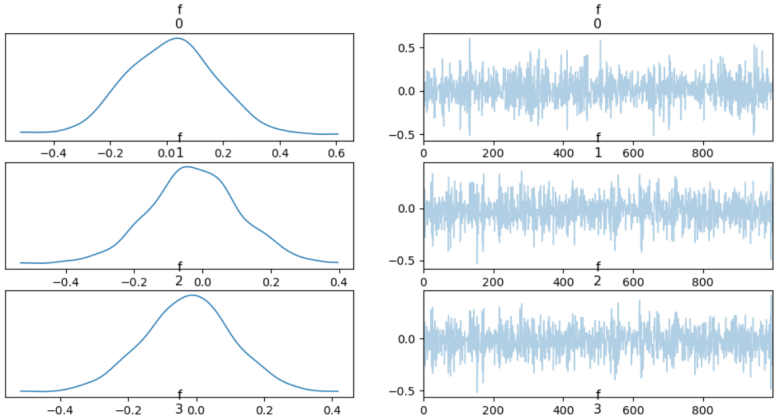

Figura 4.4: Trace plot f.

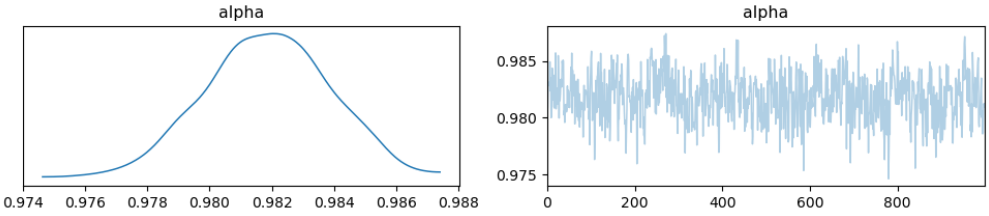

Figura 4.5: Trace plot α

4.4.2 Ranking

Una volta identificato il modello migliore si è passati allo sviluppo concreto del ranking, quest'ultimo è stato creato in due modi diversi. Il primo, basandosi sulle statistiche rilevanti (e il loro peso relativo) estratte dal miglior modello di classificazione. Il secondo, basandosi sull'area sottostante la curva prestazionale

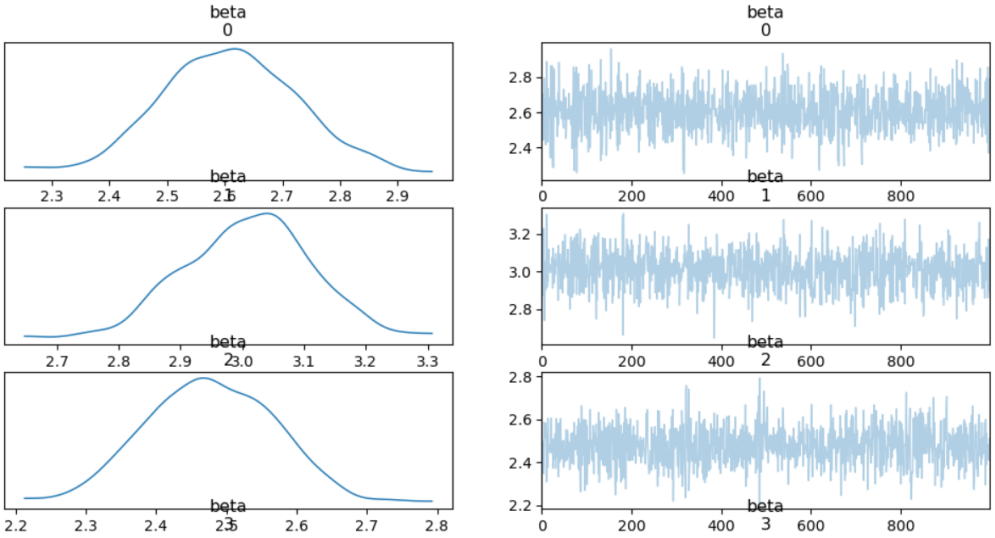

Figura 4.6: Trace plot β.

