# Eventi calcistici: modello predittivo delle reti segnate

Luca Ballarati[1], Francesco Martinelli[1], Francesco Oliviero[1], Cristiano Ruttico[1], Matteo Sala[1]

**Sommario**

La domanda di ricerca che ci siamo posti è quella di determinare nel modo più corretto possibile le condizioni migliori che portino alla realizzazione di una rete. Per fare questo, ci siamo serviti dei modelli di Machine Learning, in particolare si è optato per la costruzione di modelli predittivi in grado di restituire un esito: goal oppure no goal. Inoltre, sono stati implementati modelli di classificazione per valutare la presenza di variabili determinanti ai fini dell'output ricercato. Supponendo corretti i dati presi in considerazione, successivamente ad un processo di valutazione e comparazione dei metodi, sono stati selezionati i modelli migliori in termini di performance e attendibilità per rispondere alla domanda di ricerca. I risultati ottenuti consentiranno di fornire uno strumento valido per le future scelte tattiche del mister, con un aumento delle possibilità di concretizzare le opportunità a disposizione della squadra nel corso della partita.

*1 Università degli Studi di Milano Bicocca, CdLM Data Science*

# Indice



# 1. Introduzione

Nel gioco del calcio una squadra, per ottenere la vittoria, deve realizzare più reti della sua avversaria. A questo proposito, risulta necessario individuare nella pluralità di eventi quelli che incidono in modo significativo sull'esito di un'azione di gioco, prendendo come riferimento la fase realizzativa. Ciò rappresenta l'oggetto di studio della *match analysis,* l'approccio analitico al calcio, le cui origini possono essere individuate nel lavoro di Charles Reep, un contabile inglese della Royal Air Force[1]. Reep negli anni 30' del XX secolo iniziò ad annotarsi i dati di centinaia di partite del campionato inglese e studiare quanto raccolto. Nel 1968 pubblicò, insieme allo statistico Benjamin Bernard, il risultato dei sui studi sul *Journal of the Royal Statistical Societ*y[2]. Il suo lavoro, sebbene presentasse notevoli limiti metodologici, ebbe una notevole influenza sul calcio inglese e scandinavo. Nei decenni successivi i progressi tecnologici hanno semplificato e ottimizzato le modalità di raccolta dei dati, aumentandone in modo esponenziale la disponibilità per ogni evento sportivo, al punto che sono state fondate società come Opta,

Prozone e StatDNA, specializzate in questo tipo di servizi. In concomitanza con l'incremento dei dati a disposizione, in letteratura sono stati pubblicati molti studi con l'obiettivo di sviluppare modelli predittivi per i risultati delle partite di calcio. Alcuni di questi si sono concentrati sugli *expected goals*[3], analizzando i fattori che influenzano la probabilità che un tiro diventi un goal. La presente analisi rientra in tale ambito di ricerca in quanto si pone l'obiettivo di determinare un modello predittivo di questo tipo, utilizzando un dataset derivante da un precedente lavoro di webscraping e di integrazione di dati di diversa origine riguardanti 9,074 partite dei maggiori campionati europei. L'obiettivo di questa analisi è stato in particolare quello di prevedere sulla base dei dati forniti relativi agli eventi, se gli stessi possano condurre ad una rete o ad una mancata concretizzazione dell'occasione da goal.
La ricerca è così strutturata: dopo una breve presentazione del database e un'analisi preliminare sui dati, vengono presentati i modelli utilizzati e le misure di performance impiegate. Infine, sono riportate le analisi ed i risultati ottenuti

## 1.2 Dataset

Il set di dati[4] prende in considerazione 941.009 eventi relativi a 9.074 partite svoltesi dalla stagione 2011/2012 alla stagione 2016/2017 (sino al 25.01.2017) dei 5 maggiori campionati europei di calcio (Inghilterra, Spagna, Germania, Italia e Francia). Il dataset è composto da 22 variabili, 18 delle quali sono categoriali nominali:

