

# AI COACH

## Componenti del gruppo

Alessandro Aldo Mangione, [MAT. 776117],  
[a.mangione15@studenti.uniba.it](mailto:a.mangione15@studenti.uniba.it)

Tommaso Palumbo, [MAT. 777854],  
[t.palumbo6@studenti.uniba.it](mailto:t.palumbo6@studenti.uniba.it)

Link GitHub:

[https://github.com/Inf2425/AI\\_COACH\\_icon2425](https://github.com/Inf2425/AI_COACH_icon2425)

A.A. 2024/2025

## Indice

Introduzione .....	3
Requisiti funzionali .....	3
Installazione e avvio .....	6
Capitolo 0) Creazione del dataset .....	7
Preprocessing del dataset .....	7
Capitolo 1) Ragionamento .....	11
1.1 Estrazione delle serie temporali.....	11
1.2 Probabilità con Catene di markov.....	14
1.3 Predizione con Regression Decision tree.....	21
1.4 Conclusione fase di ragionamento.....	24
Capitolo 2) Apprendimento .....	26
2.1 Calcolo score giocatori.....	26
2.2 Costruzione del dataset per la classificazione.....	35
2.3 Definizione e assegnazione delle etichette.....	37
2.3.1 Normalizzazione delle variabili.....	38
2.3.2 Calcolo dello score composito.....	38
2.3.3 Logica di assegnazione delle etichette.....	38
2.4 Classificazione supervisionata.....	40
2.5 Clustering non supervisionato.....	45
Capitolo 3) Ricerca .....	59
Conclusioni .....	66

## **Introduzione**

L'obiettivo di questo progetto è la realizzazione di un sistema intelligente basato su conoscenza, in grado di selezionare automaticamente la formazione ideale per ciascuna squadra di Serie A, in vista della prossima giornata di campionato.

A tal proposito, è importante precisare che i dati a nostra disposizione coprono le 38 giornate del campionato di Serie A 2023/2024, e che, ai fini predittivi e sperimentali, si sta assumendo l'esistenza di una giornata aggiuntiva ancora da disputare, la 39esima.

Per raggiungere questo scopo, il progetto affronta e integra in modo coerente diverse tecniche strutturate in tre fasi principali: Ragionamento, Apprendimento e Ricerca.

La prima fase del progetto è dedicata al ragionamento, e mira a stimare il potenziale rendimento offensivo dei giocatori in vista della prossima giornata di campionato. Si tratta di una fase predittiva che, attraverso l'analisi delle prestazioni passate, fornisce indicazioni utili sul contributo che ogni calciatore potrebbe offrire.

A questa segue la fase di apprendimento, che consente di classificare i giocatori in base alla loro affidabilità e al loro profilo tattico. In questo modo è possibile identificare non solo chi merita di essere schierato, ma anche il tipo di ruolo che meglio rappresenta ciascun atleta all'interno del sistema di gioco.

Infine, la fase di ricerca sfrutta tutte le informazioni raccolte e apprese in precedenza per costruire automaticamente la formazione ideale di ogni squadra, tenendo conto di diversi vincoli tecnici e tattici, così da garantire soluzioni realistiche ed efficaci.

Dal nostro punto di vista, l'utilità di un sistema di questo tipo risulta evidente nel calcio moderno, dove le squadre si trovano ad affrontare un fitto calendario di impegni (in media una partita ogni tre giorni). Disporre di uno strumento capace di supportare automaticamente lo staff tecnico nella gestione della formazione ottimale rappresenta un concreto valore aggiunto per l'intera organizzazione sportiva.

## **Requisiti Funzionali**

Il progetto è stato interamente sviluppato in Python, un linguaggio scelto per la sua versatilità e per la disponibilità di numerose librerie utili alla manipolazione, analisi e modellazione dei dati.

L'ambiente di sviluppo utilizzato è Google Colab, che ha permesso di scrivere, eseguire e condividere il codice in modo semplice e collaborativo. La versione di Python utilizzata è la 3.11.

Librerie utilizzate:

- **pandas:**  
Importazione, manipolazione e analisi di dataset in formato .csv. Utilizzata per operazioni su tabelle di dati, come la pulizia, l'unione, la trasformazione e il salvataggio dei DataFrame durante tutte le fasi del progetto (ragionamento, apprendimento, ricerca);
- **numpy:**  
Gestione di array multidimensionali e operazioni matematiche vettoriali. Utile per elaborazioni numeriche nei modelli di regressione e classificazione;
- **google.colab.files:**  
Libreria specifica di Google Colab per caricare o scaricare file direttamente dall'ambiente notebook al computer locale. Utilizzata per esportare i risultati in formato .csv;
- **IPython.display:**  
Permette di visualizzare in maniera leggibile gli oggetti (come DataFrame) all'interno del notebook, facilitando il debug e il monitoraggio dei risultati;
- **matplotlib.pyplot:**  
Utilizzata per la creazione di grafici e visualizzazioni, ad esempio per mostrare performance dei modelli, curva del gomito per il clustering o matrici di confusione;
- **seaborn:**  
Libreria di visualizzazione statistica basata su matplotlib. Impiegata per rappresentazioni grafiche avanzate, in particolare matrici di confusione e analisi comparative;
- **sklearn.tree.DecisionTreeRegressor:**  
Modello di regressione supervisionata che prevede il numero di gol futuri per ciascun giocatore, basandosi su una struttura ad albero decisionale;
- **sklearn.tree.DecisionTreeClassifier:**  
Modello di classificazione supervisionata utilizzato per distinguere i giocatori "consigliati" da quelli "non consigliati", in base alle feature calcolate;
- **sklearn.ensemble.RandomForestClassifier:**  
Classificatore basato su un ensemble di alberi decisionali (foresta casuale), usato per migliorare le performance del modello di classificazione;

- **sklearn.model\_selection.train\_test\_split:**  
Funzione per suddividere il dataset in set di addestramento e di test in modo casuale, per garantire una corretta validazione del modello;
- **sklearn.metrics:**  
Modulo che fornisce funzioni per valutare le performance dei modelli predittivi tramite metriche come accuracy, classification report e confusion matrix;
- **sklearn.cluster.KMeans:**  
Algoritmo di clustering non supervisionato, utilizzato per raggruppare i giocatori in base a caratteristiche simili, andando oltre i ruoli tradizionali;
- **sklearn.preprocessing.StandardScaler:**  
Tecnica di normalizzazione basata su z-score per uniformare le scale delle variabili, evitando che valori numericamente più grandi influenzino il clustering;
- **collections.defaultdict:**  
Estensione del dizionario Python che assegna automaticamente un valore predefinito a nuove chiavi. Utile per aggregazioni e costruzione dinamica di strutture dati;
- **unicodedata:**  
Serve a normalizzare stringhe Unicode, ad esempio rimuovendo accenti e caratteri speciali nei nomi di giocatori e squadre per evitare errori nei merge;
- **io:**  
Gestione di flussi di input/output in memoria (buffer), utile per operazioni su file senza scriverli fisicamente su disco, come la lettura da contenuti testuali;
- **os:**  
Interazione con il file system: verifica dell'esistenza di file, navigazione nei percorsi, creazione di directory o rimozione di file temporanei;
- **ast:**  
Parsing sicuro di stringhe rappresentanti strutture dati Python (es. liste o dizionari), utilizzato per convertire dati testuali in strutture utilizzabili nel codice;
- **functools.reduce:**  
Funzione di ordine superiore che consente la combinazione iterativa di una sequenza di elementi, ad esempio per fondere più DataFrame o applicare operazioni cumulative;

- **time:**

Utilizzata per misurare il tempo di esecuzione di specifiche porzioni di codice o per introdurre ritardi controllati nell'esecuzione;

## **Installazione e avvio**

Per eseguire correttamente il progetto, è necessario seguire l'ordine logico delle fasi così come strutturato in precedenza: Ragionamento, Apprendimento e Ricerca. Ogni fase è suddivisa in uno o più file Python, e ciascun file va eseguito nell'ordine indicato, rispettando la sequenza prevista per il corretto funzionamento del sistema.

Il progetto è stato sviluppato su Google Colab, quindi non è previsto un file main.py. Tuttavia, all'interno di ciascun notebook o script sono presenti istruzioni chiaramente commentate che guidano l'utente nell'esecuzione passo-passo.

Si consiglia quindi di attenersi rigorosamente alla struttura del progetto e di seguire l'ordine dei file e delle celle come documentato.

## Capitolo 0) Creazione del dataset

Per la realizzazione di questo progetto sono stati utilizzati numerosi dataset, la maggior parte dei quali proveniente dal seguente repository Kaggle:

<https://www.kaggle.com/datasets/whisperingkahuna/serie-a-2324-team-and-player-insights>

Tali dataset fanno riferimento alla stagione di Serie A 2023/2024 e contengono un'ampia varietà di statistiche: alcuni riguardano informazioni generali sul campionato (che non sono state utilizzate nel nostro caso), mentre altri forniscono dati dettagliati sui singoli giocatori, come tiri, passaggi, contrasti, expected goals, rating, assist e così via. È proprio quest'ultima categoria che è stata integralmente sfruttata per alimentare il sistema intelligente da noi progettato.

Oltre ai file reperiti online, sono stati creati due dataset aggiuntivi manualmente, fondamentali per l'intero processo:

1. `Calendario2324.csv`: file strutturato contenente le statistiche essenziali di ogni partita del campionato, tra cui:
  - Giornata e risultato;
  - Squadra in casa e squadra in trasferta;
  - Elenco marcatori;
  - Minuto di ogni singolo gol;
  - Eventuale specifica su autogol o rigore.
2. `elenco_giocatori_per_squadre_2023_2024.csv`: si tratta di un dataset aggregato, costruito a partire da 20 file CSV (uno per ciascuna squadra di Serie A e anch'essi costruiti manualmente), ciascuno contenente:
  - Nome della squadra;
  - Numero di maglia;
  - Nome del giocatore;
  - Posizione in campo.

Tutte queste informazioni sono state unificate successivamente in questo singolo file finale, `elenco_giocatori_per_squadre_2023_2024.csv`, che comprende le colonne: Team, Number, Player, Position.

## Preprocessing del dataset

Abbiamo avviato il lavoro di preprocessing a partire dai file CSV contenenti le statistiche dei calciatori (provenienti da Kaggle) e dal file `elenco_giocatori_per_squadre_2023_2024.csv`, da noi costruito manualmente.

Come primo passo, ci siamo concentrati sulla normalizzazione dei nomi dei giocatori, ovvero del campo 'Player', per evitare problemi di inconsistenza e incompatibilità nei successivi passaggi. Questa operazione è risultata essenziale, in quanto i dati provenivano da fonti diverse e presentavano spesso varianti nei nomi (es. spazi, maiuscole/minuscole, accenti).

```
Colonna 'Player' normalizzata in
elenco_giocatori_per_squadre_2023_2024.csv
Colonna 'Player' normalizzata in
player_possessions_won_attacking_third.csv
Colonna 'Player' normalizzata in player_tackles_won.csv
Colonna 'Player' normalizzata in player_total_assists_in_attack.csv
Colonna 'Player' normalizzata in player_accurate_long_balls.csv
Colonna 'Player' normalizzata in player_big_chances_missed.csv
Colonna 'Player' normalizzata in player_effective_clearances.csv
Colonna 'Player' normalizzata in player_expected_goals.csv
Colonna 'Player' normalizzata in player_fouls_committed.csv
Colonna 'Player' normalizzata in player_interceptions.csv
Colonna 'Player' normalizzata in player_penalties_conceded.csv
Colonna 'Player' normalizzata in player_clean_sheets.csv
Colonna 'Player' normalizzata in player_expected_assists.csv
Colonna 'Player' normalizzata in player_expected_goals_on_target.csv
Colonna 'Player' normalizzata in player_goals_conceded.csv
Colonna 'Player' normalizzata in player_on_target_scoring_attempts.csv
Colonna 'Player' normalizzata in player_penalties_won.csv
Colonna 'Player' normalizzata in player_red_cards.csv
Colonna 'Player' normalizzata in player_top_assists.csv
Colonna 'Player' normalizzata in player_total_scoring_attempts.csv
Colonna 'Player' normalizzata in player_accurate_passes.csv
Colonna 'Player' normalizzata in player_top_scorers.csv
Colonna 'Player' normalizzata in player_yellow_cards.csv
Colonna 'Player' normalizzata in player_contests_won.csv
Colonna 'Player' normalizzata in player_expected_assists_per_90.csv
Colonna 'Player' normalizzata in player_expected_goals_per_90.csv
Colonna 'Player' normalizzata in player_outfielder_blocks.csv
Colonna 'Player' normalizzata in player_player_ratings.csv
Colonna 'Player' normalizzata in player_saves_made.csv
```

Successivamente, prima di procedere con l'unione dei file, abbiamo effettuato una selezione manuale delle colonne rilevanti per ciascun CSV, rimuovendo quelle ridondanti o non significative per il nostro scopo.

\*Abbiamo deciso di mantenere la colonna 'Country' in ciascun CSV perché ogni file contiene informazioni uniche non presenti negli altri. Per questo motivo, in questa fase abbiamo conservato tutte le versioni della colonna, con l'obiettivo di unirle successivamente in un'unica colonna durante il processo di integrazione dei dati.

Questo ha permesso di alleggerire il dataset finale e mantenerlo focalizzato sulle caratteristiche utili.

```
player_possessions_won_attacking_third.csv: mantenute colonne →
['Player', 'Possessions Won in Final 3rd per 90', 'Possessions Won
Midfield per 90', 'Country']
```



player\_tackles\_won.csv: mantenute colonne → ['Player', 'Tackles per 90', 'Tackle Success Rate (%)', 'Country']

player\_total\_assists\_in\_attack.csv: mantenute colonne → ['Player', 'Chances Created', 'Chances Created per 90', 'Country']

player\_accurate\_long\_balls.csv: mantenute colonne → ['Player', 'Accurate Long Balls per 90', 'Successful Long Balls (%)', 'Minutes', 'Matches', 'Country']

player\_big\_chances\_missed.csv: mantenute colonne → ['Player', 'Big Chances Missed', 'Country']

player\_effective\_clearances.csv: mantenute colonne → ['Player', 'Clearances per 90', 'Total Clearances', 'Country']

player\_expected\_goals.csv: mantenute colonne → ['Player', 'Expected Goals (xG)', 'Actual Goals', 'Country']

player\_fouls\_committed.csv: mantenute colonne → ['Player', 'Fouls Committed per 90', 'Country']

player\_interceptions.csv: mantenute colonne → ['Player', 'Interceptions per 90', 'Total Interceptions', 'Country']

player\_penalties\_conceded.csv: mantenute colonne → ['Player', 'Penalties Conceded', 'Country']

player\_clean\_sheets.csv: mantenute colonne → ['Player', 'Clean Sheets', 'Goals Conceded', 'Country']

player\_expected\_assists.csv: mantenute colonne → ['Player', 'Expected Assists (xA)', 'Actual Assists', 'Country']

player\_expected\_goals\_on\_target.csv: mantenute colonne → ['Player', 'Expected Goals on Target (xGOT)', 'Country']

player\_goals\_conceded.csv: mantenute colonne → ['Player', 'Goals Conceded per 90', 'Country']

player\_on\_target\_scoring\_attempts.csv: mantenute colonne → ['Player', 'Shots on Target per 90', 'Shot Accuracy (%)', 'Country']

player\_penalties\_won.csv: mantenute colonne → ['Player', 'Penalties Won', 'Fouls Won per 90', 'Country']

player\_red\_cards.csv: mantenute colonne → ['Player', 'Red Cards', 'Country']

player\_top\_assists.csv: mantenute colonne → ['Player', 'Secondary Assists', 'Country']

player\_total\_scoring\_attempts.csv: mantenute colonne → ['Player', 'Shots per 90', 'Shot Conversion Rate (%)', 'Country']

elenco\_giocatori\_per\_squadre\_2023\_2024.csv: mantenute colonne → ['Player', 'Team', 'Position']

player\_accurate\_passes.csv: mantenute colonne → ['Player', 'Accurate Passes per 90', 'Pass Success (%)', 'Country']

player\_top\_scorers.csv: mantenute colonne → ['Player', 'Penalties', 'Country']

player\_yellow\_cards.csv: mantenute colonne → ['Player', 'Yellow Cards', 'Country']

player\_contests\_won.csv: mantenute colonne → ['Player', 'Successful Dribbles per 90', 'Dribble Success Rate (%)', 'Country']

player\_expected\_assists\_per\_90.csv: mantenute colonne → ['Player', 'Expected Assists per 90', 'Actual Assists per 90', 'Country']

player\_expected\_goals\_per\_90.csv: mantenute colonne → ['Player', 'Expected Goals per 90', 'Goals per 90', 'Country']

player\_outfielder\_blocks.csv: mantenute colonne → ['Player', 'Blocks per 90', 'Total Blocks', 'Country']

```
player_player_ratings.csv: mantenute colonne → ['Player', 'FotMob  
Rating', 'Player of the Match Awards', 'Country']  
player_saves_made.csv: mantenute colonne → ['Player', 'Saves per 90',  
'Total Saves', 'Country']
```

Completate queste operazioni preliminari, abbiamo eseguito un merge progressivo su base 'Player', ottenendo un unico dataset unificato (StatisticheGiocatori.csv) contenente, per ogni calciatore, tutte le metriche individuali selezionate. Durante questa fase, abbiamo anche consolidato le diverse colonne Country presenti nei singoli file in un'unica colonna unificata, scegliendo per ciascun giocatore il primo valore disponibile. Il risultato finale rappresenta il cuore informativo del nostro sistema, in quanto fornisce una vista integrata e completa di ogni giocatore, da utilizzare nelle successive fasi di previsione, classificazione e ricerca.

```
Merge completato: 689 giocatori, 53 colonne
```

Una volta ottenuto il dataset unificato tramite il merge orizzontale su base Player, abbiamo gestito la presenza di valori mancanti (NaN) all'interno delle metriche numeriche. Tali valori rappresentano l'assenza di dati per alcuni giocatori in specifiche statistiche (es. un giocatore potrebbe non aver mai effettuato tiri in porta e quindi risultare privo di valori per quella colonna). Si è optato per la sostituzione di tutti i valori NaN con 0, coerentemente con l'interpretazione semantica delle metriche: l'assenza di dati in una metrica numerica indica, di fatto, una prestazione pari a zero in quel parametro.

Per quanto riguarda invece la colonna Country, abbiamo deciso di non sostituire i valori mancanti con 0, in quanto non avrebbe avuto un significato coerente. Considerata la presenza estremamente limitata di NaN in questo campo, abbiamo preferito gestirli manualmente in un secondo momento, attribuendo a ciascun giocatore la nazione corretta.

```
Tutti i NaN (tranne 'Country') sono stati sostituiti con 0.
```

In questo modo abbiamo garantito l'integrità numerica del dataset. Inoltre si evitano errori nei successivi algoritmi di machine learning o analisi statistica.

## Capitolo 1) Ragionamento

La prima fase operativa del progetto è dedicata al Ragionamento, con l'obiettivo di prevedere l'andamento futuro delle prestazioni dei giocatori. Inizialmente, l'attenzione era rivolta esclusivamente alla previsione per la prossima giornata (la 39<sup>a</sup>). Tuttavia, riflettendo sull'utilità che queste previsioni possono avere nelle fasi successive del progetto, si è deciso di estendere il calcolo delle probabilità e delle predizioni a tutte le giornate dalla 4<sup>a</sup> alla 39<sup>a</sup>. Si è deciso di partire dalla 4<sup>a</sup> giornata per avere una minima base di dati per poter effettuare i calcoli necessari per le successive fasi.

Questa scelta si è rivelata particolarmente strategica per la successiva fase di classificazione supervisionata, che richiede uno storico più ampio per l'addestramento del modello.

In particolare, per ogni giocatore e per ogni giornata, vengono calcolati due indicatori fondamentali:

- la probabilità che il giocatore segni almeno un gol nella giornata successiva;
- la predizione numerica del numero di gol attesi nella prossima partita.

Questi due valori rappresentano output chiave per le fasi successive del progetto:

- verranno utilizzati come feature di input nella Fase 3 (Classificazione);
- influenzeranno direttamente la selezione della formazione ottimale nella Fase 1 (Ricerca), fungendo da modificatori di punteggio.

Per garantire una gestione ordinata e modulare, l'intera fase è stata suddivisa in sottofasi operative distinte, che rendono il processo di previsione più chiaro, tracciabile e riutilizzabile nelle pipeline successive.

