



UNIVERSITÀ DEGLI STUDI DI
MILANO-BICOCCA

F1801Q127

MODELLI PROBABILISTICI PER LE DECISIONI

Predicting soccer results

Studenti:

Basso Matteo

Ferri Marco

Matricole:

807628

807130

Giugno 2019

Indice

1	Abstract	3
2	Introduzione	4
2.1	Dominio di riferimento	4
2.2	Obiettivi dell'elaborato	4
3	Descrizione del dataset originale	5
3.1	Formato	5
3.2	Giocatori	5
3.3	Squadre	7
3.4	Partite	8
3.5	Qualche numero	9
4	Creazione del dataset per le predizioni	10
4.1	Ipotesi e assunzioni	10
4.2	Scelta del formato CSV	11
4.3	Elaborazione dei dati	11
4.3.1	Reperimento delle informazioni sui giocatori di una partita . .	11
4.3.2	Ruolo di un giocatore	13
4.3.3	Potenzialità di una squadra in partita	14
4.3.4	Correlazione fra ruolo e <code>overall_rating</code>	15
4.3.5	Calcolo e discretizzazione dei punteggi generali per la squadra	19
5	Rete Bayesiana	20
5.1	Strumenti e librerie	20
5.2	Assunzioni	20
5.3	Generazione della rete	20
5.3.1	Autogenerazione della struttura	20
5.3.2	Vincoli	20
5.3.3	Esperimenti	20
6	Performance della rete	21
6.1	Tempi	21
6.2	Cross Fold Validation	21
6.2.1	Accuracy	21
6.2.2	Precision e Recall	21
6.2.3	Loss	21
7	Web Demo	22
7.1	Architettura	22

7.2	Interfaccia grafica	22
7.3	Predizione risultato	23
8	Conclusioni	24

1 Abstract

Il calcio è da anni lo sport più giocato e diffuso in svariati paesi del mondo. Moltissime persone seguono settimanalmente, con grande attenzione e fervore, ogni singola partita nella speranza di veder vincere la propria squadra del cuore.

In questo contesto si sono inevitabilmente sviluppati un gran numero di diversi business di enorme valore economico, fra cui ad esempio quello delle scommesse. Saper prevedere l'esito di una partita, sia in termini di vittoria che di goal fatti, rappresenta per moltissimi un argomento di particolare interesse. Durante questo studio, ci si è posti l'obiettivo di sviluppare un modello per la predizione del risultato delle partite di calcio in maniera automatica. Tale predizione deve essere basata sulla formazione delle due squadre considerate per ciascuna partita, nonché sulle caratteristiche dei giocatori coinvolti.

Il dominio di riferimento è stato modellato attraverso l'utilizzo di una Rete Bayesiana attraverso la quale fare le dovute inferenze e conseguentemente predire il risultato di una partita fra due squadre di calcio. Il medesimo approccio potrebbe essere riutilizzato in altri sport o giochi che prevedano l'interazione fra più squadre di cui si conoscono le statistiche di ciascun giocatore.

2 Introduzione

Viene qui presentata una visione generale del progetto, ovvero il dominio di riferimento e gli obiettivi che esso si pone, le scelte di design per la creazione del dataset ed eventuali ipotesi o assunzioni fatte durante lo sviluppo dell'elaborato.

2.1 Dominio di riferimento

Il calcio rappresenta, soprattutto negli ultimi anni, lo sport maggiormente diffuso in vari paesi del mondo. Molte persone seguono con grande attenzione tutte le partite cercando di capire preventivamente il vincitore, per piacere personale oppure per giocare nel mercato delle scommesse.

A seguito di questo fenomeno sono stati creati svariati portali che permettessero alle persone di entrare sempre di più in questo mondo. Un aspetto fondamentale per ogni piattaforma di scommesse è senz'altro la possibilità di predire il vincitore di un match attraverso la simulazione di partite di calcio con particolari formazioni.

2.2 Obiettivi dell'elaborato

Questo elaborato sarà suddiviso per argomenti, secondo un approccio di indagine incrementale e coerente con quanto praticamente svolto.

Verranno innanzitutto descritte le modalità di acquisizione dei dati dalla sorgente, a cui sarà associata anche una breve descrizione di quanto a disposizione. Quindi, si esploreranno i criteri e le assunzioni che hanno portato alla creazione del dataset su cui sviluppare il modello di predizione. Con questo, si intende dire che verranno inizialmente presentate alcune analisi qualitative sui dati e successivamente descritti i procedimenti svolti per l'integrazione e l'estrazione delle informazioni più rilevanti.

In seguito all'esportazione del dataset saranno descritte le modalità di suddivisione del dataset su cui fare inferenze, in particolare attraverso l'utilizzo delle Reti Bayesiane. Diversi modelli di generazione della rete verranno presentati e per ognuno saranno analizzati i risultati ed evidenziate le differenze. L'obiettivo del modello è quello di predire il team vincitore di un determinato match, utilizzando le informazioni dei giocatori presenti in campo.

Verrà infine presentata un'interfaccia utente per la simulazione di una partita di calcio, utile a mostrare l'effettivo impiego della rete in un prodotto di natura commerciale e pensato per l'utilizzo sul Web.

3 Descrizione del dataset originale

Di seguito viene presentata la descrizione del dataset originale, prelevato da Kaggle [2] e a sua volta creato attraverso l'unione di dati provenienti da diverse fonti, fra cui la più importante contenente le statistiche del famoso videogioco di calcio EA Sports FIFA. [1]. Il contenuto del dataset è rappresentato dalle principali squadre e giocatori di calcio in Europa, con le partite svolte in un periodo temporale di 6 anni.

3.1 Formato

Il dataset consiste di un database SQLite [5], ovvero un file con omonima estensione contenente un database relazionale di rapido utilizzo, in grado di essere condiviso e utilizzato facilmente su un qualsiasi dispositivo.