**id_odsp**: identificatore della partita

**id_event**: identificativo dell'evento

**sort_order**: sequenza di eventi in una partita;

**time**: minuti di gioco;

**text**: descrizione dell'evento;

**event_type**: evento primario di gioco (tiro, angolo, fallo…);

**event_type2**: evento secondario di gioco (lancio fallito, passaggio chiave, espulsione, autogol…);

**side**: home, away;

**event_team**: squadra che produce l'evento di gioco (in caso di autogol è la squadra che ne ha beneficiato);

**opponent**: squadra che ha subito l'evento;

**player**: giocatore coinvolto nell'evento primario;

**player2**: giocatore coinvolto nell'evento secondario;

**player_in**: giocatore che entra dalla sostituzione;

**player_out**: giocatore che esce dalla sostituzione;

**shot_place**: definisce la direzione del tiro (alto, bloccato, al centro…);

**shot_outcome**: risultato del tiro (in porta, fuori, bloccato, traversa…);

**is_goal**: 0=no goal, 1=goal;

**location**: posizione del campo dove è avvenuto l'evento;

**bodypart**: parte del corpo con cui viene effettuato il tiro;

**assist_method**: metodo dell'assist;

**situation**: situazione di gioco (campo aperto, punizione, angolo…);

**fast_break**: contropiede sì/no.

Nonostante siano variabili categoriche nominali il database ne presenta alcune sottoforma di numeri interi ordinali. Tali variabili sono: *event_type, event_type2, side, shot_place, shot_outcome, is_goal, location, bodypart, assist_method, situation, fast_break*. Di conseguenza, per poter applicare i vari modelli di classificazione, una volta caricato e letto il dataset, queste ultime sono state convertite in stringhe.

Con questo database è quindi possibile valutare più accuratamente tutte le varie situazioni di gioco create da una squadra e determinare quali siano le migliori occasioni che consentano ad un

giocatore di aumentare le proprie chance di segnare. Numerose variabili influenzano la realizzazione di un goal come ad esempio la posizione del campo in cui viene effettuato il tiro, la parte del corpo con cui si colpisce la palla, la direzione del tiro, la situazione di gioco stessa e così via. Molte di queste sono state analizzate in questo set di dati per poter prevedere con maggiore precisione possibile se un tiro possa produrre un effetto positivo (goal) o negativo (no goal) per la squadra che lo ha effettuato.

# 2. Preprocessing

***Input attributes***

Il database in questione registra tutti gli eventi verificatisi durante una partita: dall'ammonizione o espulsione alla sostituzione di un giocatore, da un fuorigioco ad un rigore concesso, da un tiro ad una punizione conquistata e così via. La nostra domanda di ricerca, tuttavia, si è concentrata solo sull'esito del tiro.

Per tale motivo, applicando un *row filter node*, sono state prese in considerazione solo le righe con event tipe 1 corrispondente ad attempt,

Ciò ha comportato una riduzione dei record da 941,006 a 229,135. Come variabile target (e quindi la variabile classe che costituirà l'output dei diversi modelli di classificazione) è stato posto l'attributo *is_goal*.

Come secondo step, data la nostra conoscenza del dominio, sono stati rimossi quegli attributi poco informativi come *id_odsp, id_event, text, player_in, player_out* certamente ininfluenti per il risultato di un tiro.

Successivamente sono stati scartati dal dataset altri attributi superflui per la nostra domanda di ricerca, vale a dire: *sort_order, event_team, event_2, opponent, player e player 2*..

***Missing values***

Dalle statistiche descrittive è emerso che nel nuovo set di dati formato da 229,135 sono presenti 1,683 valori mancanti, tutti appartenenti all'attributo *shot_place*. Tra questi record vi sono anche 637 valori mancanti nell'attributo *shot_outcome*, che sono perciò conseguenza dei primi.