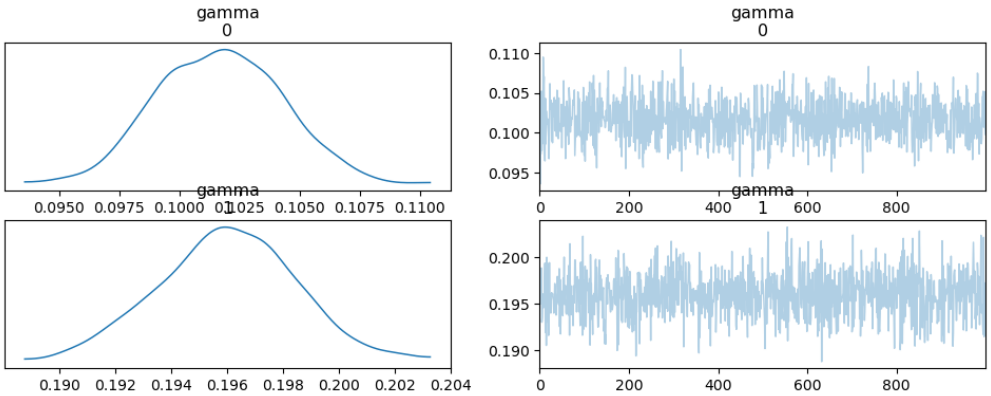

Figura 4.7: Trace plot γ.

ottenuta dal processo Bayesiano.
Ciò che accomuna le due classificazioni è la valutazione della performance singola, in quanto, in entrambi i casi, il valore di quest'ultima è stato ottenuto sommando le statistiche standardizzate ottenute nella singola partita. Le statistiche sono state divise in 3 aree fondamentali: Distribuzione palla, azioni difensive e capacità di parata.
Il punteggio delle varie aree è stato sommato, utilizzando, nei due casi pesi diversi. Nel primo caso, i pesi sono stati estratti direttamente dalla classificazione (Tabella 3.2). Nel secondo, i pesi delle aree sono stati definiti tramite un confronto con alcuni dei preparatori dei portieri del settore giovanile di uno dei club italiani

Statistica 1	Statistica 2	WAIC
gsaa		19262
Gsaa	Through balls	19001
Gsaa	Positioning errors	18984
Xg buildup		19877
Games played		NO CONVERGENZA
Xg buildup	Partite giocate	NO CONVERGENZA
Xg buildup	Gsaa	17880
Gk impact		19636
Gk impact	Xg buildup	18238
Line-breaking passes		20626
Line-breaking passes	Fwd passes	19274
Aggressive actions		20329
Aggressive actions	Line-breaking passes	19675
Lbp to space 10	Npot psxg faced	17852
Npot psxg faced	Fwd passes	14654
Tackles		21182
Tackles	Interceptions	21186
Shots blocked		21244

Tabella 4.1: WAIC ottenuti con diverse covariate

professionistici d'élite, ottenendo la seguente partizione: Distribuzione palla:45%; Capacità di parata: 20%; Azioni difensive e posizionamento: 35%.

Ranking ottenuto dalla classificazione

Il primo ranking estratto si basa sul punteggio medio ottenuto in tutte le partite. Come si può notare nella top 10 (Tabella 4.2), non c'è nessuno dei nomi che ci si aspetterebbe, portieri come Oblak (192°), Donnarumma (181°) e Courtois (118°) popolano i fondali della classifica. Il primo nome "noto" è quello di Manuel Neuer, che si posiziona 11-esimo. Questo accade principalmente a causa di tre fattori principali:

- Il punteggio medio favorisce portieri con meno partite. In quanto mantenere un certo standard di performance è molto più difficile nel lungo termine.

- Valutando la prestazione singola, si avvantaggiano i portieri di squadre minori (Liu et al. 2015). Militando in squadre meno attrezzate i portieri tendono ad essere chiamati in causa maggiormente e ad avere statistiche migliori.

