

POLITECNICO DI TORINO

Laurea di 1° Livello in Ingegneria Gestionale
Classe L-8 Ingegneria dell'Informazione



Football Analytics Tool Software per Società Calcistica

Relatore

Prof. Fulvio Corno

Candidato

Oleksandr Herashchenko

Anno Accademico
2018/2019

Indice

1. PROPOSTA DI PROGETTO	2
2. DESCRIZIONE DEL PROBLEMA	4
3. DESCRIZIONE DEL DATA-SET	6
4. DESCRIZIONE DELLE STRUTTURE DATI E DEGLI ALGORITMI	8
5. DIAGRAMMA DELLE CLASSI	12
6. FUNZIONAMENTO DELL'APPLICAZIONE	13
7. RISULTATI SPERIMENTALI OTTENUTI	17
8. VALUTAZIONI E CONCLUSIONI	18

1. PROPOSTA DI PROGETTO

Studente proponente

s238155 Herashchenko Oleksandr

Titolo della proposta

Tool per Società Calcistica

Descrizione del problema proposto

Si propone una possibile applicazione per talent scout, dirigenti o più in generale stakeholder di una società di calcio.

In particolare, il programma si presenta utile per ottimizzare la rosa: scelta una squadra, si ottiene l'elenco dei giocatori che compongono la rosa e alcune statistiche significative su di essa, in modo da capire quali caratteristiche si possono migliorare agendo sul mercato. L'utente deve inserire in input il budget massimo e scegliere quali parametri si vorrebbero migliorare.

Descrizione della rilevanza gestionale del problema

Negli ultimi anni si è assistito ad una veloce e imponente evoluzione economica delle società di calcio, le quali attualmente possono essere definite delle vere e proprie aziende. Dunque è interessante in una prima analisi andare a valutare quantitativamente e qualitativamente la rosa di giocatori disponibili per capire quali miglioramenti si possono apportare nelle successive sessioni di calcio-mercato.

In una seconda analisi si può inoltre capire come l'aggiunta/sostituzione di calciatori possa influenzare la rosa, sia dal punto di vista economico che qualitativo (fisicità, velocità, ecc..).

Descrizione dei data-set per la valutazione

La fonte dal quale deriva il data-set è il famoso video-gioco FIFA 2019.

Link: <https://www.kaggle.com/karangadiya/fifa19>

I dati sui giocatori sono abbastanza attendibili e il più possibile vicini alla realtà in quanto sono stati realizzati da specialisti del settore sulle misurazioni fatte durante tutto l'anno precedente sui giocatori.

Il formato dei dati è CSV, dal link è possibile scaricare il data-set che presenta una tabella con oltre 18 000 giocatori e oltre 80 colonne con le caratteristiche esaustive di ognuno di essi.

Quindi è necessario formattare i dati ed eliminare le informazioni che sono superflue rispetto all'analisi proposta, devo ancora valutare se estrarre dal data-set solo i calciatori della serie A piuttosto che dei 4 principali campionati europei.

Descrizione preliminare degli algoritmi coinvolti

Il principale algoritmo che verrà implementato sarà sulla ricorsione: data la lista di giocatori che rappresentano la rosa e specificando una lista di giocatori sotto osservazione si provano tutte le possibili combinazioni, fino a che non si trova la soluzione migliore, i vincoli sono i parametri inseriti dall'utente.

Oltre al budget disponibile un altro parametro importante potrebbe essere il numero di giocatori vendibili in modo da liberare spazio in rosa (pensavo ad un limite minimo e massimo di giocatori nella rosa) ed aumentare il budget. L'utente deve scegliere inoltre quale obiettivo vuole migliorare: vorrei calcolare degli indici più generali che derivano da alcuni dati che sono disponibili per ogni giocatore.

Descrizione preliminare delle funzionalità previste per l'applicazione software

L'interfaccia si divide sostanzialmente in due sezioni: una sarà dedicata alla parte di visualizzazione delle caratteristiche della rosa attuale, oltre all'elenco dei giocatori vorrei rappresentare alcune statistiche calcolate globalmente sulla rosa ed eventualmente opportuni grafici per semplificare e rendere più efficace la visualizzazione.

La seconda parte sarà incentrata sull'ottimizzazione della rosa quindi una schermata permetterà di inserire e scegliere i parametri dell'analisi, in risposta ci sarà l'elenco della rosa ipotetica con le statistiche calcolate su di essa, potrebbe essere interessante anche un confronto delle statistiche tra la rosa attuale e quella ipotetica.

