Documentazione Progetto FootballPlayers

Gruppo di lavoro

• Alessandro Aldo Boffolo, mat. 735963, a.boffolo@studenti.uniba.it

URL repo associato, contenente il materiale completo

https://github.com/Ale210501/FootballPlayers_ICon

AA 2023-24

Introduzione

Il progetto *FootballPlayers* si propone di analizzare e classificare i giocatori di calcio utilizzando tecniche avanzate di machine learning. L'obiettivo principale è duplice: prevedere il ruolo di un giocatore basato su vari indicatori di performance e raccomandare giocatori in base alle preferenze dell'utente. Il progetto si avvale di modelli di classificazione e di raccomandazione per offrire previsioni accurate e suggerimenti personalizzati.

Il dataset utilizzato è stato acquisito da Kaggle, e dopo un'accurata fase di preprocessing, è stato impiegato per addestrare e testare i modelli di machine learning. Il progetto esplora diverse tecniche di apprendimento automatico, inclusi K-Nearest Neighbors, Gaussian Naive Bayes e Random Forest, e implementa un sistema di raccomandazione basato sui dati forniti dagli utenti, grazie all'utilizzo della similarità di Manhattan.

Sommario

Il progetto *FootballPlayers* utilizza un Knowledge-Based System (KBS) per gestire e analizzare i dati relativi ai giocatori di calcio, integrando vari moduli che dimostrano competenze avanzate in diverse aree del machine learning e dell'analisi dei dati. Il sistema si articola in quattro moduli principali, ognuno dei quali affronta un aspetto specifico del progetto:

Elenco argomenti di interesse

• **Preprocessing**: Il modulo di preprocessing è cruciale per la preparazione dei dati, e si occupa di garantire che il dataset sia pulito, coerente e pronto per le fasi successive dell'analisi. Le attività di preprocessing includono la gestione dei valori mancanti, la gestione di outlier, la normalizzazione e la trasformazione delle variabili. Utilizzando librerie come Pandas e Scikit-Learn, sono stati applicati vari metodi per migliorare la qualità dei dati e assicurare che i modelli di machine learning operino su dati accurati e ben strutturati. Questa fase prepara i dati per le analisi predittive e descrittive, assicurando una base solida per le altre fasi del progetto.

• **Analisi dei Dati**: L'analisi dei dati, nell'ambito di questo progetto, si concentra sulla comprensione delle principali caratteristiche e tendenze del dataset relativo ai calciatori, al fine di supportare le fasi successive di modellazione e predizione. L'analisi esplorativa dei dati (EDA) gioca un ruolo cruciale in questa fase, permettendo di identificare pattern, correlazioni tra variabili e comportamenti anomali all'interno del dataset. Questo processo fornisce informazioni fondamentali che influenzano la selezione delle variabili, la scelta dei modelli predittivi e la gestione dei dati durante il pre-processing.

L'EDA si articola in diverse operazioni chiave, tra cui:

- **Statistica descrittiva** per comprendere la distribuzione delle variabili.
- **Visualizzazioni grafiche** (come heatmap di correlazione e grafici di distribuzione) per rappresentare e analizzare le relazioni tra le variabili.
- **Identificazione degli outliers**, ossia i valori che si discostano significativamente dalla distribuzione generale dei dati, che potrebbero influire negativamente sulla costruzione di modelli predittivi.

Il risultato finale dell'analisi è una comprensione più profonda dei dati, che consente di prendere decisioni informate riguardo alle variabili da includere nel modello, ai possibili aggiustamenti necessari (come la gestione dei valori mancanti o degli outliers) e alla costruzione di un modello di classificazione più preciso ed efficiente.

• **Classificazione**: Il modulo di classificazione si focalizza sulla previsione del ruolo dei giocatori utilizzando tecniche di machine learning supervisionato. Sono stati implementati e testati diversi algoritmi di classificazione, tra cui K-Nearest Neighbors (KNN), Gaussian Naive Bayes e Random Forest. Ogni modello è stato valutato in base a metriche di performance come accuratezza, precisione, richiamo e F1-score. Questo modulo dimostra competenze nella costruzione e nella valutazione di modelli predittivi, confrontando le performance dei diversi algoritmi per identificare quello più adatto al problema specifico. L'analisi dei risultati consente di ottimizzare le previsioni e di comprendere meglio i fattori che influenzano i ruoli dei giocatori.

• **Raccomandazione**: Il sistema di raccomandazione è progettato per suggerire giocatori in base ai dati forniti dall'utente, utilizzando tecniche di similitudine e matching. Questo modulo offre raccomandazioni personalizzate che aiutano gli utenti a trovare giocatori che soddisfano specifici criteri o preferenze. Il sistema è stato progettato e testato per garantire che le raccomandazioni siano pertinenti e utili, dimostrando competenze nella progettazione di algoritmi di raccomandazione e nella valutazione della loro efficacia. L'implementazione include l'uso di tecniche avanzate per migliorare la qualità delle raccomandazioni e fornire suggerimenti rilevanti basati sulle preferenze degli utenti.

Preprocessing dei dati

Sommario

Il modulo di preprocessing dei dati nel progetto *FootballPlayers* è progettato per preparare i dati grezzi per l'analisi e la modellazione successiva. Questo modulo include la pulizia dei dati, la gestione dei valori mancanti, la trasformazione delle variabili e la selezione delle colonne di interesse. La rappresentazione della conoscenza nel preprocessing è basata su tecniche consolidate di data cleaning e data transformation, essenziali per garantire dati coerenti e di alta qualità. La qualità dei dati è fondamentale per le fasi di classificazione, clustering e raccomandazione, poiché modelli ben progettati richiedono input pulito e ben strutturato.

