Progetto per Corso Machine Learning

Expected Goals – xG

Autore	Matricola	E-mail
Francesco Pio Cataudo	0512116773	f.cataudo1@studenti.unisa.it
Francesco Santoro	0512117079	f.santoro42@studenti.unisa.it

Sommario

Introduzione

Scopo del progetto

Il presente progetto ha come obiettivo la **costruzione e la valutazione di un modello di Expected Goals (xG)** basato su tecniche di *machine learning*, utilizzando dati evento relativi ai tiri in porta. Lo scopo principale è stimare la **probabilità che un tiro si trasformi in gol** a partire da informazioni di natura **spaziale**, derivate dalla posizione del tiro sul campo, quali la distanza e l'angolo rispetto alla porta, e da un insieme di **variabili temporali di base e descrittive dell'evento**, tra cui il minuto di gioco e alcune caratteristiche del tiro, come la parte del corpo utilizzata, la tecnica di esecuzione, il tipo di azione e la presenza di pressione avversaria.

Il progetto copre l'intera **pipeline di analisi dei dati**, includendo le fasi di caricamento e preprocessing, *feature engineering*, addestramento di diversi modelli di classificazione e valutazione delle prestazioni mediante metriche standard. Le probabilità di xG ottenute vengono infine confrontate con lo **StatsBomb xG**, utilizzato esclusivamente come **riferimento esterno di benchmark** e non come variabile di input del modello, al fine di valutare l'affidabilità e l'efficacia delle stime prodotte.

Definizione del problema

Contesto del problema

Negli ultimi anni, l'analisi dei dati sportivi ha assunto un ruolo sempre più rilevante nel supporto alle decisioni tecniche e strategiche. In particolare, nel calcio, la crescente disponibilità di **dati evento strutturati** consente di analizzare in modo quantitativo situazioni di gioco complesse, come le azioni di tiro verso la porta.

In questo progetto viene affrontato il problema della **predizione dell'esito di un tiro**, con l'obiettivo di stimare se un'azione di tiro si concluderà con la realizzazione di un goal oppure no. Tale problema è comunemente noto nell'ambito dello *sport analytics* come **Expected Goals (xG)**, ovvero la stima della probabilità che un tiro produca una rete sulla base delle sue caratteristiche osservabili.

Dal punto di vista del *Machine Learning*, il problema è formulato come un **task di classificazione binaria supervisionata**, in cui:

- l'unità di analisi è rappresentata da un singolo evento di tiro (*shot*);
- la variabile target assume valore:
    - **1 (Goal)** se il tiro termina con una rete;
    - **0 (No Goal)** in tutti gli altri casi.

Il task presenta diverse sfide rilevanti dal punto di vista metodologico. In primo luogo, il dataset risulta **sbilanciato**, poiché solo una piccola percentuale dei tiri effettuati si traduce effettivamente in un goal. Inoltre, le informazioni associate a ciascun evento di tiro possono risultare **eterogenee o parzialmente incomplete**, rendendo necessaria un'accurata fase di preprocessing e gestione dei valori mancanti. Infine, la relazione tra le caratteristiche del tiro e il suo esito è spesso **non lineare**, richiedendo l'impiego di modelli capaci di catturare interazioni complesse tra le feature disponibili.

L'obiettivo del progetto è quindi quello di **progettare e valutare una pipeline di Machine Learning completa**, che includa:

- l'analisi e la preparazione dei dati;
- la selezione e l'ingegnerizzazione delle feature più informative;
- l'addestramento di modelli di classificazione studiati durante il corso;
- la valutazione delle prestazioni tramite metriche appropriate;

- l’analisi dei principali errori commessi dai modelli.

Il progetto si propone infine di **dimostrare la corretta applicazione delle tecniche di Machine Learning apprese**, ponendo particolare attenzione alla motivazione delle scelte progettuali adottate e all’interpretazione critica dei risultati ottenuti.