Nome	Partite giocate	Squadra
Manuel Riemann	104	Bochum
Alvaro Valles	41	Las Palmas/Real Betis
Stefan Ortega	122	Arminia Belefeld/Man.City
Matias Dituro	44	Celta Vigo/Elche
Vanja Milinkovic-Savic	153	Torino/Napoli
Noah Atubolu	82	Friburgo
Arijanet Muric	40	Burnley/Ispwich/Sassuolo
Jean Butez	44	Royal Antwerp/Como
Mark Flekken	177	B.Monchenglabach/Brentford/B.Leverkusen
Mory Diaw	75	Clermont Foot/Le Havre

Tabella 4.2: Top 10 per punteggio medio ottenuto

- Il modello di classificazione attribuisce un peso molto elevato alla distribuzione palla.

Infatti, analizzando i nomi, si hanno quasi tutti portieri che hanno giocato poco (o relativamente poco), lo hanno fatto in squadre minori e sono globalmente riconosciuti come degli ottimi distributori di palla.

Nome	Partite giocate	Squadra
Alex Remiro	276	Real Sociedad
Yann Sommer	248	B.Monchenglabach/Inter/Bayern Monaco
Ederson	265	Man.City/Fenerbache
Alisson	251	Liverpool
Jan Oblak	289	Atletico Madrid
Jordan Pickford	236	Everton
Mike Maignan	252	Lille/Milan
Lucas Hradecky	239	B.Leverkusen/Monaco
Thibaut Courtois	245	Real Madrid
Manuel Neuer	213	Bayern Monaco

Tabella 4.3: Top 10 basata sulla shrunk mean

Per provare a migliorare questo ranking il valore medio puro è stato sostituito da una "shrunk mean", definita come segue:

$$\text{shrunk mean} = \frac{\alpha \cdot \mu + \text{Partite} \cdot \text{Punteggio}}{\alpha + \text{Partite}} \tag{4.9}$$

dove:

- μ : La media del punteggio del singolo portiere (calcolata come Punteggio/Partite)¦
- α : È il "peso della fiducia". Più è alto questo valore, più il portiere ha bisogno di giocare molte partite per far sì che la sua media individuale elimini l'influenza della correzione statistica. Questo valore è stato impostato a 25;
- Partite: Il numero di match giocati dal portiere;
- Punteggio: La somma totale dei punti accumulati in tutte le partite.

Così facendo si ottiene un ranking che sfavorisce chi ha giocato poche partite, dando più valore alla costanza nel tempo. In questo caso la top 10 (Tabella 4.3), contiene molti dei nomi universalmente riconosciuti come i migliori al mondo. Portieri come Alisson, Oblak, Courtois e Neuer hanno rappresentato il livello massimo del ruolo negli anni in analisi. Nonostante il miglioramento, le criticità espresse in precedenza rimangono in quanto intrinseche all'analisi. Infatti si è passati da un ranking che favoriva chi avesse registrato un minor numero di partite ad uno che favorisce chi ne ha giocate tante, senza intaccare il peso della squadra di appartenenza.

