Dipartimento di informatica
Università degli Studi di Salerno
DIPARTIMENTO DI ECCELLENZA

ANALISI PREDITTIVA DEGLI INFORTUNI SPORTIVI

Report Tecnico - Corso di Machine Learning

Data:
20 Gennaio 2026

Destinatari:
Prof. Giuseppe Polese
Prof.ssa Loredana Caruccio

Realizzato da:
Francesco Zambrino

Matricola:
0512119156

Contents

1 TASK ANALIZZATO

1.1 Obiettivo del Progetto

Il progetto affronta il problema della salute degli atleti attraverso un compito di classificazione binaria supervisionata. L'obiettivo è predire se un calciatore subirà un infortunio muscolare nella stagione successiva basandosi su dati storici, biometrici e di carico di lavoro.
Nel contesto della medicina sportiva, il valore aggiunto del modello risiede nella sua capacità di agire come sistema di *Early Warning*, permettendo allo staff tecnico di personalizzare i carichi di allenamento e ridurre i tempi di indisponibilità nel team. L'obiettivo è spostare l'orizzonte della medicina sportiva da un approccio reattivo a un approccio proattivo e anticipatorio. L'utilizzo di algoritmi di Machine Learning risponde a tre esigenze fondamentali:

- **Identificazione di pattern invisibili:** Il modello correla simultaneamente decine di variabili, identificando segnali premonitori impercettibili all'occhio umano.
- **Supporto alle decisioni cliniche:** Fornisce evidenze basate sui dati per giustificare la turnazione di un atleta o la riduzione dei carichi.
- **Standardizzazione della diagnosi precoce:** Elimina la soggettività, garantendo un monitoraggio scientifico per ogni atleta.

2 IL DATASET UTILIZZATO E LE SUE CARATTERISTICHE

2.1 Classificazione

Il dataset comprende 800 osservazioni di calciatori universitari e professionisti. Le informazioni sono strutturate in 18 variabili di input e una variabile target.

Variabili Input (18):

- **Dati personali e fisici:** `Height`, `Weight`, `BMI`, `Age`.
- **Performance atletiche:** `Sprint Speed`, `Agility Score`, `Knee Strength`, `Hamstring Flexibility`, `Reaction Time`, `Balance Score`.
- **Variabili comportamentali:** `Sleep Hours`, `Stress Levels`, `Nutrition Quality`, `Warmup Routine`.
- **Variabili storiche:** `Previous Injury`, `Matches Played`, `Training Hours`.

Variabile Target: `Injury Next Season` (Variabile binaria che indica la presenza o assenza di infortunio nella stagione successiva).

2.2 Analisi della tipologia

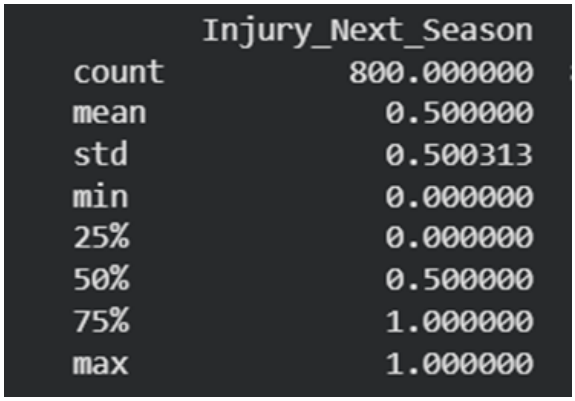
L'analisi tramite il comando `"df.info()"` ha rivelato una struttura dati caratterizzata da:

- **Variabili Numeriche:** La maggior parte degli attributi è di tipo numerico (`float64` e `int64`), permettendo l'applicazione di tecniche di standardizzazione.
- **Variabili Categoriche:** Il ruolo in campo (`Position`) è l'unica variabile categorica; verrà gestita attraverso processi di *One-Hot Encoding* per la corretta interpretazione da parte dei modelli lineari.

Attraverso l'analisi statistica condotta è stato possibile approfondire la natura della variabile target `Injury_Next_Season`. Sebbene registrata tecnicamente come variabile numerica, essa si configura a tutti gli effetti come una variabile binaria.
Il riscontro statistico fornito da `"df.describe()"`, che mostra valori minimi di 0 e massimi di 1, conferma che il target segue una logica booleana:

- **0:** indica l'assenza di infortunio.
- **1:** indica la presenza di infortunio.

Inoltre, come discusso nel paragrafo 2.3, essendo il dataset bilanciato secondo la variabile target ed avendo una media di 0.5, possiamo concludere che l'intuizione precedente è corretta.



	Injury_Next_Season
count	800.000000
mean	0.500000
std	0.500313
min	0.000000
25%	0.000000
50%	0.500000
75%	1.000000
max	1.000000

Figure 1: Statistiche descrittive della variabile target Injury_Next_Season.

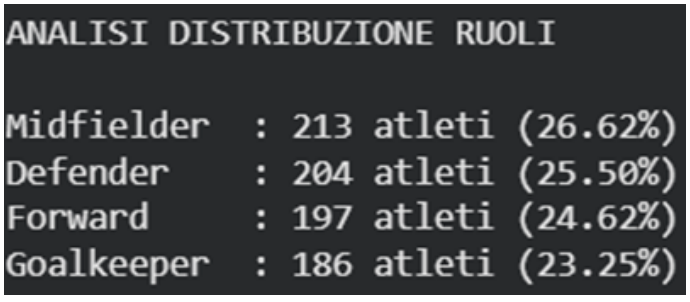
Questa distinzione tra una semplice variabile intera e una variabile binaria è fondamentale per la scelta dei modelli di classificazione utilizzati (*Logistic Regression* e *Random Forest*), che sono stati ottimizzati proprio per mappare le probabilità di appartenenza a queste due classi distinte.

2.3 Distribuzione e Bilanciamento

È stata condotta un'analisi relativa alla distribuzione dei valori dell'unica variabile categorica presente (`Position`). È stato riscontrato un buon bilanciamento tra le 4 classi presenti:

- Goalkeeper
- Defender
- Midfielder
- Forward

In particolare, l'analisi ha evidenziato il numero di istanze riportato nella Figura 2:



```
ANALISI DISTRIBUZIONE RUOLI

Midfielder  : 213 atleti (26.62%)
Defender    : 204 atleti (25.50%)
Forward     : 197 atleti (24.62%)
Goalkeeper  : 186 atleti (23.25%)
```

Figure 2: Distribuzione delle istanze per la variabile Position.

Il dataset presenta una distribuzione perfettamente bilanciata anche per la variabile target (50% per classe), condizione ideale che evita distorsioni verso la classe maggioritaria e non richiede tecniche di oversampling.