2. DESCRIZIONE DEL PROBLEMA

Le società di calcio professionistiche moderne hanno assunto negli ultimi anni, sia dal punto di vista economico che in termini strutturali, la rilevanza e la complessità organizzativa che caratterizzano le grandi aziende.

Come in ogni azienda, l'analisi degli investimenti ricopre un ruolo cruciale per la sostenibilità dell'azienda stessa, nel caso esaminato, gli investimenti si riferiscono all'acquisto/cessione dei calciatori.

Un software di supporto può essere uno strumento utile a più figure professionali all'interno della società, in particolare l'applicazione può essere rivolta ad un direttore sportivo, che si interfaccia con le altre società calcistiche in sede di calciomercato, il quale con l'ausilio dell'allenatore e degli osservatori deve analizzare le criticità della rosa, ovvero capire per quali motivi non sono stati raggiunti gli obiettivi prefissati a inizio stagione, che cosa è mancato, oppure capire come migliorare la rosa in vista di nuovi obiettivi.

Per una società di calcio moderna dunque è di vitale importanza investire il meglio possibile il budget disponibile sul mercato per comprare i giocatori che meglio si addicono alle idee dell'allenatore e che hanno caratteristiche opportune al tipo di gioco che vuole proporre.

Le specifiche sopra esposte si possono tradurre in una variante del problema dello zaino di tipo multi-obiettivo, di seguito la definizione.

Il problema dello zaino, detto anche Knapsack problem, è un problema di ottimizzazione combinatoria: dato uno zaino che possa sopportare un determinato peso e siano dati N oggetti, ognuno dei quali caratterizzato da un peso e un valore. Il problema si propone di scegliere quali di questi oggetti mettere nello zaino per ottenere il maggiore valore senza eccedere il peso sostenibile dallo zaino stesso.

Nello scenario trattato lo zaino si traduce nella rosa della squadra, che parte già da un insieme di oggetti, ovvero i giocatori, anziché da zero. Il peso dello zaino è il prezzo in euro dei giocatori da acquistare/cedere, sono inoltre presenti alcuni vincoli aggiuntivi sul numero minimo e massimo di giocatori sia della rosa globale sia dei 4 ruoli: attaccante, difensore, centrocampista, portiere. Il valore di ogni giocatore rispetto alla rosa è dato dalla somma di più componenti, si tratta di alcuni indici che sono stati calcolati sulla base delle caratteristiche fisiche e qualitative di ogni giocatore.

Sono stati calcolati 4 indici utili a valutare una rosa:

- Overall: media degli *overall* di ogni singolo giocatore;
- Potenziale: media dei *potential* di ogni singolo giocatore;
- Fisico: media sull'indice Fisico di ogni singolo giocatore, quest'ultimo è a sua volta la media di altri 4 parametri: *accelerazione, velocità, reazione, agilità*;
- Tecnica: media sull'indice Tecnica di ogni singolo giocatore, quest'ultimo è a sua volta la media di altri 4 parametri: *cross, dribbling, controllo, tiro*.

Per una descrizione più dettagliata dei parametri che si riferiscono ai singoli giocatori si rimanda al paragrafo successivo.

L'utente può scegliere a quali indici dare più rilevanza: tramite uno slider si può rispettivamente dare più rilevanza all'overall o al potenziale, questo permette di scegliere una rosa tendenzialmente competitiva da subito oppure una rosa che riserva margini di miglioramento. Discorso analogo vale per il fisico e la tecnica, ci sono allenatori che preferiscono un certo tipo di gioco più fisico rispetto ad altri che prediligono la tecnica individuale.

In definitiva, dato un budget disponibile, è possibile trovare la combinazione di calciatori da aggiungere alla rosa di partenza (a meno dei calciatori da cedere i quali vanno ad incrementare il budget iniziale) che massimizza gli indici scelti dall'utente.

