



## Panoramica

- I. Costatazione empirica
- II. Definizione dei parametri del modello
  - A. Distanza
  - B. Angolo di tiro
- III. Visualizzazione del modello
- IV. Implementazione del modello
  - A. Database
  - B. Server
  - C. Client

## Introduzione

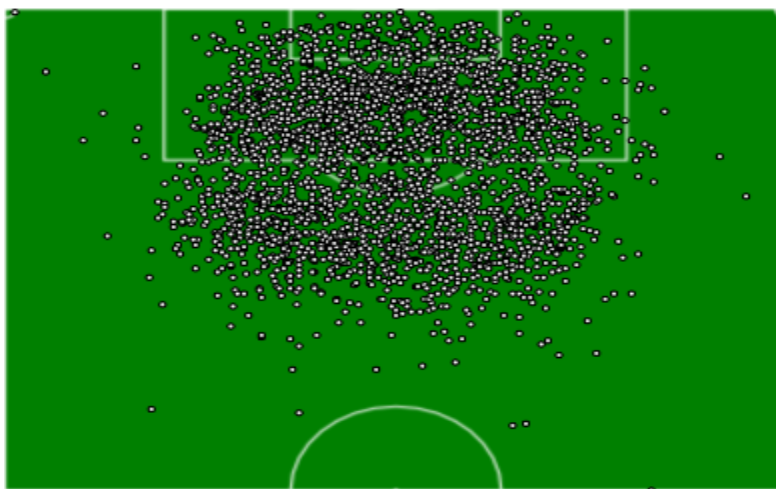
Expected Goals (xG) è un'innovativa statistica nel mondo del calcio che permette di valutare la prestazione offensiva di un individuo o di una squadra analizzando i tiri effettuati nel corso di una partita o di una stagione. Non sempre il risultato di una partita descrive esaurientemente gli eventi accaduti durante il suo svolgimento, fornendo un'immagine dettagliata ed univoca; per questo motivo possedere uno strumento che sia in grado di misurare la qualità delle opportunità create e concesse è importante per poter comprendere più a fondo le dinamiche di gioco.

## Obiettivi

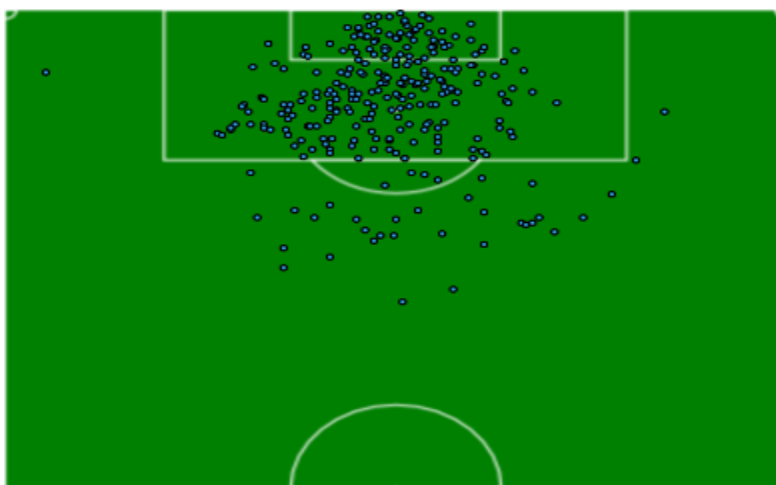
- I. Produrre un modello che sia in grado di descrivere la qualità di un tiro a rete.
- II. Implementare un'applicazione che permetta di mettere in pratica i concetti relativi al modello su un database contenente dati relativi a tutte le partite e a ~3000 tiri appartenenti alla stagione 2017/18 del campionato italiano.

## Constatazione empirica

Dopo aver popolato il database, è stata prodotta un'immagine che rappresentasse tutti i punti di origine, identificati da coordinate cartesiane (con origine (0; 0) nel vertice in alto a destra della foto), dei tiri presi in considerazione.



Successivamente è stata prodotta un'immagine simile, questa volta rappresentante solo i tiri con un esito positivo.