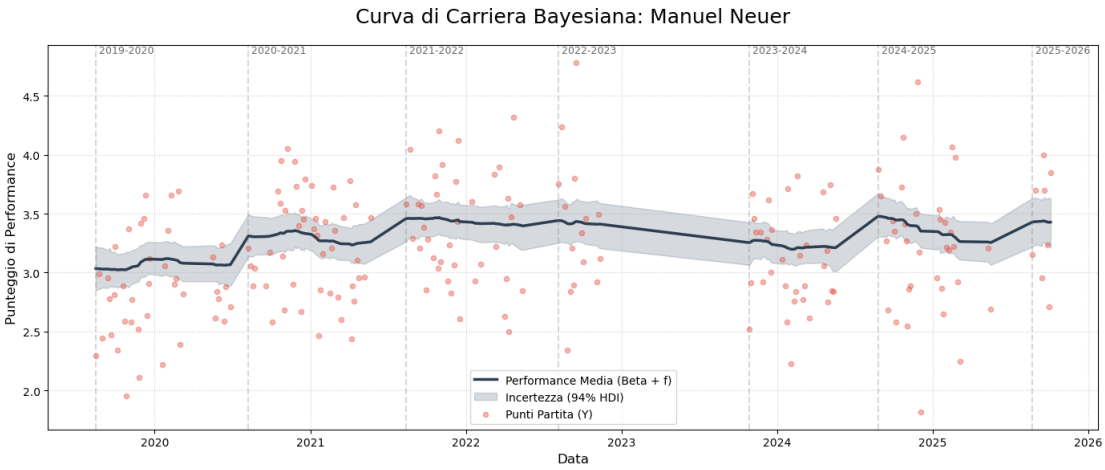

Figura 4.8: Curva prestazionale Manuel Neuer

Nome	Partite giocate	Squadra
Manuel Riemann	104	Bochum
Alvaro Valles	41	Las Palmas/Real Betis
Stefan Ortega	122	Arminia Belefeld/Man.City
Manuel Neuer	213	Bayern Monaco
Jean Butez	44	Royal Antwerp/Como
Matias Dituro	44	Celta Vigo/Elche
Noah Atubolu	82	Friburgo
Mark Flekken	177	B.Monchenglabach/Brentford/B.Leverkusen
Michael Zetterer	70	Werder Bremen/Eintracht Frankfurt
Vanja Milinkovic-Savic	153	Torino/Napoli

Tabella 4.4: Top 10 per AUC medio.

Ranking Bayesiano

Il modello bayesiano costruito ha generato, per ogni portiere, la curva prestazionale negli anni disponibili nel dataset (es. figura 4.8). Da questa curva, calcolando l'area sottostante, sono stati ricavati altri due ranking. Il primo si basa sull'AUC (Area Under the Curve) medio, quindi come per il valore medio, favorisce portieri che hanno giocato poco. Infatti, come si evince dalla top 10 (Tabella 4.4), i portieri sono pressochè gli stessi del ranking basato sul valore della prestazione medio. Le uniche differenze sono rappresentate da Neuer e Zetterer che comunque si posizionavano poco fuori la top 10. Anche in questo caso, il vantaggio statistico dei portieri di squadre minori rimane, così come, il peso della capacità di distribuire palla.

Nome	Partite giocate	Squadra
Alisson	251	Roma/Liverpool
Yann Sommer	248	B.Monchenglabach/Inter/Bayern Monaco
Marc-Andrè ter Stegen	227	Barcellona
Jordan Pickford	236	Everton
Ederson	265	Man.City
Lucas Hradecky	239	B.Leverkusen/Monaco
Oliver Baumann	218	Hoffenheim
Emiliano Martinez	228	Arsenal/Aston Villa
Sergio Herrera	192	Osasuna
Lukasz Skorupski	217	Bologna

Tabella 4.5: Top 10 per AUC totale.

Per provare a migliorare questo ranking, dunque, si sostituisce l'AUC medio con l'AUC totale. Attuando una modifica simile a quella fatta per la classificazione, in cui si cerca di individuare il valore rappresentato dal portiere per la squadra negli anni in esame. Come nel caso precedente, compaiono in top 10 (Tabella 4.5), diversi dei portieri ritenuti migliori al mondo: Alisson, ter-Stegen, Ederson. Neuer è 11-esimo. Poco fuori dalla top 10, ci sono anche Maignan e Donnarumma. Il ranking ottenuto, dipende di meno dal numero di partite giocate , e per definizione dei pesi, dipende di meno anche dalla capacità di giocare con i piedi. Nonostante questi leggeri miglioramenti, il peso di militare in squadre minori rimane, facendo comparire nella top 10 portieri come Baumann, Herrera e Skorupski che, per quanto buoni portieri, non hanno mai rappresentato il top del ruolo,