Dato che il dataset si compone di un elevato numero di record e solo lo 0.73% (1,683 su 229,135) presenta valori mancanti è stata presa la decisione di rimuovere i record (*record removal*) con almeno un valore mancante su un attributo

## 2.1 Modelli e Misure di Performance

***Modelli***

In questo studio sono state applicate diverse tecniche di classificazione supervisionata al fine di addestrare il modello per individuare la classe target "*Is_goal*"; nello specifico le nostre analisi si sono basate sull'utilizzo del software Knime, il quale utilizza sia nodi implementati internamente che nodi basati su Weka.

**-Modelli euristici**: Abbiamo utilizzato l'albero di decisione, in particolare ci siamo serviti del J48 implementato da Weka, e del Random Forest, anch'esso implementato da Weka.

**-Modelli basati su regressione**: Regressione logistica, implementata da Weka.

**-Modelli basati su separazione**: Abbiamo utilizzato le support vector machines(SVM) con kernel: Poly. Inoltre abbiamo utilizzato il percettrone multistrato.

**-Modelli probabilistici**: abbiamo utilizzato NaiveBayer e NaiveBayes tree (NB Tree).

***Misure di performance***

Ai fini di valutazione delle performance dei modelli di classificazione una misura molto importante è rappresentata dall'*Accuracy*.

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

In particolare:

TP e TN rappresentano il numero di istanze appartenenti rispettivamente alla classe positiva e negativa classificate correttamente.

FP e FN indicano il numero di istanze appartenenti alla classe negativa e positiva, classificate in

maniera errata.

Nel caso in esame ci troviamo a dover gestire una situazione di "classe sbilanciata": in particolare la classe di interesse, ovvero quella che ha come modalità del carattere "is_goal" = {0,1} il valore 1 si presenta con una proporzione del 2.6%. Il calcolo dell'Accuracy come misura di riferimento per valutare il modello è utilizzabile in maniera efficace nel qual caso tra le due classi siano bilanciate ovvero siano presenti in egual modo all'interno del dataset. Come occorre dunque procedere nel caso di dataset sbilanciato? È necessario introdurre due ulteriori misure:

$$Recall = \frac{TP}{TP + FN}$$

Anche definita come True Positive Rate o sensibilità, rappresenta la porzione di record positivi correttamente classificati dal modello; un basso valore di Recall indicherà un alto numero di falsi negativi. Nel nostro caso, questa misura è di particolare interesse poiché è in grado di dirci quante volte in percentuale il modello è in grado di predire correttamente un goal.

Avremo poi la

$$Precision = \frac{TP}{TP + FP}$$

La quale rappresenta la frazione di record che sono effettivamente positivi tra tutti quelli predetti come tali; più alto è il suo valore, minore quindi sarà il numero di falsi positivi.

Queste ultime due misure possono entrare in conflitto tra loro: è possibile che si costruiscano dei modelli che massimizzano unicamente una delle due misure. Per ovviare a questo problema calcoliamo la F1-measure data dalla media armonica tra Recall e Precision. Nel caso sia ottenga un valore elevato di questa misura si può essere ragionevolmente certi che Recall e Precision siano indicativamente alti. La formula della F1-measure è data da:

$$F1 - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Un ulteriore metodo utilizzato per valutare i modelli di classificazione è la curva ROC che riporta sull'asse delle ordinate, la percentuale del numero totale di TruePositive (TP), e sull'asse delle ascisse la percentuale di falsi positivi (FP). L'AUC rappresenta il valore dell'area sotto la curva ROC ed è una buona misura di performance perché consente di definire le prestazioni del classificatore: a valori più alti corrisponde un modello migliore.

# 3. Analisi dei risultati

L'obiettivo dell'analisi consiste nello sviluppo di un modello con capacità di predizione circa la conversione di un tiro (event_type = 1 nel dataset) in goal, rappresentata dalla variabile binaria is_goal.