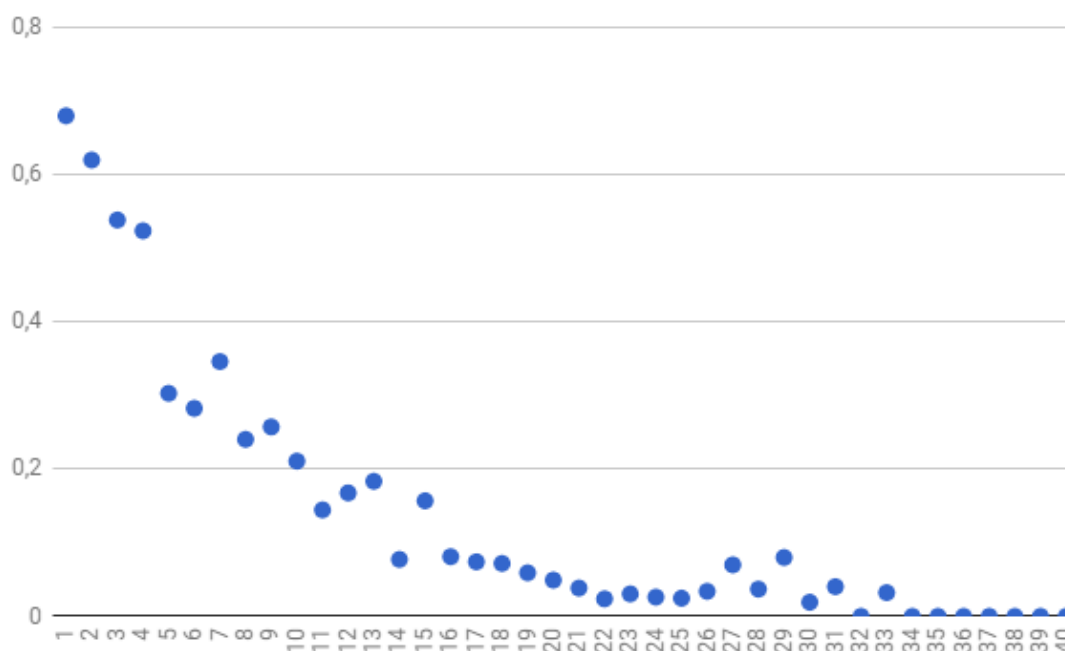


Dal confronto delle due immagini è immediatamente possibile evincere che la posizione del tiro, la sua *centralità* e vicinanza alla porta, sia uno dei principali fattori che influiscono sull'esito dell'azione. Ma per poter determinare la qualità di un'occasione è necessaria un'analisi più profonda del fenomeno: in che modo, e con che peso distanza e *centralità* influenzano l'esito del tiro?

Per comprendere più a fondo il trend si è deciso di isolare i due parametri: in primo luogo verrà studiato l'andamento della distanza dal centro della porta, e in seguito quanto riguarda la *centralità*, o più precisamente, l'angolo di tiro.

## Distanza

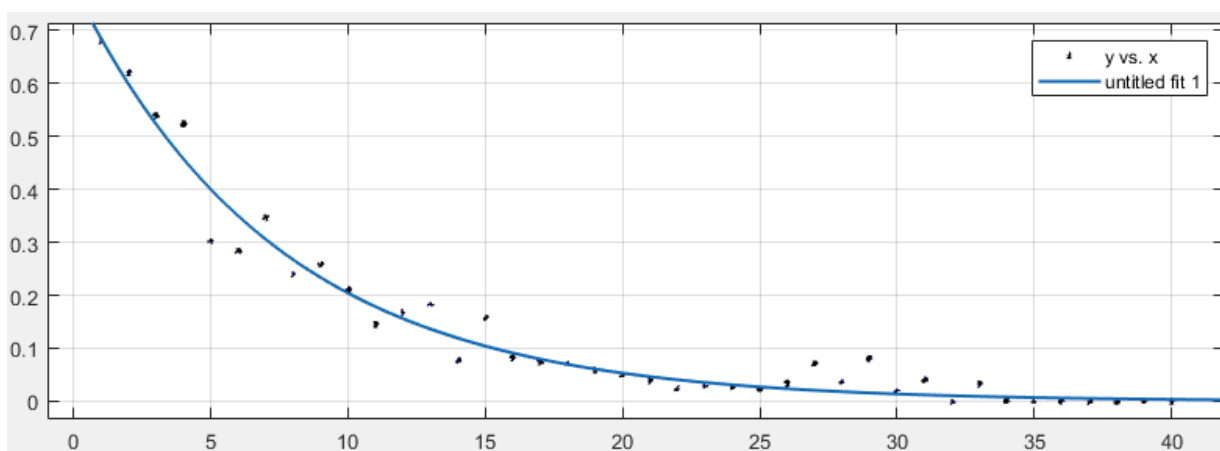
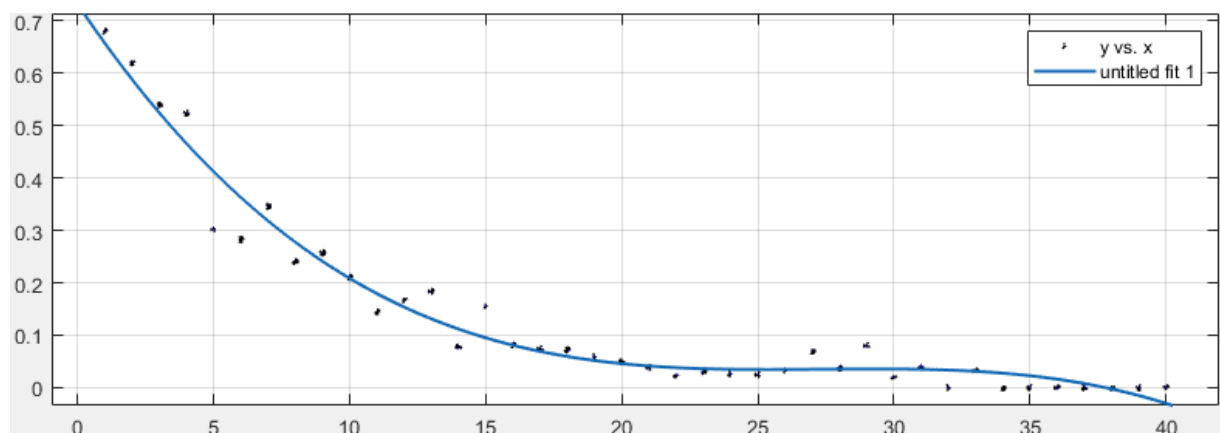
Il campo è stato diviso trasversalmente in porzioni lunghe un metro, e successivamente ogni tiro è stato etichettato ed inserito nella sezione di riferimento. Nei limiti fissati da ogni contenitore, è stato inoltre calcolato il rateo tra goal e tiri effettuati. Ad esempio, un tiro con distanza dal centro della porta pari a 14,33 metri avrà un xG pari a 0,156 perché sono stati effettuati 172 tiri tra 14 e 15 metri risultanti in 27 goal. Una volta calcolato il rapporto per tutti i tiri si può produrre un grafico che rappresenta l'xG empirico per ogni sezione.