Ranking Bayesiano normalizzato

Visti i risultati ottenuti si è provato a migliorare ulteriormente i ranking inserendo dei fattori di normalizzazione legati alla squadra di appartenenza del portiere. Le squadre sono state divise in tre fasce (Alta, Media, Bassa), inserendo nella fascia alta le squadre che nei 5 anni hanno raggiunto piazzamenti di rilievo (1°-4°) nei rispettivi campionati nazionali, per la fascia media sono stati considerati i piazzamenti dal (4°-8°). Per evitare di inserire squadre che avessero performato solo per un anno, la scelta è stata anche basata sul rendimento complessivo dei 5 anni. Definite le fasce, per ognuna di essa e per ogni area caratteristica considerata (Distribuzione palla, Azioni difensive e Capacità di parata), sono stati individuati dei coefficienti. Per la definizione di questi coefficienti si è fatto riferimento a Liu et al. 2015, estraendo i dati medi dal Table 1 del paper. Da questo studio si evince che portieri delle squadre di fascia bassa sono più sollecitati (fanno più parate) ma

Area	Fascia alta	Fascia media	Fascia bassa
Distribuzione palla	1.00	1.07	1.17
Capacità di parata	1.00	0.78	0.85
Azioni difensive	1.00	0.71	0.69

Tabella 4.6: Coefficienti di area per fascia.

Nome	Partite giocate	Squadra
Manuel Neuer	213	Bayern Monaco
Alvaro Valles	41	Las Palmas/Real Betis
Manuel Riemann	104	Bochum
Stefan Ortega	122	Arminia Belefeld/Man.City
Marc-Andrè ter Stegen	227	Barcellona
Jason Steele	57	Brigthon & Hove Albion
Janis Blaswich	70	Lipsia/Bayern Leverkusen
Alisson	251	Roma/Liverpool
Jean Butez	44	Royal Antwerp/Como
Yann Sommer	248	B.Monchenglabach/Inter/Bayern Monaco

Tabella 4.7: Top 10 per AUC medio con valori normalizzati.

hanno meno opzioni di passaggio pulite (minore precisione). Ottenuti i coefficienti in tabella 4.6, sono stati moltiplicati per le aree a cui fanno riferimento, in modo tale da ottenere un nuovo punteggio. Ottenuto il punteggio è stato effettuato lo stesso procedimento bayesiano esposto nel capitolo 4.

Come nelle sezioni precedenti, si parte dalla valutazione della top 10 basata sull'AUC medio (tabella 4.7). Appaiono in classifica Neuer, ter-Stegen e Alisson, tre tra i migliori portieri al mondo nel periodo considerato. Ma nonostante la regolarizzazione effettuata, la combo portiere bravo nella distribuzione palla che milita in squadre minori garantisce la presenza ai vertici di questa classifica.

Anche in questo caso, per valutare l'apporto totale del portiere alla squadra nel periodo in questione si è passati dall'AUC medio, all'AUC totale (tabella 4.8). I nomi sono simili a quelli della tabella 4.5 ottenuta utilizzando l'AUC totale senza l'applicazione dei coefficienti. Anche se nella top 10, le differenze non sono molte, cambia molto lo scenario al di fuori di essa, con altri portieri di livello assoluto che si avvicinano al top: Donnarumma (11°), Maignan(15°), Oblak(22°) e Courtois(25°).