Specifiche PEAS

Performance

- **Accuratezza delle previsioni**: capacità del modello di distinguere correttamente tra tiri che producono un goal e tiri che non lo producono, valutata tramite metriche di classificazione.
- **Capacità discriminativa**: elevati valori di ROC-AUC e Average Precision, indicativi di una buona separazione tra le due classi anche in presenza di dataset sbilanciati.
- **Robustezza**: mantenimento di prestazioni stabili su dati di test non visti durante la fase di addestramento.
- **Affidabilità delle probabilità**: produzione di stime xG coerenti e confrontabili con modelli di riferimento esterni (es. StatsBomb xG).
- **Efficienza computazionale**: tempi di inferenza ridotti, compatibili con analisi su grandi volumi di eventi di tiro.
- **Stabilità**: il modello mantiene invariato il proprio comportamento dopo l’addestramento iniziale, fino a un eventuale ri-addestramento con nuovi dati.

Environment

- **Parzialmente osservabile**: il modello non dispone di informazioni complete sul contesto di gioco (posizione dei difensori, del portiere, dinamica dell’azione), ma solo delle feature disponibili associate all’evento di tiro.
- **Agente singolo:** il sistema è composto da un unico agente che apprende dai dati e produce una stima della probabilità di goal.
- **Stocastico**: tiri con caratteristiche simili possono avere esiti differenti, rendendo la relazione tra feature ed output non deterministica.
- **Continuo:** l’output del modello è una probabilità continua compresa tra 0 e 1, rappresentativa del valore di Expected Goals.
- **Statico:** le caratteristiche del tiro non cambiano durante il processo di predizione.

- **Episodico:** ogni evento di tiro viene analizzato in modo indipendente dagli altri, senza dipendenze temporali tra le osservazioni.

Actuators

- Produzione in output della **probabilità xG** associata a ciascun evento di tiro.
- Visualizzazione dei risultati sotto forma di **metriche di valutazione** (confusion matrix, ROC curve, Precision–Recall curve).
- Confronto delle stime prodotte dal modello con i valori di **StatsBomb xG** a fini di analisi e validazione.

Sensors

- Acquisizione dei **dati evento relativi ai tiri** da file strutturati in formato JSON.
- Lettura delle informazioni spaziali (coordinate del tiro), temporali e descrittive dell'evento.
- Accesso ai valori di **StatsBomb xG** esclusivamente come riferimento esterno per il confronto delle prestazioni.

Definizione della soluzione

Valutazione delle soluzioni

La soluzione proposta consiste nello sviluppo di una **pipeline di Machine Learning supervisionata** finalizzata alla stima della probabilità che un evento di tiro si concluda con la realizzazione di un goal, secondo il paradigma degli **Expected Goals (xG)**. Il problema viene affrontato come un task di **classificazione binaria**, in cui ciascun evento di tiro rappresenta un'osservazione indipendente e l'output del modello è una probabilità continua compresa tra 0 e 1.

La pipeline prevede una fase iniziale di **acquisizione e preparazione dei dati**, durante la quale vengono estratti dagli eventi di gioco le informazioni rilevanti e gestiti eventuali valori mancanti. Successivamente, viene svolta un'attività di **feature engineering**, con particolare attenzione alla trasformazione delle coordinate spaziali del tiro in variabili più informative, quali la distanza e l'angolo rispetto alla porta. Le restanti caratteristiche descrittive dell'evento vengono opportunamente codificate per poter essere utilizzate dai modelli di apprendimento automatico.

Per la fase di modellazione vengono considerati **diversi algoritmi di classificazione**, tra cui Naive Bayes, Decision Tree e Random Forest, al fine di confrontare approcci con differenti livelli

di complessità e capacità di modellare relazioni non lineari. L'addestramento dei modelli avviene su un sottoinsieme dei dati, mentre una porzione separata viene riservata alla fase di test, così da valutare la capacità di generalizzazione su dati non visti.

La **valutazione delle soluzioni** è condotta mediante metriche standard per problemi di classificazione binaria e dataset sbilanciati, tra cui confusion matrix, ROC curve, ROC-AUC e Precision–Recall curve. Le probabilità xG prodotte dai modelli vengono inoltre confrontate con i valori di **StatsBomb xG**, utilizzati esclusivamente come benchmark esterno, al fine di analizzare la coerenza e l'affidabilità delle stime ottenute. Questo approccio consente di individuare il modello più efficace e di evidenziarne punti di forza e limiti, fornendo una valutazione complessiva delle soluzioni proposte.