3.2 Giocatori

I giocatori sono espressi all'interno di due tabelle differenti nello schema del database considerato. La prima relazione denominata *Player* contiene informazioni di carattere generale ed è mostrata in tabella 1 a seguire.

Tabella 1: Player

Campo	Descrizione
id	Id del record
player_api_id	Identificativo univoco del giocatore
player_name	Nome completo del giocatore
player_fifa_api_id	Id utile per reperire le informazioni da fifa
birthday	Data di nascita
height	Altezza in centimetri
weight	Peso in libbre

La seconda tabella, denominata *PlayerAttributes* contiene invece i punteggi associati alle abilità personali di ciascun giocatore. Essa contiene inoltre un campo che indica a quale data tali punteggi si riferiscono, cioè quando sono state effettuate le rilevazioni. È possibile, infatti, che i dati associati ad un giocatore cambino nel tempo; per tale motivo, nel momento in cui si rivelerà necessario recuperare i dati dei giocatori facenti partecipanti ad una partita, sarà necessario selezionare i giusti record dalla tabella *Player* proprio in base alla data della partita.

Nella tabella 2 vengono meglio elencati i campi che descrivono un giocatore. La maggior parte di essi sono espressi su una scala intera da 1 a 100.

Tabella 2: PlayerAttributes

Campo	Descrizione
id	Id del record
player_api_id	Identificativo univoco del giocatore
player_ffa_api_id	Id utile per reperire le informazioni da FIFA
date	Data di riferimento del record
overall_rating	Valutazione generale
potential	Potenziale
preferred_foot	Piede preferito
attacking_work_rate	Punteggio di attacco
defensive_work_rate	Punteggio di difesa
crossing	Cross
finishing	Capacità di finalizzare (segnare un goal)
heading_accuracy	Tiro di testa
short_passing	Passaggio corto
volleys	Tiro al volo
dribbling	Dribbling
curve	Tiro con effetto curvilineo
free_kick_accuracy	Accuratezza dei calci piazzati
long_passing	Passaggio lungo
ball_control	Controllo di palla
acceleration	Accelerazione
sprint_speed	Velocità in scatto
agility	Agilità
reactions	Reattività
balance	Equilibrio
shot_power	Potenza di tiro
jumping	Salto
stamina	Resistenza
strength	Forza
long_shots	Tiri lunghi
aggression	Aggressività
interceptions	Intercettazione
positioning	Posizionamento
vision	Vista
penalties	Falli commessi
marking	Capacità di marcare
standing_tackle	Contrasto in piedi
sliding_tackle	Scivolata
gk_diving	Tuffo (portiere)
gk_handling	Presa (portiere)
gk_kicking	Rinvio (portiere)
gk_positioning	Posizionamento (portiere)
gk_reflexes	Riflessi (portiere)

3.3 Squadre

Le squadre sono descritte analogamente ai giocatori tramite 2 tabelle, *Team* 4 e *TeamAttributes* 3 riportate qui di seguito.

Tabella 3: TeamAttributes

Campo	Descrizione
id	Id del record
team_api_id	Identificativo univoco della squadra
team_fifa_api_id	Id utile per reperire le informazioni da FIFA
team_name	Nome completo della squadra
date	Data di riferimento del record
buildUpPlaySpeed	Velocità di gioco
buildUpPlaySpeedClass	Discretizzazione velocità di gioco
buildUpPlayDribbling	Dribbling
buildUpPlayDribblingClass	Discretizzazione dribbling
buildUpPlayPassing	Passaggio
buildUpPlayPassingClass	Discretizzazione passaggio
buildUpPlayPositioningClass	Posizionamento della squadra, definito o libero
chanceCreationPassing	Creazione chance di passaggio
chanceCreationPassingClass	Discretizzazione creazione chance di passaggio
chanceCreationCrossing	Creazione chance di cross
chanceCreationCrossingClass	Discretizzazione creazione chance di cross
chanceCreationShooting	Creazione chance di tiro in porta
chanceCreationShootingClass	Discretizzazione creazione chance di tiro in porta
chanceCreationPositioningClass	Creazione chance di posizionamento
defencePressure	Pressing della difesa
defencePressureClass	Discretizzazione pressing della difesa
defenceAggression	Aggressività della difesa
defenceAggressionClass	Discretizzazione aggressività della difesa
defenceTeamWidth	Copertura della difesa
defenceTeamWidthClass	Discretizzazione copertura della difesa
defenceDefenderLineClass	Tipologia di difesa

Come osservabile, a differenza di quanto avvenga con i giocatori, per le squadre molti attributi appaiono già discretizzati attraverso una suddivisione in classi che descrive a parole la relativa scala da 1 a 100 di ciascun campo.

Tabella 4: Team

Campo	Descrizione
id	Id del record
team_api_id	Identificativo univoco della squadra
team_name	Nome completo della squadra
team_fifa_api_id	Id utile per reperire le informazioni da fifa
team_short_name	Nome abbreviato della squadra
team_long_name	Nome completo della squadra

3.4 Partite

La tabella *Match* contiene tutte le informazioni relative le squadre che hanno giocato una partita l'una contro l'altra, i giocatori coinvolti e il loro posizionamento in campo. I giocatori vengono numerati da 1 a 11 e il portiere viene sempre collocato in posizione [1;1]. Sono presenti inoltre dei valori il cui scopo è quello di indicare alcune specifiche tecniche sull'andamento della partita. Per brevità e mancanza di documentazione verranno semplicemente omessi, poiché poco rilevanti ai fini dello studio. Nella tabella 5 è presente la descrizione dei vari campi.