Sull'asse delle ascisse è presente la divisione in sezioni in metri, sull'asse delle ordinate il rapporto tra tiri segnati e tiri tentati per ogni sezione.

Per poter implementare in un modello lo studio della distanza, è necessario trovare una funzione che approssimi, e si adatti il meglio possibile ai punti che sono stati trovati. Viene quindi utilizzato uno strumento reso disponibile da *MATLAB* chiamato *Curve Fitting Tool*. Con questo programma è possibile provare diversi tipi di curve e selezionare quella che più si addice alla rappresentazione del fenomeno facendo riferimento ai principi di regressione ed al metodo dei minimi quadrati, il quale prevede di valutare la bontà del modello sulla base della somma dei quadrati di tutti gli errori di stima commessi, ossia proporzione tra la variabilità dei dati e la correttezza del modello utilizzato, introducendo il coefficiente di determinazione  $r^2$ : quando è basso significa che il modello non riesce a descrivere adeguatamente il fenomeno, non si adatta bene ai dati, e quindi anche le previsioni saranno poco affidabili; quando è elevato, prossimo all'1, il modello descrive bene i dati empirici, spiegando gran parte della variabilità osservata della variabile dipendente.

In questo caso sono state utilizzate due funzioni diverse per descrivere i punti osservati, una polinomiale di terzo grado (*figura 1*) e un'esponenziale (*figura 2*).



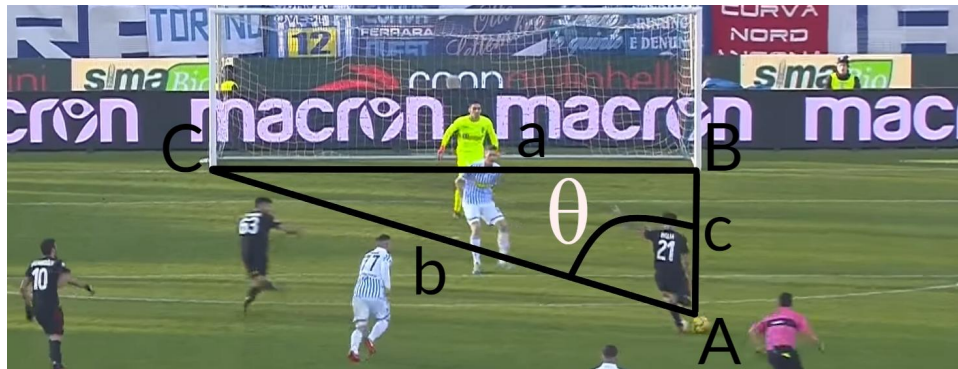
Entrambe le funzioni descrivono con grande precisione il fenomeno, tanto che il *goodness of fit*, indicato da  $r^2$ , è pari a 0.966 per la polinomiale e 0.969 per l'esponenziale. Per semplicità di implementazione, precisione teorica e pratica, è stata preferita la seguente funzione esponenziale individuata dallo strumento:

$$N(d) = 0.7836e^{(-0.1345d)}$$

La precisione teorica è data dal fatto che questo tipo di curva prende il nome di *decadimento esponenziale*, e indica la diminuzione di una quantità proporzionale al valore attuale della funzione. Intuitivamente sembrerebbe addirsi perfettamente a questo caso, e di conseguenza alle nostre esigenze.