Il metodo di classificazione ha inizialmente tenuto conto di tutte le variabili utilizzabili potenzialmente esplicative, quindi le variabili relative a: situazione tecnico-tattica della singola azione (come contropiede e tipo di assist), il momento della partita e i riferimenti spaziali del tiro (posizione da cui avviene il tiro e direzione della conclusione). Trovandoci alla presenza di un database con classi sbilanciate si è fatto riferimento a tecniche adatte a trattare database con tali caratteristiche.

Infine, essendo il proposito della ricerca la possibilità di prevedere le situazioni di gioco più pericolose all'interno della partita, la misura di valutazione a cui presteremo maggiormente attenzione sarà la Recall: l'obiettivo è infatti quello di prevedere adeguatamente il maggior numero di istanze positive possibile cercando, in secondo luogo, di mantenere livelli di Precision accettabili.

Di seguito nel dettaglio i diversi approcci impiegati nella classificazione.

## 3.1 Metodo Hold-Out

Il dataset è stato inizialmente suddiviso, con un partizionamento casuale, in un training set (formato dal 67% dei record) e in un test set (con il restante 33%), come previsto dal metodo hold-out. L'addestramento dei modelli è stato svolto con il traning set ottenuto e la validazione con il

test set. I risultati ottenuti sono mostrati nella Tabella A.

| Modello | Recall | Precision | F-Measure | Accuracy | AUC |
|---|---|---|---|---|---|
| ***J48*** | 0.605 | 0.770 | 0.677 | 0.942 | 0.959 |
| ***RdmForest*** | 0.588 | 0.676 | 0.629 | 0.931 | 0.902 |
| ***Logistic*** | 0.596 | 0.769 | 0.671 | 0.942 | 0.966 |
| ***SVM-poly*** | 0.444 | 0.886 | 0.591 | 0.939 | 0.719 |
| ***MLP*** | 0.612 | 0.763 | 0.679 | 0.942 | 0.967 |
| ***NB*** | 0.807 | 0.519 | 0.632 | 0.906 | 0.957 |
| ***NBTree*** | 0.631 | 0.752 | 0.686 | 0.942 | 0.967 |

*Tabella A*

Trovandoci in una situazione di class imbalance problem i risultati ottenuti risentono di questo fenomeno, in particolare: l'Accuracy presenta valori elevati dovuti alla presenza di molte istanze negative e alla tendenza dei modelli a classificare le istanze ambigue come appartenenti a tale classe, tuttavia, le misure per la valutazione della performance con classi sbilanciate, ovvero Recall, Precision e F-measure hanno valori meno soddisfacenti, in particolare la Recall.

Il modello NaiveBayes è l'unico con Recall (0.807) superiore alla Precision (0.519), quindi l'unico che sembra tendere a classificare le istanze ambigue come positive.

Tutti gli altri modelli hanno Precision superiore alla Recall. In particolare, il modello SVM-poly mostra un'elevata Precision a discapito della più bassa Recall fra i modelli (0.444).

L'AUC mostra valori soddisfacenti già in questa prima analisi fatta eccezione per SVM-poly.

## 3.2 Metodo Hold-out con Feature Selection

Al fine di migliorare le performance del modello si è deciso di utilizzare il metodo della feature selection, ovvero di selezionare in base alla rilevanza solo quegli attributi significativi per la previsione della variabile target. Dopo aver testato due feature selection, ovvero Filtro Univariato e Filtro Multivariato, sì è scelto di utilizzare quest'ultima via dimostratasi la più performante tra le due. Tramite l'utilizzo del nodo Weka AttributeSelectedClassifier con metodo ClassifierSubsetEval basato sul classificatore NBTree, ottimizzando la F-measure abbiamo ottenuto come subset delle variabili esplicative rilevanti {time, shot_place, shot_outcome, location, bodypart, assist_method, situation, fastbreak}.I risultati dei modelli in seguito alla feature selection sono mostrati nella Tabella B.