Tabella 5: Match

Campo	Descrizione
id	Id del record
country_id	Id della nazione di riferimento
league_id	Id della lega di riferimento
season	Stagione calcistica
date	Data della partita
match_api_id	Id univoco della partita
home_team_api_id	Id del team in casa
away_team_api_id	Id del team in trasferta
home_team_goal	Goal del team in casa
away_team_goal	Goal del team in trasferta
home_player_i	Id dell'i-esimo giocatore in casa
away_player_i	Id dell'i-esimo giocatore in trasferta
home_player_Xi	Posizione i-esimo gioc. (casa) sul lato corto del campo
home_player_Yi	Posizione i-esimo gioc. (casa) sul lato lungo del campo
away_player_Xi	Posizione i-esimo gioc. (trasferta) sul lato corto del campo
away_player_Yi	Posizione i-esimo gioc. (trasferta) sul lato lungo del campo

3.5 Qualche numero

Di seguito sono riportati i numeri costituenti il dataset d'origine:

- 25979 partite svoltesi fra il 2010 e il 2016;
- 11060 giocatori in totale;
- 183978 rilevazioni sulle abilità dei giocatori
ciò significa approssimativamente 16 rilevazioni per ciascun giocatore;
- 299 squadre di 11 diversi campionati Europei
- 1458 rilevazioni sulle caratteristiche delle squadre
ciò significa approssimativamente 5 per ciascun squadra.

4 Creazione del dataset per le predizioni

Affinché i dati a disposizione possano essere utilizzati per la predizione del risultato di una partita su base statistica, è necessario che essi vengano prima elaborati secondo un insieme di regole su cui costruire i principi per portare a termine una specifica predizione. L'elaborazione stessa è perciò determinata sia attraverso delle assunzioni volte a identificare gli elementi più rilevanti nei dati, sia tramite l'effettiva esecuzione delle operazioni necessarie a trasformare il dataset originale in quello utilizzato per modellare la Rete Bayesiana.

4.1 Ipotesi e assunzioni

Al fine di sviluppare il progetto e procedere con la stesura del seguente elaborato, sono state fatte delle assunzioni che fungono da base portante.

Si assume che il risultato di una partita, a sua volta determinato dal numero di goal segnati da una squadra e dall'altra, sia influenzato dalla bravura dei giocatori che partecipano alla partita stessa. Nel gioco del calcio, come in molti altri, esiste una netta suddivisione fra i compiti assegnati a ciascun giocatore che vanno a determinare il ruolo di quest'ultimo: **portiere**, **difensore**, **centrocampista** e **attaccante**. È chiaro che ogni ruolo riveste la sua particolare importanza e influisce in maniera diversa sull'esito della partita. Per questo progetto, si è pensato di costruire un modello basato proprio sulle potenzialità delle due squadre sfidanti nei diversi ruoli assunti dai giocatori. Poiché tale informazione è mancante all'interno del dataset originale, il primo step cui si è fatto fronte è stata la determinazione dei ruoli di ciascun giocatore in campo. Tale obiettivo è stato raggiunto attraverso l'ipotesi, più che ragionevole, secondo la quale la posizione di una persona sul campo da gioco determina anche il suo ruolo all'interno della squadra e della partita.

In secondo luogo, si è scelto inoltre di supporre che le caratteristiche del team siano determinate solo dai giocatori che ne fanno parte. Non avendo a disposizione informazioni circa l'appartenenza di un giocatore ad un determinato team nel tempo, non è possibile ottenere una suddivisione dei giocatori in squadre ben formate. Nonostante quest'ultime si potrebbero ricavare direttamente dalle partite svolte, ciò comporterebbe la creazione di dati incoerenti col mondo reale poiché i giocatori di calcio sono soggetti ad un mercato di compra-vendita su base annuale.

Per questo motivo, ai fini della predizione si è scelto di fatto di ignorare l'identità delle due squadre coinvolte in un match, ma semplicemente **limitarsi a considerare i 22 giocatori che hanno preso parte alla partita**, pesando adeguatamente le abilità che li contraddistinguono nel proprio ruolo.

4.2 Scelta del formato CSV

Per la creazione del dataset finale, utile per la modellazione della Rete Bayesiana, è stato scelto di trasformare il dataset in formato CSV, in quanto più semplice e pratico da utilizzare rispetto al database relazionale inizialmente fornito. Come si accennava, affinché possa essere creato il modello e su questo condotto delle previsioni, è necessario che i dati vengano elaborati e aggregati in un'unica tabella. Mentre SQLite consente agevolmente di lavorare sulle tabelle per effettuare *join*, selezioni e proiezioni sul modello relazionale, il DBMS non è particolarmente comodo per conservare i dati una volta terminata l'aggregazione. A tale scopo si è preferito quindi esportare successivamente il risultato ottenuto all'interno di un file CSV, decisamente più portatile e immediatamente leggibile rispetto ad un database.

4.3 Elaborazione dei dati

Avendo come principale fonte d'informazione i dati relativi alle partite calcistiche svoltesi in Europa fra il 2010 e il 2016, e sapendo per ciascuna di essa i giocatori partecipanti (con relative posizioni in campo) e l'esito del match, è chiaro che la priorità durante la fase di elaborazione dei dati è stata quella di unire tutte le informazioni a disposizione in un'unica tabella. Ciò è stato possibile attraverso il *join* delle tabelle *Match* e *PlayerAttributes* utilizzando le chiavi che le mettono in relazione. Successivamente, i dati sono stati aggregati per computare la competitività di una certa squadra in una specifica partita.

4.3.1 Reperimento delle informazioni sui giocatori di una partita

Per ogni partita si hanno a disposizione 22 chiavi esterne, relative ai giocatori della squadra di casa e di quella ospite; con queste, è stato possibile reperire per ciascun giocatore le proprie abilità in quel determinato momento temporale. Tale informazione è reperibile nella tabella *PlayerAttributes*, nella quale per la medesima persona sono contenute più rilevazioni, con date differenti. Chiaramente, la rilevazione corretta delle abilità di un certo giocatore è rappresentata da quella più recente, purché questa sia stata effettuata precedentemente alla data del match in esame.

Per fare un esempio pratico, se in una partita del 3/07/2016 ha partecipato il giocatore con id 1234 e supponendo che in *PlayerAttributes* si trovino, per tale giocatore, i dati che è possibile osservare nella tabella 6, è chiaro che il dato corretto da associare a tale partita per il giocatore 1234 è quello datato 12/12/2015. Questo rappresenta l'informazione più aggiornata alla data della partita.