### Angolo di tiro

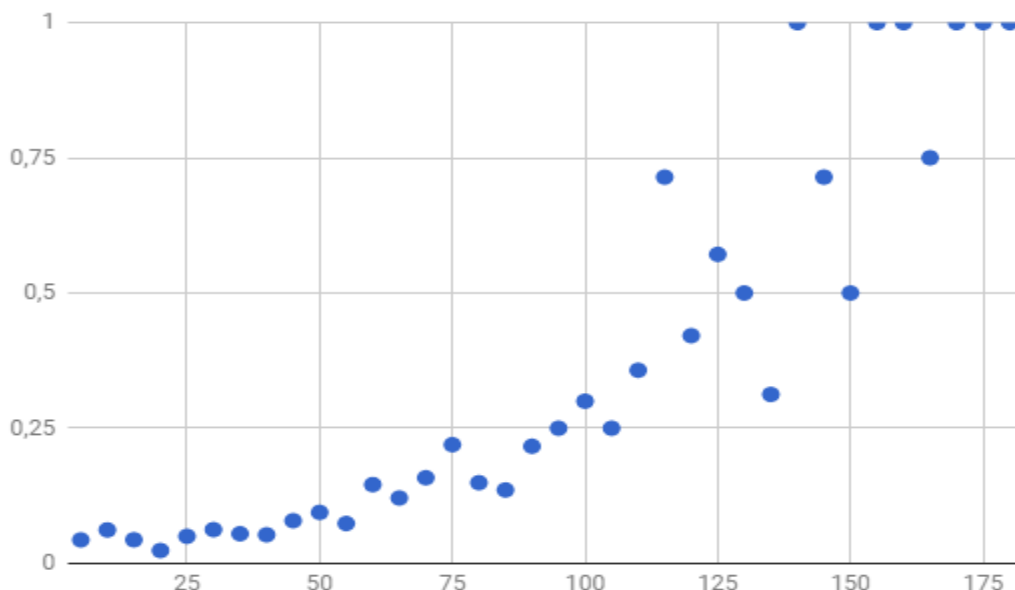
Per secondo parametro utilizzato dal modello viene calcolato l'angolo che si forma quando due rette che passano per i due pali della porta si intersecano sul punto di tiro.



Conoscendo la posizione dei tre vertici del triangolo si può calcolare la lunghezza dei lati e successivamente si può trovare il valore di un angolo opposto ad uno dei segmenti applicando il teorema dei coseni. Di conseguenza:

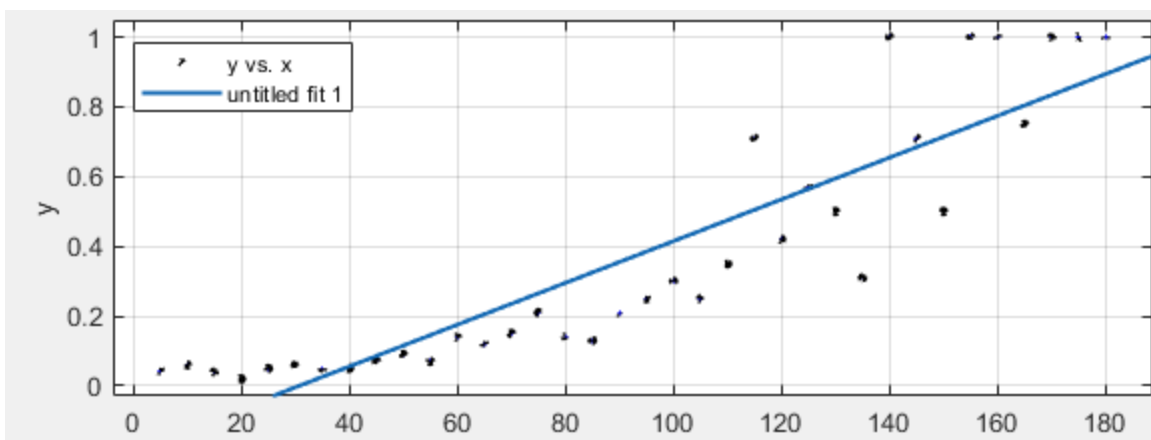
$$\theta = \arccos\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$

Analogamente a quanto fatto in precedenza, si può calcolare l'angolo per tutti i tiri presenti nel database, dividerli ad intervalli di 5° e determinare il rapporto tra tiri segnati ed effettuati all'interno dei limiti fissati da ogni sezione.



L'angolo di tiro trovato potrà assumere tutti i valori compresi tra 0° (il tiro sarà stato effettuato sulla linea di fondo) e 180° (il tiro sarà stato effettuato sulla linea di porta).

La funzione di regressione lineare che permette di avvalersi di un metodo di stima del valore atteso viene nuovamente calcolata con l'ausilio del *Curve Fitting Tool* di *MATLAB*.