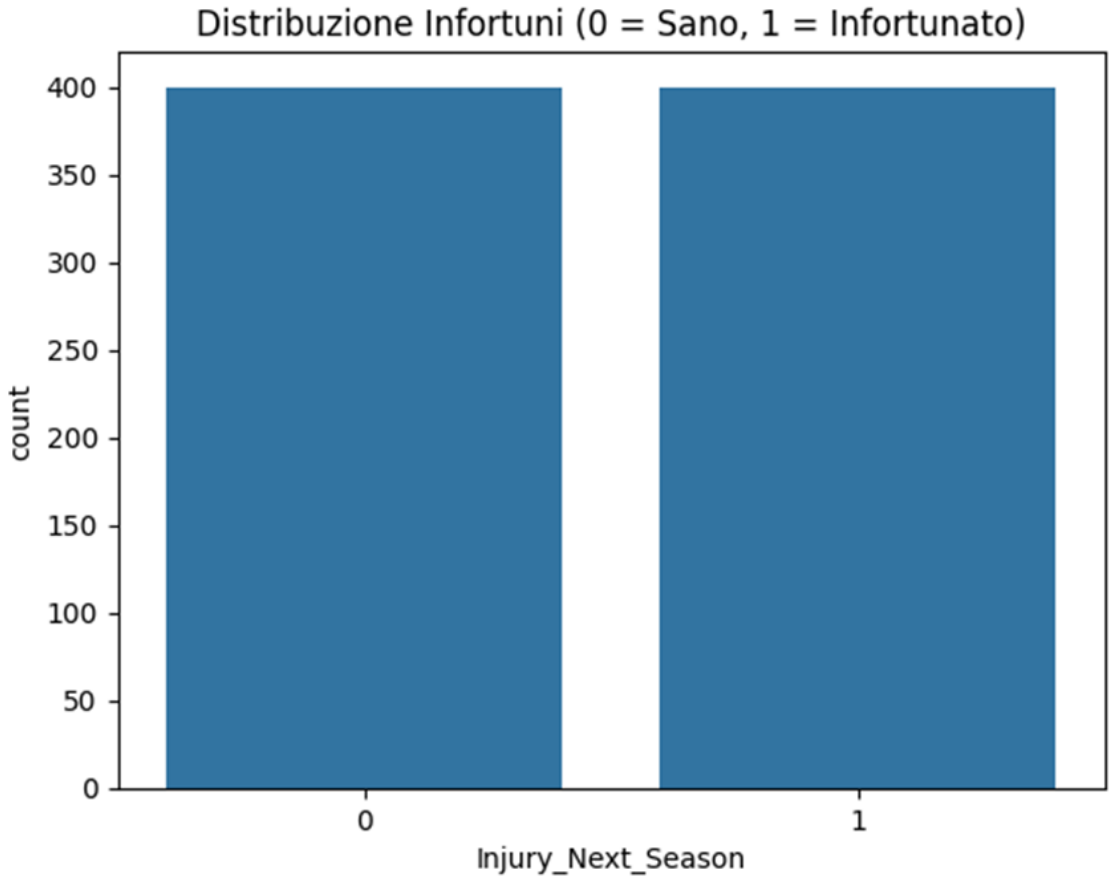


Figure 3: Distribuzione delle istanze per la variabile target.

2.4 Analisi Missing Values

Un punto di forza rilevante emerso da questa fase è la totale assenza di *Missing Values* in ogni colonna. Questa condizione di completezza del dataset è fondamentale per due ragioni:

- **Affidabilità Statistica:** Evita distorsioni derivanti da tecniche di imputazione che potrebbero introdurre rumore artificiale.
- **Qualità dell'Addestramento:** Garantisce che ogni singola istanza fornisca il massimo apporto informativo durante la fase di apprendimento dei modelli di *Logistic Regression* e *Random Forest*.

2.5 Analisi delle Ridondanze

Oltre alla verifica dei valori nulli, è stata eseguita un'analisi volta a identificare la presenza di record duplicati all'interno del dataset. Tramite l'utilizzo del comando Python `df.duplicated().sum()`, è stata confermata l'assenza di righe identiche.
Di seguito viene riportata l'analisi del dataset relativa ai valori nulli ed al tipo per ciascuna feature presente (Figura 4):

```
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         800 non-null    int64
 1   Height_cm                   800 non-null    int64
 2   Weight_kg                   800 non-null    int64
 3   Position                    800 non-null    object
 4   Training_Hours_Per_Week     800 non-null    float64
 5   Matches_Played_Past_Season  800 non-null    int64
 6   Previous_Injury_Count       800 non-null    int64
 7   Knee_Strength_Score         800 non-null    float64
 8   Hamstring_Flexibility       800 non-null    float64
 9   Reaction_Time_ms            800 non-null    float64
 10  Balance_Test_Score          800 non-null    float64
 11  Sprint_Speed_10m_s          800 non-null    float64
 12  Agility_Score               800 non-null    float64
 13  Sleep_Hours_Per_Night       800 non-null    float64
 14  Stress_Level_Score          800 non-null    float64
 15  Nutrition_Quality_Score     800 non-null    float64
 16  Warmup_Routine_Adherence    800 non-null    int64
 17  Injury_Next_Season          800 non-null    int64
 18  BMI                         800 non-null    float64
dtypes: float64(11), int64(7), object(1)
memory usage: 118.9+ KB

Numero di righe duplicate: 0
```

Figure 4: Riepilogo dei valori nulli e tipologia delle feature.

3 ISSUE ANALIZZATE E SCELTE DI PROGETTAZIONE

3.1 Rilevamento e Gestione degli Outlier

L'integrità del dataset è stata garantita attraverso un'analisi combinata di ispezione visiva (Box Plot) e rigore statistico attraverso il metodo IQR (Interquartile Range). Questa procedura ha permesso l'individuazione e la successiva rimozione di 17 campioni anomali.
A differenza di un approccio automatizzato su tutto il dataset, si è scelto di intervenire chirurgicamente solo su 4 variabili critiche, seguendo una logica di dominio specifica:

- `BMI` e `Sleep_Hours_Per_Night` **(Vincoli Biologici):** Queste variabili presentano confini fisiologici invalicabili. Valori estremi (es. ore di sonno negative o BMI incompatibili con l'attività atletica) sono stati trattati come errori di inserimento per non distorcere la profilazione fisica dell'atleta.
- `Training_Hours_Per_Week` **(Integrità dello Scaling):** Poiché le ore di allenamento sono soggette a standardizzazione (*StandardScaler*), i valori fuori scala avrebbero compresso la distribuzione dei dati reali verso lo zero, riducendo la sensibilità del modello.
- `Stress_Level_Score` **(Rilevanza per il Task):** Anomalie nel reporting dello stress sono state rimosse per assicurare che il modello impari solo da pattern di stress coerenti e realistici.

Questa "pulizia selettiva" è stata propedeutica alla fase successiva di *Feature Engineering*, assicurando che le nuove variabili sintetiche (come il *Fatigue Index*) venissero calcolate su basi solide.

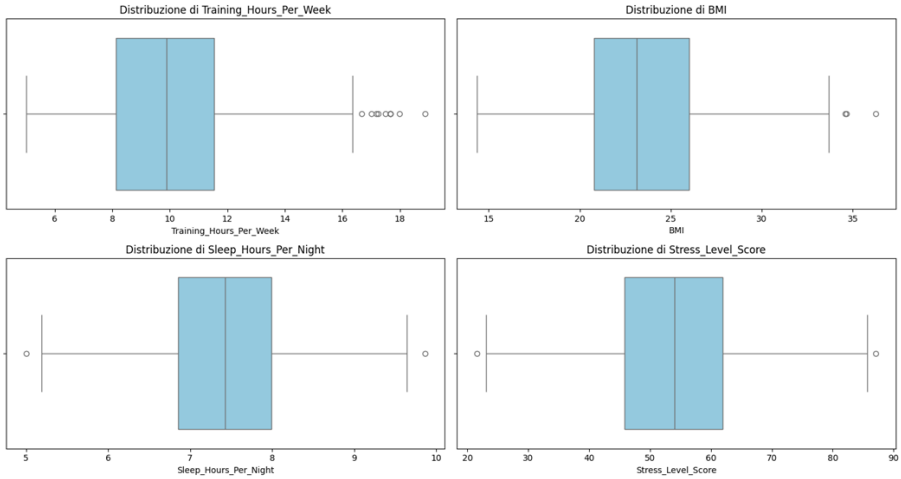


Figure 5: Distribuzione delle variabili critiche tramite Box Plot per l'individuazione degli outlier.