Infine per provare a bilanciare l'AUC medio e l'AUC totale, è stato definito un ranking utilizzando l'AUC medio "aggiustato", in modo simile alla shrunk mean

Nome	Partite giocate	Squadra
Alisson	251	Roma/Liverpool
Ederson	265	Man.City
Marc-Andrè ter Stegen	227	Barcellona
Yann Sommer	248	B.Monchenglabach/Inter/Bayern Monaco
Emiliano Martinez	228	Arsenal/Aston Villa
Lukasz Skorupski	217	Bologna
Alex Meret	194	Napoli
Lucas Hradecky	239	B.Leverkusen/Monaco
Jordan Pickford	236	Everton
Manuel Neuer	213	Bayern Monaco

Tabella 4.8: Top 10 per AUC totale normalizzato.

definita in precedenza si ha:

$$MeanAUC_{adj} = \frac{v \cdot \mu + C \cdot M}{v + C} \tag{4.10}$$

dove:

- ν = Numero di osservazioni (partite giocate) per il singolo portiere;
- μ = Media aritmetica delle prestazioni del singolo portiere;
- M = Valore medio di tutti i portieri presenti nel dataset;
- C = È il parametro di "forza" del sistema. Matematicamente, rappresenta il numero di partite "virtuali" con prestazione pari alla media globale che si aggiungono ad ogni giocatore.

In questo modo si è ottenuta la top 10 in tabella 4.9. Con questa metrica, soprattutto all'aumentare di C, i portieri che hanno giocato tanto ottengono ottimi piazzamenti. A differenza della tabella basata sulla shrunk mean (tabella 4.3), compaiono in top 10, quasi tutti i migliori portieri. Come in precedenza, altri portieri considerati di altissimo livello sono subito fuori dalle prime dieci posizioni: Martinez(16°), Donnarumma(23°).

Per quanto si possa provare ad identificare una metrica capace di rappresentare la realtà, questa avrà al suo interno sempre delle assunzioni che ne influenzano decisamente i risultati. Identificare un criterio che sia capace di considerare tutte le variabili in gioco è molto complesso. Nonostante tutte le variabili considerate il ranking non risulta precisissimo. Infatti, se si chiedesse a qualsiasi esperto del settore di creare una top 10 dei portieri nel mondo negli anni in esame, difficilmente

Nome	**Partite giocate**	**Squadra**
Manuel Neuer	213	Bayern Monaco
Alisson	251	Roma/Liverpool
Yann Sommer	248	B.Monchenglabach/Inter/Bayern Monaco
Marc-Andrè ter Stegen	227	Barcellona
Ederson	265	Man.City
Mike Maignan	252	Lille/Milan
Ivan Provedel	182	Lazio
Alex Remiro	276	Real Sociedad
Lucas Hradecky	239	B.Leverkusen/Monaco
Thibaut Courtois	245	Real Madrid

Tabella 4.9: Top 10 per AUC medio aggiustato e normalizzato.

comparirebbero nomi come Provedel, Remiro o Hradecky. Alcuni portieri come Donnarumma e Oblak, subiscono molto la loro relativamente scarsa capacità di gestire la palla con i piedi (Figura 4.9), e questo non gli permette di comparire tra le prime posizioni. Un altro esempio potrebbe essere il confronto tra Svilar e Butez, due portieri in rampa di lancio nel campionato italiano. Il primo, a differenza del secondo non compare in nessuna top 10 per cause simili a quelle che non fanno comparire Donnarumma e Oblak.