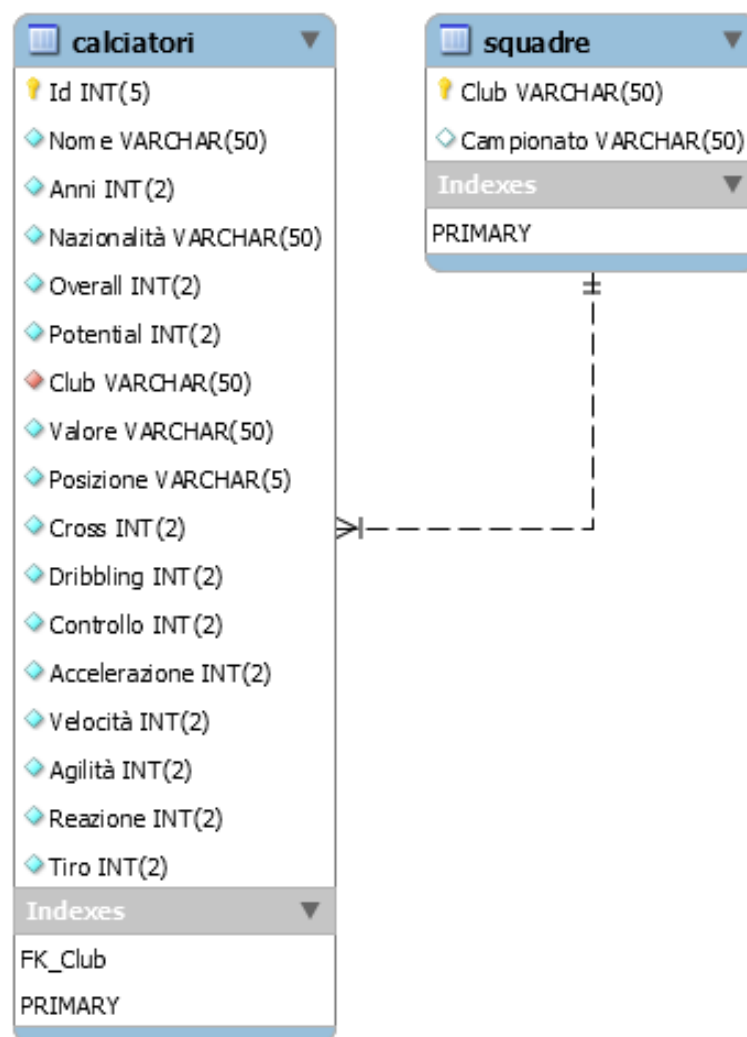
3. DESCRIZIONE DEL DATA-SET

Il data-set utilizzato dall'applicazione è stato realizzato a partire da un data-set trovato online, la cui fonte è il famoso video-gioco di calcio FIFA 2019, disponibile al link <https://www.kaggle.com/karangadiya/fifa19> .

Dal link è possibile scaricare un file CSV di grandi dimensioni. Nello specifico, si tratta di una tabella che contiene oltre 18 000 righe di calciatori con circa 80 colonne di caratteristiche per ognuno di essi.

I dati sono stati ripuliti e inseriti, con alcune aggiunte e modifiche, all'interno di un database denominato "fifa2019", il quale presenta le due tabelle di seguito specificate.

Diagramma del database:



La tabella *calciatori* rispecchia il data-set di FIFA 2019 al netto delle informazioni superflue rispetto all'analisi proposta: dalle 80 colonne iniziali sono state estratte in definitiva 17 colonne:

- *Id*: codice numerico progressivo che identifica un calciatore;
- *Nome*: cognome ed eventuale iniziale del nome del calciatore;
- *Anni*: età del calciatore;
- *Nazionalità*: nazionalità del calciatore;
- *Overall*: indice numerico (da 1 a 99) che indica la forza generale del calciatore;
- *Potential*: indice numerico (da 1 a 99) che indica l'overall potenziale che il calciatore può raggiungere nel futuro;
- *Club*: squadra per cui gioca il calciatore;
- *Valore*: valore in euro del calciatore, il data-set di fifa presenta questo parametro in forma di stringa con M che sta per milioni e K per migliaia. Convertito in valore numerico dall'applicazione per effettuare i calcoli;
- *Posizione*: parametro che indica la posizione in campo preferita del calciatore, sono possibili circa 20 valori diversi, convertiti dall'applicazione nei 4 ruoli principali: attaccante, difensore, centrocampista, portiere;
- *Cross*: indice numerico (da 1 a 99) che indica la qualità dei cross del calciatore;
- *Dribbling*: indice numerico (da 1 a 99) che indica la qualità dei dribbling;
- *Controllo*: indice numerico (da 1 a 99) che indica la qualità nel controllo palla;
- *Accelerazione*: indice numerico (da 1 a 99) che indica l'accelerazione nella corsa;
- *Velocità*: indice numerico (da 1 a 99) che indica la velocità nella corsa;
- *Agilità*: indice numerico (da 1 a 99) che indica l'agilità nei movimenti;
- *Reazione*: indice numerico (da 1 a 99) che indica la reattività nei movimenti;
- *Tiro*: indice numerico (da 1 a 99) che indica la qualità dei tiri;

In aggiunta, è stata creata la tabella *squadre* al fine di avere l'informazione del campionato a cui partecipano i vari club. L'elenco dei Club è stato estratto dalla tabella *calciatori* e sono stati inseriti manualmente i valori nel campo campionato per i 4 principali campionati europei:

- *Club*: nome della squadra;
- *Campionato*: nome del campionato a cui partecipa la squadra, i possibili valori sono Serie A, Bundesliga, Premier League, La Liga, NULL.