3.2 Feature Selection e Engineering

Per massimizzare la capacità predittiva dei modelli, non ci si è limitati all'utilizzo dei dati grezzi, ma è stata applicata una strategia di raffinamento delle variabili basata sulla conoscenza del dominio:

- **Rimozione della Ridondanza (Potatura del Peso):** È stata eliminata la variabile `Weight_kg`. L'analisi di correlazione iniziale mostrava un valore di 0.85 con il `BMI`, indicando che le due variabili fornivano quasi la stessa informazione. Rimuovere il peso riduce la multi-collinearità, rendendo i coefficienti dei modelli più stabili e affidabili.

Inoltre, sono state create tre nuove feature sintetiche per catturare pattern complessi:

- `Fatigue_Index`: Questa è la variabile più critica per il task. Dividendo lo Stress_Level_Score per le ore di sonno (Sleep_Hours_Per_Night + 1), abbiamo creato un indicatore che pesa lo stress psicofisico in relazione alla qualità del recupero. Un atleta con stress alto ma ottimo recupero risulterà meno a rischio di uno con stress medio ma poche ore di sonno.

$$\text{Fatigue Index} = \frac{\text{Stress Level Score}}{\text{Sleep Hours Per Night} + 1} \tag{1}$$

- `Total_Workload`: Ottenuta moltiplicando le ore di allenamento per le partite giocate, fornisce una misura dell'usura cumulativa. Questo valore cattura l'effetto "somma" dei carichi di lavoro che le singole variabili non esprimevano separatamente.

- `Strength_Agility_Ratio`: Questo rapporto identifica squilibri biomeccanici tra la forza del ginocchio e l'agilità. In letteratura medica, uno squilibrio tra forza pura e capacità di cambio di direzione è spesso predittivo di infortuni ai legamenti.

Di seguito la heatmap prima dell'eliminazione della variabile Weight_kg e della creazione delle tre variabili nuove:

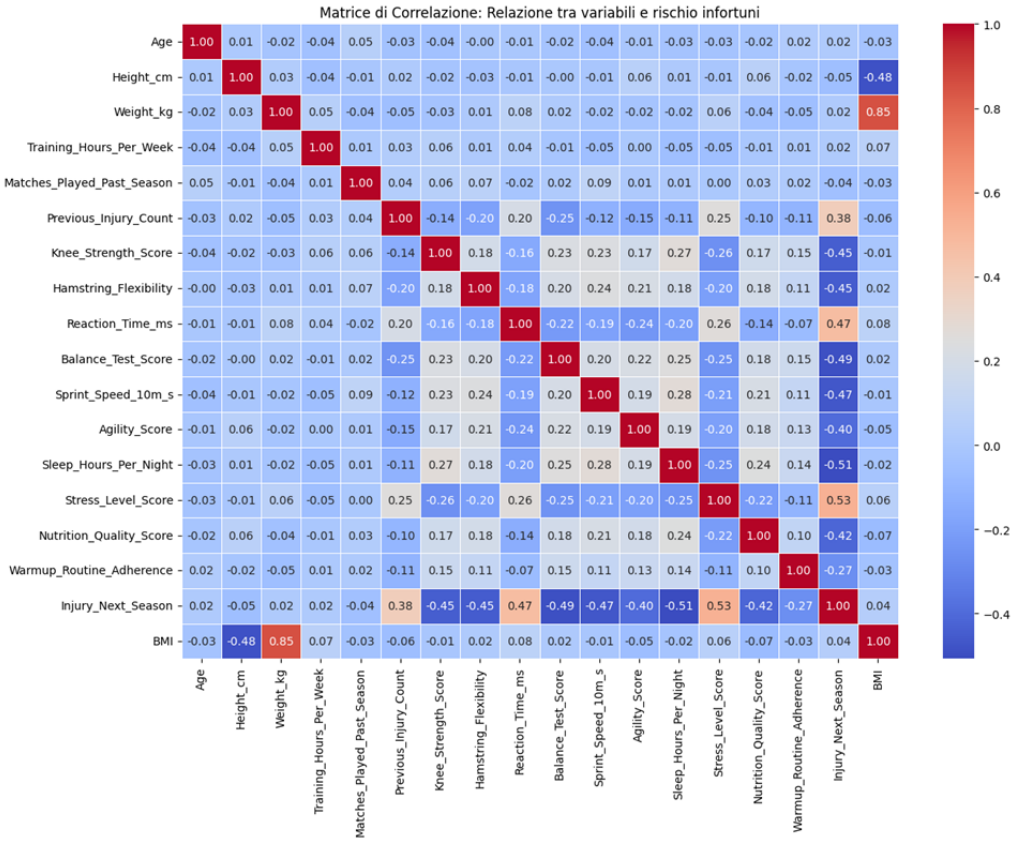


Figure 6: Matrice di Correlazione prima della fase di Feature Selection e Engineering.

Qui invece la heatmap dopo la fase di Feature Selection e Engineering:

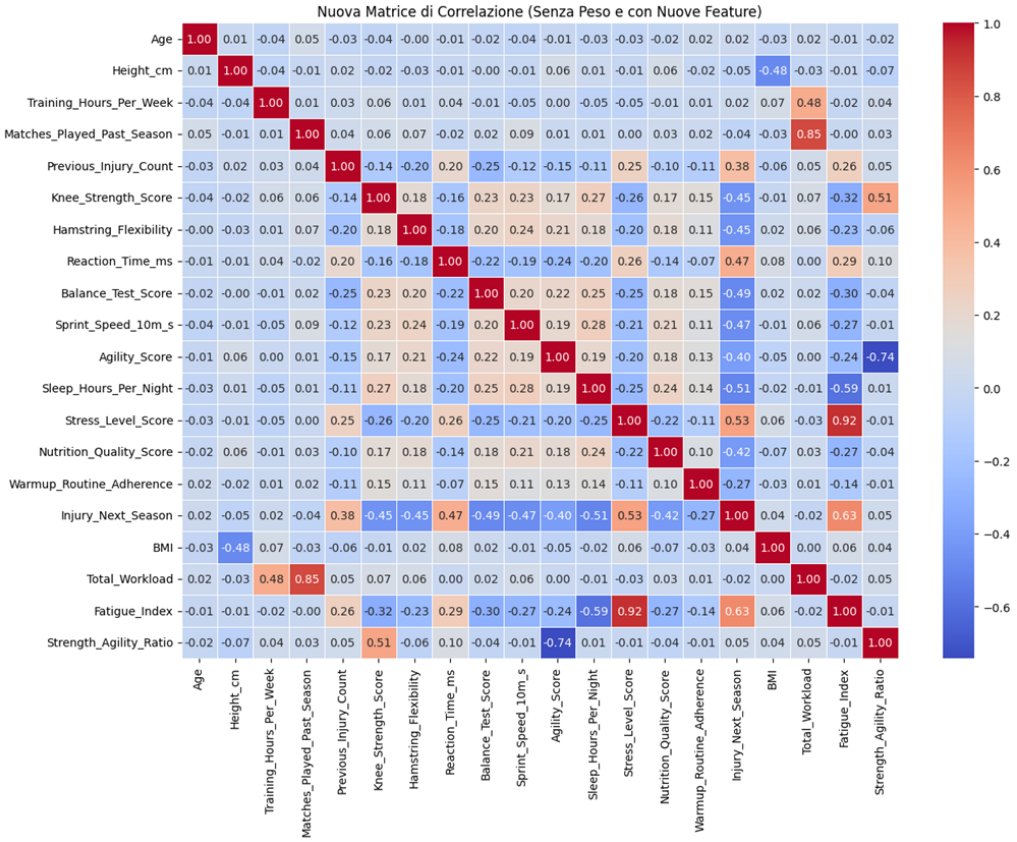


Figure 7: Nuova Matrice di Correlazione dopo la fase di Feature Selection e Engineering.