Tabella 6: PlayerAttributes WHERE Id = 1234

Id	Name	Date	Overall	Finishing	Marking	...
1234	Gabriel Marquez	12/04/2018	78	72	85	...
1234	Gabriel Marquez	12/12/2015	80	75	82	...
1234	Gabriel Marquez	05/10/2013	75	70	80	...

L'operazione di *join* da eseguire per ciascuna partita ha dovuto coinvolgere ben 22 giocatori, di cui per ognuno è stato necessario scegliere la riga corretta di *PlayerAttributes*, adeguatamente filtrata per data. È da ricordare, come accennato nel capitolo 3.5, che per ciascun giocatore si registrano mediamente 16 rilevazioni in date diverse all'interno della tabella *PlayerAttributes*.

Il primo tentativo di effettuare il *join* è quindi fallito.

Tentando di reperire tutte le informazioni necessarie attraverso l'esecuzione di un'unica query SQL ci si è accorti che, in questo specifico caso d'uso, 22 *join* fossero decisamente troppi; questo poiché per ciascuna colonna di *Match* da *joinare* (ognuna delle quali rappresenta la chiave esterna di un giocatore) era necessario computare il prodotto cartesiano fra ciascun match e mediamente 16 record di *PlayerAttributes*. La query, il cui codice è semplificato qui sotto, produceva quindi una crescita di righe esponenziale fra un *join* e il successivo, che rappresenta tutt'oggi un requisito computazionalmente impossibile da gestire per tabelle di grosse dimensioni.

```
SELECT *
FROM Match as m
JOIN Player_Attributes as h1
on h1.player_api_id = match.home_player_1 AND h1.date < m.date
JOIN Player_Attributes as h2
on h2.player_api_id = match.home_player_2 AND h2.date < m.date
...
GROUP BY m.match_api_id
HAVING h1.date = MAX(h1.date)
HAVING h2.date = MAX(h2.date)
...
```

Per ovviare il problema, si è quindi pensato di svolgere un'operazione di *join* alla volta, salvando di volta in volta il risultato in tabelle intermedie per i 22 giocatori. Ciò è stato possibile attraverso lo sviluppo di un programma Javascript per la creazione incrementale delle query e l'interfacciamento diretto con il database SQLite. L'approccio si è rivelato vincente, facendo abbassare in maniera drasticamente i tempi di esecuzione da diverse ore a poche decine di secondi.

4.3.2 Ruolo di un giocatore

La tabella risultante dallo step precedente, contenente per ogni partita i dati di ciascun giocatore coinvolto, ne è risultata di circa 17mila record dai 25mila iniziali, a causa di alcuni dati mancanti proprio relativamente ai giocatori partecipanti. A questo punto si è quindi scelto di elaborare proprio le caratteristiche dei giocatori per costruire il dataset da utilizzare per le predizioni.

Come accennato nella sezione 4.1, si è assunto che la potenzialità di vittoria di una certa squadra sia da ricondurre alle caratteristiche della squadra stessa nei diversi ruoli assunti dai giocatori a seconda della loro posizione sul campo da gioco. Tale problema è stato affrontato utilizzando le 44 colonne della tabella *Match* rappresentanti le coordinate X e Y di ciascuno dei 22 giocatori in campo.

In particolare, considerando il campo da calcio come un piano e rivolgendosi con le spalle alla porta, si è potuto evincere che le X rappresentassero la posizione di un giocatore rispetto al lato corto del campo (quindi a destra o sinistra rispetto al portiere), mentre le Y fossero rappresentative della distanza fra il portiere e il giocatore considerato, quindi la posizione di quest'ultimo rispetto al lato lungo del campo. Proprio questa coordinata, soprattutto grazie ad una conoscenza pregressa del dominio di riferimento, si è rivelata essenziale per determinare i ruoli di ciascun giocatore durante il corso di una partita.

Attraverso una scelta empirica e il confronto fra distribuzioni del numero di giocatori nei diversi ruoli, sono state fatte le seguenti assunzioni:

- La Y pari a 1 è sempre occupata dal portiere;
- Le Y da 2 a 4 costituiscono i difensori;
- Le Y da 5 a 8 sono costituenti dei centrocampisti;
- Le Y da 9 a 11 rappresentano gli attaccanti.

Tale assunzione è totalmente plausibile considerata la tipica formazione di una squadra da calcio ed ha portato all'assegnamento di uno dei 4 ruoli disponibili a ciascun giocatore in partita. Nella figura 1 è possibile quindi osservare la distribuzione assunta dal numero di giocatori per ciascun ruolo nelle varie partite presenti nel dataset. Mentre i difensori sono quasi sempre quattro, gli attaccanti sono fra uno e tre per partita mentre i centrocampisti possono essere da tre a cinque.

Le analisi e le elaborazioni sui dati presentate nel seguito di questo capitolo sono state ottenute tutte attraverso l'utilizzo del linguaggio di programmazione R.