Sulla base dei dati forniti, la funzione che meglio approssima, con  $r^2$  pari a 0.8815, è:

$$f(\theta) = 0.005989\theta - 0.1839$$

In questo caso abbiamo una relazione lineare tra  $xG$  e l'angolo di tiro. Come ci si aspetterebbe, all'aumentare della dimensione dello specchio della porta, maggiore è la probabilità di segnare.

La capacità descrittiva del modello, sulla base delle osservazioni eseguite in precedenza, è dunque buona, almeno relativamente ai dati osservati. Si può allora ritenere utilizzabile a scopi previsivi? Il passaggio dalla descrizione dei dati osservati alla previsione di nuovi dati e comportamenti è un problema di generalizzabilità dei risultati di un modello. La possibilità di operare generalizzazioni si basa su assunzioni *ragionevoli*:

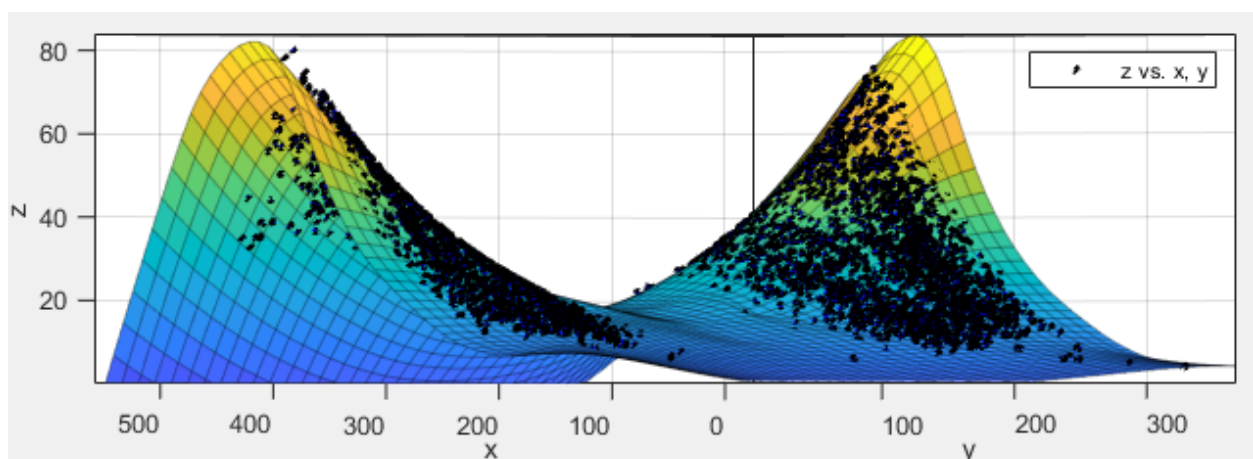
- I. **ordine** della natura: se le cose sono andate in un certo modo finora non c'è ragione per cui non debbano funzionare così anche in futuro (e in passato);
- II. **determinismo** degli eventi: nessun evento avviene per caso, ma è determinato da precedenti eventi; ovvero, se le cose funzionano in un certo modo, la ragione c'è sempre, e quindi continueranno.

Questi principi sono plausibili, ma non assicurati: nessuno può garantire che anche le prossime osservazioni si comporteranno allo stesso modo.