4. DESCRIZIONE DELLE STRUTTURE DATI E DEGLI ALGORITMI

L'applicazione è stata sviluppata in linguaggio Java, con il supporto delle interfacce JavaFX, sono implementati il pattern MVC (Model-View-Controller) ed il pattern DAO (Data Access Object).

La struttura dell'applicazione software è divisa in tre package:

- Controller: contiene la classe *Main* per l'avvio dell'applicazione, le tre classi *HomeController*, *AnalizzaController* e *OttimizzaController* che definiscono i metodi che permettono l'interazione con l'utente e i file .fxml: *Home*, *Analizza* e *Ottimizza* per la definizione dell'interfaccia grafica. E' presente anche un foglio di stile css.
- Database: contiene la classe *ConnectDB* utilizzata per la connessione al database e le classi *CalciatoreDAO* e *SquadraDAO* che tramite query SQL permettono il caricamento dei dati presenti nel database all'interno dell'ambiente Java.
- Model: contiene la classe omonima (*Model*) che possiede tutta la logica applicativa del software. Contiene anche le classi Java Bean: *Calciatore*, *Rosa*, *Campionato*.

L'algoritmo principale implementato, sull'ottimizzazione della rosa, usa un metodo ricorsivo per generare tutte le possibili combinazioni e tramite alcuni metodi riportati sotto si può identificare la soluzione migliore.

Di seguito il metodo che inizializza la ricorsione, i parametri in input sono la lista di calciatori ceduti, il budget e i due valori degli slider in percentuale.

```
59     public Rosa calcolaRosaOttimizzata(List<Calciatore> venduti, int budget, double t, double q) {
60         best = new Rosa("best");
61         Rosa parziale = new Rosa(squadra, venduti);
62         best.setCalciatori(new ArrayList<>(parziale.getCalciatori()));
63         primoBest = (1-t) * best.mediaOverall() + t * best.mediaPotenziale();
64         secondoBest = (1-q) * best.mediaTecnica() + q * best.mediaFisico();
65         this.budget = budget;
66         this.t = t;
67         this.q = q;
68
69         int somma = 0;
70         for (Calciatore calciatore : venduti)
71             somma += calciatore.getPrezzo();
72
73         filtraCalciatori(parziale, budget+somma);
74
75         ottimizza(parziale, 0);
76
77         return best;
78     }
```

Prima di iniziare la ricorsione viene avviato questo algoritmo che filtra i calciatori del campionato creando un sottoinsieme di calciatori adatti alle esigenze richieste:

```
private void filtraCalciatori(Rosa iniziale, int b) {
    calciatori.clear();
    calciatori.addAll(cdao.getCalciatori(campionato.getNome()));
    calciatori.removeAll(squadra.getCalciatori());
    List<Calciatore> rimuovi = new ArrayList<>();

    double primo = (1-t) * iniziale.mediaOverall() + t * iniziale.mediaPotenziale();
    double secondo = (1-q) * iniziale.mediaTecnica() + q * iniziale.mediaFisico();

    boolean por = true;
    if ( iniziale.numPortieri() < 3 )
        por = false;

    boolean dif = false;
    if ( iniziale.numDifensori() > 10 )
        dif = true;

    boolean att = false;
    if ( iniziale.numAttaccanti() > 10 )
        att = true;

    boolean cen = false;
    if ( iniziale.numCentrocampisti() > 6 )
        cen = true;

    for (Calciatore c : calciatori) {
        double p = (1-t) * c.getOverall() + t * c.getPotential();
        double s = (1-q) * c.getTecnica() + q * c.getFisico();
        if ( (p < primo || s < secondo) || c.getPrezzo() > b || (cen && c.getRuolo()=="centrocampista")
            || (dif && c.getRuolo()=="difensore") || (att && c.getRuolo()=="attaccante") || (por && c.getRuolo()=="portiere") )
            rimuovi.add(c);
    }

    calciatori.removeAll(rimuovi);
}
```