Strumenti utilizzati

Per il preprocessing dei dati sono stati utilizzati diversi strumenti e librerie standard nel campo della scienza dei dati:

- **Pandas**: Per la manipolazione e la pulizia dei dati.

Decisioni di Progetto

Le decisioni di progetto relative al preprocessing dei dati includono:

- **Gestione dei Valori Nulli**: Le righe contenenti valori nulli sono state rimosse.
- **Gestione dei Valori Doppi**: Nella colonna ‘Pos’ erano presenti valori doppi (es. “MF, FW” o “DF, MF”), dovuti dalla possibilità di ricoprire più ruoli da ogni giocatore. Si è scelto di eliminarli e di rimpiazzarli con il valore migliore per ogni coppia di ‘Pos’.
- **Conversione in Minuscolo**: All’interno del dataset sono presenti valori contenenti lettere, si è scelto di trasformare tutto in minuscolo.
- **Normalizzazione**: I dati numerici sono stati normalizzati utilizzando la StandardScaler di Scikit-Learn per garantire che tutte le variabili abbiano una scala comune. Per i valori non interi (colonna ‘90s’), si è scelto l’arrotondamento dei vari valori.
- **Selezione delle Colonne**: Sono state selezionate solo le colonne di interesse per l'analisi, rimuovendo le colonne non rilevanti. Le colonne scelte sono le seguenti: ‘Player’, ‘Pos’, ‘Born’, ‘90s’, ‘Gls’, ‘Ast’, ‘CrdY’, ‘CrdR’.
- **Creazione Dizionario Conversione**: È stato creato un dizionario utilizzato per la conversione dei valori presenti nella colonna ‘Player’ ed è stato salvato in un file csv. Per i valori di ‘Pos’ è stato creato un dizionario associando ad ogni ruolo un numero.

- **Conversione dei Dati Categorici**: I dati presenti nelle colonne 'Player' e 'Pos' sono stati convertiti in valori numerici utilizzando i due dizionari creati precedentemente, creando così due nuove colonne ('Pos_format' e 'Player_format') da sostituire al posto delle colonne contenenti dati non numerici.

Queste decisioni sono state prese per ottimizzare la qualità dei dati e preparare un dataset adatto per l'analisi successiva.

Il dataset ottenuto dopo la fase di preprocessing è, quindi, il seguente:

	Born	90s	Gls	Ast	CrdY	CrdR	Pos_format	Player_format
1	2000	14	0	1	1	0	1	0
2	1999	5	0	0	2	0	1	1
3	1999	14	2	1	5	0	1	1
4	1996	4	0	0	1	0	1	2
5	1997	27	3	4	6	0	1	3
6	2000	25	9	7	6	0	1	4
7	1995	17	0	0	1	0	1	5
8	1998	24	1	0	6	0	1	6
9	1992	18	0	0	2	0	1	7
10	1992	21	1	0	4	0	1	8
11	1995	24	3	2	4	0	1	9
12	2003	0	0	0	0	0	1	10
13	1994	15	0	2	3	0	1	11
14	1998	12	1	1	1	0	1	12

Valutazione

La valutazione del preprocessing è stata effettuata attraverso i seguenti metodi:

- **Analisi della Qualità dei Dati**: Sono stati utilizzati strumenti di Pandas per verificare la completezza e la coerenza dei dati dopo il preprocessing.
- **Metriche di Preprocessing**: Le metriche principali incluse sono la percentuale di valori mancanti, il range e la media delle variabili normalizzate.

```
Percentuale di valori mancanti per ogni colonna:
Born            0.0
90s             0.0
Gls             0.0
Ast             0.0
CrdY            0.0
CrdR            0.0
Pos_format      0.0
Player_format   0.0
dtype: float64
```

```
Range delle variabili numeriche prima della normalizzazione:
Born               26
90s                38
Gls                36
Ast                14
CrdY               17
CrdR                3
Player_format    2699
dtype: int64
Media delle variabili numeriche prima della normalizzazione:
Born             1997.600070
90s                13.480337
Gls                 1.722612
Ast                 1.223666
CrdY                2.658357
CrdR                0.120084
Player_format    1346.474719
dtype: float64
```

```
Range delle variabili numeriche dopo la normalizzazione:
Born               5.657725
90s                3.549925
Gls               11.233959
Ast                7.121524
CrdY               6.236114
CrdR               8.450563
Player_format      3.482600
dtype: float64
Media delle variabili numeriche dopo la normalizzazione:
Born             -1.496930e-17
90s              -4.989766e-18
Gls              -4.989766e-18
Ast               1.746418e-17
CrdY              2.993860e-17
CrdR              2.993860e-17
Player_format     0.000000e+00
dtype: float64
```

Analisi dei Dati

Sommario

Il modulo di analisi dei dati è stato creato per esplorare e comprendere le caratteristiche principali del dataset di calciatori, offrendo una panoramica statistica e visiva delle variabili. Include operazioni di analisi esplorativa dei dati (EDA) tramite statistiche descrittive, visualizzazioni di correlazione tra variabili, e grafici di distribuzione per le principali caratteristiche. L'obiettivo è estrarre conoscenze utili per comprendere meglio la distribuzione dei dati e le relazioni tra le variabili, in modo da supportare fasi successive di classificazione e modellazione predittiva.