### 1.1) Estrazione delle serie temporali

Questa sottofase è finalizzata alla creazione di un dataset temporale capace di rappresentare, giornata per giornata, l'andamento realizzativo dei singoli giocatori. L'obiettivo principale di questa fase è stato quello di costruire una matrice in cui ciascuna riga rappresentasse un giocatore e ciascuna colonna una delle 38 giornate di Serie A, con all'interno il numero effettivo di gol segnati.

Il lavoro ha preso avvio con il caricamento del file `Calendario2324.csv`, descritto nel capitolo 0 (creazione del dataset).

Tuttavia, una prima difficoltà si è presentata subito: la struttura originaria del file non era interpretata correttamente da Pandas, in quanto tutte le informazioni risultavano

contenute all'interno di una singola colonna, senza separazione tra i campi. È stato quindi necessario intervenire manualmente suddividendo la colonna principale tramite split, recuperando le singole componenti (giornata, squadra di casa, squadra ospite, risultato, marcatori di casa, marcatori ospiti). Una volta corretta la struttura, si è potuto procedere al parsing effettivo delle colonne relative ai marcatori, inizialmente rappresentate come liste di dizionari serializzati in formato stringa.

	Giornata	Casa	Ospite	Risultato	Marcatori Casa	Marcatori Trasferta
0	1	Empoli	Hellas Verona	0-1	[]	[{'Giocatore': 'Bonazzoli', 'Minuto': 75, 'Rigore': False, 'Autogol': False}]
1	1	Frosinone	Napoli	1-3	[{'Giocatore': 'Harroui', 'Minuto': 7, 'Rigore': True, 'Autogol': False}]	[{'Giocatore': 'Politano', 'Minuto': 24, 'Rigore': False, 'Autogol': False}, {'Giocatore': 'Osimhen', 'Minuto': 42, 'Rigore': False, 'Autogol': False}, {'Giocatore': 'Osimhen', 'Minuto': 79, 'Rigore': False, 'Autogol': False}]
2	1	Genoa	Fiorentina	1-4	[{'Giocatore': 'Biraschi', 'Minuto': 58, 'Rigore': False, 'Autogol': False}]	[{'Giocatore': 'Biraghi', 'Minuto': 5, 'Rigore': False, 'Autogol': False}, {'Giocatore': 'Bonaventura', 'Minuto': 11, 'Rigore': False, 'Autogol': False}, {'Giocatore': 'Gonzalez', 'Minuto': 40, 'Rigore': False, 'Autogol': False}, {'Giocatore': 'Mandragora', 'Minuto': 56, 'Rigore': False, 'Autogol': False}]

Successivamente c'è stata l'iterazione riga per riga sul dataset, estraendo per ciascuna partita la lista dei marcatori, sia della squadra di casa che della squadra ospite. Per ciascun elemento è stato verificato che non si trattasse di un autogol, e nel caso fosse valido, è stato associato al nome del calciatore normalizzato, alla relativa giornata e al numero di gol segnati. Il risultato è stato un dizionario intermedio contenente, per ogni coppia (giocatore, giornata), il conteggio cumulativo dei gol.

```
Gol per giocatore-giornata (primi 10):
('bonazzoli', 1): 1
('harroui', 1): 1
('politano', 1): 1
('osimhen', 1): 2
('biraschi', 1): 1
('biraghi', 1): 1
('bonaventura', 1): 1
('gonzalez', 1): 1
('mandragora', 1): 1
('lautaro martinez', 1): 2
```

A partire da questa struttura, è stato costruito un DataFrame in cui ogni riga corrispondeva a un calciatore e ogni colonna a una delle 38 giornate del campionato. Tutti i valori iniziali sono stati impostati a 0, per poi essere aggiornati con i gol reali aggregati a partire dal dizionario precedente. Questa trasformazione ha permesso di ottenere una rappresentazione matriciale completa dell'andamento realizzativo di ogni singolo giocatore nel corso della stagione.

	Giocatore	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15
0	a. bastoni	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	a. carboni	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	abraham	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	acerbi	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	adli	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

\*la schermata cattura la matrice sino alla 15esima giornata.

A questo punto è emersa una nuova criticità legata alla presenza di duplicati nei nomi dei giocatori, dovuta a trascrizioni differenti (es. uso di accenti, spazi, maiuscole/minuscole). In alcuni casi lo stesso calciatore era registrato con varianti lievi ma sufficienti a generare righe duplicate nella matrice.

148	lautaro martinez	2	1	2	0	0	0	4	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
149	lautaro martinez	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	2	1	1	0	0	1

\*si noti anche l'alternanza di come i valori fanno riferimento in maniera totalmente casuale ad una delle due "istanze" di giocatore.

Per risolvere questo problema è stata definita una funzione di normalizzazione che ha applicato tre trasformazioni fondamentali:

- Conversione in minuscolo;
- Rimozione di spazi superflui;
- Rimozione di tutti i caratteri accentati.

Successivamente, si è proceduto a un raggruppamento del DataFrame per nome normalizzato, aggregando automaticamente i gol sulle righe duplicate.

143	Lautaro Martinez	2	1	2	0	0	0	4	1	1	0	1	0	1	0	1	1	0	0	1	2
-----	---------------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nonostante questo primo processo abbia risolto la maggior parte dei conflitti, è stato necessario intervenire manualmente su alcuni casi residui, in cui il nome del giocatore risultava comunque distinto in più varianti semanticamente equivalenti. Per esempio, sono stati accorpati manualmente i dati riferiti a "Cheddira", "Berardi", "Gudmundsson", "Nicolas Gonzalez", "Lorenzo Pellegrini", "Rafael Leao" e "Amir Rrahmani", unificando le rispettive righe in una singola riga aggregata e sommando i gol corrispondenti per giornata.

Completata la costruzione della matrice finale e risolti tutti i conflitti di denominazione, si è proceduto all'esportazione del file in formato CSV (GolGiocatoriTemporale.csv).

Prima di passare alla sottofase successiva della fase di Ragionamento, abbiamo apportato una modifica al file GolGiocatoriTemporale.csv, aggiungendo per ogni giocatore l'eventuale autogol registrato nella relativa giornata. In parallelo, abbiamo uniformato la nomenclatura della colonna 'Giocatore' adottando la stessa convenzione utilizzata nel file StatisticheGiocatori.csv. Questa scelta ci permette, in futuro, di recuperare rapidamente le statistiche di ciascun giocatore in modo coerente tra i due file.

Tutte queste modifiche sono state salvate nel nuovo file GolGiocatoriTemporaleConAutogoal.csv.

## **1.2) Probabilità con Catene di Markov**

All'interno della fase di Ragionamento del sistema, è stata implementata una sottofase di modellazione predittiva basata su un approccio probabilistico non supervisionato, fondato sulle Catene di Markov, con l'obiettivo di stimare per ciascun giocatore la probabilità di segnare almeno un gol nella giornata successiva del campionato.

In una prima versione del sistema, l'intera procedura era stata progettata per calcolare tale probabilità esclusivamente in riferimento alla 39<sup>a</sup> giornata, ovvero quella immediatamente successiva all'ultima osservata. Questo approccio iniziale, seppur coerente con l'obiettivo di fornire previsioni aggiornate per la giornata imminente, è stato successivamente rivisto. Riflettendo infatti sull'intero flusso del progetto e sull'utilizzo che tali previsioni avrebbero avuto nelle fasi successive, si è giunti alla consapevolezza che limitare il calcolo a una sola giornata avrebbe ridotto l'efficacia del sistema, in particolare nella fase di Classificazione Supervisionata, dove la disponibilità di una serie storica più ampia di dati predittivi risulta fondamentale per l'addestramento del modello. Per questo motivo, il codice è stato modificato per estendere il calcolo delle probabilità a tutte le giornate comprese tra la 4<sup>a</sup> e la 39<sup>a</sup>, mantenendo invariata la struttura del modello e applicando le stesse correzioni già sperimentate nel caso singolo della 39<sup>a</sup>.

Le Catene di Markov si basano sull'ipotesi di Markov, secondo la quale lo stato futuro di un sistema dipende unicamente dallo stato attuale e non dalla sequenza completa degli stati precedenti. Questo le rende particolarmente adatte a contesti dinamici come quello sportivo, dove risulta difficile costruire modelli supervisionati completi, ma è possibile sfruttare i comportamenti storici per generare previsioni efficaci.

Nel nostro contesto, ogni giocatore viene modellato come un processo stocastico che attraversa una sequenza temporale di stati discreti, definiti in base alle sue performance realizzative settimanali:

- Stato 0: nessun gol segnato;
- Stato 1: un gol segnato;
- Stato 2: due o più gol segnati.

A partire dalle transizioni osservate tra stati nelle giornate precedenti, viene costruita per ciascun giocatore una matrice di transizione  $3 \times 3$ , che contiene le frequenze con cui si verificano i passaggi da uno stato all'altro tra due giornate consecutive. Questa matrice viene poi normalizzata riga per riga, trasformandola in una matrice delle probabilità condizionate: ogni riga descrive la probabilità che, dato uno stato attuale, il giocatore si trovi in ciascuno dei tre stati nella giornata successiva.

L'output del modello è la probabilità che il giocatore si trovi in uno stato attivo (cioè Stato 1 o Stato 2) nella giornata  $n + 1$ , fornendo così una stima diretta della probabilità che segni almeno un gol.

Questo approccio compatto, informativo e personalizzato per ciascun giocatore consente di cogliere il profilo evolutivo della forma realizzativa senza richiedere etichette o variabili esplicative complesse, risultando robusto e flessibile in un contesto come quello calcistico, caratterizzato da variabilità e incertezza.

Durante la prima implementazione del modello sono emerse alcune problematiche legate alla stima delle probabilità di segnare da parte dei giocatori. In particolare, in diversi casi il modello assegnava probabilità esattamente pari a 1.0 o 0.0, valori che, nel contesto calcistico, risultano sospetti e poco realistici.

Il primo caso riguarda le probabilità esattamente pari a 1.0. Questo è un segnale evidente di overfitting: il modello ha osservato sempre lo stesso comportamento da parte del giocatore e, di conseguenza, si aspetta con assoluta certezza che si ripeta. Tuttavia, nella realtà del calcio, un evento con probabilità del 100% è estremamente raro, proprio per l'elevata componente di imprevedibilità che caratterizza le partite.

Il secondo caso è quello delle probabilità esattamente pari a 0.0. Anche questa è una stima estrema, ma in alcuni casi può essere più plausibile rispetto alla precedente. Può derivare, ad esempio, da una lunga assenza di gol da parte del giocatore oppure da un modello troppo rigido, che penalizza in modo eccessivo determinati andamenti storici.

Un'altra situazione da considerare è quella di giocatori che hanno segnato un solo gol, proprio nell'ultima giornata disponibile. In questi casi, il modello si trova a gestire un unico dato "positivo" e rischia di attribuire troppo peso a quell'evento isolato, arrivando a sovrastimare la probabilità che il giocatore segni nuovamente.

Infine, la combinazione tra il primo e il terzo caso rappresenta lo scenario più problematico: il giocatore ha segnato solo una volta (nell'ultima giornata) e il modello assegna una probabilità pari a 1.0 di segnare di nuovo. Questo è un chiaro errore dovuto alla scarsità di dati storici e alla conseguente incapacità del modello di generalizzare correttamente. In fase di calcolo della forma prevista, sarà quindi necessario introdurre dei correttivi, come un peso attenuato o un flag che indichi un basso grado di attendibilità della stima.

```
Giocatori con probabilità esattamente 1.0: ['junior sambia', 'mattia viti']
```

```
Giocatori con probabilità esattamente 0.0: ['alex sandro', 'houssem aouar', 'marko arnautovic', 'arthur', 'davide biraschi', 'matteo cancellieri', 'keinan davis', 'alessandro deiola', 'federico di francesco', 'kingstone mutandwa', 'tijjani noslin', 'mario pasalic', 'simy', 'vitor oliveira', 'mattia zaccagni']
```

```
Giocatori che hanno segnato solo in G38: ['alex sandro', 'keinan davis', 'alessandro deiola', 'kingstone mutandwa', 'junior sambia', 'mattia viti']
```

```
Giocatori anomali (1.0 e unico gol nell'ultima giornata): ['junior sambia', 'mattia viti']
```

Tali valori estremi, sebbene tecnicamente corretti in caso di osservazioni ripetute identiche, sono poco plausibili in un contesto aleatorio come quello calcistico e segnalano, come detto in precedenza, un evidente rischio di overfitting.

Per correggere questa distorsione è stato introdotto il Laplace smoothing, una tecnica che consiste nell'aggiungere un'unità a ciascuna frequenza della matrice di transizione. Questo accorgimento garantisce che nessuna transizione abbia probabilità nulla o totale, favorendo stime più realistiche e meno rigide.

Nonostante l'introduzione del Laplace smoothing abbia evitato gli estremi numerici, è emersa una nuova criticità: in presenza di un numero esiguo di osservazioni (ad esempio quando un giocatore ha segnato solo nell'ultima giornata), la probabilità risultante appariva comunque eccessivamente elevata.

```
Giocatori con probabilità esattamente 1.0: []
```

```
Giocatori con probabilità esattamente 0.0: []
```

```
Giocatori che hanno segnato solo in G38: ['alex sandro', 'keinan davis', 'alessandro deiola', 'kingstone mutandwa', 'junior sambia', 'mattia viti']
```

```
Giocatori anomali (1.0 e unico gol nell'ultima giornata): []
```

6	alex sandro	0.666667
227	junior sambia	0.666667

È il caso di Alex Sandro che avendo segnato un solo gol alla giornata 38, riceveva una probabilità vicina al 66% di segnare anche nella giornata successiva. Questo



valore, sebbene formalmente derivante dal comportamento osservato, risulta sovrastimato e non coerente con il rendimento complessivo del giocatore.

Per affrontare questa ulteriore anomalia, è stata introdotta una penalizzazione dinamica basata sulla quantità di transizioni osservate dallo stato attuale. In pratica, se un giocatore ha meno di cinque osservazioni utili al calcolo della probabilità, il valore stimato viene ridotto proporzionalmente. Questo meccanismo consente di abbassare la fiducia del modello in presenza di dati deboli, attenuando il peso attribuito a comportamenti isolati o sporadici.

6	alex sandro	0.400000
227	junior sambia	0.400000

Come si può notare dagli ultimi due screenshot, è presente un'ultima criticità, la quale riguarda il fatto che il modello Markov, per sua natura, non tiene conto della produttività complessiva del giocatore lungo la stagione, basandosi unicamente sulle dinamiche locali tra gli stati. Questo limite può portare a sottostimare la probabilità di segnare per giocatori molto prolifici (come Lautaro Martínez)

140	lautaro martinez	0.391304
-----	------------------	----------

e a sovrastimarla per marcatori occasionali (come si è notato in precedenza).

Per compensare questa mancanza di prospettiva storica, la probabilità stimata è stata ulteriormente ponderata attraverso un coefficiente derivato dal numero totale di gol realizzati dal giocatore nel corso della stagione. Tale coefficiente è stato calcolato come  $\log(1 + \text{gol totali})$ , successivamente normalizzato, al fine di evitare squilibri numerici successivi. In questo modo, la previsione finale riflette non solo il comportamento recente, ma anche il valore offensivo accumulato nel tempo.

2	lautaro martinez	0.503824	24
72	alex sandro	0.110904	1
73	junior sambia	0.110904	1

A seguito di questa prima implementazione focalizzata esclusivamente sulla 39<sup>a</sup> giornata, è emersa la necessità di ampliare l'orizzonte temporale del modello. L'analisi dei risultati ottenuti, unita alla consapevolezza dell'importanza che tali stime avrebbero avuto nelle fasi successive del progetto, ha suggerito di abbandonare l'approccio limitato a una singola giornata. Si è quindi deciso di estendere il calcolo

delle probabilità a tutte le giornate comprese tra la 4<sup>a</sup> e la 39<sup>a</sup>, adottando la stessa logica modellistica e applicando, per ciascuna previsione, i medesimi correttivi già sperimentati nel caso della 39<sup>a</sup>, i quali sono:

**1. Laplace Smoothing (Add-one smoothing):**

Le matrici di transizione tra stati vengono inizializzate con valori unitari anziché con zeri. Questa tecnica impedisce che transizioni mai osservate generino probabilità nulle, stabilizzando la distribuzione e garantendo stime conservative anche in presenza di dati scarsi.

**2. Penalizzazione delle sequenze corte:**

Le sequenze con un numero molto basso di transizioni subiscono una penalizzazione: la probabilità stimata viene attenuata in modo proporzionale alla lunghezza della sequenza, prevenendo così stime distorte nei primi turni del campionato.

**3. Ponderazione tramite i gol totali realizzati fino a quel momento:**

La probabilità finale viene ulteriormente corretta moltiplicandola per un fattore di peso derivato dalla funzione logaritmica del numero totale di gol segnati dal giocatore nelle giornate precedenti, opportunamente normalizzato. In questo modo si conferisce maggiore affidabilità alle previsioni per i giocatori più prolifici, smorzando invece le stime relative a giocatori meno incisivi, pur mantenendo una crescita controllata grazie all'impiego del logaritmo.

Nonostante questi accorgimenti abbiano comportato un progressivo miglioramento della qualità delle stime, come dimostrato dalla riduzione dei casi con probabilità nulla nelle giornate successive, permangono ancora valori pari a zero nelle fasi iniziali.

Controllo valori anomali:

Prob\_Gol == 0.0 per giornata:

- G4: 222 casi
- G5: 205 casi
- G6: 198 casi
- G7: 183 casi
- G8: 175 casi
- G9: 165 casi
- G10: 157 casi
- G11: 156 casi
- G12: 145 casi
- G13: 134 casi
- G14: 128 casi
- G15: 118 casi
- G16: 112 casi
- G17: 107 casi
- G18: 100 casi
- G19: 94 casi
- G20: 88 casi

- G21: 80 casi
- G22: 73 casi
- G23: 71 casi
- G24: 65 casi
- G25: 58 casi
- G26: 51 casi
- G27: 44 casi
- G28: 37 casi
- G29: 34 casi
- G30: 31 casi
- G31: 28 casi
- G32: 26 casi
- G33: 26 casi
- G34: 24 casi
- G35: 22 casi
- G36: 19 casi
- G37: 10 casi
- G38: 6 casi

`Prob_Gol >= 1.0 per giornata: 0`

Ciò risulta comunque coerente con la natura dei modelli markoviani, fortemente dipendenti dalla disponibilità di dati storici sufficientemente profondi.

Successivamente è stato selezionato un singolo giocatore, nel caso specifico Lautaro Martinez, per effettuare un'analisi qualitativa e quantitativa approfondita. Per il giocatore selezionato è stata costruita una matrice di transizione  $3 \times 3$ , che rappresenta il numero di passaggi osservati da uno stato all'altro tra giornate consecutive. Come detto in precedenza per garantire la robustezza del modello anche in presenza di transizioni rare o non osservate, si è applicata la tecnica del Laplace smoothing, inizializzando ogni cella della matrice con un valore pari a 1. Una volta popolata la matrice con le frequenze effettive, si è proceduto alla normalizzazione riga per riga, ottenendo una matrice delle probabilità condizionate: ogni riga rappresenta la distribuzione delle probabilità di passaggio a uno dei tre stati possibili, dato lo stato attuale.

Individuato lo stato più recente del giocatore, corrispondente alla giornata più recente del dataset, è stata quindi calcolata la probabilità che il giocatore si trovi in uno stato attivo (stato 1 o 2) nella giornata successiva. Questa probabilità, definita “probabilità pura”, è stata poi corretta tramite un peso proporzionale al numero complessivo di gol segnati nella stagione: il valore del peso è stato calcolato come  $\log(1 + \text{numero\_gol})$  diviso per un coefficiente di scala scelto da noi, allo scopo di enfatizzare l'attendibilità dei giocatori con una maggiore produzione offensiva. Il risultato finale è una probabilità corretta che riflette sia le dinamiche di transizione tra stati che la produttività globale del giocatore.

Il procedimento ha previsto anche la costruzione e visualizzazione delle due matrici associate al calcolo: la matrice di transizione grezza, utile per l'ispezione diretta delle frequenze, e la matrice di probabilità normalizzata, necessaria per l'inferenza probabilistica.

Giocatore analizzato: Lautaro Martinez  
 Stato attuale (G38): 0 → 0 gol  
 Probabilità pura (Markov): 0.391304  
 Gol totali stagionali: 24  
 Peso  $\log(1+gol)/2.5$ : 1.287550  
 Probabilità finale (pura × peso): 0.503824

**Matrice di transizione** (frequenze grezze con Laplace):

	0 gol	1 gol	2+ gol
0 gol	14	7	2
1 gol	7	4	4
2+ gol	3	4	1

**Matrice di probabilità** (normalizzata):

	0 gol	1 gol	2+ gol
0 gol	0.609	0.304	0.087
1 gol	0.467	0.267	0.267
2+ gol	0.375	0.500	0.125

Una volta completato il calcolo delle probabilità per tutti i giocatori e per tutte le giornate dalla 4<sup>a</sup> alla 39<sup>a</sup> si è proceduto al salvataggio del file Probabilita\_Storica.csv contenente per ogni riga il nome de giocatore e la probabilità che esso segni nella giornata indicata.