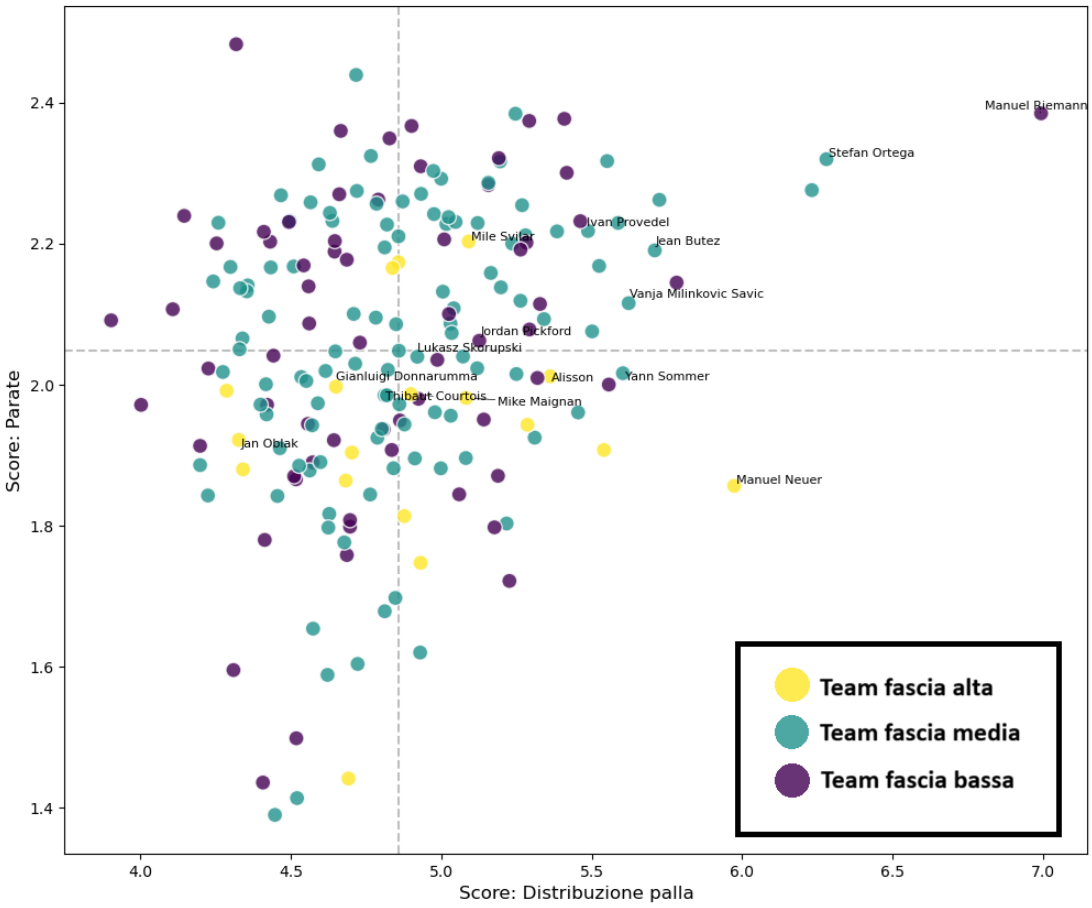

Figura 4.9: Scatter plot Distribuzione palla vs Capacità di parata

Capitolo 5

Conclusioni

Lo scopo di questo elaborato era quello di provare a creare un ulteriore ponte tra la statistica e il calcio. Nella prima parte, in cui è stato addestrato un modello di classificazione capace di riconoscere le caratteristiche di un portiere d'élite, sono stati ottenuti dei buoni risultati. L'individuazione di queste caratteristiche può tornare utile in fase di scouting, permettendo di avere una prima idea, basata sui dati, dei profili da esaminare. Oltre che in fase di scouting, può essere utile nella formazione dei ragazzi, esaminando i profili degli stessi ed individuando le aree in cui sono carenti, creando allenamenti ad hoc per migliorarli in fondamentali specifici. Ovviamente l'utilità di questi mezzi proposti è collegata al connubio con il professionista, in quanto, non si possono sostituire integralmente ad esso ma possono sicuramente fornire un ottimo supporto alle decisioni. Nella seconda parte è stato utilizzato un approccio innovativo al calcolo del valore delle carriere dei portieri. In questo approccio sono state riscontrate diverse problematiche, una su tutte, la difficoltà di racchiudere in una sola metrica tutte le variabili che determinano il valore di un portiere. Nonostante ciò sono stati ottenuti discreti risultati. Entrambe le parti sono sicuramente migliorabili: la parte di classificazione, se arricchita con ulteriori dati potrebbe essere capace di discernere con maggiore precisione le caratteristiche dei portieri di alto livello; la parte di creazione del ranking potrebbe essere migliorata provando a modificare ulteriormente i pesi delle statistiche e dei coefficienti in questione, fino a trovare una configurazione ottimale. Inoltre, si potrebbe inserire una valutazione della prestazione in base al livello dell'avversario. In ogni caso, alcune variabili rimarrebbero sicuramente fuori. La capacità dei calciatori di alto livello non sta solo nel quanto bene fanno, ma anche, e forse soprattutto, nel quando lo fanno. Ci sono delle parate, che in momenti particolari di partite importanti, pesano più che una parata in una "normale" partita di campionato. Infine, come già detto, per quanto si possa provare a rappresentare la realtà utilizzando una metrica, questa risulterà sempre riduttiva e ci sarà sempre qualche portiere "fuori posto". Al di là della creazione del ranking, questo elaborato