Strumenti utilizzati

Per l'analisi dei dati sono stati usati:

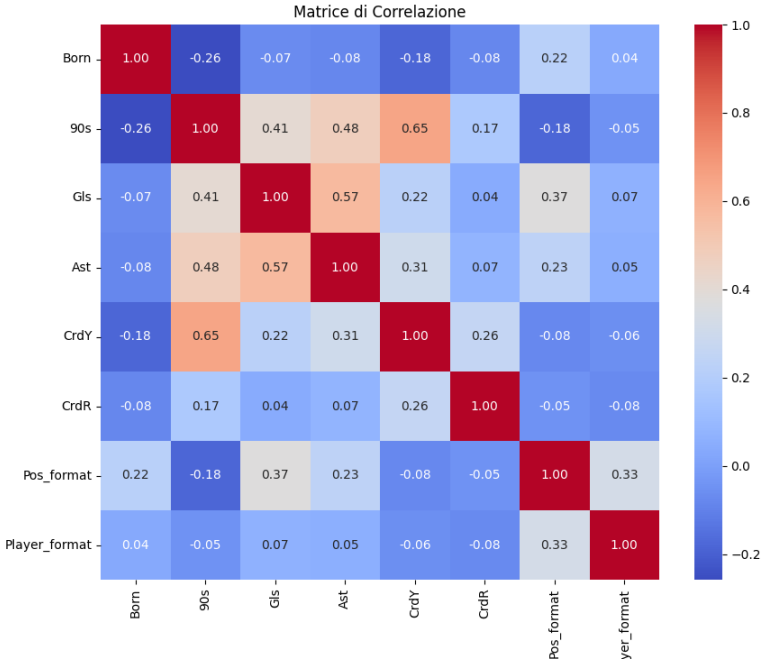
- **Pandas:** per la manipolazione e l'elaborazione dei dati tabellari.
- **Matplotlib:** per creare grafici di distribuzione e altre visualizzazioni base.
- **Seaborn:** per grafici più complessi come la heatmap della matrice di correlazione.

Decisione di progetto

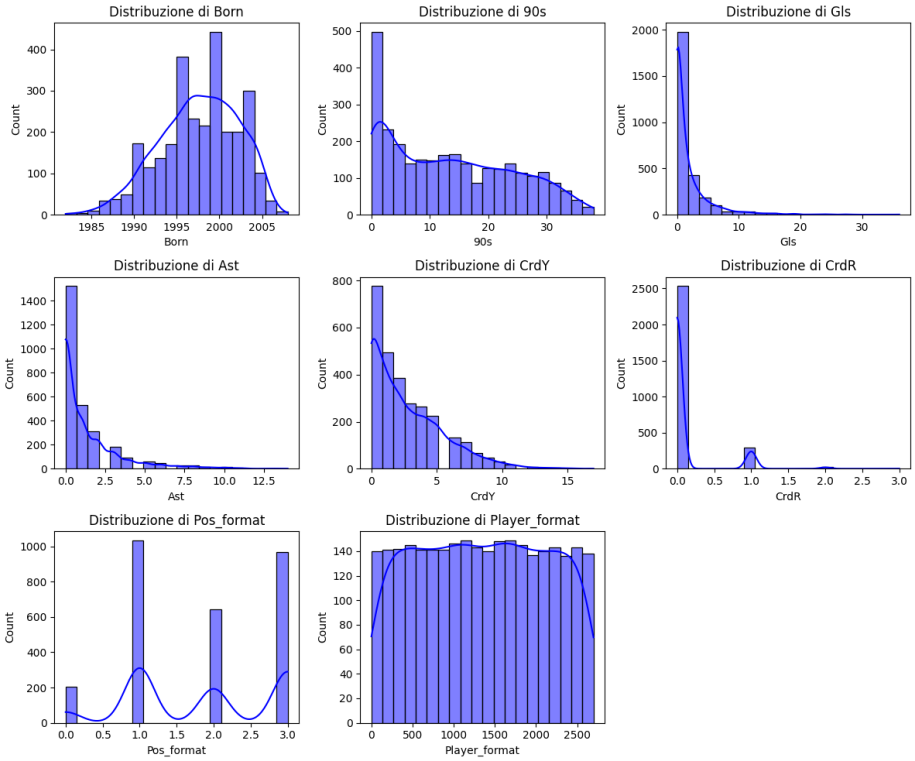
La scelta delle visualizzazioni e delle statistiche descrittive è stata guidata dall'obiettivo di comprendere il dataset nei suoi aspetti principali:

- **Analisi Esplorativa dei Dati (EDA)**: La heatmap della matrice di correlazione è una rappresentazione grafica che mostra il grado di correlazione tra le variabili di un dataset, utilizzando una scala di colori per visualizzare i valori di correlazione. La correlazione misura la relazione lineare tra due variabili e varia tra -1 e 1:
    - **1** indica una correlazione positiva perfetta: quando una variabile aumenta, anche l'altra aumenta in proporzione.
    - **0** indica nessuna correlazione lineare: non esiste una relazione lineare tra le due variabili.