Giocatore	Giornata	Prob_Gol
federico gatti	G4	0.0
alessandro bastoni	G4	0.0
andrea carboni	G4	0.0
tammy abraham	G4	0.0
francesco acerbi	G4	0.0
yacine adli	G4	0.0
alex sandro	G4	0.0
pontus almqvist	G4	0.1109
amir rrahmani	G4	0.0
houssem aouar	G4	0.1664

...

Giocatore	Giornata	Prob_Gol
nehuen perez	G18	0.1034
matteo pessina	G18	0.0462
andrea petagna	G18	0.0
roberto piccoli	G18	0.1034
niccolo pierozzi	G18	0.0
andrea pinamonti	G18	0.4753
matteo politano	G18	0.256
stefan posch	G18	0.0
matteo prati	G18	0.0
christian pulisic	G18	0.2048

...

Giocatore	Giornata	Prob_Gol
vitor oliveira	G39	0.1758
dusan vlahovic	G39	0.3238
nikola vlastic	G39	0.0749
cristian volpato	G39	0.0213
walace	G39	0.0463
shon weissman	G39	0.0213
kenan yildiz	G39	0.0463
mattia zaccagni	G39	0.1946
duvan zapata	G39	0.4004
gabriele zappa	G39	0.0213

Questo cambiamento ha reso il sistema più completo e versatile, garantendo un insieme di dati predittivi più ampio e coerente, fondamentale sia per la successiva fase di classificazione supervisionata sia per un'analisi retrospettiva più solida.

### 1.3) Predizione con Regression Decision Tree

Forte dell'esperienza maturata nella fase precedente, in cui inizialmente si era commesso l'errore di limitare la previsione a una sola giornata (la 39<sup>a</sup>), in questa seconda parte si è deciso sin dall'inizio di adottare una strategia più estesa e sistematica.

Parallelamente alla stima probabilistica tramite catene di Markov, è stata implementata una predizione regressiva del numero di gol attesi per ciascun giocatore in ogni giornata.

In particolare, è stato impiegato un modello di regressione ad albero decisionale (Decision Tree Regressor), un algoritmo in grado di apprendere relazioni non lineari tra input e output a partire da dati etichettati. Il modello sfrutta sequenze temporali delle performance recenti di ciascun giocatore (es. numero di gol segnati nelle ultime giornate) per produrre una stima continua del numero di gol che il giocatore potrebbe realizzare nella giornata successiva.

L'obiettivo è stimare il numero atteso di gol per la giornata  $G_{i+1}$ , allenando il modello sui valori osservati nelle giornate precedenti, da  $G_{i-4}$  a  $G_i$ .

Il dataset utilizzato è, come nel caso precedente, GolGiocatoriTemporale.csv. Una matrice costituita da righe (giocatori) e colonne (giornate) nella cui intersezione viene indicato se c'è stato un gol o meno da parte di quel giocatore nella rispettiva giornata.

In questo file, un calciatore è presente soltanto se ha realizzato almeno un gol nel corso del campionato. Questa scelta è stata adottata al fine di costruire un modello predittivo mirato, capace di stimare sia la probabilità che il giocatore segni, sia il numero atteso di reti nelle giornate successive. Per quanto riguarda invece la gestione degli autogol, è stata adottata una strategia differente, in quanto tali eventi non contribuiscono positivamente alla prestazione offensiva del giocatore. Nella fase successiva del progetto, verrà assegnato uno score complessivo a ciascun calciatore, calcolato sulla base delle sue caratteristiche e delle performance registrate lungo tutte le giornate disputate. In tale punteggio, eventuali autogol verranno considerati come fattori penalizzanti, influenzando negativamente il valore finale attribuito al giocatore.

Per realizzare questo schema, sono stati sviluppati tre modelli regressivi distinti:

- un modello dedicato per la giornata G4, addestrato su una finestra ridotta con padding artificiale per compensare la scarsità di dati iniziali;
- un secondo modello per G5, strutturato in modo analogo ma con finestra leggermente più ampia;
- un modello globale, utilizzato dalla giornata G6 in poi, basato sulla strategia di sliding window, con una finestra temporale di dimensione 5, che permette di costruire esempi supervisionati dinamici a partire dalle sequenze storiche dei gol dei giocatori.

Il codice realizza questi tre modelli iterando su tutte le righe del dataset (GolGiocatoriTemporale.csv) e creando per ciascuna una serie di coppie input/output composte da:

- $X_{train}$ : una finestra temporale di gol precedenti (come input),
- $Y_{train}$ : il numero di gol segnati nella giornata successiva (come target).

Ogni modello viene quindi addestrato separatamente e applicato in base alla giornata di previsione:

- il modello per G4 usa gli ultimi 3 dati con padding,
- quello per G5 usa 4 dati,
- da G6 in poi si impiega il modello globale.

Per ciascuna giornata da G4 a G39, il modello predice il numero atteso di gol grezzo per ogni giocatore. Tuttavia, l'analisi preliminare dei risultati ha evidenziato alcune anomalie e distorsioni nei valori predetti, in particolare in corrispondenza di giocatori con picchi di prestazione nelle ultimissime giornate, come ad esempio Calafiori, che pur avendo segnato solo 2 gol in tutta la stagione, concentrati nella penultima giornata, risultava tra i giocatori con la previsione più alta.

	Giocatore	Pred_Gol_G39	Gol_Stagionali
0	riccardo calafiori	0.400000	2
1	nicolas gonzalez	0.375000	11

Per correggere questo squilibrio e ottenere una stima più affidabile abbiamo scelto di combinare questo valore con una misura della forma storica del giocatore, calcolata come il numero cumulato di gol segnati fino alla giornata precedente, normalizzato rispetto al massimo della stagione.

La formula applicata per la predizione pesata finale è:

$$\text{Pred\_Gol\_Pesata} = 0.7 \times \text{Pred\_Gol} + 0.3 \times \text{Gol\_Normalizzati}$$

A valle della generazione delle predizioni, è stato eseguito un controllo mirato sui valori pari a 0.0. Tali predizioni indicano un'assenza totale di aspettativa di rete per quel giocatore in quella giornata. Il codice calcola:

- il numero totale di predizioni nulle su tutte le giornate,
- il conteggio per giornata, utile per evidenziare eventuali concentrazioni anomale.

```
ANALISI PRED_GOL == 0.0
```

```
Totale predizioni con valore 0.0: 0
```

```
Pred_Gol_Pesata == 0.0 per giornata:0
```

Tale comportamento è coerente con le aspettative teoriche, confermando che il modello regredisce inizialmente verso stime conservative per poi affinarsi progressivamente.

Inoltre, abbiamo verificato che utilizzando la formula per il calcolo della colonna: Pred\_Gol\_Pesata, non possono esistere valori pari a 0.

	Giocatore	Giornata	Pred_Gol	Pred_Gol_Pesata	Gol_Stagionali	Gol_Normalizzati
0	federico chiesa	G4	1.0000	0.8200	2	0.4
1	abdou harroui	G4	0.5000	0.4700	2	0.4
2	nicolas gonzalez	G4	0.5000	0.4700	2	0.4
...						
275	jordan zemura	G4	0.0795	0.0556	0	0.0
276	szymon zurkowski	G4	0.0795	0.0556	0	0.0
277	leo ostigard	G4	0.0795	0.0556	0	0.0
...						
	Giocatore	Giornata	Pred_Gol	Pred_Gol_Pesata	Gol_Stagionali	Gol_Normalizzati
3614	lautaro martinez	G17	0.2500	0.4750	15	1.0000
3615	lorenzo lucca	G17	0.3415	0.3190	4	0.2667
3616	hakan calhanoglu	G17	0.2500	0.3150	7	0.4667
...						
3889	oier zarraga	G17	0.0694	0.0486	0	0.0000
3890	jordan zemura	G17	0.0694	0.0486	0	0.0000
3891	szymon zurkowski	G17	0.0694	0.0486	0	0.0000
...						
	Giocatore	Giornata	Pred_Gol	Pred_Gol_Pesata	Gol_Stagionali	Gol_Normalizzati
9730	nicolas gonzalez	G39	0.3750	0.4000	11	0.4583
9731	lautaro martinez	G39	0.1111	0.3778	24	1.0000
9732	olivier giroud	G39	0.2500	0.3750	16	0.6667
...						
10005	gabriele zappa	G39	0.0694	0.0611	1	0.0417
10006	jordan zemura	G39	0.0694	0.0611	1	0.0417
10007	leo ostigard	G39	0.0694	0.0611	1	0.0417

## 1.4) Conclusione fase di Ragionamento

I risultati ottenuti dalle elaborazioni predittive, sia in termini di probabilità (modello Markoviano) sia di predizione regressiva (Decision Tree con sliding window), sono stati salvati nel file `Giocatori_Pred_Prob_Storiche.csv`, che raccoglie le stime per ciascun giocatore e per ogni giornata compresa tra la G4 e la G39. Tutto ciò sarà di



sostegno alla fase di Apprendimento, in particolare nell'ambito della Classificazione e nel poter stabilire quale giocatore sia consigliabile o meno nella fase di Ricerca.

Giocatore	Giomata	Prob_Gol	Pred_Gol_Pesata
federico gatti	G4	0	0.0556
alessandro bastoni	G4	0	0.0556
andrea carboni	G4	0	0.0556

...

federico gatti	G12	0.0693	0.0963
alessandro bastoni	G12	0	0.0486
andrea carboni	G12	0	0.0486

...

federico gatti	G24	0.1008	0.0936
alessandro bastoni	G24	0	0.0486
andrea carboni	G24	0	0.0486

...

federico gatti	G39	0.0894	0.0986
alessandro bastoni	G39	0.0213	0.0611
andrea carboni	G39	0.0213	0.0611

## Capitolo 2) Apprendimento

Questa fase prevede due tipi di apprendimento:

- Apprendimento Supervisionato - > Classificazione Supervisionata: L'obiettivo è quello di predire se un giocatore dovrebbe essere schierato o evitato, basandosi su statistiche e probabilità/predizione dei gol per la prossima giornata;
- Apprendimento Non Supervisionato - > Clustering: L'obiettivo è quello di indentificare gruppi funzionali di giocatori all'interno dello stesso ruolo.

Tutto ciò verrà utilizzato a supporto della fase di Ricerca.

### 2.1) Calcolo Score dei giocatori

Una sottofase iniziale della fase di Apprendimento, con particolare riferimento alla Classificazione Supervisionata, prevede l'assegnazione di un punteggio denominato "score" a ciascun giocatore della Serie A 2023/2024 presente nel file StatisticheGiocatori.csv

Tale punteggio viene calcolato in modo differenziato in base al ruolo del giocatore e ha l'obiettivo di rappresentare un indicatore sintetico della qualità complessiva del calciatore. Oltre a rappresentare, come detto, un valore generale di qualità, questo score assume anche un ruolo compensativo, andando a colmare quanto non è stato possibile realizzare nella fase di Ragionamento, ovvero la stima di probabilità e predizione per quei giocatori che non hanno mai segnato nelle giornate precedenti e che, per questo motivo, non erano gestibili dai modelli predittivi implementati in quella fase.

In vista dell'obiettivo finale del progetto, la formazione automatica della Top 11 per ogni squadra durante la fase di Ricerca, lo score costituirà uno dei principali criteri di selezione, affiancandosi, ad altri indicatori statistici e predittivi.

Per poter procedere al calcolo dello score per ciascun giocatore, sono state svolte alcune azioni preliminari. In particolare, si è reso necessario estrarre il numero di autogol commessi da ogni calciatore, utilizzando il file GolGiocatoriTemporaleConAutogol.csv, nel quale ogni autogol era indicato con la lettera "a" nella giornata corrispondente. Per rendere questa informazione utilizzabile a fini statistici, si è effettuata una conversione simbolico-numerica: i valori "a" sono stati trasformati in 1 (presenza di autogol), mentre i restanti valori sono rimasti 0 (assenza di autogol). A partire da questa trasformazione, per ogni riga è stata calcolata la somma totale degli autogol realizzati dal giocatore durante il campionato.

	<b>Giocatore</b>	<b>Autogoals</b>
<b>0</b>	Davide Calabria	1
<b>1</b>	Alessandro Zanolì	1
<b>2</b>	Angelino	1
<b>3</b>	Alberto Grassi	1
<b>4</b>	Matias Vina	2
<b>5</b>	Lorenzo Pirola	1

Successivamente, questo conteggio è stato integrato nel dataset delle statistiche dei giocatori (StatisticheGiocatori.csv) mediante un'operazione di join effettuata sul nome del calciatore. Prima dell'unione, è stata svolta un'operazione di uniformazione dei nomi tra i due file per garantire la corretta corrispondenza.

Al termine di questa fase, è stato generato il file StatisticheConAutogoal.csv, contenente per ogni giocatore tutte le caratteristiche originarie più una nuova colonna denominata "Autogoals". Questo dataset arricchito ha rappresentato la base informativa per il successivo calcolo dello score individuale di ciascun calciatore.

Completata la fase di integrazione dei dati, il dataset è stato suddiviso in quattro sottoinsiemi, uno per ciascun ruolo: portieri, difensori, centrocampisti e attaccanti.

Per ognuna di queste categorie è stata definita una formula specifica per il calcolo dello score, basata su una ponderazione lineare di diverse metriche individuali registrate nel corso della stagione. La selezione delle metriche da includere è stata guidata da un processo di analisi esplorativa: per ciascuna caratteristica statistica, è stata calcolata la media per ruolo. Questa analisi ha permesso di comprendere, anche grazie a una successiva valutazione qualitativa e realistica (basata sulla conoscenza del gioco e sull'esperienza pratica), quali fossero le metriche più significative e rilevanti da utilizzare nel calcolo dello score per ciascun ruolo.

I risultati di questa analisi sono stati raccolti nel file Media\_Statistiche\_per\_Ruolo.csv.

## Media\_Statistiche\_per\_Ruolo.csv

A	B	C	D	E
	P	D	C	A
Possessions Won In F	0	0.09778761062	0.2537383178	0.3075144509
Possessions Won Mic	0	0.8283185841	1.375700935	0.8485549133
Tackles per 90	0.001315789474	0.539380531	0.5518691589	0.2913294798
Tackle Success Rate	4.605263158	33.04734513	30.33504673	29.56358382
Chances Created	0.1842105263	8.460176991	13.8364486	14.08092486
Chances Created per	0.006578947368	0.5402654867	1.130373832	1.278612717
Accurate Long Balls	0	1.326548673	0.9677570093	0.3947976879
Successful Long Bal	0	25.40575221	28.46775701	28.5433526
Minutes	0	1095.632743	999.5700935	825.7283237
Matches	0	15.33185841	15.49065421	14.76300578

Infine, a ogni metrica selezionata è stato assegnato un peso specifico, scelto in funzione della sua rilevanza funzionale rispetto al ruolo del calciatore. Questi pesi sono stati utilizzati nella costruzione della formula finale di score per ciascun sottoinsieme di ruolo.

Inoltre, per ogni categoria di ruolo, escluso il ruolo del portiere, è stato introdotto un coefficiente moltiplicativo finale, denominato “fattore di continuità”, con l’obiettivo di valorizzare la costanza di impiego del giocatore lungo l’intero arco del campionato. Questo coefficiente, compreso tra 0.75 e 1 per i difensori e tra 0.85 e 1 per centrocampisti e attaccanti, è stato calcolato rapportando i minuti effettivamente giocati dal calciatore rispetto al totale teorico massimo di 3420 minuti (equivalente a 38 giornate da 90 minuti ciascuna).

Al termine del calcolo dello score (basato sulla formula specifica per ruolo), si è deciso di normalizzarne il valore su una scala da 0 a 10, così da renderlo confrontabile tra giocatori appartenenti a ruoli diversi. Questa normalizzazione si è resa necessaria per poter unificare rendere lo score confrontabile trasversalmente tra i ruoli e integrabile con gli altri punteggi già calcolati per ciascun giocatore, ovvero la probabilità di segnare (derivata dal modello Markov) e la predizione numerica dei gol attesi (calcolata tramite Decision Tree Regressor). Tutti e tre questi valori, ovvero, score, probabilità, predizione, sono fondamentali per alimentare la successiva fase di classificazione supervisionata.

La procedura di normalizzazione è stata effettuata tramite una trasformazione lineare sull’intervallo 0 – 10, secondo la seguente formula:

Score normalizzato =  $10 \times (\text{Score finale} - \text{Score minimo}) / (\text{Score massimo} - \text{Score minimo})$

In questa espressione, lo score finale è il punteggio ottenuto da ciascun giocatore dopo l'eventuale applicazione del fattore di continuità, mentre score minimo e score massimo rappresentano rispettivamente il valore più basso e quello più alto tra tutti i punteggi della categoria considerata.

Per quanto riguarda i portieri, la formula di scoring è stata costruita dando particolare rilievo a clean sheet, numero e media di parate effettuate, premi individuali ricevuti e al rating sintetico fornito da FotMob. Sono inoltre state introdotte penalizzazioni proporzionali al numero di gol subiti per 90 minuti e agli eventuali autogol commessi.

In questo caso, non è stato applicato il fattore di continuità, in quanto le statistiche riguardavano principalmente i 20 portieri titolari, che avevano un minutaggio molto simile. Sebbene fossero presenti anche portieri di riserva (secondi e terzi), la loro incidenza complessiva sul calcolo dello score era marginale e non tale da giustificare l'introduzione di un coefficiente correttivo.

Il calcolo dello score per i portieri è stato eseguito tramite la seguente formula:

```
df_portieri['Score_P'] = (  
    6 * df_portieri['Clean Sheets'] +  
    0.02 * df_portieri['Total Saves'] +  
    2 * df_portieri['Saves per 90'] +  
    1.5 * df_portieri['Player of the Match Awards'] +  
    15 * df_portieri['FotMob Rating'] -  
    8 * df_portieri['Goals Conceded per 90'] -  
    5 * df_portieri['Autogoals']  
)
```

Il risultato di questa operazione è stato salvato nel file Score\_Portieri.csv.

	Player	Team	Score_P
40	Vanja Milinkovic-Savic	Torino	222.51
0	Yann Sommer	Inter	221.20
60	Wojciech Szczesny	Juventus	194.62
20	Michele Di Gregorio	Monza	185.49

Come già anticipato in precedenza, anche per i portieri si è deciso di normalizzare lo score calcolato, al fine di garantire l'uniformità del punteggio rispetto a quello degli altri ruoli. Questa normalizzazione, applicata su una scala da 0 a 10, ha permesso di rendere confrontabili gli score tra giocatori con caratteristiche e ruoli differenti.

Il risultato è stato salvato nel file Score\_Portieri\_Normalizzato.csv.

	Player	Team	Score_P_normalizzato
40	Vanja Milinkovic-Savic	Torino	10.000000
0	Yann Sommer	Inter	9.941126
60	Wojciech Szczesny	Juventus	8.746573
20	Michele Di Gregorio	Monza	8.336255

Nel caso dei difensori, si è valorizzato non solo il contributo difensivo diretto, espresso attraverso intercetti, contrasti, respinte e blocchi, ma anche l'apporto alla manovra offensiva, evidenziato da assist, passaggi precisi e occasioni create. Una componente rilevante della formula ha riguardato i clean sheet di squadra, calcolati in modo indiretto sommando i clean sheet stagionali dei portieri appartenenti alla medesima squadra del difensore. Tale valore è stato considerato rappresentativo della solidità collettiva del rapporto arretrato. Le penalizzazioni hanno incluso le metriche negative come falli commessi, rigori concessi, ammonizione, espulsioni e autogol.

Il punteggio finale è stato poi moltiplicato per un coefficiente di continuità, il quale, come già detto, premia i giocatori maggiormente impiegati durante la stagione.