Modello di base

Come **modello di base** (*baseline*) per il problema di stima degli Expected Goals (xG) viene adottato un **classificatore Naive Bayes**, utilizzato come punto di riferimento iniziale per la valutazione delle prestazioni. Tale scelta è motivata dalla **semplicità del modello**, dalla sua efficienza computazionale e dalla capacità di fornire rapidamente una prima stima delle probabilità di goal, anche in presenza di dataset sbilanciati.

Il modello Naive Bayes assume l'**indipendenza condizionata** tra le feature date la variabile target e utilizza una rappresentazione probabilistica del problema. Nel contesto del progetto, le variabili categoriche vengono opportunamente codificate tramite *one-hot encoding*, mentre le variabili numeriche vengono utilizzate direttamente in forma continua. Nonostante le forti assunzioni semplificative, il modello rappresenta un **valido punto di partenza** per valutare l'efficacia delle feature selezionate e per confrontare le prestazioni con modelli più complessi.

Il modello di base viene addestrato sugli stessi dati utilizzati dagli altri classificatori e valutato tramite le medesime metriche, consentendo un confronto equo in termini di capacità predittiva e di qualità delle probabilità xG stimate. I risultati ottenuti dal Naive Bayes costituiscono quindi una **baseline di riferimento**, rispetto alla quale è possibile misurare i miglioramenti apportati da modelli più sofisticati, come Decision Tree e Random Forest.

Preprocessing e rappresentazione delle feature

Prima della fase di predizione, i dati evento relativi ai tiri vengono sottoposti a un processo di **preprocessing e trasformazione delle feature**, necessario per rendere le informazioni compatibili con i modelli di Machine Learning adottati. In questa fase vengono gestiti i valori mancanti e selezionate le variabili rilevanti per il problema di stima degli Expected Goals.

Le coordinate spaziali del tiro vengono trasformate in feature derivate più informative, quali la **distanza** e l'**angolo rispetto alla porta**, al fine di catturare in modo più efficace la relazione tra la posizione del tiro e la probabilità di realizzazione. Le variabili categoriche descrittive dell'evento vengono invece codificate tramite **one-hot encoding**, mentre le variabili numeriche vengono mantenute nella loro rappresentazione continua. Questo processo consente di ottenere una rappresentazione vettoriale omogenea degli eventi di tiro, che costituisce l'input per i modelli di classificazione utilizzati nella fase di predizione.

Flusso di predizione

Il flusso di predizione, nel contesto di questo progetto, descrive la sequenza di operazioni che consente di trasformare un evento di tiro (shot) in una stima probabilistica **xG**, cioè la **probabilità di goal** compresa tra 0 e 1 prodotta da un modello addestrato. Il processo è implementato come pipeline: prima si costruisce la rappresentazione finale delle feature e poi si applica il classificatore.

1. **Acquisizione dell'evento di tiro dai file JSON**
   Gli eventi vengono letti da file JSON e filtrati mantenendo solo quelli con type == "Shot". Per ciascun tiro vengono estratti:
   - la variabile target goal (1 se l'outcome è "Goal", altrimenti 0);
   - le coordinate x e y;
   - variabili descrittive: body_part, shot_type, technique, play_pattern;
   - variabile booleana under_pressure;
   - informazioni temporali minute e second.
     Il valore statsbomb_xg viene acquisito **solo** per confronto successivo, non come input del modello.
2. **Pulizia minima e feature engineering (trasformazione spaziale)**
   Prima della predizione, gli eventi con coordinate mancanti vengono rimossi (dropna su x, y). A partire da x e y si calcolano:
   - distance: distanza euclidea del punto di tiro dal centro porta;
   - angle: ampiezza dell'angolo di tiro rispetto ai due pali (calcolata tramite prodotto scalare e arccoseno).
     Dopo il calcolo, le coordinate x e y vengono eliminate e rimangono solo le feature finali (incluse distance e angle).

3. **Rappresentazione delle feature nella pipeline (preprocessing "automatico")**
   La predizione non richiede che l'utente applichi manualmente la codifica: essa è incorporata nella **Pipeline** del modello. In particolare:

   - le feature categoriche vengono trasformate con **One-Hot Encoding** (handle_unknown="ignore", quindi categorie non viste non rompono la predizione);
   - le feature numeriche (es. distance, angle, minute, second, under_pressure) passano in *passthrough* (restano numeriche).
     Nota implementativa: per **Naive Bayes** l'encoding viene forzato in formato **denso**, mentre per Decision Tree e Random Forest può essere **sparse**.