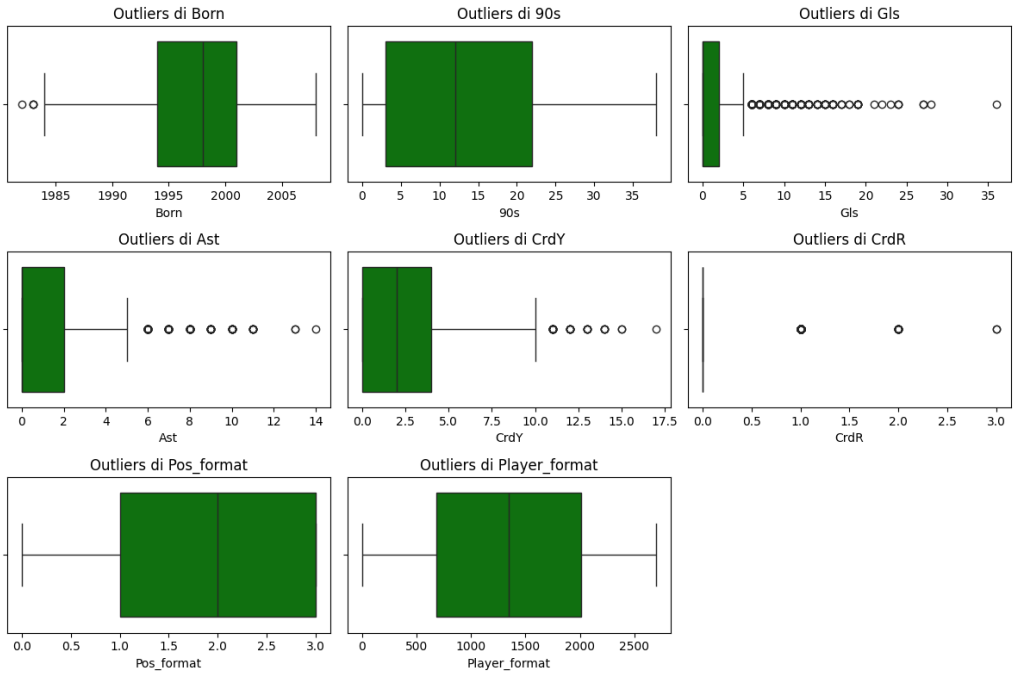
- **-1** indica una correlazione negativa perfetta: quando una variabile aumenta, l'altra diminuisce.



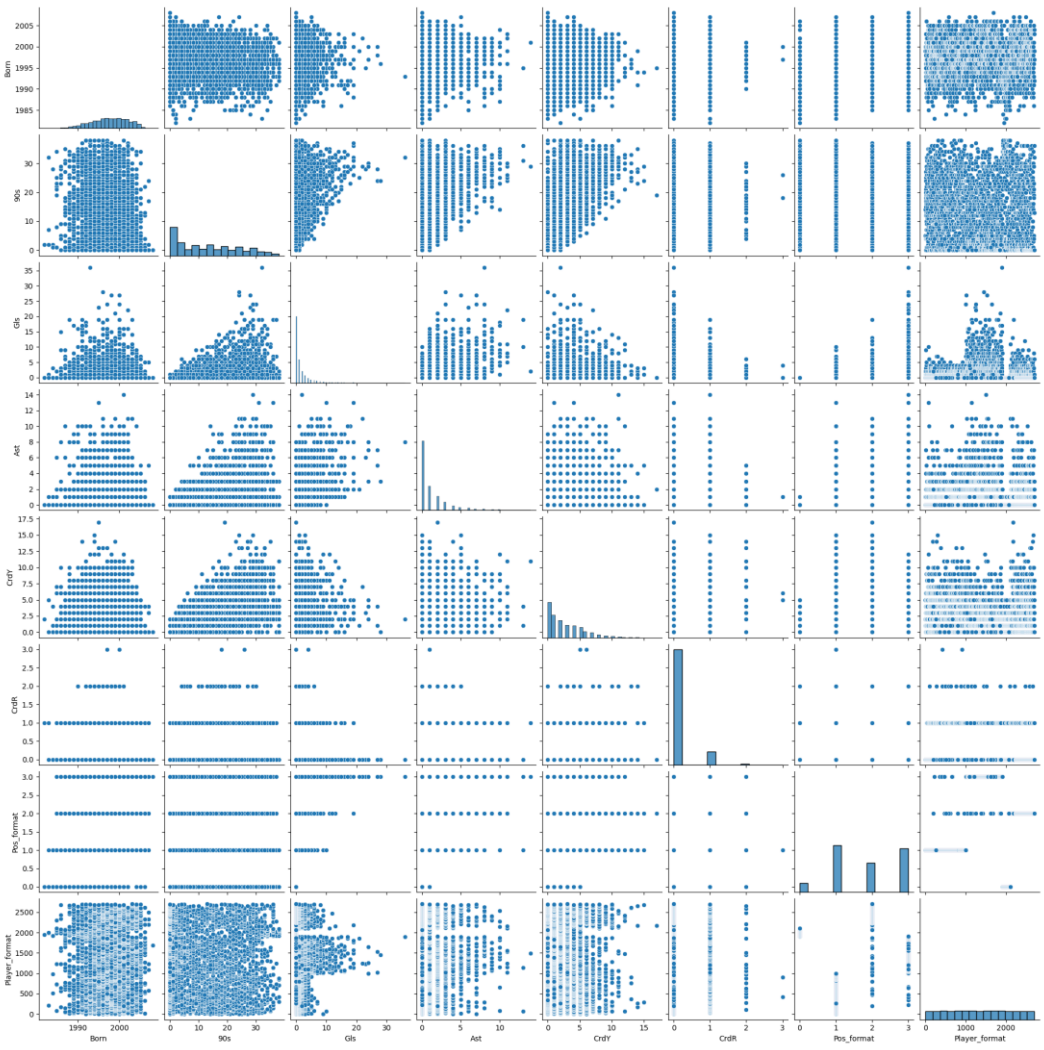
- **Distribuzione delle Variabili Numeriche**: Serve per comprendere la forma, il range e la variabilità dei dati per ciascuna variabile numerica nel dataset.