Il calcolo dello score per i difensori è stato eseguito tramite la seguente formula:

```
df_difensori["Score_D"] = (  
    6 * df_difensori["Tackles per 90"] +  
    6 * df_difensori["Interceptions per 90"] +  
    4 * df_difensori["Clearances per 90"] +  
    4 * df_difensori["Blocks per 90"] +  
    1.5 * df_difensori["Tackle Success Rate (%)"] +  
    2 * df_difensori["Accurate Passes per 90"] +  
    6 * df_difensori["FotMob Rating"] +  
    4 * df_difensori["Actual Assists"] +  
    4 * df_difensori["Actual Goals"] +  
    4 * df_difensori["Clean Sheets_Squadra"] -  
    1 * df_difensori["Goals Conceded"] +  
    1.5 * df_difensori["Pass Success (%)"] +  
    2 * df_difensori["Player of the Match Awards"] -  
    2 * df_difensori["Fouls Committed per 90"] -  
    2 * df_difensori["Penalties Conceded"] -  
    3 * df_difensori["Red Cards"] -  
    2 * df_difensori["Yellow Cards"] +  
    1 * df_difensori["Chances Created per 90"] +  
    1 * df_difensori["Accurate Long Balls per 90"] +  
    1 * df_difensori["Expected Goals (xG)"] +  
    1 * df_difensori["Expected Assists (xA)"] -  
    1 * df_difensori["Big Chances Missed"] +  
    1.5 * df_difensori["Successful Long Balls (%)"] -
```

```

    4 * df_difensori["Autogoals"]
)
df_difensori["Score_D_finale"] = df_difensori["Score_D"] * (0.75 + 0.25 *
(df_difensori["Minutes"] / 3420))

```

Il tutto è stato salvato nel file Score\_Difensori.csv.

	Player	Team	Minutes	Autogoals	Score_D_finale
272	Alessandro Bastoni	Inter	2283.0	0	575.923581
291	Riccardo Calafiori	Bologna	2338.0	1	568.051924
174	Francesco Acerbi	Inter	2386.0	0	566.712741
247	Jhon Lucumi	Bologna	2214.0	0	559.934882

Anche per i difensori, si è proceduto alla normalizzazione dei punteggi su una scala da 0 a 10. Il risultato è stato salvato nel file Score\_Difensori\_Normalizzato.csv

	Player	Team	Score_D_normalizzato
272	Alessandro Bastoni	Inter	10.000000
291	Riccardo Calafiori	Bologna	9.863339
174	Francesco Acerbi	Inter	9.840089
247	Jhon Lucumi	Bologna	9.722418

Per i centrocampisti, la valutazione ha contemplato un equilibrio tra produzione offensiva e responsabilità difensive. Sono state ponderate le reti segnate, gli assist diretti e indiretti, le occasioni create, la qualità del passaggio e l'efficacia nei dribbling. Allo stesso modo, sono stati inclusi parametri relativi a tackle, intercetti e disciplina tattica. Anche per questa categoria, gli autogol sono stati integrati nella formula come fattore di penalizzazione e il punteggio è stato ulteriormente affinato attraverso l'applicazione del coefficiente di continuità.

Il calcolo dello score per i centrocampisti è stato eseguito tramite la seguente formula:

```

df_centrocampisti['Score_C'] = (
    8 * df_centrocampisti['Actual Goals'] +
    8 * df_centrocampisti['Actual Assists'] +

```

```

6 * df_centrocampisti['Secondary Assists'] +
6 * df_centrocampisti['Expected Assists (xA)'] +
6 * df_centrocampisti['Expected Goals (xG)'] +
5 * df_centrocampisti['FotMob Rating'] +
4 * df_centrocampisti['Chances Created'] +
3 * df_centrocampisti['Accurate Passes per 90'] +
3 * df_centrocampisti['Pass Success (%)'] +
3 * df_centrocampisti['Successful Dribbles per 90'] +
3 * df_centrocampisti['Interceptions per 90'] +
3 * df_centrocampisti['Tackles per 90'] +
2 * df_centrocampisti['Shot Conversion Rate (%)'] +
2 * df_centrocampisti['Shot Accuracy (%)'] +
1 * df_centrocampisti['Player of the Match Awards'] -
6 * df_centrocampisti['Red Cards'] -
4 * df_centrocampisti['Yellow Cards'] -
3 * df_centrocampisti['Fouls Committed per 90'] -
3 * df_centrocampisti['Penalties Conceded'] -
2 * df_centrocampisti['Big Chances Missed'] -
3 * df_centrocampisti['Autogoals']
)
df_centrocampisti['Score_C_finale'] = df_centrocampisti['Score_C'] *
(0.85 + 0.15 * df_centrocampisti['Minutes'] / 3420)

```

Punteggio salvato nel file Score\_Centrocampisti.csv.

	Player	Team	Minutes	Score_C_finale
374	Hakan Calhanoglu	Inter	2576.0	1060.147386
340	Teun Koopmeiners	Atalanta	2633.0	919.284121
379	Luis Alberto	Lazio	2323.0	872.546270
369	Antonio Candreva	Salernitana	2757.0	824.554704

Anche per i centrocampisti, lo score finale è stato normalizzato su una scala da 0 a 10. Il risultato è stato salvato nel file Score\_Centrocampisti\_Normalizzato.csv.

	Player	Team	Score_C_normalizzato
374	Hakan Calhanoglu	Inter	10.000000
340	Teun Koopmeiners	Atalanta	8.678702
379	Luis Alberto	Lazio	8.240300
369	Antonio Candreva	Salernitana	7.790139



Gli attaccanti sono stati valutati principalmente in base alla loro incidenza offensiva, mediante un sistema di pesi che ha privilegiato i gol segnati, gli expected goals, le conclusioni a rete e le occasioni create. Sono stati presi in esame anche parametri di efficienza tecnica, quali la precisione al tiro, la conversione delle occasioni e l'abilità nel dribbling. Oltre alle penalizzazioni standard per falli, cartellini e rigori concessi, è stata inserita anche la penalizzazione per gli autogol, sebbene con peso inferiore rispetto a quanto previsto per i ruoli difensivi ed è stato applicato il coefficiente di continuità.

Il calcolo dello score per gli attaccanti è stato eseguito tramite la seguente formula:

```
df_attaccanti['Score_A'] = (  
    30 * df_attaccanti['Actual Goals'] +  
    10 * df_attaccanti['Expected Goals (xG)'] +  
    10 * df_attaccanti['Expected Goals on Target (xGOT)'] +  
    8 * df_attaccanti['Actual Assists'] +  
    6 * df_attaccanti['Expected Assists (xA)'] +  
    6 * df_attaccanti['Actual Assists per 90'] +  
    6 * df_attaccanti['Expected Assists per 90'] +  
    4 * df_attaccanti['Chances Created'] +  
    4 * df_attaccanti['Chances Created per 90'] +  
    3 * df_attaccanti['Shots on Target per 90'] +  
    3 * df_attaccanti['Shots per 90'] +  
    3 * df_attaccanti['Secondary Assists'] +  
    3 * df_attaccanti['Player of the Match Awards'] +  
    2 * df_attaccanti['Shot Accuracy (%)'] +  
    2 * df_attaccanti['Shot Conversion Rate (%)'] +  
    2 * df_attaccanti['Successful Dribbles per 90'] +  
    2 * df_attaccanti['Dribble Success Rate (%)'] +  
    2 * df_attaccanti['Accurate Passes per 90'] +  
    2 * df_attaccanti['Accurate Long Balls per 90'] +  
    1.5 * df_attaccanti['Pass Success (%)'] +  
    1.5 * df_attaccanti['Successful Long Balls (%)'] +  
    1 * df_attaccanti['Fouls Won per 90'] +  
    1 * df_attaccanti['Penalties Won'] +  
    1 * df_attaccanti['Goals per 90'] +  
    1 * df_attaccanti['Expected Goals per 90'] +  
    0.5 * df_attaccanti['Possessions Won in Final 3rd per 90'] +  
    0.5 * df_attaccanti['Possessions Won Midfield per 90'] +  
    0.5 * df_attaccanti['Tackles per 90'] +  
    0.5 * df_attaccanti['Tackle Success Rate (%)'] +  
    0.5 * df_attaccanti['Interceptions per 90'] +  
    6 * df_attaccanti['FotMob Rating'] -  
    3 * df_attaccanti['Big Chances Missed'] -  
    2 * df_attaccanti['Fouls Committed per 90'] -  
    2 * df_attaccanti['Penalties Conceded'] -  
    2 * df_attaccanti['Yellow Cards'] -  
    4 * df_attaccanti['Red Cards'] -  
    2 * df_attaccanti['Autogols'] # penalizzazione inclusa  
)
```

```
# 4. Calcola Score_A_finale in base ai minuti giocati
df_attaccanti['Score_A_finale'] = df_attaccanti['Score_A'] * (0.85 + 0.15
* (df_attaccanti['Minutes'] / 3420))
```

Punteggio salvato sul file Score\_Attaccanti.csv.

	Player	Team	Minutes	Score_A_finale
645	Lautaro Martinez	Inter	2668.0	1746.549726
518	Albert Gudmundsson	Genoa	3022.0	1609.829336
564	Matias Soule	Frosinone	3136.0	1464.063398
569	Paulo Dybala	Roma	1973.0	1431.222286

Come per le altre categorie, anche lo score finale degli attaccanti è stato normalizzato su scala 0–10, e salvato nel file Score\_Attaccanti\_Normalizzato.csv.

	Player	Team	Score_A_normalizzato
645	Lautaro Martinez	Inter	10.000000
518	Albert Gudmundsson	Genoa	9.217350
564	Matias Soule	Frosinone	8.382918
569	Paulo Dybala	Roma	8.194921

Il passo successivo è stato quello di unificare tutti gli score normalizzati (portieri, difensori, centrocampisti e attaccanti) in un unico file, ordinato per ruolo e valore di score.

Il risultato di questa operazione è stato salvato nel file Score\_Giocatori\_Normalizzato.csv.

	Player	Team	Score	Role
0	vanja milinkovic-savic	Torino	10.000000	P
1	yann sommer	Inter	9.941126	P
2	wojciech szczesny	Juventus	8.746573	P

.....

76	alessandro bastoni	Inter	10.000000	D
----	--------------------	-------	-----------	---

77	riccardo calafiori	Bologna	9.863339	D
----	--------------------	---------	----------	---

78	francesco acerbi	Inter	9.840089	D
----	------------------	-------	----------	---

...

302	hakan calhanoglu	Inter	10.000000	C
-----	------------------	-------	-----------	---

303	teun koopmeiners	Atalanta	8.678702	C
-----	------------------	----------	----------	---

304	luis alberto	Lazio	8.240300	C
-----	--------------	-------	----------	---

....

516	lautaro martinez	Inter	10.000000	A
-----	------------------	-------	-----------	---

517	albert gudmundsson	Genoa	9.217350	A
-----	--------------------	-------	----------	---

518	matias soule	Frosinone	8.382918	A
-----	--------------	-----------	----------	---

## 2.2) Costruzione del dataset per la classificazione

Una volta completate le fasi di ragionamento e scoring, si è proceduto alla preparazione del dataset necessario per la classificazione supervisionata.

Tale attività ha previsto una serie di passaggi ben strutturati, con l'obiettivo di fornire, per ogni giocatore e per ogni giornata, un insieme di caratteristiche aggiornate che potessero sostenere un processo decisionale automatizzato circa la sua selezionabilità.

Il primo step ha riguardato l'unione dei file contenenti le predizioni e le probabilità (output della fase di Ragionamento) con il file contenente gli score normalizzati di tutti i giocatori di Serie A. È stato scelto di considerare come universo di riferimento l'intero insieme dei giocatori dotati di uno score, includendo dunque anche i portieri, pur in assenza, per la maggior parte di essi, di valori di predizione e probabilità. L'operazione di merge è stata effettuata tramite le colonne 'Giocatore' e 'Giornata', e i valori mancanti (NaN) nelle colonne Prob\_Gol e Pred\_Gol\_Pesata sono stati sostituiti con 0.0, per garantire uniformità nel dataset risultante. Questo ha permesso di ottenere, per ogni giornata, un dataset esteso (Giocatori\_PredProbStoriche\_Score\_Completo.csv) in cui ogni giocatore possiede almeno uno score, e, ove disponibili, anche le informazioni predittive.

Giocatore	Giornata	Team	Role	Score	Prob_Gol	Pred_Gol_Pesata
junior ajayi	G29	Verona	A	0.001946316329406	0.0	0.0
assan ceesay	G29	Lecco	A	0.001946316329406	0.0	0.0
alphadjo cisse	G29	Verona	A	0.001946316329406	0.0	0.0
erik botheim	G29	Salemitana	A	0.0	0.0	0.0
lautaro martinez	G30	Inter	A	10.0	0.678	0.86
albert gudmundsson	G30	Genoa	A	9.217349622639194	0.3349	0.1802
matias soule	G30	Frosinone	A	8.382918373067875	0.2773	0.1399
paulo dybala	G30	Roma	A	8.194920748790997	0.3123	0.1565
khvicha kvaratskhelia	G30	Napoli	A	8.164231058664397	0.3753	0.2577
olivier giroud	G30	AC Milan	A	7.9886033517632	0.3908	0.2365

Nel secondo passaggio è stata aggiunta, per ciascun giocatore e per ogni giornata, una nuova informazione: il numero complessivo di gol segnati nelle tre giornate precedenti (per i giocatori di movimento) e, nel caso dei portieri, il numero di gol subiti nelle ultime tre giornate.

Per i giocatori di movimento, questa informazione è stata calcolata utilizzando la matrice GolGiocatoriTemporale.csv, contenente per ciascun calciatore il numero di reti segnate in ognuna delle 38 giornate della Serie A. Il valore ottenuto è stato memorizzato nella nuova variabile GolUlt3. In assenza della giornata di riferimento (es. G39), è stata utilizzata come limite superiore l'ultima giornata disponibile (G38) per determinare l'intervallo utile.

Per quanto riguarda i portieri, i dati completi sono disponibili principalmente per i 20 titolari delle rispettive squadre, mentre per i portieri secondari le informazioni risultano più frammentarie o assenti. Per garantire coerenza e rappresentatività, si è quindi scelto di attribuire a tutti i portieri di una stessa squadra lo stesso valore, ricavato dal numero complessivo di gol subiti dalla squadra nelle tre giornate precedenti. Questo approccio consente di considerare tale dato come una proxy della prestazione difensiva collettiva, rendendo l'indicatore GolUlt3 significativo anche per i portieri non titolari.

Il calcolo dei gol subiti da ciascuna squadra è stato effettuato a partire dal file Calendario2324.csv, contenente per ogni giornata di campionato le due squadre coinvolte (casa e trasferta) e il relativo risultato (es. "2-1"). Per ciascuna squadra e per ogni giornata a partire dalla quarta (G4), è stata calcolata la somma dei gol subiti nelle tre giornate precedenti. Il risultato è stato salvato nel file Gol\_Subiti\_Ultime3\_G4\_G39.csv.

Questa integrazione è stata effettuata allo scopo di stimare una proxy temporale dello stato di forma recente, utile nella successiva fase di etichettatura supervisionata, in cui verranno considerate più variabili per classificare i giocatori. In particolare, per i portieri, il parametro che influenzerà maggiormente l'assegnazione dell'etichetta sarà lo score complessivo, ritenuto più rappresentativo della loro performance rispetto al solo numero di gol subiti.

File risultante: Giocatori\_PredProbStoriche\_Score\_GolUlt3\_Con\_Portieri.csv.

Giocatore	Giornata	Team	Role	Score	Prob_Gol	Pred_Gol_Pesata	GolUlt3
vanja milinkovic-savic	G4	Torino	P	10.0	0.0	0.0	4
yann sommer	G4	Inter	P	9.941126241517232	0.0	0.0	0
wojciech szczesny	G4	Juventus	P	8.7465731877219	0.0	0.0	1
michele di gregorio	G4	Monza	P	8.336254550357287	0.0	0.0	5
lukasz skorupski	G4	Bologna	P	8.150195496831603	0.0	0.0	4
mike maignan	G4	AC Milan	P	7.2239449912363485	0.0	0.0	0
ivan provedel	G4	Lazio	P	7.020808053570625	0.0	0.0	4
marco carnesecchi	G4	Atalanta	P	6.906655880634578	0.0	0.0	2
josep martinez	G4	Genoa	P	6.847332704148128	0.0	0.0	5
pietro terracciano	G4	Fiorentina	P	6.720596827108893	0.0	0.0	7

...

Giocatore	Giornata	Team	Role ▼	Score	Prob_Gol	Pred_Gol_Pesata	GolUlt3
arijon ibrahimovic	G4	Frosinone	C	0.4959209233230413	0.0	0.0	0
gaetano castrovilli	G4	Fiorentina	C	0.4943263222834173	0.0	0.0556	0
viktor kovalenko	G4	Empoli	C	0.4911371202041695	0.0	0.0556	0
simone bastoni	G4	Empoli	C	0.4735965087683064	0.0	0.0556	0
toma basic	G4	Salernitana	C	0.4688127056494344	0.0	0.0	0
hans nicolussi caviglia	G4	Juventus	C	0.4241638765399646	0.0	0.0	0
oussama el azzouzi	G4	Bologna	C	0.4177854723814689	0.0	0.0556	0
szymon zurkowski	G4	Empoli	C	0.3683528401531271	0.0	0.0556	0
martin hongla	G4	Verona	C	0.3524068297568878	0.0	0.0	0
eljif elmas	G4	Napoli	C	0.34921762767764	0.0	0.0556	0

...

Giocatore	Giornata	Team	Role	Score	Prob_Gol	Pred_Gol_Pesata	GolUlt3
florian thauvin	G12	Udinese	A	5.151711554124309	0.0	0.0486	0
ciro immobile	G12	Lazio	A	5.087514727829896	0.2218	0.1378	1
cyril ngonge	G12	Napoli	A	4.990652635055614	0.1198	0.0986	0
lucas beltran	G12	Fiorentina	A	4.86950125264437	0.0	0.0486	0
nikola vlastic	G12	Torino	A	4.802855811270323	0.1109	0.1023	1
dany mota	G12	Monza	A	4.6930126924409725	0.0	0.0486	0
antonio sanabria	G12	Torino	A	4.675122241620843	0.1109	0.1023	1
luka jovic	G12	AC Milan	A	4.657185663706282	0.0	0.0486	0
andrea belotti	G12	Fiorentina	A	4.51878010669156	0.1512	0.1463	0
mateo retegui	G12	Genoa	A	4.505252030670766	0.3577	0.18	0

2.3) Definizione e assegnazione delle etichette

Questa fase ha previsto la definizione e l’assegnazione delle etichette supervisionate a ciascun giocatore, con l’obiettivo di classificare il livello di consigliabilità sulla base delle informazioni precedentemente elaborate. Il processo è stato articolato in tre fasi: normalizzazione dei dati, calcolo di uno score composito e logica di assegnazione dell’etichetta.

### 2.3.1) Normalizzazione delle variabili

Per garantire un confronto coerente tra variabili con scale diverse, sono state normalizzate le seguenti:

- Score  $\rightarrow$  Score\_norm, ottenuto dividendo lo score per 10 (valore massimo atteso).
- GolUlt3  $\rightarrow$  GolUlt3\_norm, ottenuto dividendo per 3 (massimo numero di gol realizzabili nelle tre giornate precedenti).

Il file intermedio generato a valle di questa operazione è stato salvato come "Giocatori\_Normalizzati.csv".

### 2.3.2) Calcolo dello score composito

È stato quindi calcolato uno score composito differenziato per ruolo:

- Per i portieri (Role = "P"), lo score composito è stato determinato come:  
$$\text{Score\_Composito} = 0.7 * \text{Score\_norm} + 0.3 * \text{GolUlt3\_norm}$$

dove GolUlt3\_norm rappresenta, in questo caso, i gol subiti dalla squadra nelle ultime tre giornate (normalizzati).

- Per i giocatori di movimento, lo score composito è stato definito come combinazione pesata di:
  - probabilità di segnare (Prob\_Gol)
  - predizione pesata dei gol (Pred\_Gol\_Pesata)
  - score normalizzato (Score\_norm)
  - forma recente (GolUlt3\_norm)

secondo la formula:

$$\text{Score\_Composito} = 0.3 * \text{Prob\_Gol} + 0.3 * \text{Pred\_Gol\_Pesata} + 0.5 * \text{Score\_norm} + 0.2 * \text{GolUlt3\_norm}$$

### 2.3.3) Logica di assegnazione delle etichette

L'assegnazione finale delle etichette supervisionate (Etichetta) ha seguito criteri diversi per portieri e giocatori di movimento.