4. **Applicazione del modello addestrato e generazione dell'output xG**
   Una volta ottenuta (internamente alla pipeline) la rappresentazione numerica dell'evento di tiro, il classificatore produce:

   - la classe prevista (predict, Goal/No Goal);
   - soprattutto la probabilità della classe positiva (predict_proba), che viene interpretata come **xG**.

5. **Uso dell'output e confronto con benchmark**
   Le probabilità xG prodotte vengono poi utilizzate per la valutazione (ROC, PR, confusion matrix) e, dove disponibile, confrontate con **StatsBomb xG** esclusivamente come riferimento esterno (senza leakage, perché è escluso dalle feature di training/predizione).

Dataset

Il dataset utilizzato per lo sviluppo del progetto è lo **StatsBomb Open Data**, un insieme di dati pubblici rilasciati da StatsBomb che raccoglie eventi dettagliati relativi a partite di calcio professionistico. Il dataset è ampiamente utilizzato in ambito accademico e applicativo per attività di analisi e modellazione dei dati sportivi.
I dati sono forniti in formato **JSON** e organizzati in più file, ciascuno dei quali descrive differenti livelli informativi, tra cui competizioni, partite ed eventi di gioco. In particolare, per questo progetto vengono utilizzati i file relativi agli **eventi di partita**, contenenti la sequenza cronologica delle azioni avvenute durante ciascun match.
Il dataset in questione è stato recuperato da: https://github.com/statsbomb/open-data

Pre-processing dei dati

Il dataset utilizzato presenta alcune criticità tipiche dei **dati evento sportivi**, legate principalmente alla presenza di **valori mancanti** e alla necessità di trasformare informazioni grezze in feature numeriche utilizzabili dai modelli di Machine Learning. In particolare, non tutti gli eventi di tiro contengono coordinate spaziali valide, rendendo necessaria una fase preliminare di pulizia dei dati.

Per questo motivo, il dataset è stato sottoposto a una fase di **pre-processing**, che ha previsto:

- la **rimozione degli eventi di tiro con coordinate mancanti** (x, y), in quanto non utilizzabili per il calcolo delle feature spaziali;
- la **selezione delle sole variabili rilevanti** per il task di predizione dell'esito del tiro;
- la **trasformazione delle coordinate spaziali** in feature derivate più informative, come la distanza e l'angolo rispetto alla porta;
- la conservazione delle variabili descrittive dell'evento (parte del corpo, tecnica, tipo di azione, pressione avversaria e informazioni temporali).

Questa fase di pulizia e preparazione ha permesso di ottenere un dataset **coerente, strutturato e privo di informazioni incomplete**, migliorando la qualità dell'input fornito ai modelli durante la fase di addestramento e garantendo una maggiore affidabilità delle stime di Expected Goals prodotte.

Mapping delle labels

Nel contesto del problema di stima degli Expected Goals, la variabile target è di natura **binaria** e rappresenta l'esito dell'evento di tiro. Il mapping delle label viene definito in fase di caricamento dei dati, a partire dall'informazione sull'outcome del tiro fornita dal dataset StatsBomb.

In particolare, a ciascun evento di tiro viene assegnata una label secondo la seguente codifica:

- **1 (Goal)** se l'outcome del tiro è pari a *"Goal"*;
- **0 (No Goal)** in tutti gli altri casi.

Questo mapping consente di formulare il problema come un **task di classificazione binaria supervisionata**, in cui il modello apprende a stimare la probabilità che un tiro si concluda con una rete. La scelta di una codifica binaria risulta coerente con l'obiettivo del progetto e con i modelli di classificazione utilizzati, che producono in output una probabilità associata alla classe positiva (Goal), interpretata come valore di Expected Goals (xG

Downsample

Il dataset utilizzato presenta un **naturale sbilanciamento tra le classi**, in quanto solo una piccola percentuale degli eventi di tiro si conclude con la realizzazione di un goal. Tale caratteristica è tipica dei problemi di stima degli Expected Goals e può influenzare le prestazioni dei modelli di classificazione.