3.3 Pipeline e Prevenzione del Data Leakage

Per garantire il rigore metodologico, è stata utilizzata una *Pipeline* di *Scikit-Learn* che incapsula i seguenti passaggi di pre-processing:

- `StandardScaler`: Normalizzazione delle feature numeriche per portarle su una scala comune.
- `OneHotEncoder`: Codifica della variabile categorica `Position` in variabili dummy.

L'applicazione dello scaling all'interno della pipeline assicura che i parametri vengano calcolati esclusivamente sul **Training Set**, evitando rigorosamente il *Data Leakage* verso il **Test Set**. In questo modo, le informazioni statistiche del set di test non influenzano minimamente la fase di addestramento, garantendo una valutazione imparziale delle performance del modello.

4 ANALISI DELLE PRESTAZIONI

4.1 Selezione del Modello e Hyperparameter Tuning

Il processo di sviluppo ha previsto il confronto tra una baseline lineare (*Logistic Regression*) e un modello non lineare (*Random Forest*). Quest'ultimo è stato ottimizzato tramite `GridSearchCV` con *5-fold Cross-Validation* per trovare la combinazione ottimale di profondità degli alberi e numero di stimatori.
L'analisi dei tipi di errore, condotta tramite le matrici di confusione, ha rivelato una differenza qualitativa fondamentale tra i due modelli:

- **Analisi dei Falsi Negativi:** Nonostante la complessità del *Random Forest*, questo ha mancato la predizione di 10 infortuni reali. La *Logistic Regression* si è dimostrata superiore, riducendo i Falsi Negativi a soli 6 casi (corrispondenti a una *Recall* del 93.3%). In ambito sportivo, questo è il risultato più importante: minimizzare i casi in cui un atleta a rischio viene erroneamente classificato come sano, evitando un mancato intervento preventivo.
- **Analisi dei Falsi Positivi:** Entrambi i modelli presentano un numero estremamente esiguo di Falsi Positivi (solo 2 per la *Logistic Regression*). Questi errori sono considerati "cautelativi" nel contesto di un sistema di Early Warning, poiché comporterebbero al massimo un riposo precauzionale senza rischi per l'atleta.

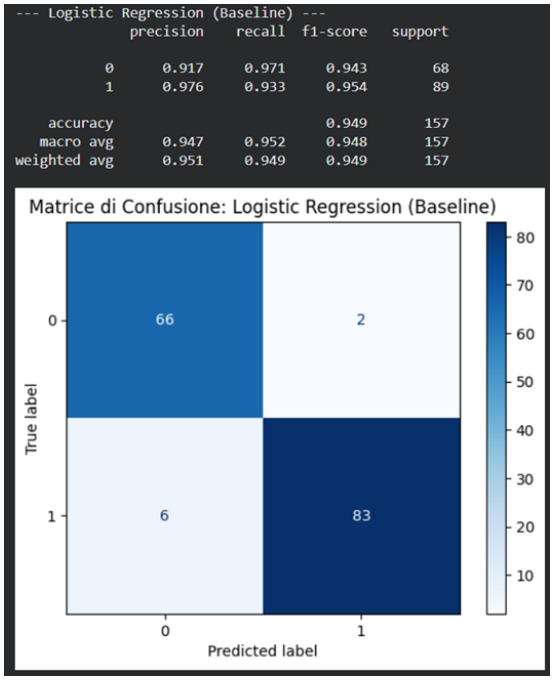


Figure 8: Matrice di Confusione: Logistic Regression

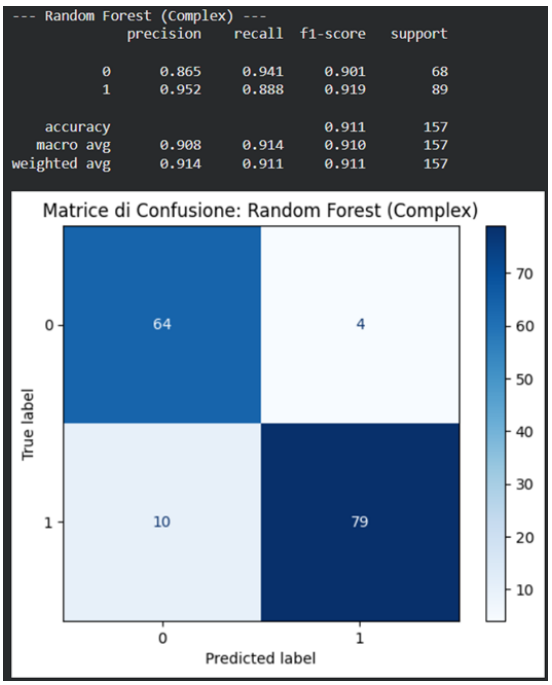
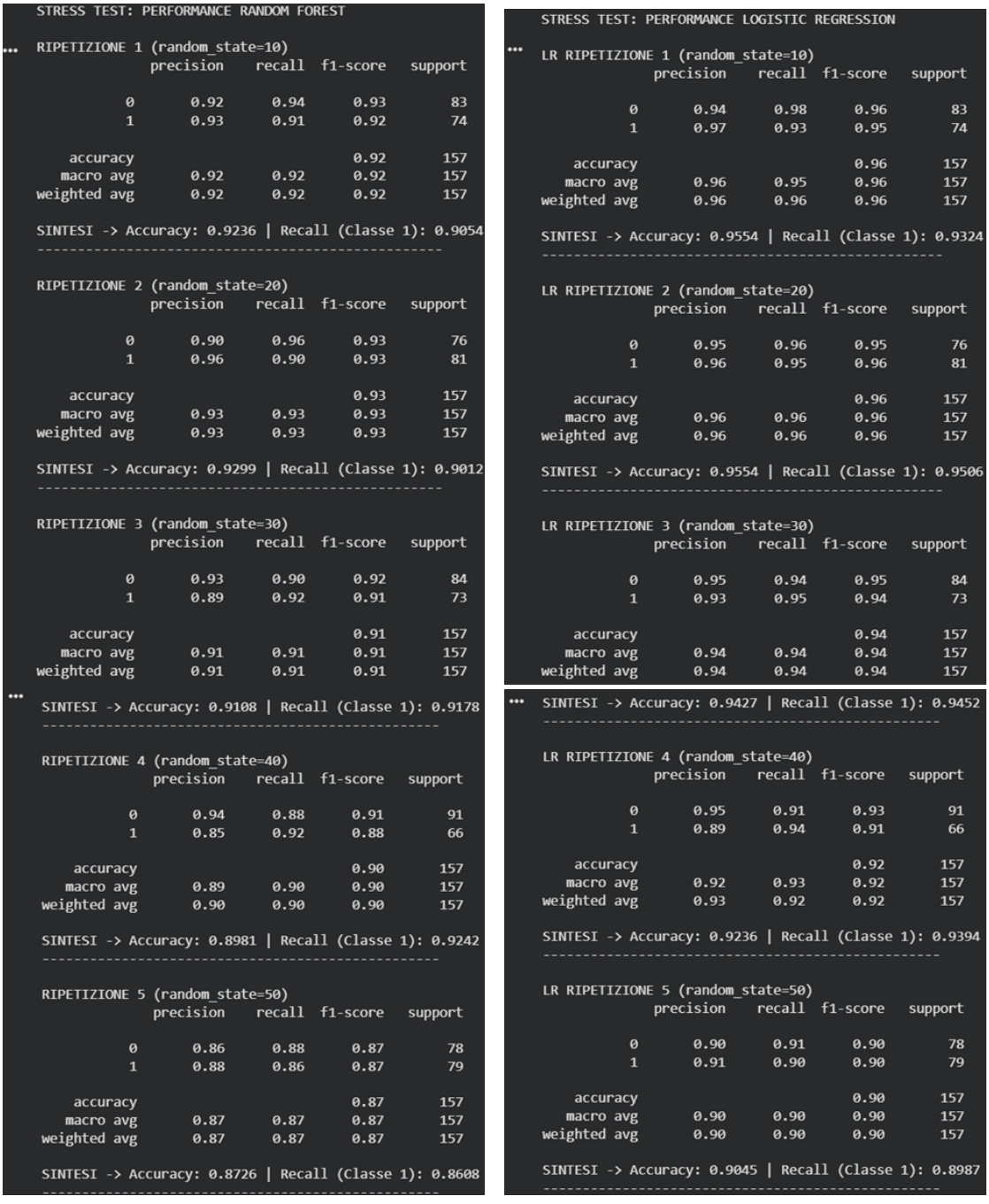