- Portieri:
  - 0  $\rightarrow$  non consigliato, se Score\_norm == 0 (portiere mai utilizzato) oppure Score\_Composito < 0.50 e GolUlt3\_norm  $\geq$  1
  - 1  $\rightarrow$  consigliato, se Score\_Composito  $\geq$  0.50
  - 2  $\rightarrow$  sorpresa, se GolUlt3\_norm < 1 e Score\_Composito < 0.50
- Giocatori di movimento:

- Se il giocatore ha segnato almeno un gol nelle ultime tre giornate (GolUlt3\_norm > 0):
  - 1 → consigliato, se Score\_Composito  $\geq 0.40$
  - 2 → sorpresa, se Score\_Composito  $\geq 0.15$  ma  $< 0.40$
  - 0 → non consigliato, se Score\_Composito  $< 0.15$
- Se non ha segnato recentemente (GolUlt3\_norm == 0):
  - 1 → consigliato, solo se Score\_Composito  $\geq 0.45$
  - 0 → non consigliato, altrimenti

Il file finale generato, contenente le etichette, è stato salvato come "Giocatori\_Con\_Etichetta.csv".

Giocatore	Giornata	Team	Role	Score	Prob_Gol	Pred_Gol_Pesata	GolUlt3	Score_norm	GolUlt3_norm	Score_Composito	Etichetta
juan musso	G4	Atalanta	P	1.6179048132668192	0.0	0.0	2	0.16179048132668192	0.6666666666666666	0.3132533369286773	2
mile sviar	G4	Roma	P	1.3482540110556829	0.0	0.0	6	0.1348254011055683	2.0	0.6943777807738978	1
federico ravaglia	G4	Bologna	P	1.0786032088445463	0.0	0.0	4	0.10786032088445463	1.3333333333333333	0.4755022246191182	0
marco silvestri	G4	Udinese	P	1.0786032088445463	0.0	0.0	4	0.10786032088445463	1.3333333333333333	0.4755022246191182	0
christos mandas	G4	Lazio	P	1.0786032088445463	0.0	0.0	4	0.10786032088445463	1.3333333333333333	0.4755022246191182	0
etrit berisha	G4	Empoli	P	1.0786032088445463	0.0	0.0	5	0.10786032088445463	1.6666666666666667	0.5755022246191183	1
boris radunovic	G4	Cagliari	P	0.5393016044222732	0.0	0.0	4	0.053930160442227315	1.3333333333333333	0.437751123095591	0
michele cerofolini	G4	Frosinone	P	0.5393016044222732	0.0	0.0	4	0.053930160442227315	1.3333333333333333	0.437751123095591	0
marco sportiello	G4	AC Milan	P	0.5393016044222732	0.0	0.0	0	0.053930160442227315	0.0	0.0377511230955912	2
emil audero	G4	Inter	P	0.5393016044222732	0.0	0.0	0	0.053930160442227315	0.0	0.0377511230955912	2

...

Giocatore	Giornata	Team	Role	Score	Prob_Gol	Pred_Gol_Pesata	GolUlt3	Score_norm	GolUlt3_norm	Score_Composito	Etichetta
alexis sanchez	G31	Inter	A	4.293440294623524	0.1758	0.1238	1	0.4293440294623524	0.3333333333333333	0.37121868139784286	2
zito iuvumbo	G31	Cagliari	A	4.218342271585198	0.1536	0.1452	0	0.42183422715851976	0.0	0.30055711357925985	0
stephan el shaarawy	G31	Roma	A	4.18244125420769	0.0956	0.0877	0	0.41824412542076905	0.0	0.2641120627103845	0
lameck banda	G31	Lecce	A	4.155316875643998	0.0586	0.0746	0	0.41553168756439984	0.0	0.24772584378219992	0
arkadiusz milik	G31	Juventus	A	4.105186360170173	0.0956	0.1169	1	0.41051863601701727	0.3333333333333333	0.3356759846751753	2
pontus almqvist	G31	Lecce	A	4.09262020178492	0.0439	0.0746	0	0.40926202017849195	0.0	0.24018101008924597	0
valentin castellanos	G31	Lazio	A	4.056707085209281	0.111	0.3322	2	0.40567070852092807	0.6666666666666666	0.46912868759379733	1
nicolo cambiaghi	G31	Empoli	A	3.973908584137184	0.0	0.0486	0	0.3973908584137184	0.0	0.2132754292068592	0
federico bonazzoli	G31	Verona	A	3.962712806190207	0.1758	0.1033	1	0.39627128061902067	0.3333333333333333	0.348532306976177	2
darko lazovic	G31	Verona	A	3.928057418709881	0.0268	0.0616	0	0.3928057418709881	0.0	0.22292287093549407	0

Tutto ciò costituisce la base necessaria per l'avvio della fase di classificazione supervisionata, poiché consente di associare a ogni giocatore, per ciascuna giornata, un insieme di variabili descrittive (feature) e un'etichetta esplicita (label). Questo rende possibile l'addestramento di modelli predittivi in grado di generalizzare la consigliabilità dei giocatori anche in giornate future o su nuovi dati. Le etichette sono state calcolate dalla 4 fino alla 38esima giornata, per la 39esima il campo etichetta è vuoto per ogni giocatore proprio perché questo è ciò che desideriamo ottenere nella fase successiva, in quella di Apprendimento.



Giocatore	Giornata	Team	Role	Score	Prob_Gol	Pred_Gol_Pesata	GolUlt3	Score_norm	GolUlt3_norm	Score_Composito	Etichetta
gennaro borrelli	G39	Frosinone	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
hirving lozano	G39	Napoli	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
giacopo dosogus	G39	Cagliari	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
luigi cherubini	G39	Roma	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
sekou diawara	G39	Udinese	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
leonardo cerri	G39	Juventus	A	0.001946316329406	0.0213	0.0611	0	0.0001946316329406	0.0	0.0248173158164703	
joaquin correa	G39	Inter	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
soufiane bidaoui	G39	Frosinone	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
andrea ferraris	G39	Monza	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	
joel voelkerling persson	G39	Lecce	A	0.001946316329406	0.0	0.0	0	0.0001946316329406	0.0	9.73158164703e-05	

## 2.4) Classificazione Supervisionata

La classificazione supervisionata è una tecnica dell'apprendimento automatico che consente a un algoritmo di apprendere una funzione di mappatura tra input (feature) e output (etichette) a partire da un insieme di dati etichettati. L'obiettivo è costruire un modello in grado di assegnare correttamente una classe a nuove istanze mai viste, sulla base delle informazioni fornite durante la fase di addestramento. Questo approccio richiede quindi la disponibilità di una base storica contenente, per ogni esempio, sia le caratteristiche osservabili sia la classe di appartenenza corretta (ground truth).

Nel contesto del nostro progetto, la classificazione supervisionata è stata utilizzata per prevedere, partendo dalla 38ª giornata, la previsione per la 39ª, sulla base delle loro prestazioni storiche, delle previsioni di rendimento e della forma recente. L'algoritmo è stato addestrato a riconoscere pattern significativi nei dati al fine di assegnare automaticamente, per ciascun giocatore, un'etichetta rappresentativa del suo stato atteso nella giornata successiva.

La fase di classificazione supervisionata ha avuto come obiettivo principale l'attribuzione automatica, per ciascun giocatore e per la giornata futura, di un'etichetta che rappresentasse il suo stato prestazionale atteso nella giornata immediatamente successiva.

Le classi previste dal sistema sono tre:

- “non consigliato” (etichetta 0);
- “consigliato” (etichetta 1);
- “sorpresa” (etichetta 2).

Quest'ultima categoria è stata introdotta per evidenziare quei giocatori che, pur avendo uno score aggregato stagionale non elevato, avevano mostrato nelle ultime giornate segnali di forte crescita in termini di rendimento (es. gol improvvisi o consistenti). L'introduzione di questa classe ha implicato una maggiore complessità nella definizione dei pattern di classificazione, richiedendo un'attenzione specifica alla costruzione del dataset e alla scelta del modello predittivo più adatto.



Il dataset supervisionato è stato costruito includendo, per ogni giocatore e per ogni giornata a partire dalla G4 fino alla G38, un insieme di feature esplicative, selezionate sulla base di criteri tecnici e semantici. Le variabili utilizzate come input sono state:

- lo score normalizzato (una misura statistica aggregata delle prestazioni complessive del giocatore durante l'intero campionato);
- la probabilità stimata che il giocatore segnasse nella giornata specifica (ottenuta tramite un modello probabilistico, ad es. catena di Markov);
- la predizione del numero di gol attesi per quella stessa giornata (calcolata con un regressore come Decision Tree Regressor);
- il numero di gol segnati nelle ultime tre giornate precedenti (variabile temporale, con ruolo informativo cruciale per la classe "sorpresa").

La preparazione del dataset è stata seguita da un'attenta fase di separazione tra set di addestramento e set di test. Per garantire una valutazione realistica e non distorta, si è deciso di utilizzare come set di addestramento tutte le giornate dalla G4 alla G38, mentre la giornata G39 è stata esclusa da questa fase in quanto oggetto della predizione finale. Le etichette presenti per le giornate G4–G38 sono state definite sulla base di criteri combinati tra score statico e performance recenti, già integrate nel file di origine, e sono state trattate come ground truth per l'addestramento del modello.

In fase di modellazione, sono stati presi in considerazione due approcci classificatori: un Decision Tree Classifier, scelto per la sua elevata interpretabilità e la coerenza logica con la struttura semantica delle classi, e un Random Forest Classifier, selezionato per la sua maggiore robustezza e capacità di generalizzazione rispetto a dati complessi o leggermente rumorosi. Entrambi i modelli sono stati addestrati sul medesimo training set e successivamente valutati sullo stesso validation set, al fine di garantire un confronto equo e oggettivo.

La valutazione delle prestazioni è stata condotta attraverso un insieme di metriche numeriche standard, ciascuna delle quali fornisce un'informazione complementare sul comportamento del modello:

- **Accuracy:** indica la proporzione complessiva di predizioni corrette rispetto al totale degli esempi valutati. È una misura globale, utile quando le classi sono bilanciate, ma meno affidabile in presenza di classi sbilanciate.
- **Precision macro:** calcola la precisione (cioè la percentuale di predizioni corrette tra tutte quelle fatte per una determinata classe) per ciascuna classe e poi ne fa la media aritmetica. Questa metrica è particolarmente utile per valutare quanto il modello sia affidabile nel fare affermazioni positive su ogni classe, indipendentemente dalla loro frequenza.

- **Recall macro:** misura, per ciascuna classe, la capacità del modello di individuare correttamente tutti i casi appartenenti a quella classe, e successivamente ne calcola la media. È cruciale in scenari dove la capacità di intercettare tutte le istanze di una classe, come nel caso dei giocatori “sorpresa”, è prioritaria.
- **F1-score macro:** rappresenta la media armonica tra precision e recall per ciascuna classe, calcolata poi come media delle classi. Questo indicatore sintetizza il bilanciamento tra falsi positivi e falsi negativi, ed è particolarmente utile quando si desidera un equilibrio tra completezza e precisione.
- **Cohen’s Kappa:** misura l’accordo tra le etichette predette e le etichette reali, correggendo per il caso. A differenza dell’accuracy, considera la probabilità che l’accordo tra predizioni e realtà avvenga per puro caso, fornendo così una stima più robusta della qualità del modello.
- **Log-loss** (o cross-entropy loss): valuta la bontà delle probabilità predette dal modello. Penalizza maggiormente le predizioni sbagliate quando vengono fatte con elevata confidenza. Un log-loss più basso indica non solo che il modello predice correttamente, ma che è anche ben calibrato nel fornire stime probabilistiche affidabili.
- **Tempo di inferenza:** rappresenta il tempo computazionale necessario affinché il modello produca le predizioni sui dati di test. Pur non influenzando direttamente la qualità delle predizioni, è un parametro rilevante in applicazioni dove la velocità di risposta è un requisito operativo.

L’uso congiunto di queste metriche ha permesso una valutazione approfondita dei due modelli, garantendo non solo una misura dell’accuratezza generale, ma anche una comprensione più dettagliata della loro capacità di trattare correttamente tutte le classi, incluse quelle meno rappresentate e più complesse da identificare.

### Confronto tra modelli:

	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)	\
Random Forest	0.997719	0.992111	0.986811	0.989446	
Decision Tree	0.987145	0.986547	0.899341	0.935786	

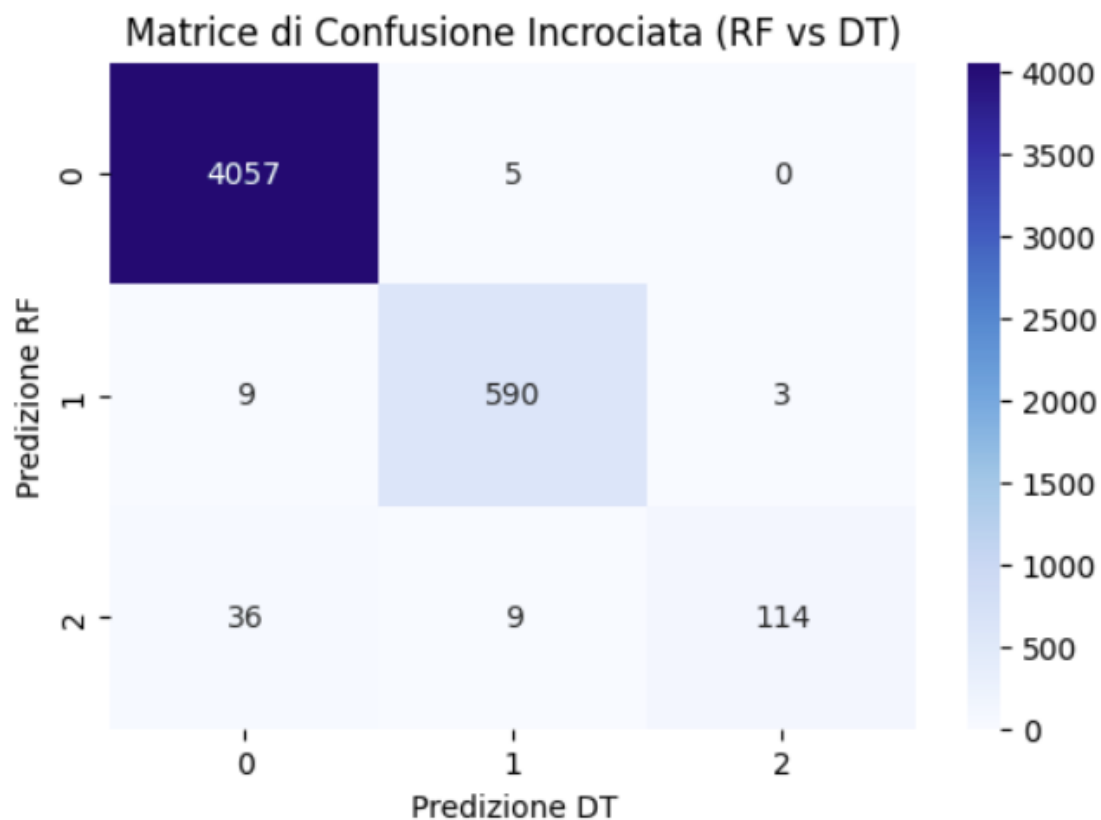
	Cohen Kappa	Log Loss	Prediction Time (s)
Random Forest	0.991700	0.043299	0.038019
Decision Tree	0.952043	0.069273	0.001889

L’analisi dei risultati ha mostrato un evidente vantaggio da parte del modello Random Forest in quasi tutte le metriche. In particolare, l’accuracy raggiunta è stata del 99.77% contro il 98.71% del Decision Tree, ma il divario più significativo si è registrato nel recall macro (98.68% vs 89.93%) e nel F1-score macro (0.989 vs

0.935). Questi due indicatori, fondamentali in un contesto multi-classe con una classe minoritaria e strategicamente rilevante come la “sorpresa”, hanno evidenziato una maggiore capacità del Random Forest di riconoscere correttamente i giocatori non solo affermati, ma anche emergenti. La metrica Cohen’s Kappa, che misura l’accordo tra le predizioni del modello e le etichette reali corretto per il caso, ha ulteriormente confermato la superiorità del Random Forest (0.9917 vs 0.9520), indicando un accordo quasi perfetto. Anche il log loss è risultato inferiore per il Random Forest, denotando una miglior calibratura delle probabilità predette.

Oltre alle metriche di valutazione standard applicate a ciascun modello singolarmente, è stato introdotto un ulteriore strumento di confronto basato sulla matrice di confusione incrociata tra modelli. Tale matrice, costruita a partire dalle predizioni effettuate da entrambi i modelli sul medesimo validation set, consente di analizzare in modo dettagliato il grado di accordo o disaccordo tra le decisioni prese dal Random Forest e dal Decision Tree per ogni singolo esempio.

Nella matrice incrociata, le righe rappresentano le predizioni del modello Random Forest e le colonne quelle del Decision Tree. Una distribuzione fortemente diagonale indica che i modelli tendono a restituire la stessa etichetta, mentre la presenza significativa di valori fuori diagonale segnala divergenze predittive, potenzialmente rilevanti in termini di interpretazione o strategia operativa.



Predizioni identiche: 4761 su 4823 (98.71%)

Il valore di accordo pari al 98.71% è stato ottenuto confrontando, istanza per istanza, le etichette predette dal modello Random Forest con quelle predette dal Decision

Tree sul medesimo validation set. In particolare, si è costruito un vettore binario che, per ogni esempio, assume valore 1 se i due modelli hanno fornito la stessa predizione e 0 in caso contrario. Calcolando la media aritmetica di questo vettore, si ottiene la proporzione di predizioni coincidenti tra i due modelli.

Formalmente, la metrica è definita come:

$$\text{Accordo \%} = (\text{Numero di predizioni identiche} / \text{Numero totale di esempi}) * 100$$

Nel caso specifico, su un totale di 4.823 esempi nel validation set, i due modelli hanno prodotto la stessa etichetta in 4.761 casi. Pertanto:


$$(4761 / 4823) * 100 = 98.71\%.$$

Il risultato ottenuto evidenzia un'elevata coerenza tra i due modelli, pur lasciando spazio a una differenza qualitativa nelle predizioni, che ha poi guidato la scelta del modello finale.

L'analisi della matrice di confusione incrociata ha mostrato che le principali discrepanze tra i due classificatori si sono concentrate nella classe “sorpresa” (etichetta 2). In particolare, in 36 casi il Random Forest ha correttamente etichettato un giocatore come sorpresa, mentre il Decision Tree ha assegnato erroneamente l'etichetta 0 (“non consigliato”), riducendone così il potenziale valore predittivo. Questa tendenza conservativa del Decision Tree, che emerge anche da altri valori fuori diagonale, ha contribuito in modo significativo alla scelta finale del modello da adottare.

Nonostante il Decision Tree abbia mostrato un tempo di inferenza significativamente più breve (0.0024s contro 0.1131s), tale differenza non è stata considerata rilevante ai fini applicativi, trattandosi di un contesto offline o comunque batch-oriented. La leggibilità e la trasparenza del Decision Tree, pur rappresentando un valore aggiunto, non hanno compensato il gap prestazionale, soprattutto in termini di recall, ritenuto centrale per evitare la sottovalutazione di giocatori potenzialmente decisivi.

Alla luce dei risultati numerici, della stabilità statistica del modello e della sua capacità di gestire situazioni complesse come quella della classe “sorpresa”, si è giunti alla decisione finale di adottare il Random Forest Classifier come algoritmo ufficiale per l'assegnazione delle etichette nella giornata G39. Tale classificazione è stata salvata in un file CSV contenente, per ciascun giocatore, le feature previsionali e l'etichetta assegnata, pronto per essere integrato nella Fase 1 (CSP) del sistema.

351 to 360 of 689 entries  

Giocatore	Team	Role	Score_norm	Prob_Gol	Pred_Gol_Pesata	Etichetta_Predetta
razvan marin	Empoli	C	0.5358576940183092	0.0	0.0	non consigliato
valentin carboni	Monza	C	0.5343029826889025	0.0749	0.0861	non consigliato
davide fratesi	Inter	C	0.5335925583046782	0.1831	0.1528	consigliato
michel aebischer	Bologna	C	0.5320367732114135	0.0	0.0	non consigliato
tomas suslov	Verona	C	0.5303053228194105	0.3327	0.1375	consigliato
matias vecino	Lazio	C	0.5286469109969563	0.1603	0.215	consigliato
enzo barrenechea	Frosinone	C	0.5240230513814849	0.0	0.0	non consigliato
maxime lopez	Fiorentina	C	0.520438778255191	0.0	0.0	non consigliato
grigoris kastanos	Salernitana	C	0.5199983499417313	0.0749	0.0861	non consigliato
milan badelj	Genoa	C	0.5165731358007173	0.0213	0.0903	sorpresa

L'intera fase si è così conclusa con un modello affidabile, validato e coerente con la semantica sportiva del progetto, in grado di offrire suggerimenti automatizzati con alto grado di accuratezza e significatività tecnica. La scelta del Random Forest è quindi motivata non solo dalle sue prestazioni superiori in termini quantitativi, ma anche dalla sua maggiore sensibilità nella rilevazione di quei profili meno ovvi e più volatili – come la classe “sorpresa” – che risultano strategici nel contesto applicativo del sistema. Tale caratteristica, assente nel modello ad albero singolo, ha rappresentato l'elemento determinante nell'identificazione del classificatore più adatto allo scopo progettuale.