| Modello | Recall | Precision | F-Measure | Accuracy | AUC |
|---|---|---|---|---|---|
| ***J48*** | 0.604 | 0.77 | 0.677 | 0.942 | 0.959 |
| ***RdmForest*** | 0.591 | 0.691 | 0.637 | 0.933 | 0.91 |
| ***Logistic*** | 0.588 | 0.774 | 0.668 | 0.942 | 0.966 |
| ***SVM-poly*** | 0.444 | 0.886 | 0.591 | 0.939 | 0.719 |
| ***MLP*** | 0.61 | 0.759 | 0.676 | 0.942 | 0.967 |
| ***NB*** | 0.808 | 0.519 | 0.632 | 0.906 | 0.957 |
| ***NBTree*** | 0.63 | 0.753 | 0.686 | 0.942 | 0.967 |

*Tabella B*

Come evidenziato dai risultati, il miglioramento delle performance del modello in seguito alla Feature selection (con la metodologia sopra descritta) è stato pressoché nullo.

## 3.3 Metodo Hold-out con Feature Selection e Equal Size Sampling

Una delle possibili problematiche più rilevanti con dataset sbilanciati consiste nella tendenza dei classificatori a classificare le istanze ambigue come negative (ovvero appartenenti alla classe maggiormente rappresentata).

Questo avviene perché il modello apprende su un training set in cui il numero di istanze negative è sensibilmente maggiore di quelle positive. Nel nostro caso, tuttavia, c'è maggior interesse per la corretta classificazione delle istanze positive (TruePositive), piuttosto che nella limitazione dei falsi positivi. Un approccio possibile è l'utilizzo della tecnica dell'Equal Size Sampling applicata sul training set: questa tecnica ha la funzione di ribilanciare le istanze del training set basandosi sull'attributo target.In questo modo il modello, apprendendo su unsubset con una classe positiva più numerosa, acquisisce una maggiore tendenza a favorire la classificazione delle istanze ambigue come appartenenti alla classe positiva.

Nella Tabella C si mostrano i risultati adottando la tecnica dell'Equal Size Sampling.

| Modello | Recall | Precision | F-Measure | Accuracy | AUC |
|---|---|---|---|---|---|
| ***J48*** | 0.953 | 0.417 | 0.58 | 0.862 | 0.962 |
| ***RdmForest*** | 0.883 | 0.431 | 0.58 | 0.872 | 0.938 |
| ***Logistic*** | 0.967 | 0.4 | 0.566 | 0.851 | 0.966 |
| ***SVM-poly*** | 0.966 | 0.4 | 0.566 | 0.852 | 0.903 |
| ***MLP*** | 0.939 | 0.435 | 0.595 | 0.872 | 0.965 |
| ***NB*** | 0.999 | 0.327 | 0.492 | 0.794 | 0.957 |
| ***NBTree*** | 0.968 | 0.401 | 0.567 | 0.852 | 0.958 |

*Tabella C*

Si può notare dai risultati una consistente crescita della Recall ed una contestuale riduzione della Precision. I risultati sono in linea con quanto previsto dalla teoria.

In particolare: tutti i modelli (tranne RandomForest) forniscono adesso una Recall superiore a 0.90 ed una Precision con media e mediana di circa 0.4.

Il modello NaiveBayes, che già nelle precedenti elaborazioni si era dimostrato il modello con maggiore tendenza alla classificazione positiva, presenta la più elevata Recall tra i modelli con lo 0.997, al contempo ha valori inferiori agli altri modelli in tutte le altre misure di valutazione.

I modelli Logistic, SimpleLogistic, NBTree e SVM-poly hanno misure di valutazione molto simili tra di loro con valori medi tra i modelli trattati. MLP e RandomForest offrono i risultati con Recall inferiore (rispettivamente 0.939 e 0.883), la Precision e la F-measure sono invece superiori agli altri modelli. L'AUC della SVM è migliorata.

Il MLP è il modello con i valori più alti di Precision (0.435), F-Measure (0.595) e AUC (0.966) con una Recall pari a 0.939.

## 3.4 Comparazione tra i modelli

Avendo ottenuto risultati molto simili tra di loro per i vari modelli si è deciso di implementare varie modalità di comparazione tra i modelli al fine di identificare i più performanti.