Figure 9: Matrice di Confusione: Random Forest

In sintesi, la *Logistic Regression* non è stata scelta solo per l'Accuracy del 94.9%, ma perché rispetto al *Random Forest*, abbatte drasticamente il numero di Falsi Negativi (6 vs 10), offrendo la massima protezione possibile all'integrità fisica dei calciatori.
Oltre alle performance puntuali, è fondamentale analizzare la stabilità degli algoritmi tramite lo Stress Test effettuato su 5 split casuali. Questa analisi ha rivelato una criticità strutturale del Random Forest: nonostante l'ottimizzazione iper-parametrica via GridSearchCV, il modello ha mostrato un'elevata varianza. Le sue prestazioni sono oscillate sensibilmente, scendendo fino a un'accuratezza minima dell'87.90%. Ciò dimostra che il Random Forest è fortemente dipendente dalla specifica combinazione di dati ricevuti in addestramento, rischiando l'overfitting su campioni specifici. Al contrario, la Logistic Regression ha dimostrato una robustezza superiore. Sebbene anch'essa abbia mostrato una naturale sensibilità alla variazione dei dati, ha mantenuto un'accuratezza media e minima più elevata e costante rispetto alla controparte complessa. Indipendentemente dalla difficoltà dello split (ovvero dalla presenza di casi limite nel test set), il modello lineare ha garantito una coerenza nei risultati che lo rende, per questo dataset, molto più affidabile per un'applicazione reale in ambito medico-sportivo.
Successivamente gli andamenti dello stress test del Random Forest e del Logistic Regression:



```
STRESS TEST: PERFORMANCE RANDOM FOREST

RIPETIZIONE 1 (random_state=10)
              precision    recall  f1-score   support

           0       0.92      0.94      0.93        83
           1       0.93      0.91      0.92        74

    accuracy                           0.92       157
   macro avg       0.92      0.92      0.92       157
weighted avg       0.92      0.92      0.92       157

SINTESI -> Accuracy: 0.9236 | Recall (Classe 1): 0.9054
--------------------------------------------------

RIPETIZIONE 2 (random_state=20)
              precision    recall  f1-score   support

           0       0.90      0.96      0.93        76
           1       0.96      0.90      0.93        81

    accuracy                           0.93       157
   macro avg       0.93      0.93      0.93       157
weighted avg       0.93      0.93      0.93       157

SINTESI -> Accuracy: 0.9299 | Recall (Classe 1): 0.9012
--------------------------------------------------

RIPETIZIONE 3 (random_state=30)
              precision    recall  f1-score   support

           0       0.93      0.90      0.92        84
           1       0.89      0.92      0.91        73

    accuracy                           0.91       157
   macro avg       0.91      0.91      0.91       157
weighted avg       0.91      0.91      0.91       157

SINTESI -> Accuracy: 0.9108 | Recall (Classe 1): 0.9178
--------------------------------------------------

RIPETIZIONE 4 (random_state=40)
              precision    recall  f1-score   support

           0       0.94      0.88      0.91        91
           1       0.85      0.92      0.88        66

    accuracy                           0.90       157
   macro avg       0.89      0.90      0.90       157
weighted avg       0.90      0.90      0.90       157

SINTESI -> Accuracy: 0.8981 | Recall (Classe 1): 0.9242
--------------------------------------------------

RIPETIZIONE 5 (random_state=50)
              precision    recall  f1-score   support

           0       0.86      0.88      0.87        78
           1       0.88      0.86      0.87        79

    accuracy                           0.87       157
   macro avg       0.87      0.87      0.87       157
weighted avg       0.87      0.87      0.87       157

SINTESI -> Accuracy: 0.8726 | Recall (Classe 1): 0.8608
--------------------------------------------------
```

```
STRESS TEST: PERFORMANCE LOGISTIC REGRESSION

LR RIPETIZIONE 1 (random_state=10)
              precision    recall  f1-score   support

           0       0.94      0.98      0.96        83
           1       0.97      0.93      0.95        74

    accuracy                           0.96       157
   macro avg       0.96      0.95      0.96       157
weighted avg       0.96      0.96      0.96       157

SINTESI -> Accuracy: 0.9554 | Recall (Classe 1): 0.9324
--------------------------------------------------

LR RIPETIZIONE 2 (random_state=20)
              precision    recall  f1-score   support

           0       0.95      0.96      0.95        76
           1       0.96      0.95      0.96        81

    accuracy                           0.96       157
   macro avg       0.96      0.96      0.96       157
weighted avg       0.96      0.96      0.96       157

SINTESI -> Accuracy: 0.9554 | Recall (Classe 1): 0.9506
--------------------------------------------------

LR RIPETIZIONE 3 (random_state=30)
              precision    recall  f1-score   support

           0       0.95      0.94      0.95        84
           1       0.93      0.95      0.94        73

    accuracy                           0.94       157
   macro avg       0.94      0.94      0.94       157
weighted avg       0.94      0.94      0.94       157

SINTESI -> Accuracy: 0.9427 | Recall (Classe 1): 0.9452
--------------------------------------------------

LR RIPETIZIONE 4 (random_state=40)
              precision    recall  f1-score   support

           0       0.95      0.91      0.93        91
           1       0.89      0.94      0.91        66

    accuracy                           0.92       157
   macro avg       0.92      0.93      0.92       157
weighted avg       0.93      0.92      0.92       157

SINTESI -> Accuracy: 0.9236 | Recall (Classe 1): 0.9394
--------------------------------------------------

LR RIPETIZIONE 5 (random_state=50)
              precision    recall  f1-score   support

           0       0.90      0.91      0.90        78
           1       0.91      0.90      0.90        79

    accuracy                           0.90       157
   macro avg       0.90      0.90      0.90       157
weighted avg       0.90      0.90      0.90       157

SINTESI -> Accuracy: 0.9045 | Recall (Classe 1): 0.8987
--------------------------------------------------
```

Figure 10: Andamenti dello stress test per ambo i modelli.

Qui invece i dati relativi al Random Forest con i parametri forniti dal GridSearchCV:

```
--- PERFORMANCE MODELLO OTTIMIZZATO ---
              precision    recall  f1-score   support

           0       0.83      0.93      0.88        68
           1       0.94      0.85      0.89        89

    accuracy                           0.89       157
   macro avg       0.88      0.89      0.88       157
weighted avg       0.89      0.89      0.89       157

STRESS TEST: MODELLO OTTIMIZZATO (Best RF)
Ripetizione 1 (random_state=10): Accuracy = 0.9172
Ripetizione 2 (random_state=20): Accuracy = 0.9172
Ripetizione 3 (random_state=30): Accuracy = 0.9299
Ripetizione 4 (random_state=40): Accuracy = 0.8981
Ripetizione 5 (random_state=50): Accuracy = 0.8790

--- RIEPILOGO CONFRONTO STABILITÀ ---
RF COMPLESSO (Precedente) - Varianza: 0.0004 | Minimo: 0.8726
RF OTTIMIZZATO (Attuale)   - Varianza: 0.0003 | Minimo: 0.8790
```

Figure 11: Performance del modello Random Forest ottimizzato e riepilogo del confronto di stabilità.