---

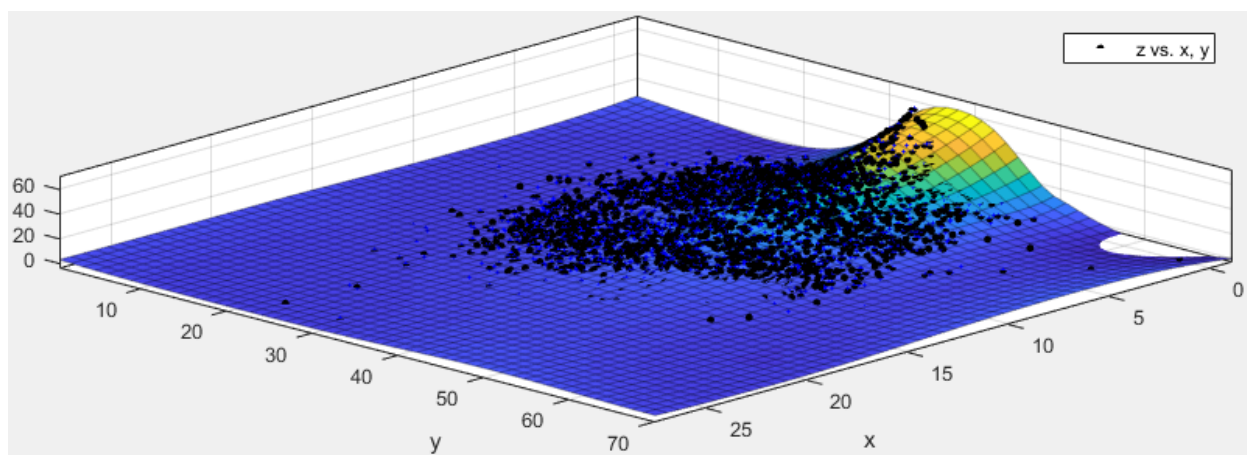
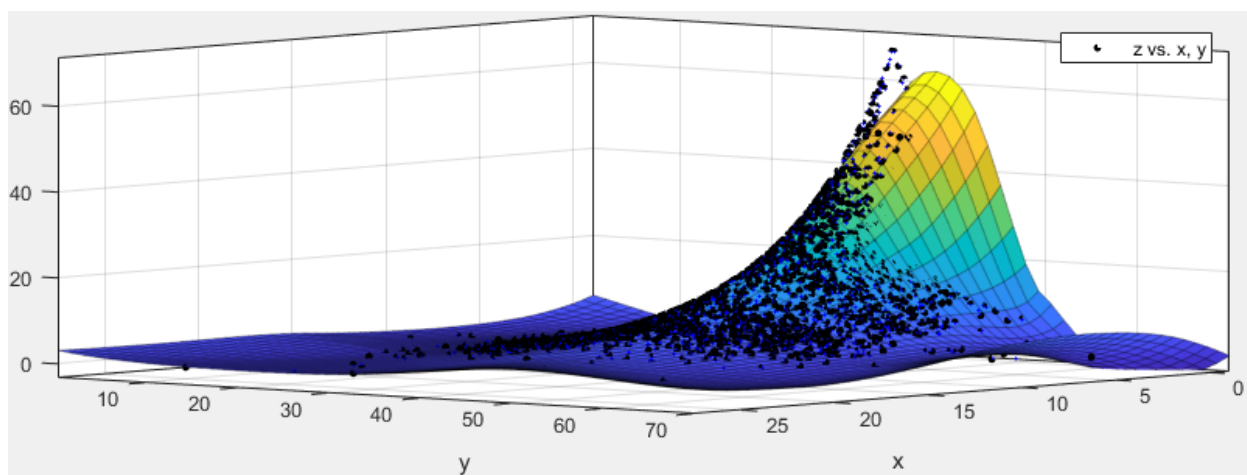
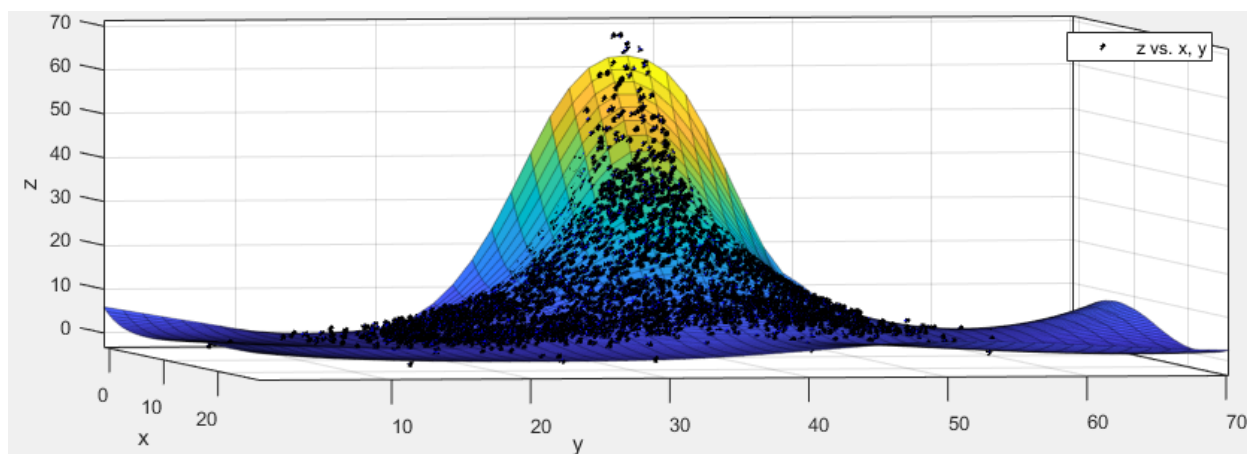
## Visualizzazione del modello

Una volta definito il modello, possiamo applicarlo per ogni posizione sul campo. I due punti di massimo che ricaviamo dal modello sono, come si potrebbe intuire, nei pressi delle due porte.