Il metodo ricorsivo che genera tutte le possibili combinazioni e salva la migliore:

```
80     private void ottimizza(Rosa parziale, int L) {
81
82         if ( L == calciatori.size() ) {
83             if ( controlloMinCalciatori(parziale) && migliore(parziale) ) {
84                 best.setCalciatori(new ArrayList<>(parziale.getCalciatori()));
85                 primoBest = (1-t) * best.mediaOverall() + t * best.mediaPotenziale();
86                 secondoBest = (1-q) * best.mediaTecnica() + q * best.mediaFisico();
87             }
88             return ;
89         }
90
91         ottimizza(parziale, L+1);
92
93
94         parziale.addCalciatore(calciatori.get(L));
95
96         if ( controlloBudget(parziale) && controlloMaxCalciatori(parziale) )
97             ottimizza(parziale, L+1);
98
99         parziale.removeCalciatore(calciatori.get(L));
100
101     }
```

Questo è il metodo che permette di valutare se una rosa è migliore di un'altra:

```
149      /*
150      * Multi-obiettivo da massimizzare
151      */
152      private boolean migliore(Rosa completa) {
153          double primo = (1-t) * completa.mediaOverall() + t * completa.mediaPotenziale();
154          double secondo = (1-q) * completa.mediaTecnica() + q * completa.mediaFisico();
155
156          if ( (primo*0.5+secondo*0.5) >= (primoBest*0.5+secondoBest*0.5) )
157              return true;
158          else
159              return false;
160      }
```

Vincolo sul budget:

```
103      /*
104      * Vincolo sul Budget
105      */
106      private boolean controlloBudget(Rosa parziale) {
107          if ( parziale.valoreTot() > (squadra.valoreTot() + budget) )
108              return false;
109          else
110              return true;
111      }
```

Vincoli dinamici, data la diversità dei campionati, sul numero massimo e minimo di Calciatori:

```
113      /*
114      * Vincoli aggiuntivi sul numero massimo di Calciatori
115      */
116      private boolean controlloMaxCalciatori(Rosa parziale) {
117          if ( parziale.numCalciatori() > max(30, squadra.numCalciatori()) )
118              return false;
119          if ( parziale.numAttaccanti() > max(6, squadra.numAttaccanti()) )
120              return false;
121          if ( parziale.numCentrocampisti() > max(10, squadra.numCentrocampisti()) )
122              return false;
123          if ( parziale.numDifensori() > max(10, squadra.numDifensori()) )
124              return false;
125          if ( parziale.numPortieri() > max(3, squadra.numPortieri()) )
126              return false;
127
128          return true;
129      }
130
131      /*
132      * Vincoli aggiuntivi sul numero minimo di Calciatori
133      */
134      private boolean controlloMinCalciatori(Rosa completa) {
135          if ( completa.numCalciatori() < min(25, squadra.numCalciatori()) )
136              return false;
137          if ( completa.numAttaccanti() < min(4, squadra.numAttaccanti()) )
138              return false;
139          if ( completa.numCentrocampisti() < min(8, squadra.numCentrocampisti()) )
140              return false;
141          if ( completa.numDifensori() < min(8, squadra.numDifensori()) )
142              return false;
143          if ( completa.numPortieri() < min(3, squadra.numPortieri()) )
144              return false;
145
146          return true;
147      }
```

Si riporta come sono stati calcolati i due indici Tecnica e Fisico per ogni calciatore.

```
172         public double getTecnica() {
173             return (cross+dribbling+controllo+tiro)/4;
174         }
175
176         public double getFisico() {
177             return (accelerazione+velocità+reazione+agilità)/4;
178         }
```