***ROC Curve***

La ROC curve può essere utilizzata come metodo per la comparazione dei modelli. Con questo metodo abbiamo valutato, in prima battuta, la differenza tra i modelli derivanti dall'apprendimento con Equal Size Sampling (*Figura 2*) e senza (*Figura 1*) e poi la differenza tra i vari modelli nelle singole figure.

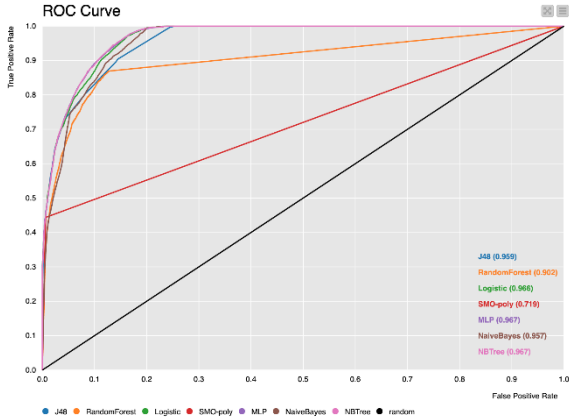


*Figura 1*

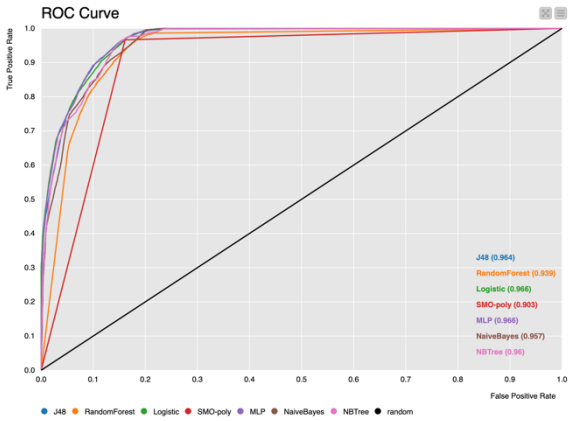


*Figura 2*

I risultati ottenuti da tutti i modelli in entrambi i casi sono soddisfacenti, in quanto tutte le curve sono al disopra della retta rappresentante il modello casuale.

Si può notare un tendenziale miglioramento derivante dall'utilizzo dell'Equal Size Sampling, in particolare per RandomForest e SVM-poly. Le ROC Curve dei modelli nelle due figure presentano valori molto simili tra di loro, fanno eccezione il modello RandomForest in *Figura 1* e SVM-poly, che è il modello meno prestante in entrambe le figure come confermato anche dai valori dell'AUC. Questo metodo di valutazione non consente di identificare un modello migliore in assoluto, in quanto in entrambi i casi nessuna delle curve sovrasta le altre indipendentemente dall'ascissa: giudicando i grafici e le AUC i tre modelli più performanti sembrano essere, tuttavia, MLP, Logistic e NBTree.

### *Lineplot*

Per svolgere un'analisi maggiormente approfondita e individuare il classificatore migliore è stata presa la decisione di adottare un partizionamento hold-out su due fasi. Il prodotto finale consiste in un training set e in due test set (test set A e B). I classificatori sono stati addestrati sulla partizione training set e testati sia sulla partizione A che sulla partizione B. Una volta implementato il nodo scorer, che calcola le misure di performance, è stato effettuato un confronto tra le diverse misure di Recall