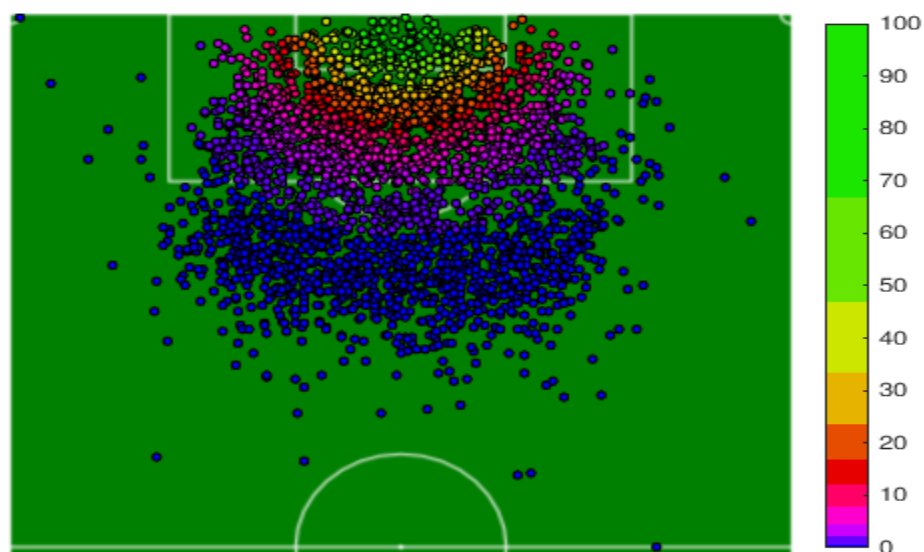




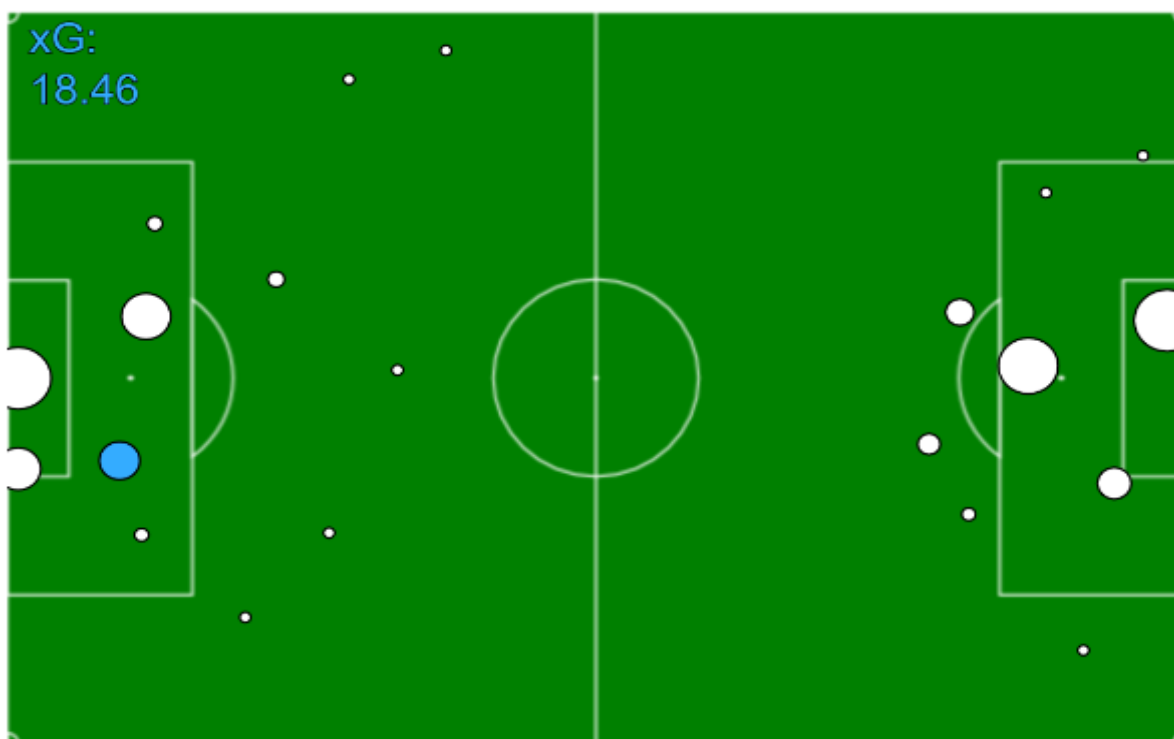
Si possono anche creare viste più dettagliate che si limitano ad una metà di campo, e di conseguenza ad una singola porta.



Una volta sviluppato il modello si può produrre una vista due dimensionale analoga a quella precedente che rappresenti la probabilità che ogni tiro finisca in rete.



Con il canvas interattivo si possono verificare dinamicamente nuove posizioni di tiro, ricevendo in tempo reale la probabilità calcolata in quel punto.