Nel presente progetto **non è stata applicata alcuna tecnica di downsampling o oversampling** delle classi. Questa scelta è motivata dall'obiettivo di **preservare la distribuzione reale degli eventi** e di ottenere stime probabilistiche coerenti con la frequenza effettiva dei goal. Alterare artificialmente il bilanciamento del dataset avrebbe potuto compromettere l'interpretabilità e la calibrazione delle probabilità xG prodotte.

Lo sbilanciamento è stato gestito attraverso l'utilizzo di **metriche di valutazione appropriate per dataset sbilanciati**, come ROC-AUC e Precision–Recall, e mediante una suddivisione stratificata dei dati in fase di training e test.

Suddivisione del dataset

Il dataset pre-processato è stato suddiviso in **insieme di addestramento** e **insieme di test** al fine di valutare la capacità di generalizzazione dei modelli su dati non visti. La suddivisione è stata effettuata utilizzando una **stratificazione rispetto alla variabile target**, così da preservare la distribuzione delle classi in entrambi gli insiemi.

In particolare, l'80% dei dati è stato utilizzato per l'addestramento dei modelli, mentre il restante 20% è stato riservato alla fase di test. L'insieme di test è stato mantenuto separato durante l'intero processo di addestramento e selezione dei modelli, garantendo una valutazione corretta e non distorta delle prestazioni.

Addestramento

La fase di addestramento ha l'obiettivo di costruire modelli di Machine Learning in grado di stimare la probabilità che un evento di tiro si concluda con la realizzazione di un goal. A partire dal dataset pre-processato, il problema viene affrontato come un **task di classificazione binaria supervisionata**, in cui la variabile target rappresenta l'esito del tiro (Goal / No Goal).

L'addestramento viene effettuato utilizzando un insieme di training ottenuto tramite una suddivisione stratificata del dataset, così da preservare la distribuzione delle classi. Durante questa fase, i modelli apprendono la relazione tra le feature di input (spaziali, temporali e descrittive dell'evento) e la variabile target. Al termine dell'addestramento, i modelli risultanti

vengono successivamente valutati su un insieme di test separato, al fine di analizzarne la capacità di generalizzazione su dati non visti.

Funzionamento del Trainer

Il processo di addestramento è implementato tramite una funzione dedicata che funge da **trainer** del sistema. Il trainer riceve in input il dataset pre-processato e si occupa di orchestrare tutte le operazioni necessarie alla costruzione dei modelli.

In particolare, il trainer:

- separa le feature dalla variabile target;
- individua automaticamente le variabili numeriche e categoriche;
- definisce le pipeline di preprocessing, includendo la codifica delle variabili categoriche tramite *One-Hot Encoding*;
- suddivide il dataset in training set e test set utilizzando una strategia stratificata;
- addestra i diversi modelli di classificazione sui dati di training;
- restituisce i modelli addestrati insieme ai dati di test necessari per la fase di valutazione.

L'utilizzo di pipeline consente di garantire coerenza tra le fasi di addestramento e predizione, evitando errori dovuti a trasformazioni incoerenti delle feature.

Argomenti del Trainer

Il trainer opera sui seguenti elementi principali:

- **Dataset di input**: insieme di eventi di tiro pre-processati, contenente sia le feature di input sia la variabile target.
- **Feature matrix (X)**: matrice delle variabili indipendenti, ottenuta rimuovendo la variabile target e le informazioni non utilizzabili in fase di addestramento.
- **Variabile target (y)**: etichetta binaria che indica l'esito del tiro (1 = Goal, 0 = No Goal).
- **Pipeline di preprocessing**: componenti responsabili della codifica delle variabili categoriche e del passaggio diretto delle variabili numeriche.
- **Modelli di classificazione**: algoritmi utilizzati per l'apprendimento supervisionato (Naive Bayes, Decision Tree e Random Forest).

- **Parametri di suddivisione**: percentuale di split train/test e strategia di stratificazione utilizzata.

Questi argomenti consentono al trainer di eseguire in modo sistematico e riproducibile l'intero processo di addestramento, producendo modelli confrontabili e valutabili secondo metriche comuni.

Valutazione

Valutazione dei risultati

Dall'esecuzione del training e dalla successiva fase di valutazione sul set di test è possibile analizzare in modo dettagliato le prestazioni dei modelli di Machine Learning adottati per il task di predizione dell'esito di un tiro. Considerata la natura del problema, caratterizzato da un **forte sbilanciamento delle classi** (goal rate ≈ 11%), l'analisi non si è basata esclusivamente sull'accuracy, ma ha incluso metriche più informative per la classe positiva, come precision, recall e ROC-AUC.