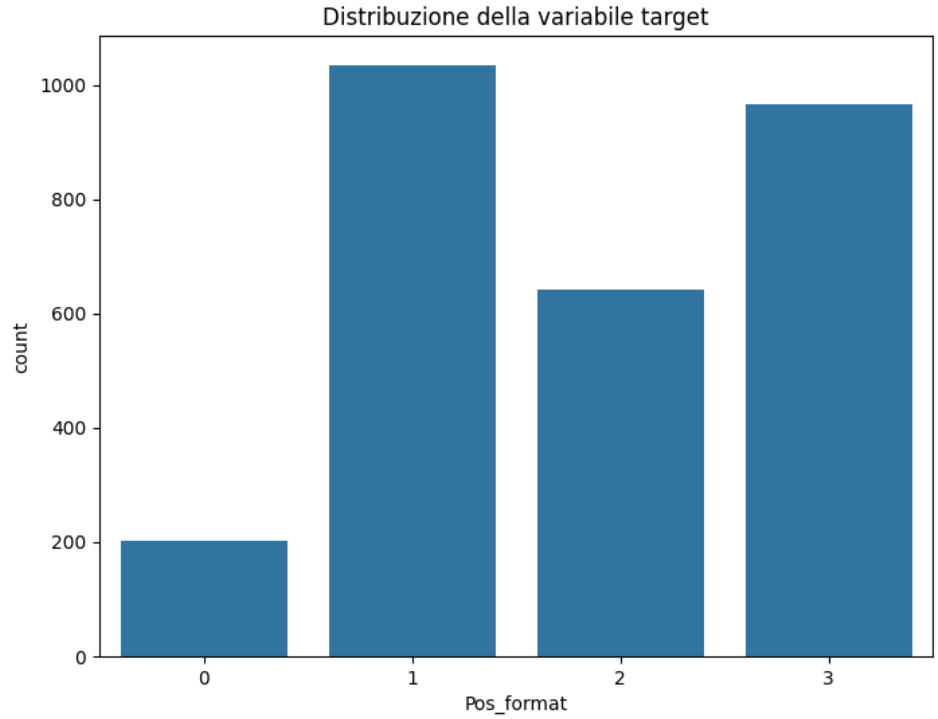
- **Identificazione di Outliers:** Serve a individuare i valori anomali o estremi che si discostano significativamente dalla maggior parte dei dati in un dataset. Gli outliers possono avere un grande impatto sui risultati dell'analisi e dei modelli di machine learning, quindi è importante rilevarli e gestirli.



- **Relazioni tra Variabili Numeriche**: È un passo importante nell'analisi esplorativa dei dati (EDA), poiché permette di capire come le variabili numeriche del dataset interagiscono tra loro. Questa analisi è utile per individuare correlazioni, semplificare il modello, individuare pattern o gruppi, selezionare feature per il modello.



- **Analisi della Variabile Target:** È utilizzata per comprendere le caratteristiche della variabile che si vuole prevedere o classificare. Questa analisi serve a capire la distribuzione dei valori, individuare squilibri di classe, identificare pattern importanti, definire strategie di valutazione



- **Analisi delle Correlazioni tra la Variabile Target e le Altre Variabili:** Aiuta a comprendere le relazioni esistenti tra la variabile che si desidera prevedere (la variabile target) e le altre caratteristiche del dataset.

```
Target_corr:
Pos_format       1.000000
Gls              0.371261
Player_format    0.326666
Ast              0.234023
Born             0.215457
CrdR            -0.049535
CrdY            -0.079641
90s             -0.183101
Name: Pos_format, dtype: float64
```

L'uso di queste tecniche permette di capire meglio come ogni variabile potrebbe influenzare la classificazione nel modello finale.

Valutazione

L'analisi offre utili insight per lo sviluppo di modelli predittivi, aiutando a:

- Identificare possibili variabili target e input significativi per i modelli di classificazione.
- Escludere variabili con correlazioni troppo alte o troppo basse, migliorando così l'efficienza del modello.
- Ottimizzare il pre-processing per la pulizia e gestione dei dati, includendo strategie di gestione dei valori mancanti e outlier.

Grazie all'EDA, è possibile avere una migliore comprensione dei dati, riducendo il rischio di errori nella fase di modellazione predittiva e migliorando la performance dei classificatori.

Classificazione dei ruoli dei giocatori

Sommario

Il modulo di classificazione nel progetto *FootballPlayers* si concentra sulla predizione del ruolo dei giocatori, utilizzando dati storici e statistici per apprendere modelli in grado di assegnare un giocatore a una determinata posizione (ad esempio attaccante, centrocampista, difensore). Il modello di apprendimento supervisionato scelto per rappresentare la conoscenza del dominio si basa su tecniche di classificazione come K-Nearest Neighbors (KNN), Random Forest e Naive Bayes. I dati utilizzati includono variabili statistiche sui giocatori come gol, assist, e cartellini, che vengono processati per costruire una Knowledge Base (KB) utile a predire correttamente il ruolo del giocatore.

Strumenti utilizzati

Sono stati utilizzati i seguenti strumenti e librerie per la classificazione:

- **Scikit-Learn**: Utilizzato per implementare i modelli di KNN, Naive Bayes e Random Forest.
- **Pandas**: Per gestire il dataset e preparare i dati per il modello.
- **NumPy**: Per supporto a operazioni numeriche complesse.