## 2.5) Clustering – non supervisionato

Nel nostro progetto è stata implementata una fase di clustering non supervisionato finalizzata all'identificazione di profili funzionali all'interno di ciascun ruolo di movimento, ovvero, quello dei difensori, centrocampisti e attaccanti.

Il clustering non supervisionato è una tecnica di apprendimento automatico che permette di raggruppare elementi simili tra loro sulla base delle loro caratteristiche, senza richiedere etichette o classi predefinite.

Nel contesto calcistico, questa tecnica consente di individuare sottocategorie di giocatori all'interno di uno stesso ruolo, con lo scopo di arricchire l'analisi tattica e migliorare la costruzione della formazione ideale.

A partire dal dataset StatisticheGiocatori.csv, sono stati selezionati tutti i giocatori con ruolo “D” (difensori) che avessero giocato almeno 500 minuti nel corso della stagione. Tale soglia è stata introdotta per garantire una base statistica affidabile ed evitare che giocatori con scarso minutaggio venissero assegnati erroneamente a un cluster.

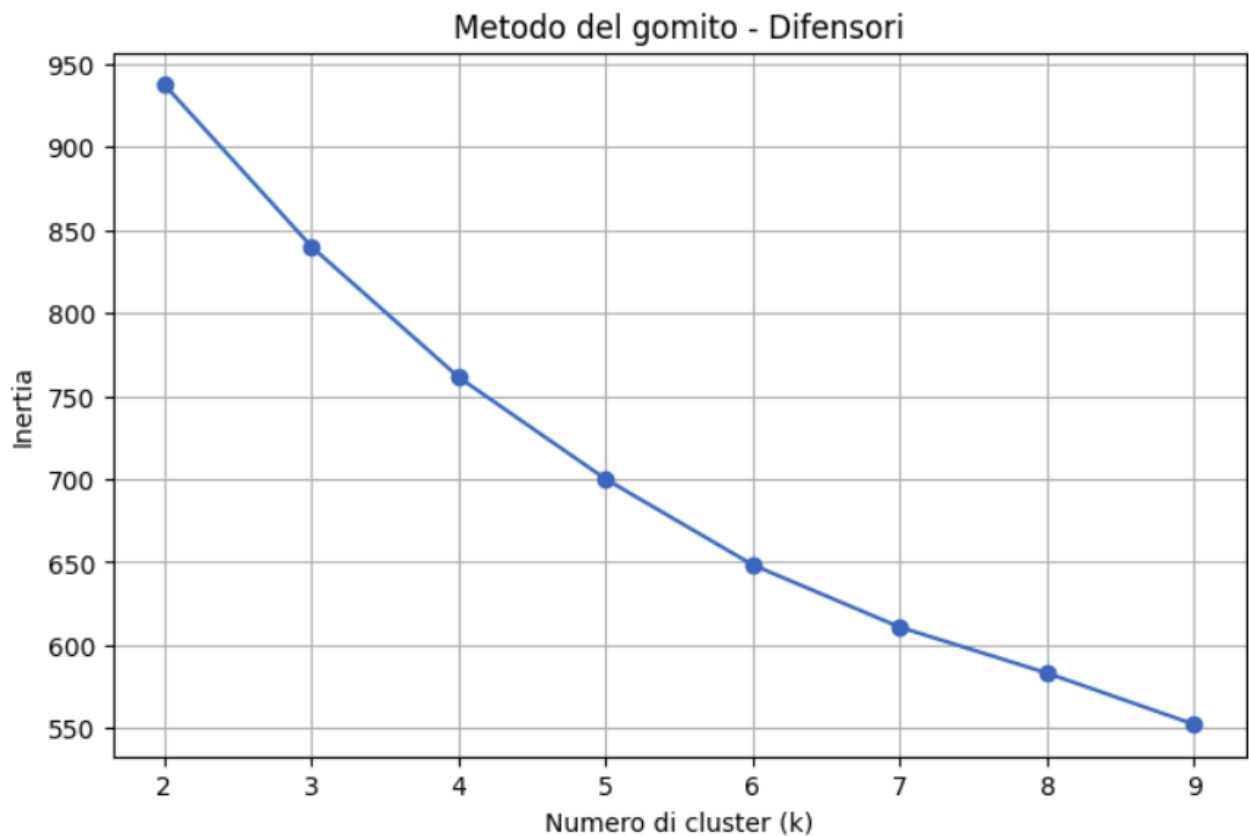
Per la rappresentazione dei difensori sono state selezionate dieci feature significative, in grado di descrivere sia aspetti difensivi che contributi offensivi e qualità tecniche:

- Tackles per 90;

- Tackles Success Rate (%);
- Interceptions per 90;
- Clearances per 90;
- Blocks per 90;
- Actual Goals;
- Shots per 90;
- Successful Dribbles per 90;
- Actual Assists;
- Pass Success (%)

Queste variabili sono state sottoposte a una normalizzazione standard (z-score) utilizzando la funzione `StandardScaler` di `sklearn`, al fine di uniformare le scale dei valori ed evitare che le feature con valori numerici più grandi influenzassero maggiormente il calcolo delle distanze euclidee tra i punti, operazione centrale nel clustering perché determina quanto due elementi sono simili o dissimili, influenzando così l'assegnazione ai diversi gruppi.

Per determinare il numero ottimale di cluster da generare è stato applicato il metodo del gomito. Sono stati calcolati i valori di inerzia (somma delle distanze intra-cluster) per diversi valori di  $k$  compresi tra 2 e 9.



Il grafico risultante ha mostrato una significativa riduzione dell'inerzia fino a  $k = 5$ , oltre il quale il miglioramento risultava marginale. È stato quindi scelto  $k = 5$  come valore ottimale, che ha permesso di ottenere una buona differenziazione tra i profili mantenendo al contempo l'interpretabilità dei gruppi.

L'algoritmo KMeans è stato quindi applicato con  $k = 5$ . Ogni difensore è stato assegnato a uno dei cinque cluster individuati. Per ciascun cluster sono state calcolate le medie delle feature utilizzate, al fine di identificare le caratteristiche dominanti.

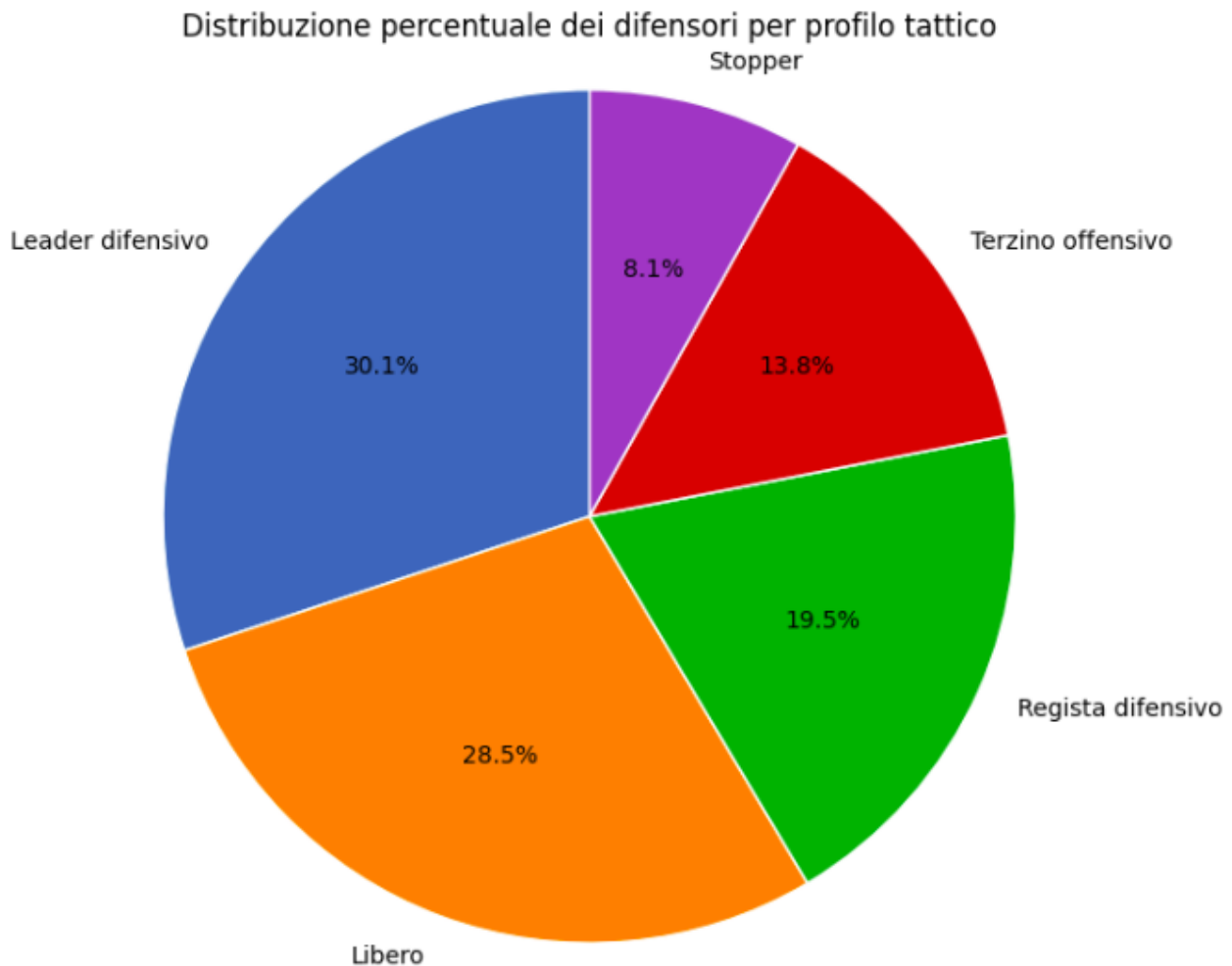
Cluster	Tackles per 90	Tackle Success Rate (%)	Interceptions per 90	Clearances per 90	Blocks per 90	Actual Goals	Shots per 90	Successful Dribbles per 90	Actual Assists	Pass Success (%)
0	1.04	63.51	0.75	1.76	0.34	2.41	1.14	0.58	3.53	85.55
1	0.67	51.92	0.82	3.06	0.76	1.08	0.44	0.15	0.67	88.20
2	1.46	67.30	1.09	2.12	0.26	0.40	1.05	1.30	0.50	73.15
3	1.01	60.63	0.73	1.75	0.28	0.46	0.53	0.63	1.89	80.44
4	1.04	63.45	1.32	3.76	0.76	0.59	0.51	0.17	0.57	84.53

L'analisi dei profili risultanti ha consentito di assegnare manualmente a ciascun gruppo un'etichetta semantica rappresentativa del ruolo tattico svolto dal giocatore all'interno della squadra. Le etichette assegnate sono le seguenti:

- Cluster 0 = Terzino offensivo: difensore laterale con forte propensione alla spinta, al dribbling e alla partecipazione alla manovra offensiva;
- Cluster 1 = Regista difensivo: centrale tecnico con alto tassi di precisione nei passaggi, fondamentale nell'impostazione da dietro;
- Cluster 2 = Stopper: difensore aggressivo, con numeri elevati in tackle e intercetti, poco coinvolto nella fase di costruzione;
- Cluster 3 = Libero: difensore equilibrato, con valori medi in tutte le statistiche, adatto a più contesti;
- Cluster 4 = Leader difensivo: centrale dominante nell'area di rigore, efficace nelle chiusure e nelle respinte, ma meno presente in fase offensiva.

A ciascun giocatore è stata quindi assegnata una colonna "Etichetta\_Tattica" che rappresenta il profilo funzionale di appartenenza. Il file risultante, contenente tutte le informazioni e le etichette assegnate, è stato salvato localmente con il nome Clustering\_Difensori.csv.

Player	Team	Position	Etichetta_Tattica
adam masina	Torino	D	Leader difensivo
johan vasquez	Genoa	D	Leader difensivo
joao ferreira	Udinese	D	Stopper
alessio romagnoli	Lazio	D	Leader difensivo
victor kristiansen	Bologna	D	Libero
valentino lazaro	Torino	D	Libero
valentin gendrey	Lecce	D	Libero
alessandro zanoli	Salernitana	D	Libero
alessandro vogliacco	Genoa	D	Leader difensivo
alessandro florenzi	AC Milan	D	Terzino offensivo



Lo stesso procedimento è stato fatto sia per i centrocampisti che per gli attaccanti.

A partire dal dataset StatisticheGiocatori.csv, sono stati selezionati i giocatori con ruolo "C" (centrocampisti) che avessero accumulato almeno 500 minuti in stagione. Questo filtro è stato introdotto per garantire che le statistiche utilizzate avessero una base solida e non fossero alterate da campioni troppo piccoli.

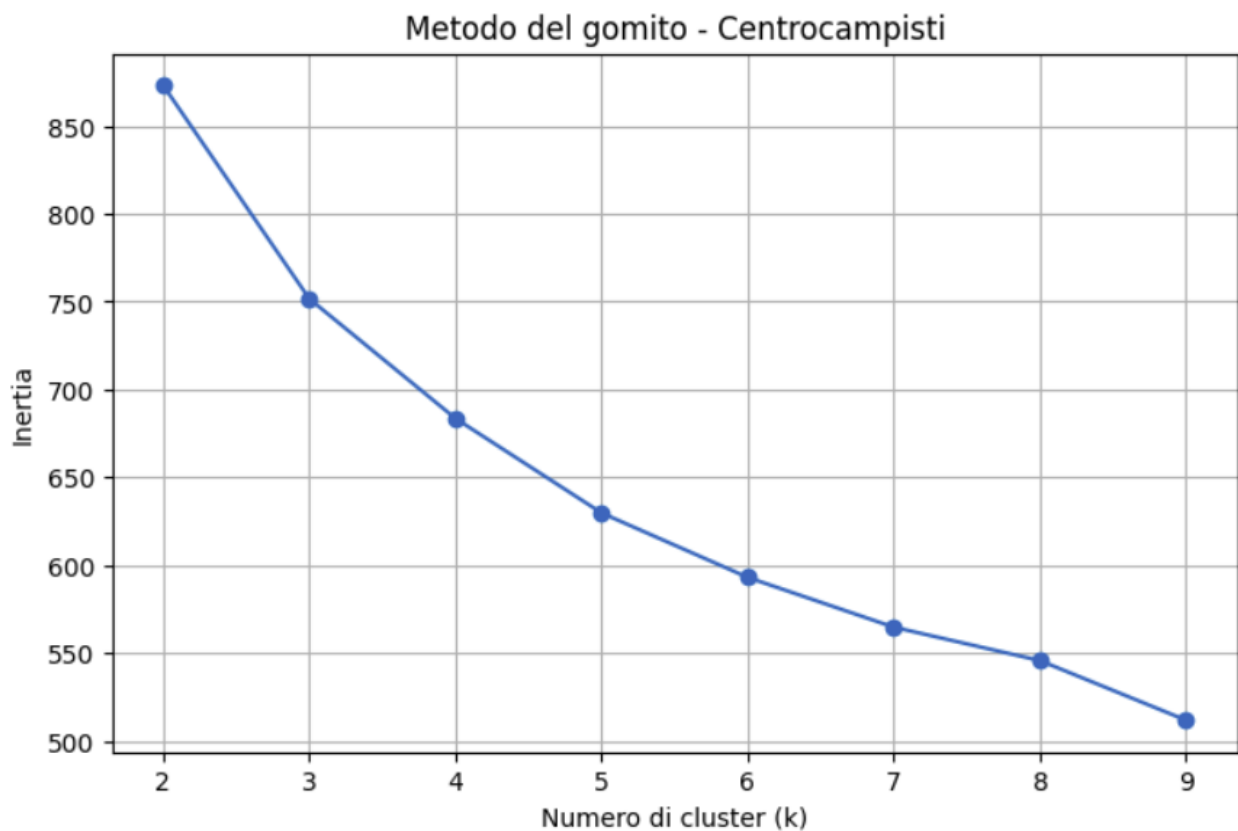
Sono state individuate dieci variabili descrittive significative per il ruolo di centrocampista, scelte per rappresentare sia le qualità offensive che quelle difensive e distributive:



- Actual Assists
- Expected Assists (xA)
- Shots per 90
- Actual Goals
- Pass Success (%)
- Accurate Passes per 90
- Successful Dribbles per 90
- Dribble Success Rate (%)
- Tackles per 90
- Interceptions per 90

Tutte le feature sono state sottoposte a una normalizzazione standard (z-score) utilizzando la funzione `StandardScaler` di `sklearn`, al fine di uniformare le scale dei valori ed evitare che le feature con valori numerici più grandi influenzassero maggiormente il calcolo delle distanze euclidee tra i punti, operazione centrale nel clustering perché determina quanto due elementi sono simili o dissimili, influenzando così l'assegnazione ai diversi gruppi.

Per determinare il numero ottimale di cluster, è stato impiegato il metodo del gomito anche qui.



L'analisi ha suggerito che  $k = 5$  fosse un buon compromesso tra coesione interna dei gruppi e differenziazione tra i cluster.

L'algoritmo KMeans è stato quindi applicato con  $k = 5$ , e ogni centrocampista è stato assegnato a uno dei cinque cluster.

Cluster	Actual Assists	Expected Assists (xA)	Shots per 90	Actual Goals	Pass Success (%)	Accurate Passes per 90	Successful Dribbles per 90	Dribble Success Rate (%)	Tackles per 90	Interceptions per 90
0	1.47	1.28	0.84	0.90	83.79	35.50	0.53	49.92	1.20	0.99
1	1.83	1.70	1.88	3.42	77.80	25.10	0.71	47.53	0.73	0.46
2	1.76	1.39	0.90	1.48	88.90	51.71	0.63	67.52	1.16	0.95
3	4.24	4.14	1.64	5.05	84.52	40.18	0.81	51.02	0.86	0.59
4	1.60	1.44	1.61	1.47	78.32	24.53	1.57	56.24	1.43	0.66

Le medie delle feature all'interno di ciascun cluster sono state successivamente analizzate per identificare le caratteristiche distintive di ogni gruppo e attribuire un'etichetta tattica coerente con il profilo emerso. Le etichette sono state assegnate manualmente, sulla base del comportamento statistico medio dei giocatori appartenenti a ciascun gruppo.