vuole dimostrare come i dati possano essere utili e fuorvianti allo stesso tempo. Una classifica, una metrica o qualsiasi altro dato è valido all'interno di un contesto specifico e con precise assunzioni, se non si scende in profondità interpretando questi fattori, il dato rimarrà un numero incapace di descrivere la realtà.

Bibliografia

Botev, Zdravko, Dirk P Kroese, Thomas Taimre, and Radislav Vaisman (2019). *Data science and machine learning: mathematical and statistical methods*. Chapman and Hall/CRC (cit. on pp. 8–10).

Piyadasa, Tharinda Dilshan and Kasun Gunawardana (2023). "A review on oversampling techniques for solving the data imbalance problem in classification". In: *The International Journal on Advances in ICT for Emerging Regions* 16.1 (cit. on p. 11).

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on pp. 14, 15, 27).

Darst, Burcu F, Kristen C Malecki, and Corinne D Engelman (2018). "Using recursive feature elimination in random forest to account for correlated variables in high dimensional data". In: *BMC genetics* 19.Suppl 1, p. 65 (cit. on p. 14).

Lemaître, Guillaume, Fernando Nogueira, and Christos K. Aridas (2017). "Imbalanced-learn: A Python Toolbox to Tackle the Curated List of Selection Bias in Imbalanced Data Sets". In: *Journal of Machine Learning Research* 18.17, pp. 1–5. URL: `http://jmlr.org/papers/v18/16-365.html` (cit. on p. 15).

Betancourt, Michael and Mark Girolami (2015). "Hamiltonian Monte Carlo for hierarchical models". In: *Current trends in Bayesian methodology with applications* 79.30, pp. 2–4 (cit. on p. 29).

Liu, Hongyou, Miguel A Gómez, and Carlos Lago-Peñas (2015). "Match performance profiles of goalkeepers of elite football teams". In: *International Journal of Sports Science & Coaching* 10.4, pp. 669–682 (cit. on pp. 33, 37).

Jamil, Mikael, Ashwin Phatak, Saumya Mehta, Marco Beato, Daniel Memmert, and Mark Connor (2021). "Using multiple machine learning algorithms to classify elite and sub-elite goalkeepers in professional men's football". In: *Scientific reports* 11.1, p. 22703.

Pappalardo, Luca, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti (2019). "PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.5, pp. 1–27.

Spyropoulou, M, JG Hopker, and JE Griffin (2024). "Modelling between-and within-season trajectories in elite athletic performance data". In: *arXiv preprint arXiv:2405.17214*.

Griffin, Jim, Maria-Zafeiria Spyropoulou, and James Hopker (2026). "Modelling between-and within-season trajectories in elite athletic performance data". In: *The Journal of the Royal Statistical Society, Series C (Applied Statistics)*.

Griffin, Jim E, Laurenţiu C Hinoveanu, and James G Hopker (2022). "Bayesian modelling of elite sporting performance with large databases". In: *Journal of Quantitative Analysis in Sports* 18.4, pp. 253–268.

Wolf, Stephan, Maximilian Schmitt, and Björn Schuller (2021). "A football player rating system". In: *Journal of Sports Analytics* 6.4, pp. 243–257.

Alvo, Mayer and LH Philip (2014). *Statistical methods for ranking data*. Vol. 1341. Springer.