Decisioni di Progetto

Le seguenti decisioni sono state prese durante lo sviluppo del modulo di classificazione:

- **Target e Set di Addestramento**: Si è definito il target come la colonna 'Pos_format' e il set di addestramento come le altre colonne.
- **Modelli Utilizzati**: Sono stati implementati tre diversi modelli di classificazione: K-Nearest Neighbors, Gaussian Naive Bayes e Random Forest. Questi modelli sono stati scelti per la loro capacità di gestire dati eterogenei e la loro efficienza in diversi contesti di classificazione:
    - **K-Nearest Neighbors**: KNN è semplice da implementare e funziona bene quando si ha una quantità di dati limitata o di media entità. È un modello non parametrico, cioè non assume nessuna distribuzione specifica dei dati. KNN basa la classificazione sulla "vicinanza" dei punti nello spazio delle caratteristiche; quindi, le caratteristiche dei giocatori (come goal, assist, ammonizioni) possono raggruppare giocatori simili nella stessa classe (es.

difensori, attaccanti). KNN funziona bene per compiti in cui la somiglianza tra i punti è rilevante per la classificazione.

- **Gaussian Naive Bayes:** Naive Bayes assume che le caratteristiche siano indipendenti. Il Gaussian Naive Bayes si applica bene a dati con caratteristiche numeriche continue che approssimativamente seguono una distribuzione normale. Le variabili nel dataset dei giocatori (ad es. il numero di goal, assist, ammonizioni, ecc.) possono spesso essere distribuite normalmente per alcune classi di giocatori.

- **Random Forest:** I giocatori hanno diverse caratteristiche (goal, assist, età, posizione) che possono non essere linearmente correlate. Random Forest costruisce un insieme di alberi, ciascuno specializzato in vari aspetti dei dati, risultando in un modello potente e adatto a relazioni complesse. Grazie all'approccio ensemble, Random Forest tende a ridurre l'overfitting rispetto ad altri modelli basati sugli alberi, il che è utile se ci sono variabili irrilevanti o rumore. Random Forest permette di valutare l'importanza delle varie caratteristiche, aiutando a capire quali statistiche dei giocatori sono più rilevanti per predire la posizione, un aspetto utile per analisi più approfondite.

- **Configurazione dei Parametri**: Grazie ad una funzione, è stata calcolata l'accuratezza media e i parametri ottimali al variare della cross-validation. È stato, quindi, ricercato il miglior modello con GridSearchCV. Per ogni classificatore si è partiti da una definizione iniziale di parametri:

  - **KNN**:

    - n_neighbors: [5, 10]
    - weights: ['uniform', 'distance']
    - metric: ['manhattan', 'cosine']

  - **Random Forest**:

    - n_estimators: [100, 200]
    - max_depth: [10, 20]
    - min_samples_split: [2, 5]
    - min_samples_leaf: [1, 2]
    - criterion: ['gini', 'entropy']

  - **Gaussian Naive Bayes**:

    - var_smoothing: np.logspace(0, -9, num=10)

Successivamente, grazie ai vari risultati ottenuti utilizzando i vari parametri, si è trovato che il miglior classificatore da utilizzare è il Random Forest con una 15-fold cross validation con i seguenti parametri: criterion: entropy, max_depth: 10, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 100.

L'accuratezza media di 96.55% indica che il modello ha classificato correttamente circa il 96.55% degli esempi di test su tutte le 15 iterazioni di cross-validation. Questo è un buon risultato, indicando che il modello sta generalizzando bene e non si adatta troppo ai dati di addestramento (evitando l'overfitting). In altre parole, il Random Forest sembra essere in grado di fare previsioni molto accurate sui dati di test che non ha mai visto prima.

Random Forest è il miglior classificatore per il dataset utilizzato perché è in grado di gestire una combinazione di variabili numeriche e categoriche, gestisce interazioni complesse tra le variabili, è robusto agli outliers e al rumore e ha una buona capacità di generalizzazione. Inoltre, grazie alla sua capacità di esplorare più profondamente le interazioni tra le caratteristiche, può ottenere ottimi risultati anche in presenza di dati complessi come quelli che potrebbero caratterizzare le statistiche di calcio.

Valutazione

La valutazione del modello è stata eseguita utilizzando diverse metriche di classificazione standard:

- **Accuratezza Media:** L'accuratezza misura quanto spesso il classificatore ha ragione nel complesso. Si misura con la percentuale di predizioni corrette rispetto al totale delle predizioni fatte.
- **Precisione Media:** La precisione misura la qualità delle predizioni positive. In altre parole, tra tutte le predizioni che il modello ha fatto come positive, conta quanti sono effettivamente corretti. Si misura con la proporzione di predizioni positive corrette rispetto al totale delle predizioni fatte per la classe positiva.
- **Richiamo Medio:** Il richiamo misura la capacità del modello di identificare correttamente tutti gli esempi positivi. In altre parole, quanti dei veri positivi sono stati effettivamente rilevati dal modello. Si misura con la proporzione di veri positivi corretti rispetto al totale di tutti gli esempi effettivamente positivi.
- **F1-Score Medio:** L'F1-score è una metrica che cerca di trovare un buon compromesso tra precisione e richiamo. È utile quando c'è bisogno di un equilibrio tra le due, e quando non si vuole privilegiare l'una a discapito dell'altra. Si misura con la media armonica tra precisione e richiamo, che bilancia le due metriche.

- **ROC AUC:** La ROC-AUC misura la capacità di un modello di distinguere tra le classi. Un valore più alto significa che il modello è migliore nel separare le classi. Un AUC di 0.5 indica che il modello è indistinguibile da un classificatore casuale, mentre un AUC di 1.0 significa un classificatore perfetto. Viene definita come l'area sotto la curva della caratteristica operativa del ricevitore (Receiver Operating Characteristic - ROC). La curva ROC è un grafico che rappresenta il tasso di veri positivi (True Positive Rate, TPR) contro il tasso di falsi positivi (False Positive Rate, FPR) per vari threshold.