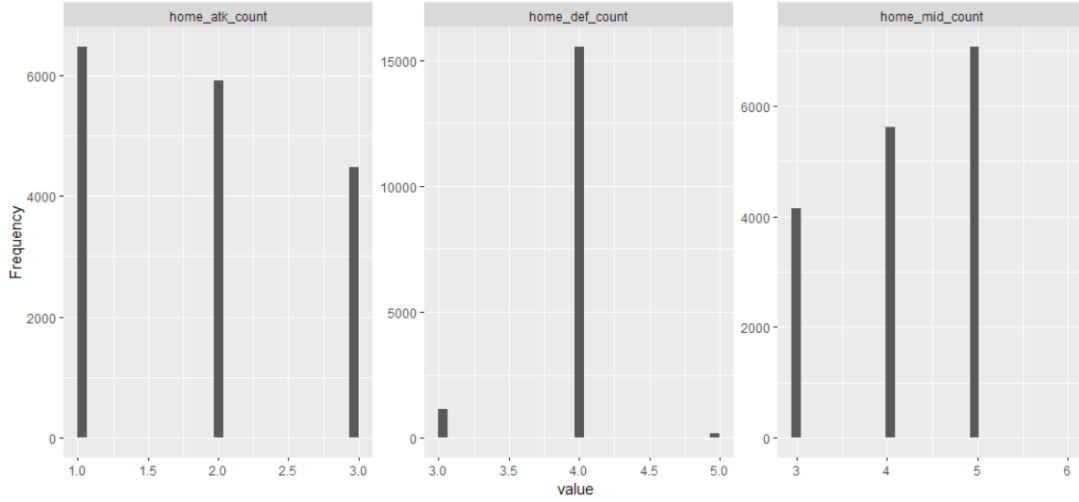


Figura 1: Distribuzione dei ruoli nelle partite

4.3.3 Potenzialità di una squadra in partita

Se in *PlayerAttributes* (le cui colonne sono meglio descritte nella tabella 2) vengono espresse quasi 40 diverse abilità tipiche dei giocatori di calcio, non tutte sono essenziali per ciascun ruolo. Si trovano ad esempio **finishing** e **shot_power** come caratteristiche degli attaccanti oppure **marking** e **standing_tackle** per i difensori, e ancora altri attributi espressamente dedicati ai portieri.

In questo contesto, si è pensato di computare per ciascuna partita e squadra partecipante diversi punteggi per esprimere la potenza di una certa squadra, sulla base dei 4 ruoli disponibili nel calcio: **portiere**, **difesa**, **centrocampo** e **attacco**. Tali punteggi costituiranno in seguito i nodi della Rete Bayesiana su cui fare le dovute predizioni; si è deciso di calcolarli sfruttando i valori registrati dai giocatori che costituiscono i 4 ruoli della squadra durante una partita.

Inizialmente, si era pensato di assegnare a ciascun giocatore della partita un determinato punteggio calcolato attraverso una media pesata delle abilità più rilevanti per il ruolo che si sta giocando. Questo avrebbe comportato la necessità di aggregare per ciascun giocatore diverse abilità a seconda del suo ruolo in partita, per successivamente andare a costituire un'ulteriore aggregazione di tutti i giocatori in una certa posizione al fine di computare le potenzialità della squadra, per ciascun ruolo, durante la partita.

Seppur il calcolo sia logicamente sensato, conoscendo il dominio si è tuttavia ragionato sul fatto che ciascun giocatore del calcio moderno è sempre intrinsecamente

legato ad un ruolo, che assume in tutte le partite da egli giocate. Seguendo questo ragionamento si è conseguentemente provato a verificare l'attendibilità del punteggio `overall_rating`, già espresso all'interno del dataset per ciascun giocatore, che dovrebbe rappresentare il punteggio generale del giocatore con particolare attenzione al proprio ruolo.

È stata pertanto condotta un'analisi di correlazione fra `overall_rating` e gli altri attributi di *PlayerAttribute* per dimostrare che il primo fosse in grado di adattarsi efficacemente al tipico ruolo del giocatore considerato. Tale analisi, presentata nella sezione successiva, ha chiaramente evidenziato che l'attributo `overall_rating` della tabella *PlayerAttribute* è sufficiente a descrivere efficacemente un giocatore, pertanto si è scelto di utilizzare questa colonna come unica caratteristica per costituire i punteggi aggregati di difesa, centrocamp e attacco.

4.3.4 Correlazione fra ruolo e `overall_rating`

Per analizzare la correlazione fra ruolo di un giocatore e `overall_rating`, al fine di comprendere se quest'ultimo sia adatto a descrivere efficacemente un giocatore, si è innanzitutto tentato di suddividere i giocatori in ruoli, sulla base dei valori assunti in determinate abilità e studiando la distribuzione dei valori assunti da quest'ultime.

Facendo riferimento a `gk_diving` e la figura 2, si è facilmente evidenziata una distribuzione nettamente suddivisa fra valori sopra 50 (su una scala di 100) e sotto 40, a scindere inequivocabilmente i giocatori portieri da tutti gli altri. Per questi, è chiaramente evidente in figura 3 che `overall_rating` sia totalmente dipendente dalle abilità legate al portiere e dalla capacità di reagire rapidamente.