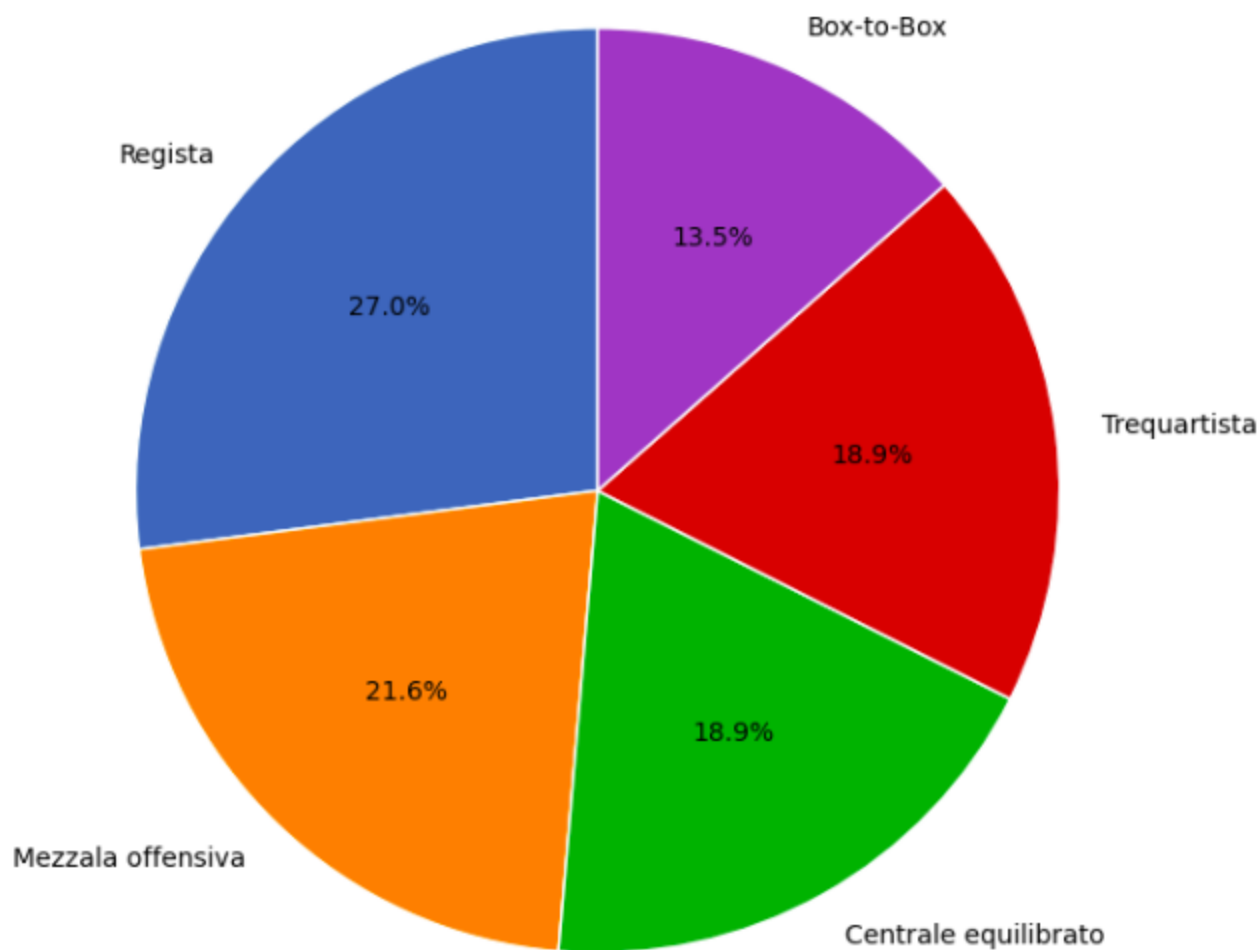
I cluster sono stati così interpretati:

- Cluster 0 – **Regista**: centrocampista con alta percentuale di passaggi completati e buon volume di gioco; partecipa poco alla fase offensiva ma è essenziale nella costruzione.
- Cluster 1 – **Mezzala offensiva**: centrocampista dinamico con numeri elevati in termini di gol, tiri e assist; partecipa attivamente alla fase offensiva.
- Cluster 2 – **Centrale equilibrato**: profilo bilanciato, eccelle nella distribuzione palla ma anche nelle coperture; passaggi precisi, contrasti e intercetti ben distribuiti.
- Cluster 3 – **Trequartista**: centrocampista offensivo puro, alto numero di assist, expected assist e goal; crea occasioni e finalizza.
- Cluster 4 – **Box-to-Box**: giocatore completo, molto attivo sia nella fase di attacco che in quella difensiva, con alto numero di dribbling e tackle.

Il risultato finale è un dataset arricchito con una nuova colonna "Etichetta\_Tattica", che riporta il profilo funzionale del centrocampista secondo il cluster di appartenenza. Il file risultante, contenente tutte le informazioni e le etichette assegnate, è stato salvato localmente con il nome Clustering\_Centrocampisti.csv.

Player	Team	Position	Etichetta_Tattica
adrien tameze	Torino	C	Centrale equilibrato
adrien rabiot	Juventus	C	Trequartista
alfred duncan	Fiorentina	C	Trequartista
alexis blin	Lecce	C	Regista
warren bondo	Monza	C	Centrale equilibrato
kacper urbanski	Bologna	C	Box-to-Box
jonathan ikone	Fiorentina	C	Mezzala offensiva
joan gonzalez	Lecce	C	Box-to-Box
karol linetty	Torino	C	Regista
uros racic	Sassuolo	C	Centrale equilibrato

Distribuzione percentuale dei centrocampisti per profilo tattico

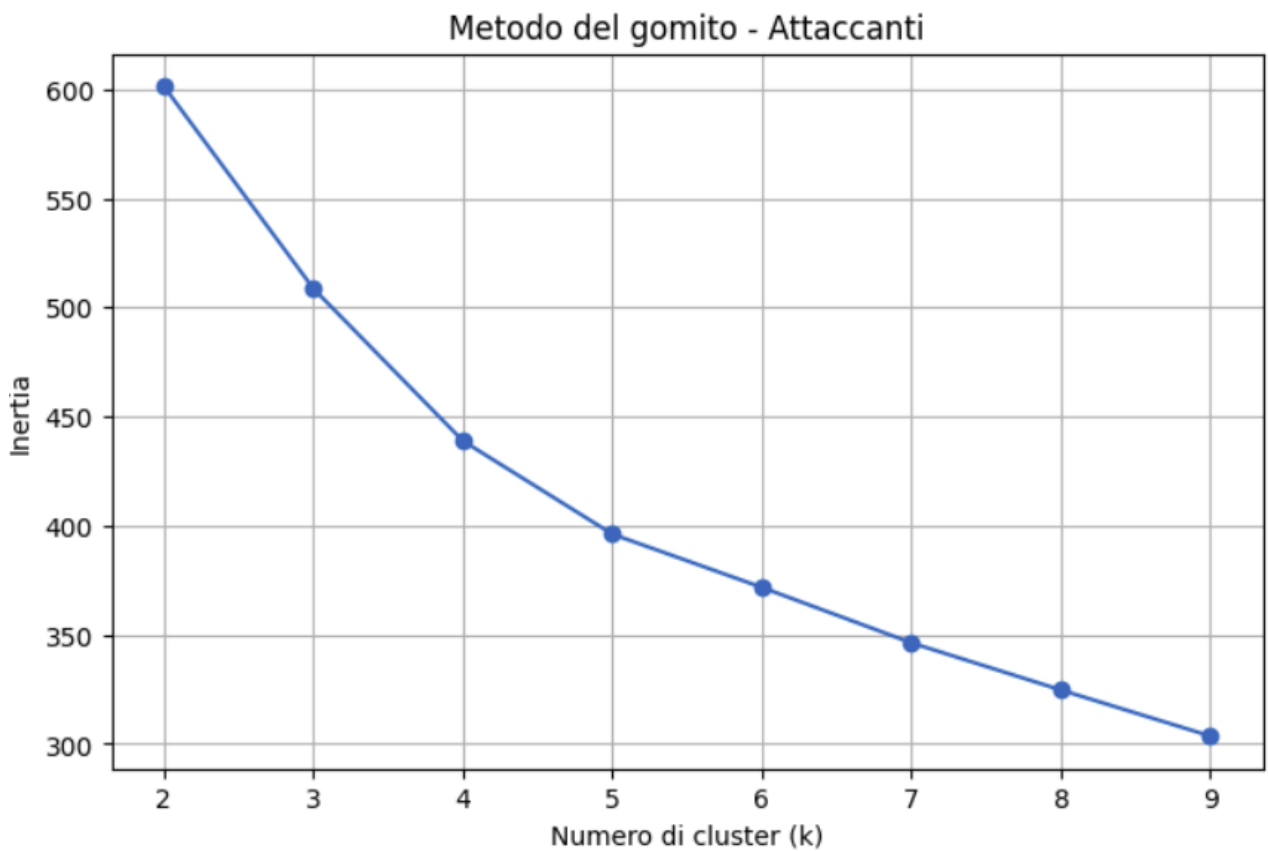


Lo stesso procedimento ora è stato fatto per gli attaccanti. In una prima fase del lavoro, è stato effettuato un clustering utilizzando una selezione iniziale di 10 variabili chiave relative al ruolo offensivo, le quali sono:

- Actual Goals;
- Expected Goals (xG);
- Shots per 90;
- Shot Conversion Rate (%);
- Actual Assists;
- Expected Assists (xA);

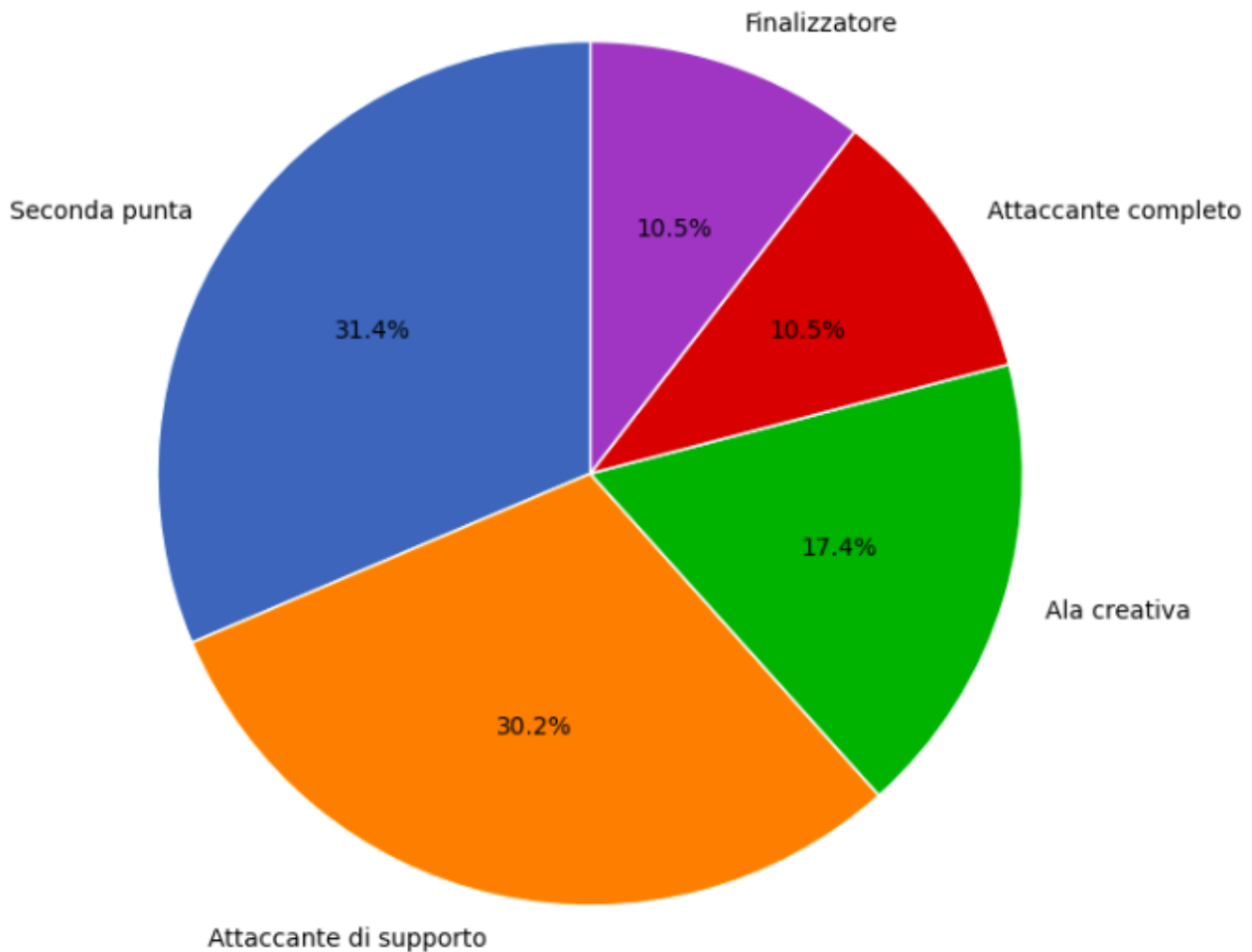
- Pass Success (%);
- Successful Dribbles per 90;
- Fouls Won per 90;
- Chances Created.

Il numero di cluster è stato fissato a  $k = 5$  sulla base dell'analisi del metodo del gomito, che ha mostrato un buon equilibrio tra coesione interna e separabilità. Sebbene i risultati ottenuti fossero coerenti sul piano tecnico, i cluster emersi non restituivano sempre profili chiaramente interpretabili o tatticamente distintivi, in particolare per la distinzione tra punte centrali, ali e attaccanti secondari.



	Actual Goals	Expected Goals (xG)	Shots per 90	Shot Conversion Rate (%)	Actual Assists	Expected Assists (xA)	Pass Success (%)	Successful Dribbles per 90	Fouls Won per 90	Chances Created
Cluster										
0	14.78	12.92	3.40	17.36	4.44	2.02	72.31	0.80	0.54	29.11
1	1.46	2.18	1.77	6.18	1.58	1.63	78.60	1.30	0.07	17.31
2	5.40	5.61	2.67	9.45	2.87	3.11	76.55	1.55	1.18	36.40
3	4.78	5.20	2.49	13.79	1.26	0.84	70.13	0.54	0.70	12.89
4	11.00	9.34	2.63	16.58	6.78	5.41	81.59	1.99	1.16	61.00

Distribuzione percentuale degli attaccanti per profilo tattico



Per migliorare la qualità del clustering, è stato deciso di estendere la lista delle variabili includendo nuove statistiche in grado di rappresentare meglio la completezza di un attaccante, il suo impatto creativo e la sua capacità di rifinire. Le nuove feature inserite sono state:

- Shot Accuracy (%);
- Big Chances Missed;
- Secondary Assists;
- Chances Created.

Con la nuova lista composta da 13 variabili è stato rieseguito il clustering con  $k = 5$ . Il risultato è stato nettamente più soddisfacente: i cluster emersi hanno mostrato profili distinti, interpretabili e pienamente coerenti con le tipologie offensive moderne.

cluster	Actual Goals	Expected Goals (xG)	Shots per 90	Shot Conversion Rate (%)	Shot Accuracy (%)	Big Chances Missed	Actual Assists	Expected Assists (xA)	Pass Success (%)	Successful Dribbles per 90	Fouls Won per 90	Chances Created	Secondary Assists
0	3.89	4.03	2.29	8.76	34.04	3.74	2.84	2.69	78.45	1.42	0.92	29.47	2.69
1	1.35	2.07	1.71	6.66	30.42	2.09	1.17	1.12	76.62	1.22	0.09	13.61	0.89
2	14.00	12.79	3.41	16.38	39.90	11.60	4.10	1.99	71.82	0.83	0.69	30.30	1.99
3	5.18	5.73	2.63	14.10	41.85	5.64	1.23	0.85	69.80	0.51	0.70	13.41	0.68
4	10.08	8.62	2.71	15.54	40.32	7.00	5.83	5.22	80.30	1.85	1.00	57.00	5.22

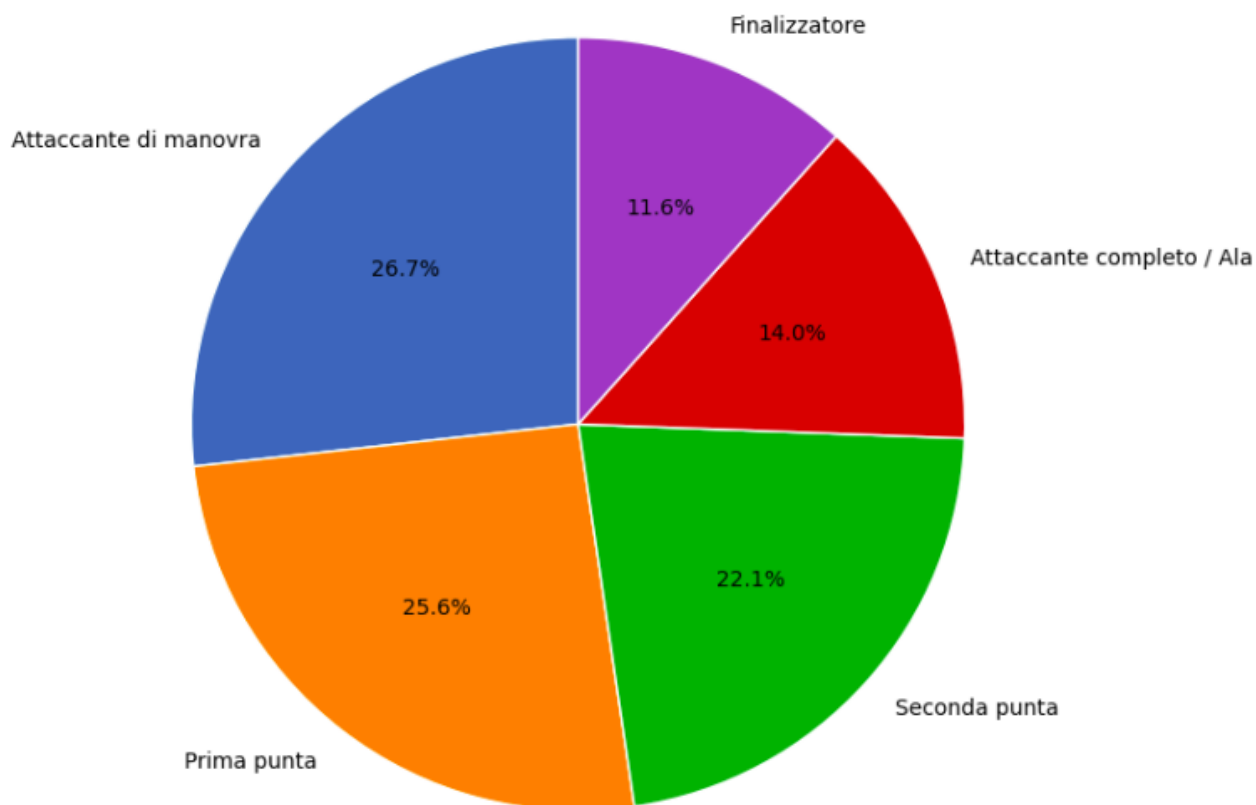
In particolare, sono stati identificati i seguenti gruppi:

- Cluster 0 – **Seconda punta**: buon contributo associativo, crea occasioni e serve assist, senza essere un finalizzatore puro.
- Cluster 1 – **Attaccante di manovra**: partecipa al gioco offensivo ma non eccelle in termini di produzione; profilo di raccordo tra centrocampo e attacco.
- Cluster 2 – **Finalizzatore**: alto numero di gol, tiri e conversione; vive per segnare, profilo da punta centrale classica.
- Cluster 3 – **Prima punta**: presenza in area, discreta efficacia al tiro ma poco coinvolto nella manovra creativa.
- Cluster 4 – **Attaccante completo / Ala**: elevati valori in tutte le metriche, sia nella finalizzazione che nella creazione; rappresenta un profilo moderno, completo, potenzialmente anche esterno offensivo.

Il risultato finale è un dataset arricchito con una nuova colonna “Etichetta\_Tattica”, che riporta il profilo funzionale del centrocampista secondo il cluster di appartenenza. Il file risultante, contenente tutte le informazione e le etichette assegnate, è stato salvato localmente con il nome Clustering\_Attaccanti.csv.

Player	Team	Position	Etichetta_Tattica
valentin castellanos	Lazio	A	Prima punta
albert gudmundsson	Genoa	A	Attaccante completo / Ala
andrea belotti	Fiorentina	A	Prima punta
tommaso baldanzi	Roma	A	Attaccante di manovra
stephan el shaarawy	Roma	A	Attaccante di manovra
emmanuel quartsin gyasi	Empoli	A	Attaccante di manovra
moise kean	Juventus	A	Attaccante di manovra
nicola sansone	Lecce	A	Attaccante di manovra
nicolas gonzalez	Fiorentina	A	Finalizzatore
nicolo cambiaghi	Empoli	A	Seconda punta

Distribuzione percentuale degli attaccanti per profilo tattico



Per valutare la capacità generalizzante del modello di clustering, è stato realizzato un riepilogo della distribuzione dei giocatori per squadra e ruolo, dopo l'assegnazione delle etichette tattiche. I risultati hanno mostrato che ogni squadra della Serie A presenta una distribuzione coerente di attaccanti, centrocampisti e difensori, con valori medi compresi tra 4 e 7 per ciascun ruolo. Questo conferma che il modello ha saputo distinguere ruoli in maniera equilibrata, evitando squilibri e assegnazioni errate. La consistenza tra le squadre è indice di una buona copertura del dataset e di una corretta segmentazione funzionale.