In particolare, per quanto riguarda il KNN:

```
K-Nearest Neighbors Results:

Cross-validation: 5-fold
Migliori parametri: {'metric': 'manhattan', 'n_neighbors': 10, 'weights': 'distance'}
Accuratezza media: 0.9055406530385719
Precisione media: 1.0
Richiamo medio: 1.0
F1-Score medio: 1.0
ROC AUC: 1.0

Cross-validation: 10-fold
Migliori parametri: {'metric': 'manhattan', 'n_neighbors': 10, 'weights': 'distance'}
Accuratezza media: 0.9272831727205336
Precisione media: 1.0
Richiamo medio: 1.0
F1-Score medio: 1.0
ROC AUC: 1.0

Cross-validation: 15-fold
Migliori parametri: {'metric': 'manhattan', 'n_neighbors': 10, 'weights': 'distance'}
Accuratezza media: 0.9381602153531978
Precisione media: 1.0
Richiamo medio: 1.0
F1-Score medio: 1.0
ROC AUC: 1.0
```

Invece, per quanto riguarda il Gaussian Naive Bayes:

```
Gaussian Naive Bayes Results:

Cross-validation: 5-fold
Migliori parametri: {'var_smoothing': np.float64(0.0001)}
Accuratezza media: 0.8809539666389172
Precisione media: 0.9338778010381169
Richiamo medio: 0.9273174157303371
F1-Score medio: 0.9283274038227824
ROC AUC: 0.9835030620416766

Cross-validation: 10-fold
Migliori parametri: {'var_smoothing': np.float64(0.0001)}
Accuratezza media: 0.9086533234494686
Precisione media: 0.9338778010381169
Richiamo medio: 0.9273174157303371
F1-Score medio: 0.9283274038227824
ROC AUC: 0.9835030620416766

Cross-validation: 15-fold
Migliori parametri: {'var_smoothing': np.float64(0.001)}
Accuratezza media: 0.9156613756613758
Precisione media: 0.9333102400455455
Richiamo medio: 0.9269662921348315
F1-Score medio: 0.927940338431931
ROC AUC: 0.9843744556007441
```

Infine, per quanto riguarda il Random Forest:

```
Random Forest Results:

Cross-validation: 5-fold
Migliori parametri: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100}
Accuratezza media: 0.9097135633459749
Precisione media: 0.9892399976149056
Richiamo medio: 0.9891151685393258
F1-Score medio: 0.9890649644045876
ROC AUC: 0.9998318076031508

Cross-validation: 10-fold
Migliori parametri: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 100}
Accuratezza media: 0.9462157153446997
Precisione media: 0.9892399976149056
Richiamo medio: 0.9891151685393258
F1-Score medio: 0.9890649644045876
ROC AUC: 0.9999026179117126

Cross-validation: 15-fold
Migliori parametri: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Accuratezza media: 0.9655267799127448
Precisione media: 0.9919940510109375
Richiamo medio: 0.9919241573033708
F1-Score medio: 0.9918945787109054
ROC AUC: 0.9999595180616555
```

Ciò ha portato a scegliere il modello Random Forest per prevedere il ruolo di un giocatore. Un esempio di utilizzo è il seguente:

```
Ciao, cosa vuoi fare?
Vuoi che ti suggerisca un giocatore? - Premi 1
Vuoi sapere il ruolo di un giocatore? - Premi 2
Vuoi uscire? - Premi 3
2
Ti chiederò un po' di cose. Iniziamo...
Qual è il nome del giocatore che vuoi classificare?
 sergi roberto
In che anno è nato il giocatore che vuoi classificare?
1992
Quante partite ha giocato il giocatore che vuoi classificare? (se non sai come calcolare il numero di partite giocate digita 0)
10
Quanti goal ha segnato il giocatore che vuoi classificare?
3
Quanti assist ha fatto il giocatore che vuoi classificare?
2
Quanti cartellini gialli ha preso il giocatore che vuoi classificare?
4
Quanti cartellini rossi ha preso il giocatore che vuoi classificare?
0
Il ruolo del giocatore sergi roberto è:  df
```

Raccomandazione dei giocatori

Sommario

Il modulo di raccomandazione dei giocatori del progetto FootballPlayers suggerisce calciatori in base ai dati forniti dall'utente, sfruttando le statistiche dei giocatori e le informazioni storiche. Il sistema utilizza la **distanza di Manhattan** per calcolare la similarità tra i giocatori. Questa metrica consente di misurare quanto due profili di giocatori siano simili tra loro, basandosi su attributi come gol segnati, assist, cartellini e ruoli. La Knowledge Base (KB) è costituita dal dataset pre-elaborato dei giocatori, che include le statistiche rilevanti per la raccomandazione. Questo approccio consente di fornire suggerimenti personalizzati e pertinenti, migliorando la precisione delle raccomandazioni.

Strumenti utilizzati

Il modulo di raccomandazione è stato costruito utilizzando il seguente strumento:

- **Scikit-Learn:** In modo particolare, pairwise_distances. Serve per calcolare le distanze tra i dati, in questo caso specificamente la distanza di Manhattan, che misura la similarità tra i giocatori basandosi sulle caratteristiche specificate.