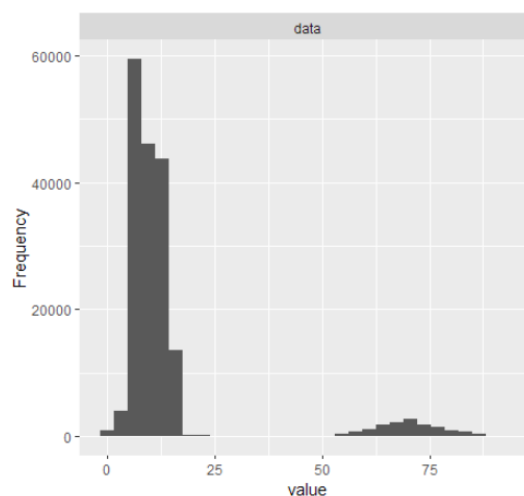


Figura 2: Distribuzione di `gk_diving`

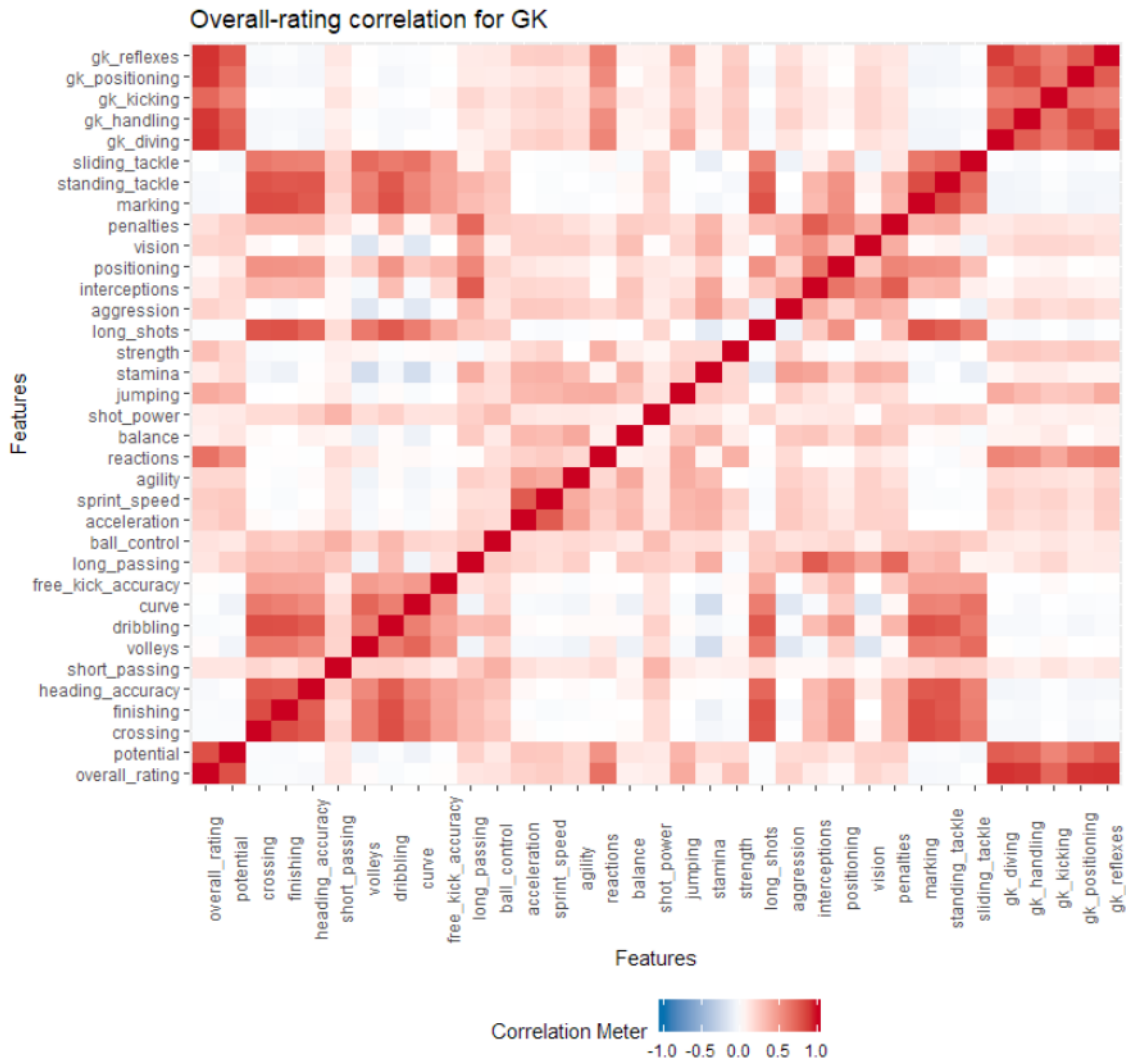


Figura 3: Correlazione fra `overall_rating` (prima riga dal basso) e le abilità di un portiere (a destra)

Studiando invece la correlazione fra le diverse abilità, si è evinto che due fra gli attributi meno correlati fra loro fossero `marking` e `finishing`, associati rispettivamente al ruolo difensivo e quello d'attacco. Sfruttando l'incorrelazione di -0.62 qui mostrata, si è quindi potuto sfruttare i due campi per scindere il dataset dei giocatori in attaccanti e difensori, per studiarne il comportamento di `overall_rating`.

```
cor(PlayerAttributes[PlayerAttributes$gk_diving <= 40, ]$marking,
PlayerAttributes[PlayerAttributes$gk_diving <= 40, ]$finishing,
use = "pairwise.complete.obs"); RESULT = -0.6204823
```

In figura 4 è possibile osservare la matrice di correlazione applicata sui difensori, cioè i giocatori non-portieri con un punteggio minimo di 50 su **marking** e massimo di 50 su **finishing**. Si evince come la maggior parte della correlazione di **overall_rating** (prima riga dal basso) la si ottenga con le abilità tipiche della difesa come **marking**, **sliding_tackle** e **interceptions**. Le abilità del portiere sono totalmente ininfluenti ed in generale anche le caratteristiche d'attacco non sono molto rilevanti.