AC Milan  
6 attaccanti  
5 centrocampisti  
7 difensori

Atalanta  
4 attaccanti  
5 centrocampisti  
7 difensori

Bologna  
4 attaccanti  
6 centrocampisti  
6 difensori

Cagliari

4 attaccanti  
8 centrocampisti  
5 difensori

-----  
Empoli

4 attaccanti  
4 centrocampisti  
6 difensori

-----  
Fiorentina

6 attaccanti  
7 centrocampisti  
6 difensori

-----  
Frosinone

5 attaccanti  
5 centrocampisti  
5 difensori

-----  
Genoa

3 attaccanti  
5 centrocampisti  
7 difensori

-----  
Inter

4 attaccanti  
5 centrocampisti  
8 difensori

-----  
Juventus

6 attaccanti  
6 centrocampisti  
4 difensori

-----  
Lazio

4 attaccanti  
9 centrocampisti  
7 difensori

-----  
Lecce

5 attaccanti  
7 centrocampisti  
5 difensori

-----  
Monza

4 attaccanti  
6 centrocampisti  
8 difensori

-----  
Napoli

6 attaccanti  
4 centrocampisti  
7 difensori

-----  
Roma

5 attaccanti  
5 centrocampisti



```

5 difensori
-----
Salernitana
1 attaccanti
6 centrocampisti
4 difensori
-----
Sassuolo
4 attaccanti
5 centrocampisti
6 difensori
-----
Torino
4 attaccanti
4 centrocampisti
7 difensori
-----
Udinese
3 attaccanti
5 centrocampisti
8 difensori
-----
Verona
4 attaccanti
4 centrocampisti
5 difensori
-----

```

La fase di apprendimento è stata conclusa con successo. Il risultato finale è rappresentato dal file `Giocatori_Predetti_Con_Ruolo_Tattico.csv`, che integra in un unico dataset le informazioni previste dal modello (etichetta predetta) con i dati anagrafici e il ruolo tattico assegnato tramite clustering non supervisionato. Il file contiene, per ciascun giocatore, il nome, la squadra, il ruolo (es. attaccante, centrocampista, difensore o portiere), lo score normalizzato, la probabilità di segnare, la predizione regressiva pesata e l'etichetta di classificazione assegnata (es. "consigliato", "non consigliato" o "sorpresa"), oltre alla corrispondente etichetta tattica derivata dal clustering (es. "Finalizzatore", "Regista difensivo", "Portiere", ecc.).

Questo file costituisce l'input finale per la successiva fase di ricerca, in cui si dovranno selezionare i giocatori ottimali in base a vincoli e preferenze definite (es. moduli, squadre, combinazioni tattiche).

Player	Team	Role	Score_norm	Prob_Gol	Pred_Gol_Pesata	Etichetta_Predetta	Etichetta_Tattica
vanja milinkovic-savic	Torino	P	1.0	0.0	0.0	consigliato	Portiere
yann sommer	Inter	P	0.9941126241517232	0.0	0.0	consigliato	Portiere
wojciech szczesny	Juventus	P	0.87465731877219	0.0	0.0	consigliato	Portiere
alessandro bastoni	Inter	D	1.0	0.0213	0.0611	consigliato	Terzino offensivo
riccardo calafiori	Bologna	D	0.986333894871479	0.0338	0.305	consigliato	Terzino offensivo
francesco acerbi	Inter	D	0.984008919094826	0.0749	0.0861	consigliato	Regista difensivo

manuel lazzari	Lazio	C	0.4500337894150118	0.0	0.0	non consigliato	Regista
kevin strootman	Genoa	C	0.4444744840054523	0.0	0.0	non consigliato	Centrale equilibrato
fabio miretti	Juventus	C	0.4432896102724298	0.0213	0.0611	non consigliato	Box-to-Box
riccardo orsolini	Bologna	A	0.6332960829678862	0.1744	0.1736	non consigliato	Attaccante completo / Ala
armand lauriente	Sassuolo	A	0.5762828461167271	0.0995	0.1338	non consigliato	Seconda punta
andrea pinamonti	Sassuolo	A	0.5712719829198509	0.3538	0.2278	consigliato	Prima punta

## Capitolo 3) Ricerca

Per la fase di ricerca del progetto è stato progettato e implementato un sistema euristico di selezione, basato su una logica greedy con vincoli tattici, orientata all'individuazione della formazione titolare ottimale per ogni squadra. Il problema affrontato può essere formalizzato come una variante del Constraint Satisfaction Problem (CSP), in cui si devono assegnare giocatori a ruoli compatibili rispettando un insieme di vincoli (numero di elementi per ruolo, coerenza tattica, esclusività dei giocatori), ma con la particolarità che le variabili non sono simboliche bensì valutate attraverso punteggi numerici, ottenuti da fasi precedenti di classificazione e regressione.

La scelta di un approccio greedy è stata motivata dalla necessità di garantire efficienza e scalabilità, dato il numero limitato ma eterogeneo di squadre da processare. In ciascun passo della selezione, viene effettuata una scelta locale ottima, individuando il miglior candidato per ogni ruolo in base a un punteggio totale, calcolato combinando: prestazione attesa (score normalizzato), probabilità di segnare, predizione dei gol futuri e tipo di etichetta assegnata dal classificatore (“consigliato”, “sorpresa”, “non consigliato”). Questa funzione obiettivo guida le decisioni a ogni livello del processo.

Il greedy è stato applicato modulo per modulo, esplorando in parallelo tre configurazioni tattiche comuni (4-3-3, 4-2-3-1, 3-5-2), selezionando per ciascuna i giocatori migliori secondo vincoli tattici espliciti (ad esempio, evitare due terzini adattati al centro della difesa, o preferire trequartisti puri nei ruoli offensivi centrali). Sono state così costruite tre formazioni complete e coerenti, ne è stato valutato il punteggio aggregato e infine è stata selezionata quella con lo score più elevato come soluzione ottimale.

L'intero processo è stato configurato come un sistema di ricerca euristica con conoscenza incorporata, in cui i vincoli non vengono imposti rigidamente come in un CSP puro, ma sono gestiti attraverso filtri progressivi e strategie di fallback, che permettono di garantire sempre un esito anche in caso di dati parziali. Il greedy si adatta quindi perfettamente alla natura del problema: sfrutta al meglio la conoscenza disponibile, reagisce alla scarsità informativa, ed è facilmente scalabile a tutte le squadre presenti nel dataset.

Per la fase di ricerca, è stato progettato e implementato un algoritmo capace di selezionare automaticamente la formazione ottimale per ciascuna squadra presente all'interno del file `Giocatori_Predetti_Con_Ruolo_Tattico.csv`. L'obiettivo principale è stato quello di scegliere, per ogni team, il miglior undici titolare possibile valutando tre diversi moduli tattici (4-3-3, 4-2-3-1 e 3-5-2), sulla base sia del ruolo reale dei giocatori, sia della loro etichetta tattica predetta, sia infine di una funzione di scoring derivata da metriche di performance predittive calcolate nella fase di apprendimento.

In primo luogo, è stata definita una funzione di punteggio denominata `punteggio_totale`, applicata a ogni giocatore, con lo scopo di ottenere un valore sintetico che combinasse tre componenti fondamentali: un bonus fisso basato sull'etichetta di classificazione predetta (100 punti se “consigliato”, 50 se “sorpresa”, 0 se “non consigliato”), il valore normalizzato dello score tecnico (moltiplicato per 10), la probabilità che il giocatore segni nella prossima partita (pesata per 5) e la predizione del numero di gol attesi (anch'essa pesata per 5). Questa funzione restituisce una metrica numerica continua utilizzabile per l'ordinamento e la selezione automatica.

Una volta assegnato questo punteggio a tutti i giocatori, si è proceduto con un ciclo su ogni squadra (for team in squadre), estraendo il sottoinsieme dei dati relativi a quel team e generando tre possibili formazioni, una per ciascun modulo. La formazione per ciascun modulo è stata costruita seguendo criteri precisi e tatticamente coerenti, utilizzando una logica gerarchica di preferenze che tiene conto delle etichette tattiche predette e, laddove necessario, si è fatto ricorso a strategie di fallback per garantire comunque la formazione di una squadra completa.

Nel modulo **4-3-3**, è stata prevista la selezione di:

- un portiere con il punteggio più alto;
- quattro difensori: due terzini (preferibilmente con etichetta “Terzino offensivo” o “Libero”) e due centrali (tra “Stopper”, “Leader difensivo”, ecc.);
- tre centrocampisti, dove il primo viene scelto in assoluto come miglior centrocampista, e gli altri due tra quelli con etichetta tattica diversa dal primo;
- tre attaccanti: uno centrale (con priorità a “Finalizzatore”, “Attaccante di manovra” o “Prima punta”) e due esterni (“Attaccante completo / Ala” o “Seconda punta”), con eventuale recupero da altri ruoli in caso di mancanza.

Nel modulo **4-2-3-1** sono stati inclusi:

- sempre il portiere migliore e gli stessi 4 difensori del modulo 4-3-3 per coerenza e confronto equo;
- due centrocampisti centrali con priorità a “Regista”, “Centrale equilibrato” e “Box-to-Box”, ma con fallback progressivo a “Mezzala offensiva” o “N.C.” in caso di carenza;
- tre trequartisti, selezionati con priorità a “Trequartista”, “Ala”, “Seconda punta”, poi eventualmente “Mezzala offensiva” e infine “N.C.”, evitando ripetizioni;

- una punta centrale “pura” (tra i ruoli finalizzatori), escludendo giocatori già scelti tra i trequartisti.

Nel modulo **3-5-2** la struttura è stata più articolata:

- sono stati selezionati tre difensori centrali (tra “Leader difensivo”, “Regista difensivo”, ecc.);
- sono stati identificati fino a due giocatori “spostabili” sugli esterni tra i terzini e i liberi non già selezionati;
- è stato costruito il centrocampo con un playmaker (tra “Centrale equilibrato”, “Regista”, “Box-to-Box”), due interni offensivi (priorità a “Mezzala”, fallback su altri ruoli coerenti), e i due esterni ottenuti dalla logica precedente;
- gli ultimi due slot sono stati assegnati agli attaccanti migliori rimasti non ancora utilizzati.

Per ogni modulo, è stato calcolato lo score totale sommando i punteggi individuali dei giocatori selezionati. Il modulo con lo score più alto è stato considerato il modulo ottimale per quella specifica squadra. Questo modulo e la formazione corrispondente sono stati visualizzati tramite display() e contemporaneamente salvati in un dataframe cumulativo risultati, che raccoglie le formazioni ottimali di tutte le squadre.

Al termine del processo, tutte le formazioni migliori sono state unite in un unico file finale denominato `Formazioni_Ottimali_Per_Squadra.csv`, contenente per ogni squadra i nomi dei giocatori titolari, il loro ruolo, l’etichetta tattica, e il modulo selezionato.

===== TEAM: Torino =====

MODULO 4-3-3 – SCORE: 732.50

	Player	Role	Etichetta_Tattica
0	vanja milinkovic-savic	P	Portiere
1	raoul bellanova	D	Terzino offensivo
2	ricardo rodriguez	D	Libero
3	alessandro buongiorno	D	Leader difensivo
4	valentino lazaro	D	Libero
5	ivan ilic	C	Trequartista
6	samuele ricci	C	Regista
7	adrien tameze	C	Centrale equilibrato
8	nikola vlastic	A	Seconda punta
9	pietro pellegri	A	Attaccante di manovra
10	duvan zapata	A	Finalizzatore

MODULO 4-2-3-1 – SCORE: 680.32

	Player	Role	Etichetta_Tattica
0	vanja milinkovic-savic	P	Portiere
1	raoul bellanova	D	Terzino offensivo
2	ricardo rodriguez	D	Libero
3	alessandro buongiorno	D	Leader difensivo
4	valentino lazaro	D	Libero
5	samuele ricci	C	Regista
6	adrien tameze	C	Centrale equilibrato
7	ivan ilic	C	Trequartista
8	nikola vlastic	A	Seconda punta
9	zanos savva	A	N.C.
10	duvan zapata	A	Finalizzatore

MODULO 3-5-2 - SCORE: 630.04

	Player	Role	Etichetta_Tattica
0	vanja milinkovic-savic	P	Portiere
1	raoul bellanova	D	Terzino offensivo
2	ricardo rodriguez	D	Libero
3	alessandro buongiorno	D	Leader difensivo
4	samuele ricci	C	Regista
5	adrien tameze	C	Centrale equilibrato
6	karol linetty	C	Regista
7	valentino lazaro	D	Libero
8	mergim vojvoda	D	Libero
9	duvan zapata	A	Finalizzatore
10	pietro pellegri	A	Attaccante di manovra

MIGLIOR MODULO PER Torino: 4-3-3 - SCORE: 732.50

...

...

...

===== TEAM: AC Milan =====

MODULO 4-3-3 - SCORE: 900.14

	Player	Role	Etichetta_Tattica
0	mike maignan	P	Portiere
1	theo hernandez	D	Terzino offensivo
2	alessandro florenzi	D	Terzino offensivo
3	fikayo tomori	D	Regista difensivo
4	malick thiaw	D	Regista difensivo
5	tijjani reijnders	C	Trequartista
6	ismael bennacer	C	Centrale equilibrato
7	ruben loftus-cheek	C	Mezzala offensiva
8	rafael leao	A	Attaccante completo / Ala
9	christian pulisic	A	Attaccante completo / Ala
10	olivier giroud	A	Finalizzatore

MODULO 4-2-3-1 - SCORE: 899.81

	Player	Role	Etichetta_Tattica
0	mike maignan	P	Portiere
1	theo hernandez	D	Terzino offensivo
2	alessandro florenzi	D	Terzino offensivo
3	fikayo tomori	D	Regista difensivo
4	malick thiaw	D	Regista difensivo
5	ismael bennacer	C	Centrale equilibrato
6	yacine adli	C	Centrale equilibrato
7	tijjani reijnders	C	Trequartista
8	rafael leao	A	Attaccante completo / Ala
9	christian pulisic	A	Attaccante completo / Ala
10	olivier giroud	A	Finalizzatore

MODULO 3-5-2 - SCORE: 798.44

	Player	Role	Etichetta_Tattica
0	mike maignan	P	Portiere
1	theo hernandez	D	Terzino offensivo
2	fikayo tomori	D	Regista difensivo
3	alessandro florenzi	D	Terzino offensivo
4	ismael bennacer	C	Centrale equilibrato
5	ruben loftus-cheek	C	Mezzala offensiva
6	yacine adli	C	Centrale equilibrato
7	davide calabria	D	Terzino offensivo
8	rafael leao	A	Attaccante completo / Ala
9	olivier giroud	A	Finalizzatore
10	christian pulisic	A	Attaccante completo / Ala

MIGLIOR MODULO PER AC Milan: 4-3-3 - SCORE: 900.14

...

Questa implementazione rappresenta un sistema di ricerca euristica avanzata, orientata all'ottimizzazione delle risorse disponibili per ciascuna squadra. Le scelte implementate sono state giustificate sia da criteri calcistici (coesione tattica, copertura del campo), sia da criteri algoritmici (preservazione della completezza del risultato, massimizzazione di un punteggio quantitativo). L'approccio è risultato robusto a incompletezze nei dati (grazie ai fallback) e altamente adattivo. Inoltre, è stata mantenuta la possibilità di estensione futura verso vincoli più rigidi (come quelli di un CSP puro) o moduli personalizzati.

Nella fase conclusiva del progetto è stato implementato un ulteriore algoritmo di ricerca di tipo euristico, con l'obiettivo specifico di individuare la miglior formazione teorica assoluta del campionato di Serie A, selezionando i top 11 tra tutti i giocatori presenti nel dataset complessivo. In continuità con quanto realizzato nella fase precedente, l'approccio adottato si basa ancora una volta su un processo di ottimizzazione euristica vincolata, in cui la selezione viene guidata da una funzione obiettivo che integra, in maniera bilanciata, i risultati ottenuti dalle fasi di classificazione, regressione e stima probabilistica.

L'algoritmo, mantenendo inalterata la logica modulare già applicata alla selezione squadra-per-squadra, costruisce tre diverse formazioni ideali seguendo i moduli 4-3-3, 4-2-3-1 e 3-5-2, ciascuna composta rispettando ruoli reali, etichette tattiche e vincoli funzionali. Una volta generate le tre possibili formazioni, viene calcolata per ciascuna la somma dei punteggi totali dei giocatori selezionati. La formazione con lo score aggregato più elevato viene selezionata come formazione ideale assoluta del

campionato, ovvero la combinazione teorica più efficace in termini di rendimento atteso.

Al termine dell'esecuzione, tutte e tre le formazioni generate vengono esportate in un file CSV, che riporta per ciascun giocatore selezionato il modulo di riferimento, la squadra di appartenenza, il ruolo e l'etichetta tattica.

MODULO 4-3-3 – SCORE: 1223.38

	Player	Team	Role	Etichetta_Tattica
0	vanja milinkovic-savic	Torino	P	Portiere
77	riccardo calafiori	Bologna	D	Terzino offensivo
76	alessandro bastoni	Inter	D	Terzino offensivo
78	francesco acerbi	Inter	D	Regista difensivo
80	amir rrahmani	Napoli	D	Terzino offensivo
302	hakan calhanoglu	Inter	C	Trequartista
308	arthur	Fiorentina	C	Centrale equilibrato
306	matteo pessina	Monza	C	Centrale equilibrato
517	albert gudmundsson	Genoa	A	Attaccante completo / Ala
523	rafael leao	AC Milan	A	Attaccante completo / Ala
516	lautaro martinez	Inter	A	Finalizzatore

MODULO 4-2-3-1 – SCORE: 1223.38

	Player	Team	Role	Etichetta_Tattica
0	vanja milinkovic-savic	Torino	P	Portiere
77	riccardo calafiori	Bologna	D	Terzino offensivo
76	alessandro bastoni	Inter	D	Terzino offensivo
78	francesco acerbi	Inter	D	Regista difensivo
80	amir rrahmani	Napoli	D	Terzino offensivo
308	arthur	Fiorentina	C	Centrale equilibrato
306	matteo pessina	Monza	C	Centrale equilibrato
302	hakan calhanoglu	Inter	C	Trequartista
517	albert gudmundsson	Genoa	A	Attaccante completo / Ala
523	rafael leao	AC Milan	A	Attaccante completo / Ala
516	lautaro martinez	Inter	A	Finalizzatore



MODULO 3-5-2 – SCORE: 1217.93

	Player	Team	Role	Etichetta_Tattica
0	vanja milinkovic-savic	Torino	P	Portiere
77	riccardo calafiori	Bologna	D	Terzino offensivo
78	francesco acerbi	Inter	D	Regista difensivo
76	alessandro bastoni	Inter	D	Terzino offensivo
308	arthur	Fiorentina	C	Centrale equilibrato
306	matteo pessina	Monza	C	Centrale equilibrato
330	mattia zaccagni	Lazio	C	Box-to-Box
80	amir rrahmani	Napoli	D	Terzino offensivo
86	federico dimarco	Inter	D	Terzino offensivo
516	lautaro martinez	Inter	A	Finalizzatore
521	olivier giroud	AC Milan	A	Finalizzatore

MIGLIOR MODULO DEL CAMPIONATO: 4-3-3 – SCORE: 1223.38

Questa ultima fase conferma l'efficacia e la versatilità dell'approccio euristico adottato, dimostrando come, a partire da modelli predittivi differenti ma complementari, sia possibile costruire soluzioni tattiche complesse che tengano conto non solo del potenziale individuale ma anche della coerenza e funzionalità complessiva del sistema.

## Conclusioni

Il progetto *AI Coach* ha rappresentato un caso concreto di integrazione tra diverse tecniche di intelligenza artificiale basate su conoscenza, applicate al contesto sportivo della Serie A italiana. A partire dalla costruzione di un dataset ricco e strutturato, passando per la modellazione predittiva delle prestazioni individuali, fino alla classificazione e alla successiva selezione ottimale della formazione titolare per ogni squadra, è stato sviluppato un sistema completo, coerente e modulare.

L'architettura progettuale si è basata su tre pilastri fondamentali: **Ragionamento**, **Apprendimento** e **Ricerca**. Ogni fase ha dialogato in modo sinergico con le altre, fornendo output raffinati e contestualizzati per alimentare le decisioni delle fasi successive. In particolare, l'approccio probabilistico (Catene di Markov), la predizione regressiva (Decision Tree Regressor), la classificazione supervisionata (Random Forest), il clustering funzionale (KMeans) e l'algoritmo greedy di ricerca euristica si sono integrati in un flusso informativo continuo, robusto e adattivo.

La scelta di un modello a vincoli euristici, anziché un CSP rigido, ha permesso di mantenere alta la flessibilità del sistema, garantendo comunque la produzione di soluzioni sempre complete e tatticamente plausibili, anche in presenza di dati parziali o inconsistenti.

Dal punto di vista applicativo, *AI Coach* dimostra come un sistema intelligente possa effettivamente supportare lo staff tecnico di una squadra nella gestione delle rotazioni e nella selezione ottimale dei giocatori. Le soluzioni generate non si limitano a replicare ranking statici, ma riflettono andamenti, forme recenti, e ruoli funzionali, offrendo suggerimenti personalizzati e strategicamente rilevanti.

Il progetto si presta infine a numerose estensioni future, tra cui l'integrazione di fattori contestuali (es. avversario, stadio, condizioni meteo), l'adattamento in tempo reale con aggiornamenti live, e l'applicazione a contesti diversi dal calcio, mantenendo inalterata la pipeline generale. In sintesi, *AI Coach* rappresenta un primo passo concreto verso l'adozione di sistemi intelligenti nel decision-making sportivo.