I modelli sono stati valutati su un insieme di test separato, ottenuto tramite suddivisione stratificata del dataset, al fine di garantire una stima affidabile della capacità di generalizzazione.

Risultati quantitativi

La tabella seguente riassume le principali metriche ottenute sul set di test per ciascun modello:

Modello	Accuracy	Precision (Goal)	Recall (Goal)	F1-score (Goal)	ROC-AUC
Naive Bayes	0.889	0.505	0.210	0.297	0.755
Decision Tree	0.834	0.278	0.307	0.292	0.604
Random Forest	0.897	0.613	0.210	0.313	0.777

I valori ricavati consentono di trarre alcune considerazioni rilevanti:

- **Naive Bayes** fornisce una baseline solida, con una buona capacità discriminante complessiva (ROC-AUC 0.755), ma risulta fortemente sbilanciato verso la classe

maggioritaria. Il basso recall sulla classe Goal indica che molti tiri che producono una rete non vengono individuati dal modello.

- **Decision Tree** mostra prestazioni inferiori rispetto agli altri modelli. Pur ottenendo un recall leggermente più elevato sulla classe positiva, il modello presenta una precision molto bassa e una ROC-AUC prossima a un classificatore casuale. Questo comportamento è indicativo di una **scarsa capacità di generalizzazione** e di una possibile tendenza all'overfitting.
- **Random Forest** risulta il modello con le migliori prestazioni complessive. Con una ROC-AUC pari a 0.777 e una precision sulla classe Goal superiore al 60%, il modello fornisce stime di Expected Goals più affidabili. Il recall rimane contenuto, ma questo limite è coerente con la natura sbilanciata del problema e con l'assenza di feature contestuali avanzate.

Nel complesso, i risultati mostrano come l'utilizzo di modelli ensemble consenta di migliorare la robustezza e la qualità delle predizioni rispetto a modelli più semplici. La Random Forest riesce infatti a catturare in modo più efficace la relazione tra le caratteristiche del tiro e la probabilità di realizzazione, rappresentando una soluzione adeguata per la stima degli Expected Goals su dati reali.

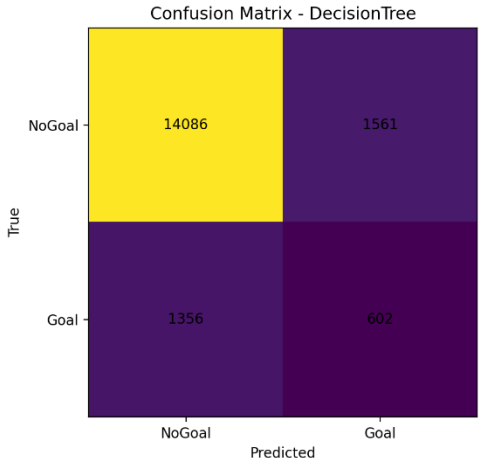
Valutazione del Validation Set

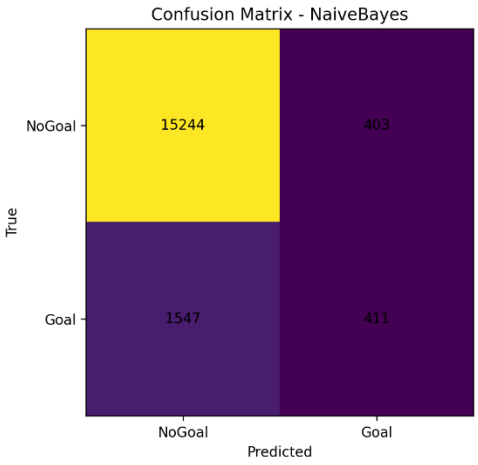
Il validation set, mantenuto separato durante l'intera fase di addestramento, è composto da **17.605 eventi di tiro** e presenta una distribuzione delle classi fortemente sbilanciata, con circa **11% di goal**. I risultati ottenuti confermano la capacità dei modelli di **generalizzare su dati non visti**, producendo stime probabilistiche affidabili.