4.1.1 Logica di Valutazione e Scelta delle Metriche

L'utilizzo della Recall come metrica fondamentale è legato a una priorità clinica: in ambito preventivo, l'obiettivo primario è minimizzare i Falsi Negativi. Non identificare un atleta a rischio è infinitamente più grave che suggerire un riposo precauzionale a un atleta sano. Una Recall elevata (pari al 90% nel modello finale) garantisce che la quasi totalità degli infortuni venga intercettata in anticipo, permettendo allo staff di intervenire con protocolli di recupero mirati.
Al contempo, è stata utilizzata l'Accuracy per validare la robustezza generale del modello. Poiché l'analisi preliminare ha confermato che il dataset è perfettamente bilanciato (50% per classe), questa metrica risulta estremamente affidabile e priva delle distorsioni che si presenterebbero in caso di sbilanciamento delle classi.
Metriche più complesse, come la ROC Curve, sono state scartate poiché avrebbero aggiunto un livello di astrazione non necessario, rischiando di compromettere la spiegabilità dei risultati, che rimane uno dei pilastri centrali di questo progetto.

4.2 Valutazione Slice-based

Per garantire l'affidabilità clinica del sistema, è stata eseguita una valutazione granulare per sottogruppi basata sul ruolo in campo. Come evidenziato nell'analisi esplorativa iniziale, l'assenza di sampling bias e la distribuzione omogenea dei reparti hanno permesso di confermare la Fairness del modello, il quale ha mostrato prestazioni eccellenti e consistenti:

- **Portieri:** 97% Accuracy.
- **Attaccanti:** 98% Accuracy.
- **Difensori:** 95% Accuracy.

Si è evidenziata una lieve flessione per i Centrocampisti (90%); dato il perfetto bilanciamento numerico di questo sottogruppo, tale scostamento non è imputabile a una carenza di dati, ma suggerisce che per questo ruolo, caratterizzato da un dinamismo ibrido tra fase difensiva e offensiva, i fattori di rischio siano più eterogenei e complessi da modellare rispetto ai ruoli più specializzati.

4.3 Feature Selection e Modello Finale

Dopo aver identificato la Logistic Regression come modello vincente, è stata eseguita una "potatura" delle variabili per rendere il sistema più efficiente.

- **Rimozione Feature Secondarie:** Sono state eliminate variabili come `Age`, `Height_cm` e `Agility Score`, che l'analisi di correlazione e l'importanza delle feature indicavano come poco determinanti.

```
CONFRONTO DEFINITIVO: MODELLI SNELLI (Senza Age, Height, Agility)

[ 1. LOGISTIC REGRESSION (Dataset Snello) ]
              precision    recall  f1-score   support

           0       0.88      0.97      0.92        68
           1       0.98      0.90      0.94        89

    accuracy                           0.93       157
   macro avg       0.93      0.93      0.93       157
weighted avg       0.93      0.93      0.93       157


[ 2. RANDOM FOREST (Ottimizzato - Dataset Snello) ]
              precision    recall  f1-score   support

           0       0.85      0.93      0.89        68
           1       0.94      0.88      0.91        89

    accuracy                           0.90       157
   macro avg       0.90      0.90      0.90       157
weighted avg       0.90      0.90      0.90       157
```

Figure 12: Performance dei modelli dopo la rimozione delle feature secondarie.

- **Performance del Modello Snello:** Il modello finale, addestrato sul set di dati ridotto, ha mantenuto una performance eccellente (93% Accuracy, 90% Recall). La lieve flessione rispetto al 95% (Accuracy) iniziale è giustificata da una maggiore robustezza e dalla riduzione del rischio di catturare rumore statistico, rendendo lo strumento più affidabile per l'applicazione clinica reale.

4.4 Analisi della Spiegabilità

Come sottolineato dai moderni standard dell'Intelligenza Artificiale in ambito medico, l'Explainability è fondamentale. Non basta che il modello fornisca una previsione accurata; per essere utile a uno staff medico, il sistema deve essere trasparente. L'uso della Logistic Regression è stato preferito proprio per la sua capacità di mostrare chiaramente quali fattori stiano pesando di più sulla valutazione del rischio.
L'analisi dei coefficienti della Logistic Regression (rappresentati in valore assoluto nel grafico) permette di identificare i principali driver del rischio infortunio. Ogni barra rappresenta quanto pesantemente quella variabile sposta la decisione del modello.

- `Hamstring_Flexibility` **e** `Reaction_Time_ms`**:** Emergono come i fattori più determinanti nel dataset. Biomeccanicamente, una scarsa flessibilità dei muscoli flessori e tempi di reazione rallentati sono segnali critici di vulnerabilità neuromuscolare e affaticamento.

- `Previous_Injury_Count`**:** Questo coefficiente elevato conferma che la storia clinica è un predittore robusto. Il modello riconosce correttamente che un atleta già infortunato in passato ha una probabilità statistica molto più alta di subire recidive.

- `Fatigue_Index`**:** La presenza di questa variabile tra le prime posizioni convalida il lavoro di Feature Engineering. Dimostra che il rapporto tra stress e recupero (sonno) ha un impatto diretto e quantificabile sul rischio, superiore a singole variabili grezze come lo Stress_Level_Score o le Sleep_Hours_Per_Night prese isolatamente.

- **Indicatori di Performance (**`Knee_Strength`**,** `Sprint_Speed`**):** Il fatto che queste variabili abbiano pesi rilevanti suggerisce che il modello non guarda solo allo stress, ma anche allo stato di forma fisica dell'atleta per determinare se il suo corpo può reggere il carico di lavoro.

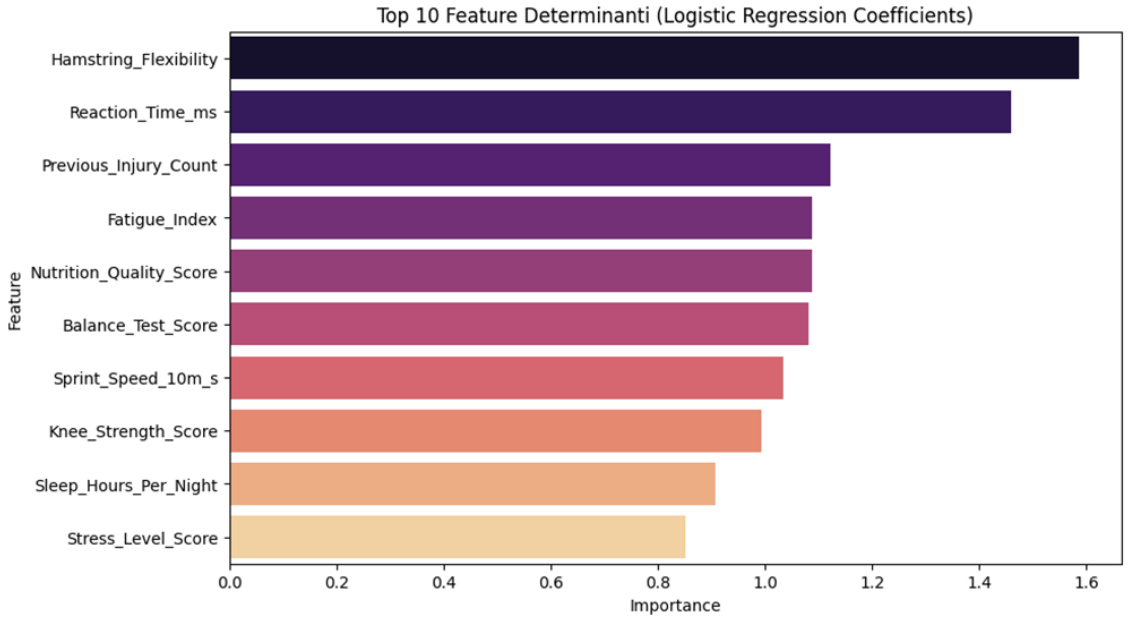


Figure 13: Top 10 Feature Determinanti basate sui coefficienti della Logistic Regression.