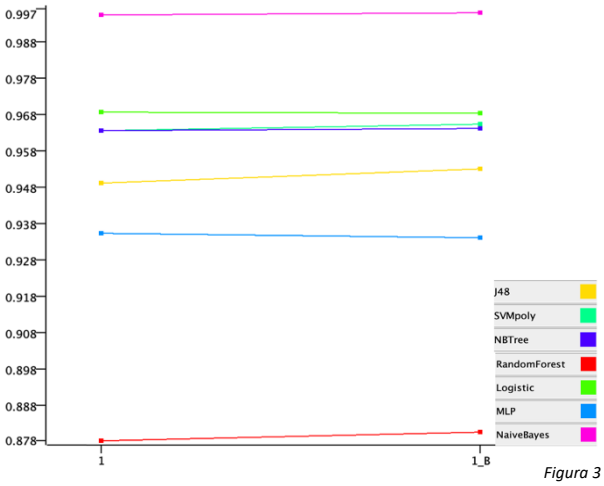


*Figura 3*

I migliori risultati sono stati ottenuti per il modello NaiveBayes, in quanto la sua linea sovrasta tutte le altre. Dopo di questo troviamo, in ordine, Logistic, SVM-poly e NBtree con risultati simili, J48, MLP e infine con valori considerevolmente inferiori RandomForest come si puó notare nella *Figura 3*

### *Intervalli di Confidenza*

Come modalità di comparazione più approfondita si è scelto infine di utilizzare gli intervalli di confidenza. Con questo metodo si quantifica il grado di confidenza con cui si può affermare che il reale valore della misura di interesse si trova nell'intervallo trovato, si sfrutta una stima ad intervallo rispetto ad una stima puntuale. Il grado di confidenza scelto è del 99% e per il calcolo degli intervalli è stata utilizzata la formula degli intervalli di Wilson. Gli intervalli sono significativi se il limite inferiore di un modello è superiore al limite superiore dell'altro. Nel caso oggetto di studio ci si è concentrati su Recall e F-measure. I risultati sono mostrati in *Figura 4* per la Recall e in *Figura* 5 per l'F-measure.

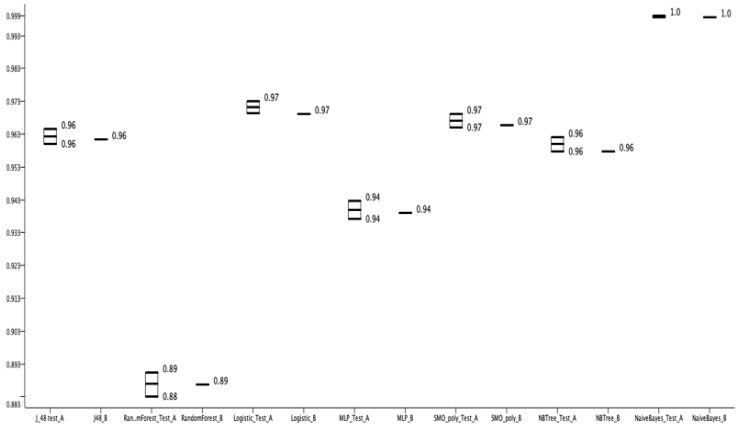

*Figura 4*

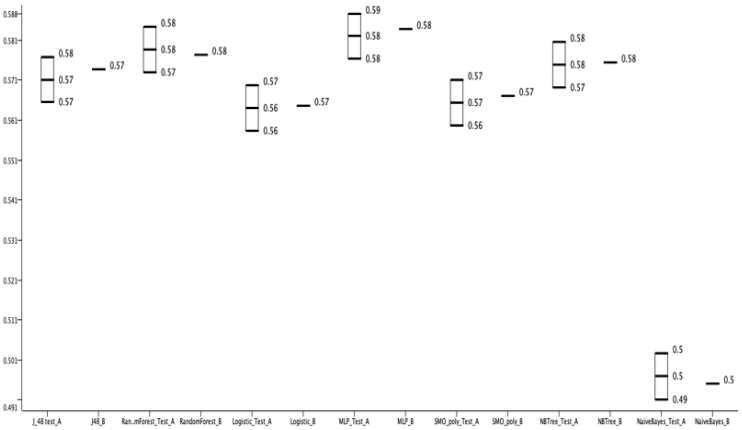

*Figura 5*

Come si evince dai risultati mostrati in *Figura 4* il miglior modello in termini di Recall è il modello NaiveBayes, mentre SVM, Logistic, J48 e NBTree non hanno valori tali da consentire un ranking univoco con questo approccio. Seguono MLP e infine RandomForest.