In particolare, la **Random Forest** ottiene una **ROC-AUC pari a 0.777**, risultando il modello con la migliore capacità discriminante sul set di test. L'analisi delle metriche sulla classe positiva evidenzia una **precision elevata (0.613)** a fronte di un **recall contenuto (0.210)**, comportamento coerente con la natura del problema e con l'obiettivo di produrre stime xG affidabili.

Le Precision–Recall curve e le confusion matrix sul validation set confermano una **riduzione significativa dei falsi positivi** rispetto al Decision Tree, migliorando l'affidabilità complessiva delle predizioni. Nel complesso, la stabilità delle prestazioni tra training e test set indica che il modello apprende **pattern generalizzabili**, risultando adeguato per applicazioni reali di stima degli Expected Goals.

Comparazione grafica



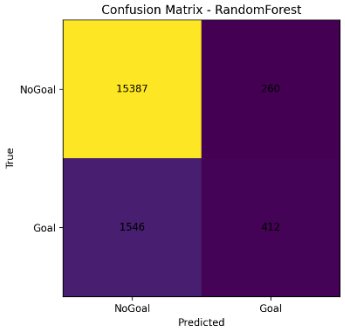


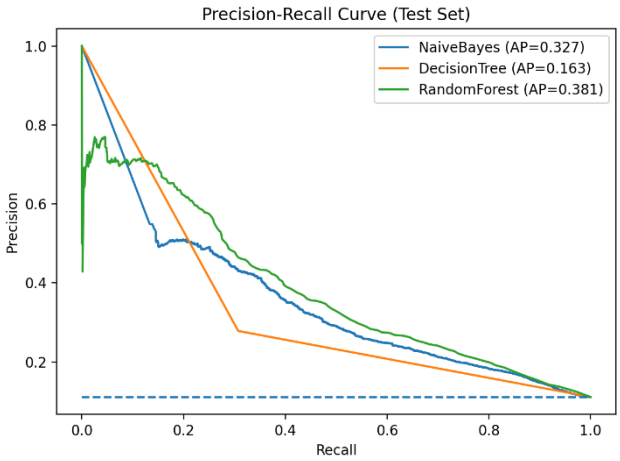
Le confusion matrix permettono di analizzare in dettaglio il comportamento dei modelli nella classificazione degli eventi di tiro, evidenziando la distribuzione di veri positivi, veri negativi, falsi positivi e falsi negativi.

Il Naive Bayes mostra un comportamento fortemente conservativo: il numero di falsi positivi è molto ridotto (403), mentre risulta elevato il numero di falsi negativi (1547). Ciò indica che il modello tende a classificare la maggior parte dei tiri come *No Goal*, riducendo gli errori di sovrastima ma penalizzando il riconoscimento dei goal effettivi.

Il Decision Tree presenta una situazione opposta: il numero di falsi positivi è elevato (1561), segno che il modello tende a classificare come goal un numero eccessivo di tiri. Questo comportamento, unito alla riduzione dei veri negativi, suggerisce una scarsa capacità di generalizzazione e una maggiore sensibilità al rumore nei dati.

La Random Forest fornisce il miglior compromesso complessivo. Pur mantenendo un numero di falsi negativi simile al Naive Bayes, riduce drasticamente i falsi positivi (260), risultando il modello più affidabile quando predice un evento di goal. Questo aspetto è particolarmente rilevante in applicazioni pratiche, dove una sovrastima della pericolosità dei tiri può portare a valutazioni fuorvianti.



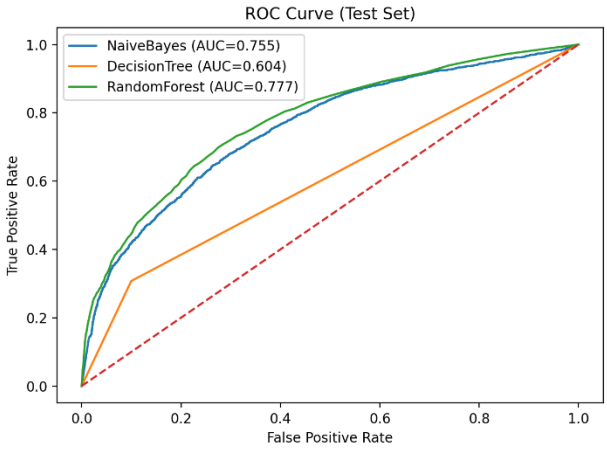


Le Precision–Recall curve sono particolarmente informative in presenza di dataset sbilanciati, come nel caso degli Expected Goals, dove la classe positiva rappresenta solo circa l'11% degli esempi.