Decisioni di Progetto

Le principali decisioni progettuali per il sistema di raccomandazione includono:

- **Algoritmo di Similarità**: È stata scelta la distanza di Manhattan per calcolare la similarità tra i dati. Questa metrica è utile per dati numerici che potrebbero non essere distribuiti in modo uniforme e per gestire dati sparsi o mancanti senza penalizzarli troppo. La formula utilizzata è una versione invertita della distanza di Manhattan: 1 / (1 + distanza), il che significa che valori più piccoli della distanza (maggiore similarità) avranno una similarità più alta. I dati dei giocatori potrebbero avere scale diverse. La distanza di Manhattan non è influenzata da differenze nei valori numerici di variabili che potrebbero non avere una distribuzione uniforme, come succede invece nella distanza euclidea. Poiché la distanza di Manhattan somma le differenze assolute, è meno sensibile a valori estremi rispetto alla distanza Euclidea. Questo è utile per la presenza di dati che includono valori estremi, non volendo che distorcano il calcolo della similarità.
- **Filtraggio delle Colonne in Base ai Dati Utente**: A seconda dei parametri forniti dall'utente (ad esempio, se l'utente non fornisce una posizione o un numero di gol), le colonne corrispondenti vengono rimosse dal dataset, evitando di utilizzare dati non

necessari per la raccomandazione. Questo approccio rende il sistema dinamico e flessibile, adattandosi ai dati disponibili e alle preferenze dell'utente.

- **Suggerimento dei Giocatori**: Una volta calcolata la similarità tra l'utente e i giocatori nel dataset, i giocatori vengono ordinati in base alla loro similarità con l'utente. I primi 10 giocatori con la maggiore similarità sono suggeriti all'utente. Un controllo aggiuntivo viene fatto per escludere i giocatori che sono già stati selezionati dall'utente, garantendo che i suggerimenti siano pertinenti.
- **Uso di un Dizionario di Conversione**: Un dizionario di conversione (dizionario.csv) viene caricato per mappare le categorie di variabili, come la posizione o il nome del giocatore, in un formato utilizzabile dal sistema. Viene utilizzato per convertire i valori di input dell'utente nel formato corretto per essere confrontato con il dataset.

Valutazione

Il sistema di raccomandazione è stato valutato attraverso vari scenari di input per testare la precisione e l'utilità delle raccomandazioni fornite. Le metriche adottate includono:

- **Precisione delle Raccomandazioni:** Valutata osservando la pertinenza dei suggerimenti rispetto alle preferenze inserite dall'utente.
- **Feedback dell'Utente:** Gli utenti hanno fornito feedback sulla coerenza dei suggerimenti rispetto alle loro aspettative. Le raccomandazioni risultano pertinenti e ben allineate con le preferenze degli utenti.

Il sistema dimostra una solida integrazione della distanza di Manhattan per la raccomandazione di giocatori, permettendo una selezione più accurata e mirata basata sui dati forniti dall'utente.

Esempio di raccomandazione di giocatori:

```
Ciao, cosa vuoi fare?
Vuoi che ti suggerisca un giocatore? - Premi 1
Vuoi sapere il ruolo di un giocatore? - Premi 2
Vuoi uscire? - Premi 3
1
Ti farò delle domande per poterti suggerire un giocatore.
Suggeriscimi il nome di un giocatore che ti piace
alphonso davies
Sai dirmi in che ruolo gioca questo giocatore? (Sì o No)(se non sai i ruoli esistenti digita 0)
df
Sai dirmi in che anno è nato questo giocatore? (Sì o No)
no
Sai dirmi quante partite ha giocato questo giocatore? (Sì o No)(se non sai come calcolare il numero di partite giocate digita 0)
si
Quante partite ha giocato?
23
Sai dirmi quanti goal ha segnato questo giocatore? (Sì o No)
si
Quanti goal ha segnato?
2
Sai dirmi quanti assist ha fatto questo giocatore? (Sì o No)
si
Quanti assist ha fatto?
5
Sai dirmi quanti cartellini gialli ha preso questo giocatore? (Sì o No)
si
Quanti cartellini gialli ha preso?
0
```

```
Sai dirmi quanti cartellini rossi ha preso questo giocatore? (Sì o No)
si
Quanti cartellini rossi ha preso?
0
I giocatori suggeriti in base alle tue preferenze sono:

1) jeremie frimpong
2) christopher operi
3) ben osborn
4) matteo gabbia
5) przemysław frankowski
6) max aarons
7) roberto gagliardini
8) antonino gallo
9) juan foyth
10) caleb okoli
```

Conclusioni

Il progetto FootballPlayers ha dimostrato un'efficace integrazione di tecniche di machine learning e analisi dei dati per la classificazione, il clustering e la raccomandazione dei giocatori. Attraverso l'uso di algoritmi di classificazione come Gaussian Naive Bayes e Random Forest, è stata ottenuta una buona accuratezza nella previsione dei ruoli dei giocatori, sebbene alcune metriche di precisione richiedano ulteriori ottimizzazioni. Il modulo di raccomandazione ha implementato la distanza di Manhattan per suggerire giocatori simili, mostrando risultati promettenti nella personalizzazione delle raccomandazioni.

Tuttavia, il progetto potrebbe essere migliorato nel tempo sotto differenti aspetti. Ad esempio, l'implementazione di algoritmi di clustering più avanzati o la rifinitura dei parametri del modello di raccomandazione potrebbero migliorare ulteriormente la precisione e la qualità delle raccomandazioni. Inoltre, l'integrazione di fonti di dati aggiuntive e l'adozione di tecniche di deep learning potrebbero arricchire il sistema, offrendo nuove prospettive per l'analisi e la predizione delle prestazioni dei giocatori. Questi aspetti rappresentano aree promettenti per future estensioni e miglioramenti, offrendo opportunità per approfondire la comprensione e l'analisi del mondo del calcio attraverso approcci più sofisticati e dati più completi.

Riferimenti Bibliografici

[1] Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson. [Link al libro]