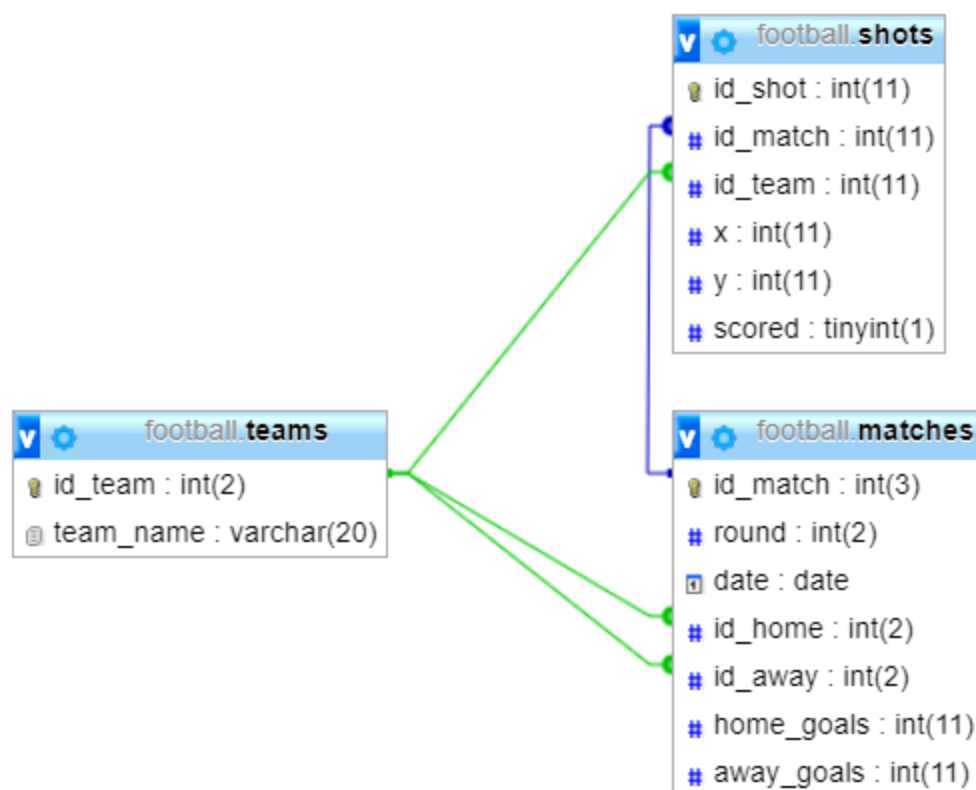


## Implementazione del modello

L'obiettivo dell'applicazione è accedere al database e produrre delle mappe per partita che rappresentino l'xG di ogni tiro effettuato. L'architettura dell'applicazione è basata sul sistema Client/Server: il client si occuperà solamente della parte grafica, il server si interfacerà con il database e su richiesta del client invierà i dati selezionati.

### Database

Il database prevede tre tabelle: teams, matches e shots.



## Server

Il cuore dell'applicazione è scritto in *Node.js*, un runtime asincrono e event driven costruito sul motore *JavaScript V8* di Chrome, che rende il programma estremamente efficiente ed in grado di gestire migliaia di connessioni concorrenti con minimo overhead per singolo processo. La comunicazione tra Client e Server è attuata attraverso un websocket. Per poter implementare tutte le funzioni del server vengono utilizzate anche librerie e moduli ausiliari come *Express.js*, *mysql*, *Socket.io*, *http-server* e *body parser*; ognuno di essi ha una mansione diversa, ma vivono tutti in simbiosi e si appoggiano interamente su *Node.js*.

Node.js	Framework che consente di utilizzare JavaScript sul lato server.
Express.js	Middleware per gestire richieste HTTP.
mysql	Un driver Node.js per mysql.
Socket.io	Comunicazione real-time bidirezionale event-based. Sia per Server che per Client.
http-server	Trasferimento dati con il protocollo HTTP.
body parser	Gestione richieste HTTP Post.

La gestione e l'interazione tra framework e librerie è resa possibile attraverso il *Node Packet Manager (NPM)*. I packets vengono gestiti interamente, automatizzando il processo: tutte le dependency del progetto vengono specificate nel file *package.json*.

## Client

Per ottenere i dati necessari all'elaborazione grafica il client ha a disposizione tre diverse richieste da fare al server (parametri tra parentesi):

- I. **list** [team\_name] → match list for the selected team.
- II. **info** [id\_match] → information on the selected match.
- III. **map** [id\_match] → xG map of the match.
- IV. **canvas** → empty interactive canvas.

Dopo aver formulato la richiesta, una funzione *jQuery* chiama la funzione *emit* del socket che invia al server il corpo del messaggio. A questo punto si mette in ascolto della risposta del server, che una volta arrivata richiama uno dei handler predisposti tra *messageClient*, *messageServer*, *messageServerInfo* e *messageServerMap*. Quest'ultimo è l'handler che gestisce la creazione della mappa. Viene creato uno sketch e viene utilizzata la libreria *p5.js*; i dati passati dal Server, in formato *JSON*, vengono analizzati e viene prodotta una mappa cerchi nella posizione di origine del tiro di dimensione proporzionale al xG nel punto.

jQuery	Libreria lightweight di JavaScript.
p5.js	Gestione esperienze grafiche ed interattive. Esclusivamente lato Client.
Socket.io — Client	Estensione Client della libreria Socket.io.

L'implementazione prevede la possibilità di eseguire le quattro diverse richieste su una riga di comando. Il Server gestisce la richiesta e invia i dati selezionati dall'utente, il Client si occupa di visualizzarli graficamente.