Per quanto riguarda l'F-measure la questione cambia, in particolare l'ottima Recall ottenuta dal modello NaiveBayes e il basso valore in F-measure è frutto di un forte sbilanciamento tra Recall e Precision nel modello. Il modello in apparenza più performante in termini di F-measure risulta essere il MLP anche se non c'è sufficiente certezza: infatti con il 99% di confidenza gli intervalli di tutti i modelli eccetto SVM, Logistic e NaiveBayes hanno all'interno del proprio intervallo il valore corrispondente al limite inferiore dell'intervallo del modello MLP.

I vari metodi di confronto hanno rimarcato come il modello NaiveBayes sia il più prestante in termini di Recall, allo stesso modo con l'ultima analisi

sull'F-measure si è messo in evidenza come il limite di questo modello sia un basso grado di Precision e conseguentemente della F-measure.

# 4. Conclusioni e sviluppi futuri

Per rispondere alla domanda di ricerca inizialmente sono state utilizzate tutte le features e quindi un database sbilanciato. Successivamente, a causa delle performance deludenti, soprattutto in termini di Recall, è stata effettuata una feature selection con il filtro multivariato CfsSubsetEval. Con questo approccio non sono stati riscontrati particolari miglioramenti in termini di performance a causa dello sbilanciamento delle classi. L'approccio dell'Equal Size Sampling in questo senso ha offerto una valida soluzione come mostrato dai risultati ottenuti. I valori di Recall raggiunti sono stati particolarmente soddisfacenti per tutti i modelli, fatta eccezione per la RandomForest. Infine, dati i risultati simili tra i vari modelli si è scelto di approfondire l'analisi attraverso vari metodi di comparazione tra modelli. Da questa ulteriore analisi è emerso che i migliori classificatori sono NBTree e Naive Bayes. Su questi modelli sono stati registrati alti valori di Recall, ma bassi valori di Precision. Questo significa che i modelli tendono a identificare con ottimi *rate* i tiri che si convertono in goal, ma al contempo considerano pericolose anche un discreto numero di azioni di tiro che si rivelano inconcludenti. Se si considerano accettabili le performance di Recall di tutti i modelli, il MLP è il modello che offre le migliori performance in termini di Precision e *F*−measure.

L'analisi svolta era finalizzata ad individuare un modello generale applicabile per l'analisi di ogni match. Un potenziale sviluppo di questo approccio potrebbe essere quello di implementare un'analisi simile, basata però su dataset relativi a un'unica squadra in modo tale da ottenere modelli specifici per match specifici. In particolare, questo approccio permetterebbe di utilizzare le variabili scartate nell'analisi poiché categoriche e caratterizzate da un eccesso di valori unici, inadatti ad essere discretizzati o binarizzati (nel dettaglio gli attributi relativi a giocatori e squadre). In questo modo si dovrebbero poter ottenere performance migliori. La principale problematica sarebbe senz'altro rappresentata dall'insufficienza di dati. Tuttavia, un'analisi che coinvolga più stagioni in società che vivono *cicli* (ovvero più stagioni consecutive con tipo di gioco, staff tecnico e rosa simili) potrebbe sopperire alla problematica.

## 4.1 Riferimenti bibliografici


[1] https://www.ilpost.it/2016/11/15/errore-che-rovino-il-calcio-inglese/

[2] C . Reep, B . Benjamin. *Skill and Chance in Association Football. Journal of the Royal Statistical Society* ., 131(4):581–585, 1968

[3] Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. "*Quality vs Quantity": Improved Shot Prediction in Soccer using Strategic Features from Spatiotemporal Data*. In Proc. 9th Annual MIT Sloan Sports Analytics Conference, pages 1–9, 2015. 4, 23

[4] https://www.kaggle.com/secareanualin/football-event