5 CONSIDERAZIONI FINALI E SVILUPPI FUTURI

5.1 Riflessioni Metodologiche e Valore del Modello

Il progetto ha dimostrato che, nel dominio della medicina sportiva, l'efficacia di un modello non risiede esclusivamente nella sua complessità algoritmica, ma soprattutto nella qualità del dato e nella sua interpretabilità. Attraverso un attento lavoro di Feature Engineering e lo Stress Test dei modelli, è emerso come la Logistic Regression offra il miglior bilanciamento tra accuratezza predittiva (93%) e stabilità statistica. A differenza del Random Forest, soggetto a maggiori fluttuazioni nelle performance, il modello lineare garantisce una coerenza nei risultati fondamentale per supportare decisioni critiche sulla salute degli atleti.
Un valore aggiunto di questo studio risiede nell'utilizzo esclusivo di dati reali. A differenza di molti approcci nel Machine Learning clinico che ricorrono a dati sintetici (oversampling) per bilanciare i dataset, questo lavoro si basa su un campione naturalmente equilibrato. Tale scelta ha permesso di eliminare il rischio di bias artificiali, garantendo un'aderenza superiore alla realtà del campo e una maggiore affidabilità nelle diagnosi precoci.
È stata data un'importanza prioritaria all'adozione di un approccio basato sulla *Explainability*. Questo concetto è essenziale per strumenti destinati a fornire supporto a uno staff medico: comprendere i fattori che guidano una predizione permette infatti di attuare interventi preventivi mirati e personalizzati.

5.2 Sviluppi futuri

In ottica futura, il progetto pone le basi per lo sviluppo di un modello capace di monitorare il rischio infortunio in tempo reale durante lo svolgimento della stagione. L'integrazione di dati biometrici e prestazionali raccolti tramite i dispositivi wearable (le "pettorine" GPS/inerziali indossate dai calciatori) permetterebbe di analizzare carichi di lavoro e stress fisico in modo istantaneo, aprendo scenari di prevenzione dinamica estremamente avanzati.
Oltre all'integrazione dei dati GPS, il sistema è predisposto per essere implementato in un'interfaccia di dashboarding (come un prototipo in Figma o una web-app), permettendo allo staff tecnico di visualizzare il rischio infortunio tramite semafori intuitivi che integrano la predizione algoritmica con la gestione operativa della squadra:

- **Verde (Sano):** L'atleta presenta parametri biometrici e di carico ottimali. Il modello lo classifica come "non a rischio", confermando la piena disponibilità per le sessioni di allenamento e le gare.
- **Giallo (A Rischio - Early Warning):** Il modello ha identificato pattern critici nei dati settimanali. Questa allerta funge da sistema di Early Warning, suggerendo allo staff interventi preventivi come carichi differenziati o sessioni di recupero specifiche.

- **Rosso (Infortunato):** Indica l'atleta attualmente non disponibile. Questo stato permette di chiudere il ciclo di feedback del sistema, registrando l'effettiva occorrenza dell'evento e validando l'efficacia delle predizioni fornite dal modello nel periodo precedente.

5.3 Prototipo

Di seguito vengono riportate le interfacce grafiche sviluppate per il prototipo funzionale, progettate per tradurre l'output del modello in uno strumento operativo di supporto alle decisioni cliniche.

5.3.1 Dashboard di monitoraggio

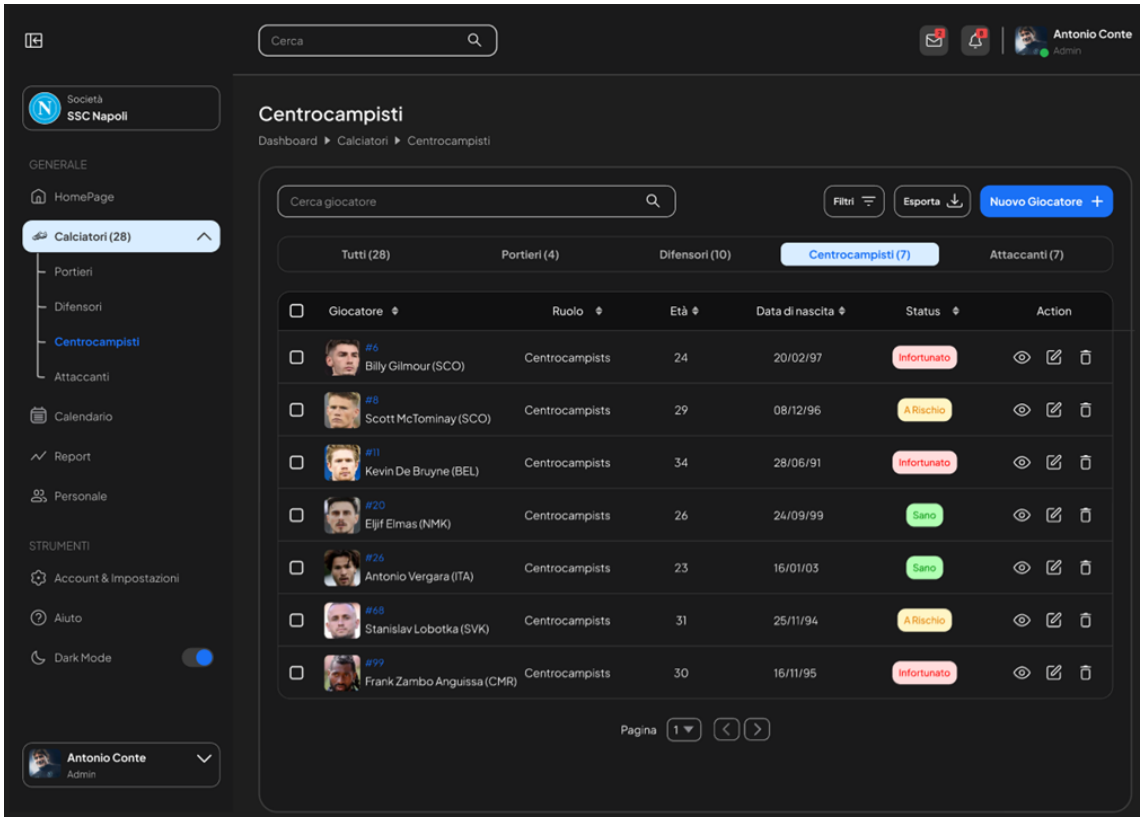


Figure 14: Interfaccia principale di monitoraggio della squadra.