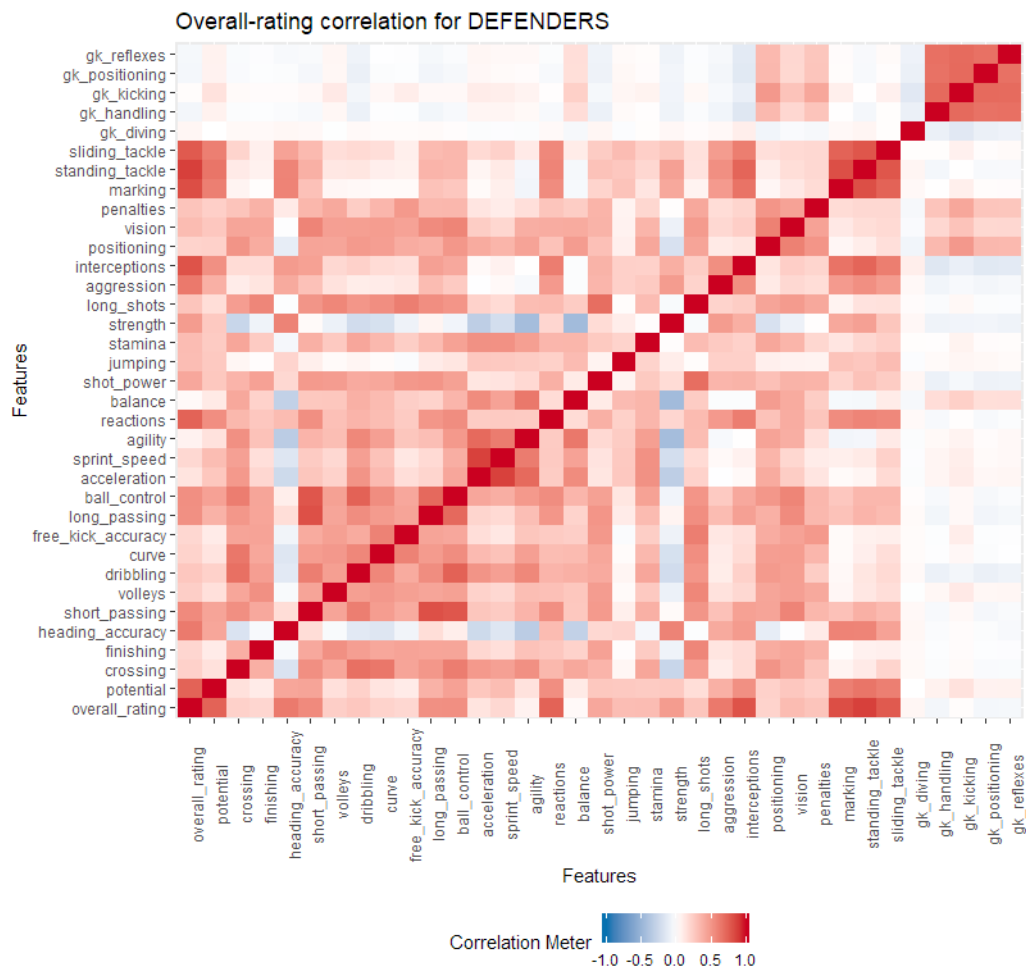


Figura 4: Correlazione fra **overall_rating** (prima riga dal basso) e le abilità di un difensore

Infine, nella figura 5 viene presentata la correlazione fra **overall_rating** e le abilità di un attaccante, cioè un giocatore non-portiere con un punteggio massimo di 50 su **marking** e minimo di 50 su **finishing**. Questa volta si nota che le abilità difensive

non hanno alcuna influenza sul punteggio globale, a differenza di quanto avvenga invece per le abilità più legate ai tiri in porta, al dribbling e alla velocità di corsa, che erano decisamente meno influenti per i difensori.

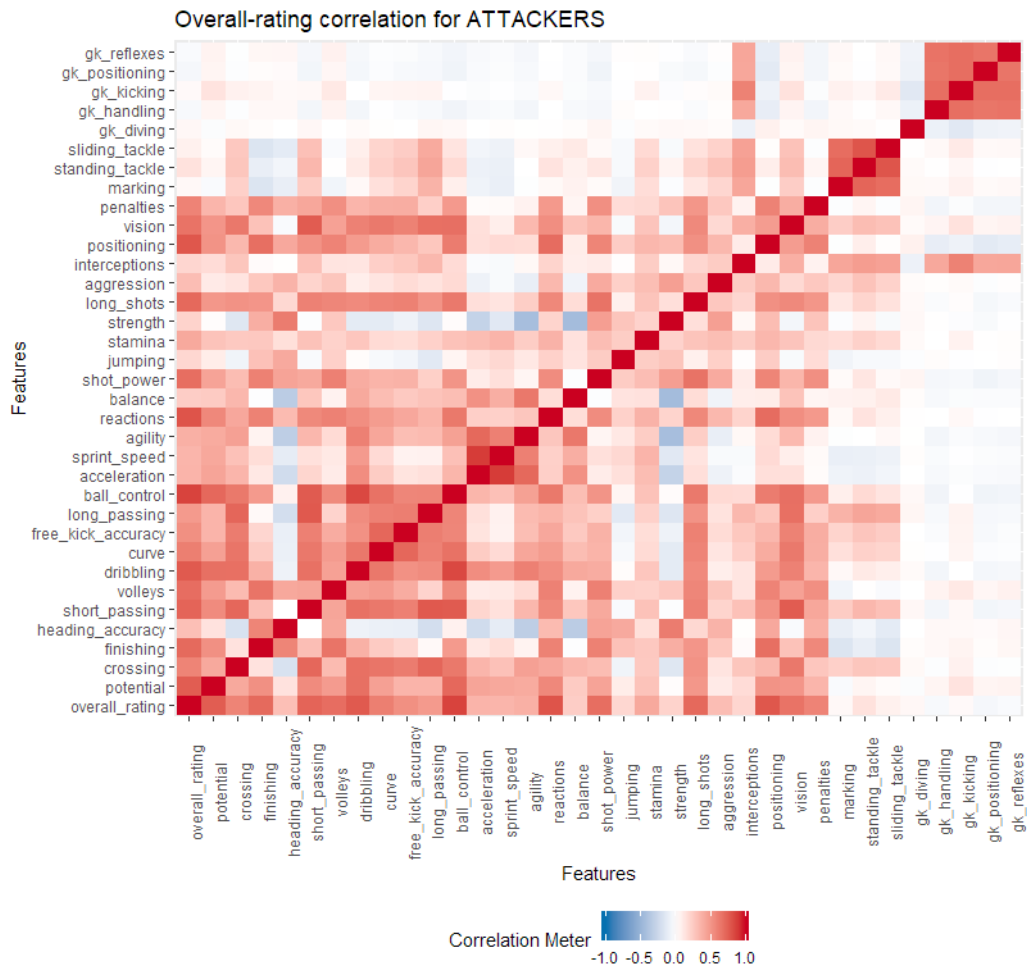


Figura 5: Correlazione fra `overall_rating` (prima riga dal basso) e le abilità di un attaccante

4.3.5 Calcolo e discretizzazione dei punteggi generali per la squadra

Tramite le precedenti considerazioni si è dimostrato che è possibile utilizzare il solo attributo `overall_rating` per descrivere efficacemente un giocatore. Per calcolare quindi i punteggi ottenuti da una certa squadra, composta dagli 11 giocatori, in una certa partita si è quindi pensato di mediare fra loro gli `overall_rating` dei giocatori facenti parte dei 4 diversi ruoli del calcio.