Il grafico mostra che:

- il Decision Tree ottiene le prestazioni peggiori, con una *Average Precision* pari a 0.163, poco superiore alla baseline casuale;
- il Naive Bayes migliora sensibilmente, raggiungendo una *Average Precision* di 0.327;
- la Random Forest ottiene il valore più elevato (0.381), mantenendo una precision maggiore a parità di recall.

Questo risultato indica che la Random Forest è il modello più efficace nella gestione della classe minoritaria, fornendo predizioni di goal più affidabili su un ampio intervallo di soglie decisionali.



L'analisi delle ROC curve conferma le osservazioni precedenti. Tutti i modelli superano la baseline casuale, ma con differenze significative:

- **Decision Tree**: AUC = **0.604**, capacità discriminante limitata;
- **Naive Bayes**: AUC = **0.755**, buona separazione tra goal e no-goal;
- **Random Forest**: AUC = **0.777**, miglior performance complessiva.

Il valore più elevato di AUC ottenuto dalla Random Forest indica una maggiore capacità di ordinare correttamente i tiri in base alla loro probabilità di trasformarsi in goal, indipendentemente dalla soglia scelta.

L'analisi congiunta delle confusion matrix e delle curve ROC e Precision–Recall mostra chiaramente come la **Random Forest rappresenti il modello più robusto ed equilibrato** tra quelli analizzati. Pur non massimizzando il recall sulla classe positiva, il modello garantisce una maggiore affidabilità delle predizioni di goal e una migliore capacità discriminante complessiva.

Nel complesso, i risultati grafici confermano quanto osservato a livello quantitativo e supportano la scelta della Random Forest come modello di riferimento per la stima degli Expected Goals nel contesto del progetto.

Comparazione con lo StatsBomb xG

A fini esplorativi, le probabilità di Expected Goals prodotte dai modelli sono state confrontate con la stima proprietaria fornita da **StatsBomb xG**, presente nel dataset ma **non utilizzata in fase di addestramento**, al fine di evitare leakage informativo.
Il confronto sul set di test mostra che la Random Forest sviluppata raggiunge una ROC-AUC pari a 0.777, inferiore rispetto a quella dello StatsBomb xG (0.832), come atteso considerando la maggiore complessità del modello proprietario. Tuttavia, l'elevata **correlazione tra le stime** indica che il modello proposto è in grado di ordinare correttamente i tiri in modo coerente con il benchmark di riferimento, pur utilizzando un insieme di feature e tecniche più semplici.

Possibili sviluppi futuri

Nonostante i risultati ottenuti siano soddisfacenti, il progetto presenta alcuni margini di miglioramento che potrebbero essere esplorati in sviluppi futuri. In particolare, l'ampliamento e l'arricchimento delle informazioni disponibili rappresentano una direzione naturale per incrementare le prestazioni del modello e la sua capacità di generalizzazione.

Un primo possibile sviluppo riguarda l'**estensione del dataset** a un numero maggiore di competizioni e stagioni. L'inclusione di un volume più ampio e diversificato di eventi di tiro

consentirebbe di ridurre la dipendenza del modello da specifici contesti di gioco e di migliorare la robustezza delle stime di Expected Goals.

Un'ulteriore direzione di sviluppo consiste nell'**introduzione di feature contestuali aggiuntive**. Ad esempio, informazioni semplificate sulla posizione del portiere o sul numero di difensori tra il tiratore e la porta potrebbero contribuire a catturare aspetti rilevanti della dinamica di gioco attualmente non modellati.

Dal punto di vista metodologico, sarebbe inoltre possibile esplorare **strategie di validazione più articolate**, come la validazione incrociata su partite o stagioni differenti. Questo approccio permetterebbe di valutare in modo più rigoroso la capacità del modello di generalizzare a contesti temporali diversi, riducendo il rischio di overfitting a specifiche competizioni.

Infine, ulteriori sviluppi potrebbero includere una **calibrazione più approfondita delle probabilità stimate**, al fine di migliorare l'interpretabilità dei valori di xG prodotti e la loro coerenza con le frequenze empiriche osservate.