5.3.2 Modulo di inserimento

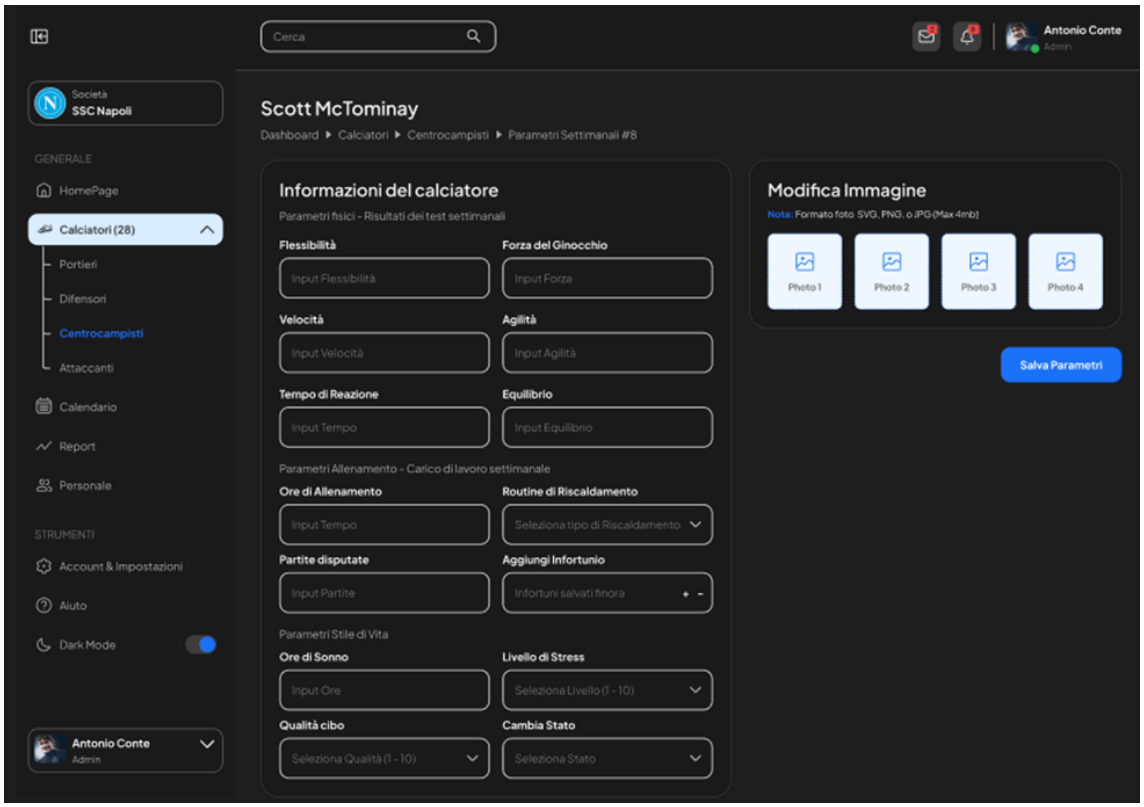


Figure 15: Schermata per l'input dei dati biometrici e prestazionali.

5.3.3 Explainability

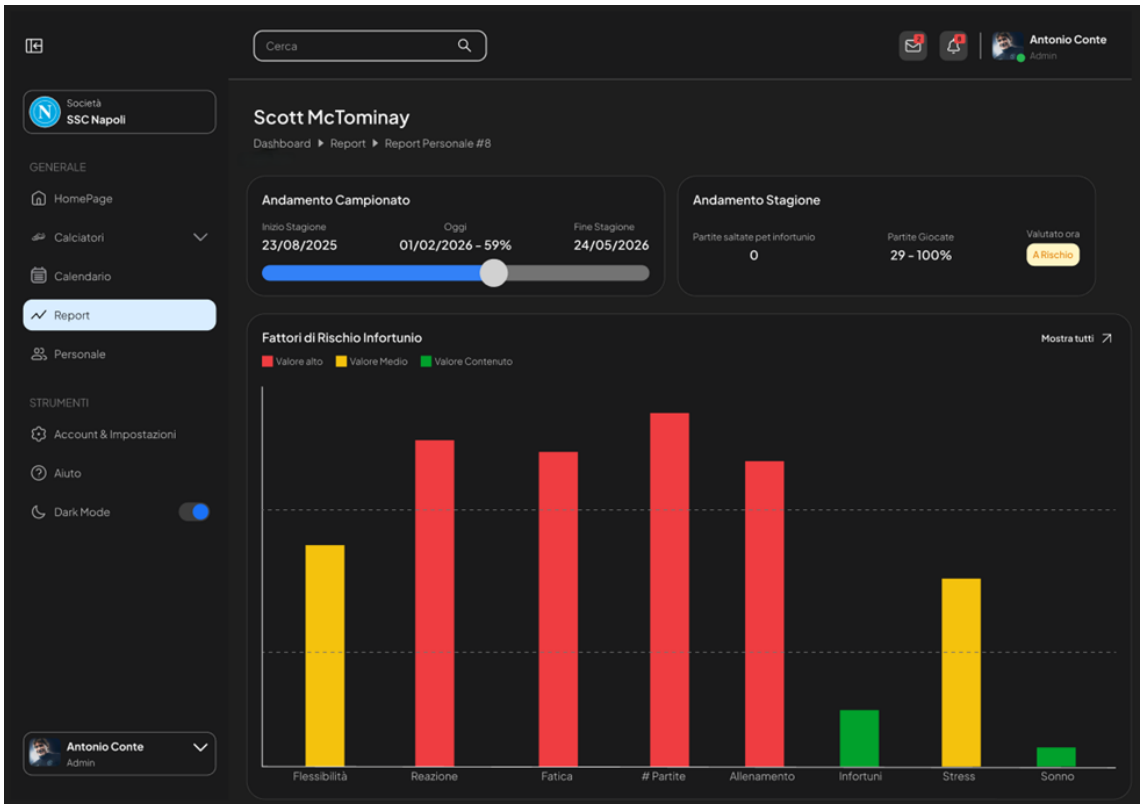


Figure 16: Visualizzazione dei fattori di rischio per singolo atleta.

5.4 Riferimenti e Strumenti Tecnici

Dataset: Football Injury Prediction Dataset (Kaggle - Aman Kansal)

L'intero workflow di analisi e modellazione è stato sviluppato in ambiente Google Colab, sfruttando la potenza del calcolo in cloud e la flessibilità del linguaggio Python 3.x. Per garantire la riproducibilità e la solidità scientifica del progetto, sono state impiegate le seguenti librerie standard del settore:

- **Analisi e Manipolazione Dati (`Pandas` & `NumPy`):** Questi strumenti sono stati fondamentali per la gestione delle 800 osservazioni del dataset. Sono stati utilizzati per il caricamento dei dati, la gestione del bilanciamento perfetto della variabile target (media 0.5) e, soprattutto, per la fase di *Feature Engineering* che ha permesso la creazione del *Fatigue Index* e degli altri indicatori biomeccanici.
- **Modellazione e Machine Learning (`Scikit-Learn`):** La libreria è stata il cuore pulsante dello studio, impiegata per l'implementazione della *Logistic Regression* e del *Random Forest*. Attraverso l'uso delle *Pipeline*, è stato possibile prevenire il *Data Leakage*, mentre il modulo `GridSearchCV` ha permesso l'ottimizzazione degli iper-parametri per garantire la massima stabilità statistica durante gli stress test.
- **Visualizzazione Scientifica (`Seaborn` & `Matplotlib`):** Oltre alla generazione delle matrici di correlazione (*Heatmap*) necessarie per identificare le ridondanze tra variabili, questi strumenti sono stati essenziali per l'analisi della *Explainability*. Hanno permesso di visualizzare graficamente l'importanza delle feature, trasformando i coefficienti matematici in evidenze cliniche comprensibili (es. l'impatto della flessibilità e del tempo di reazione).

La prototipazione UI/UX
Figma: Piattaforma utilizzata per lo sviluppo dell'interfaccia di dashboarding e la visualizzazione del sistema di *Early Warning*.