Così facendo si sono ottenuti 4 valori continui, da 1 a 100, rappresentanti le caratteristiche generali della squadra sulla base dei giocatori coinvolti nella partita. Successivamente verrà dimostrato che, per il caso in esame, considerare anche il numero di giocatori in un determinato ruolo è influente.

Al fine di sottoporre queste informazioni ad una Rete Bayesiana, si è quindi proceduto con una discretizzazione dei valori. Inizialmente, si è scelto di utilizzare 4 intervalli di discretizzazione; più avanti in questo elaborato si mostrerà come questa scelta possa avere un'influenza sulle performance del modello di predizione.

I dati ottenuti sono stati ripuliti delle informazioni non più rilevanti, come i dati sui singoli giocatori, e quindi esportati in formato CSV per essere sottoposti alla Rete Bayesiana con lo scopo di fare previsioni sul risultato di una partita e inferenze di diverso genere sui possibili valori assunti dai 4 punteggi per ciascuna squadra.

5 Rete Bayesiana

Lo scopo di questo progetto è la predizione del risultato di una partita di calcio. Sfruttando i dati ottenuti dal dataset d'origine attraverso l'elaborazione esplicita nel dettaglio alla sezione 4.3, è possibile costruire un modello di apprendimento che utilizzi le caratteristiche tecniche delle due squadre in competizione per predire, su base statistica, chi sarà il vincitore.

A questo scopo si è scelto di utilizzare una Rete Bayesiana, un modello grafico probabilistico che rappresenta un insieme di variabili stocastiche con le loro dipendenze condizionali attraverso l'uso di un grafo aciclico diretto (DAG). Nel caso in esame, le variabili stocastiche sono rappresentate proprio dai punteggi assunti dalla squadra in casa e quella in trasferta nei 4 ruoli tipici del calcio.

In questo capitolo vi si farà riferimento con la seguente nomenclatura:

HOME_GK, HOME_DEF, HOME_MID e HOME_ATK per la squadra che gioca in casa e

AWAY_GK, AWAY_DEF, AWAY_MID e AWAY_ATK per la squadra ospite.

5.1 Strumenti e librerie

5.2 Assunzioni

5.3 Generazione della rete

5.3.1 Autogenerazione della struttura

5.3.2 Vincoli

5.3.3 Esperimenti

6 Performance della rete

6.1 Tempi

6.2 Cross Fold Validation

6.2.1 Accuracy

6.2.2 Precision e Recall

6.2.3 Loss

7 Web Demo

Al fine di dimostrare l'utilizzo della rete in un applicativo software, viene qui illustrato lo sviluppo e il funzionamento di un'applicazione web che permette la configurazione delle rose dei due team coinvolti in una partita e la conseguente predizione del vincitore.

7.1 Architettura

L'interfaccia web è stata sviluppata utilizzando l'architettura a 3 layer, separando frontend, backend e database.

Il database utilizzato risulta quello fornito inizialmente senza alcuna modifica. Esso consiste quindi in un file sqlite interrogabile e modificabile semplicemente tramite un web server. Esso risulta particolarmente utile per fornire i dettagli dei giocatori ed eventualmente dei team così che l'utente possa visualizzarli e sceglierli opportunamente.

Per lo sviluppo del backend è stato deciso di utilizzare l'engine javascript tramite il popolare progetto Node.js [3]. Esso è in grado di agire come middleware tra il frontend e il database, separando al meglio le logiche di manipolazione del dato. È inoltre incaricato di chiamare adeguatamente lo script R per la predizione del vincitore della partita e per svolgere l'inferenza.

Il frontend risulta sviluppato utilizzando la libreria javascript React.js [4]

7.2 Interfaccia grafica

Di seguito vengono mostrate brevemente le principali schermate dell'applicazione e il loro funzionamento.

All'avvio dell'app, utilizzando semplicemente un browser web, è possibile notare la schermata principale con 3 tab di selezione: *Home team*, *Away team* e *Results*. Mentre i primi 2 permettono di modificare le squadre, il terzo consente invece di visualizzare i risultati data la configurazione precedente.

La schermata di configurazione della squadra mostra sulla sinistra l'immagine di un campo da calcio su cui è possibile spostare i giocatori, semplicemente trascinandoli col mouse e opportunamente aggiunti tramite il pulsante sulla destra, creando così la formazione da utilizzare.

Una volta finita la configurazione cliccando sul tab dei risultati è possibile vedere in verde la squadra vincitrice e in rosso quella che invece è stata sconfitta.

Per non dover ricreare configurazioni dall'inizio ogni qualvolta si avvii l'applicazione inoltre, nella parte superiore destra dello schermo sono presenti dei tasti per salvare e caricare di precedenti.

7.3 Predizione risultato

Per predire il vincitore della partita, come mostrato precedentemente, è stato necessario utilizzare la rete Bayesiana ottenuta a seguito degli esperimenti. Tale rete è stata infatti salvata su un file che è stato possibile poi ricaricare nell'ambiente R tramite l'apposito comando. Il web server javascript è dunque in grado di chiamare lo script R da riga di comando fornendogli in input una rappresentazione in formato JSON dei giocatori in campo con le loro caratteristiche. Lo script non deve far altro che manipolare i dati che gli sono stati forniti e applicare l'inferenza.

Per mostrare inoltre il puro funzionamento della rete è stato sviluppato un semplice form che permette l'inserimento delle evidenze e fornisce in output le distribuzioni di probabilità degli altri nodi.

8 Conclusioni

Lorem ipsum dolor sit amet.

Riferimenti bibliografici

- [1] Ea sports fifa videogame.
- [2] Kaggle dataset source.
- [3] Node.js.
- [4] React.
- [5] Sqlite.