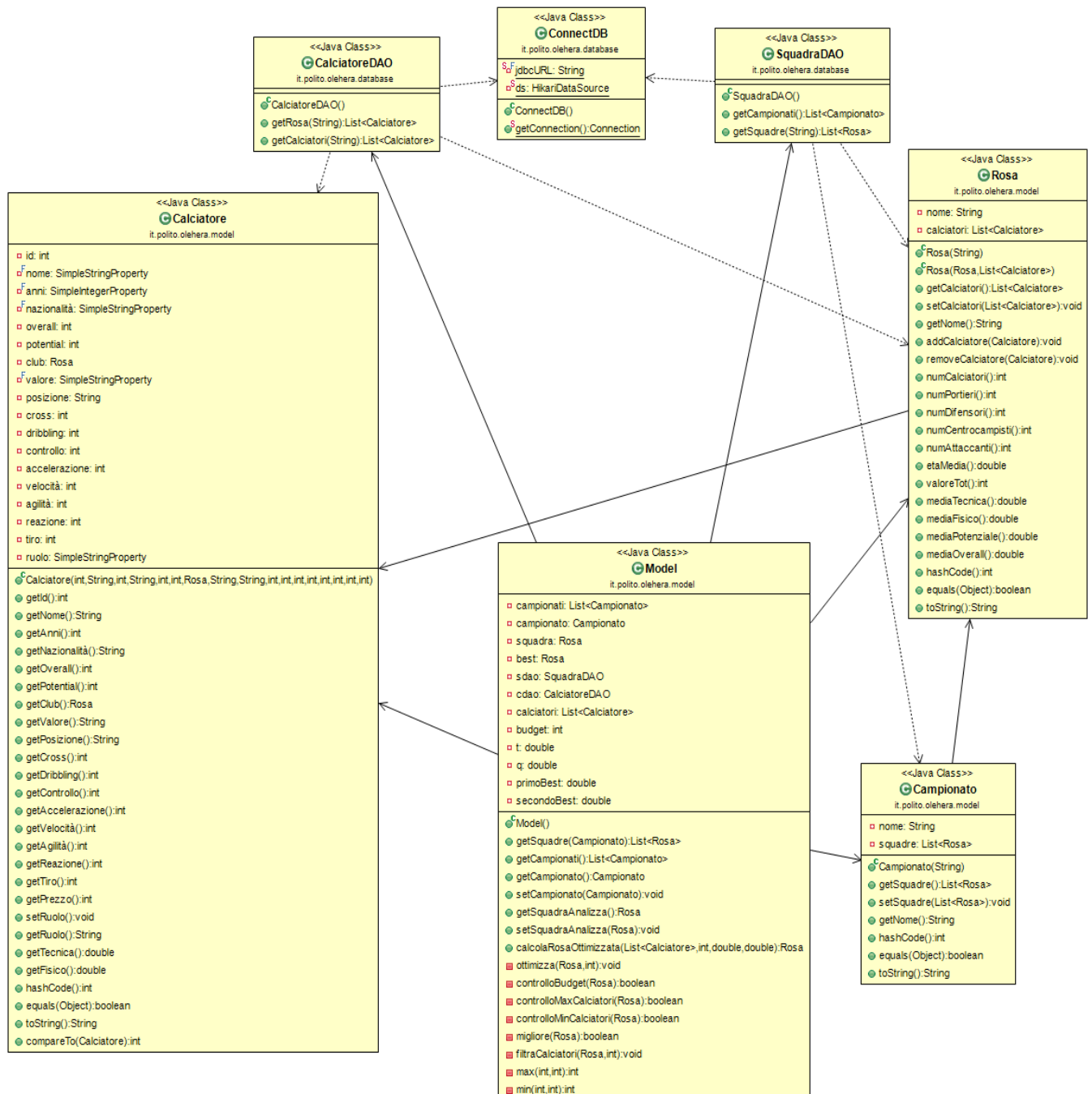
Metodo che trasforma la posizione preferita di ogni giocatore, prelevata dal dataset e la trasforma in uno dei 4 ruoli principali:

```
136         public void setRuolo() {
137             switch (posizione.trim()) {
138                 case "GK": ruolo = new SimpleStringProperty("portiere"); break;
139                 case "RB": ruolo = new SimpleStringProperty("difensore"); break;
140                 case "CB": ruolo = new SimpleStringProperty("difensore"); break;
141                 case "LCB": ruolo = new SimpleStringProperty("difensore"); break;
142                 case "RCB": ruolo = new SimpleStringProperty("difensore"); break;
143                 case "LB": ruolo = new SimpleStringProperty("difensore"); break;
144                 case "RWB": ruolo = new SimpleStringProperty("difensore"); break;
145                 case "LWB": ruolo = new SimpleStringProperty("difensore"); break;
146                 case "CDM": ruolo = new SimpleStringProperty("centrocampista"); break;
147                 case "LDM": ruolo = new SimpleStringProperty("centrocampista"); break;
148                 case "RDM": ruolo = new SimpleStringProperty("centrocampista"); break;
149                 case "CM": ruolo = new SimpleStringProperty("centrocampista"); break;
150                 case "CAM": ruolo = new SimpleStringProperty("centrocampista"); break;
151                 case "RAM": ruolo = new SimpleStringProperty("centrocampista"); break;
152                 case "LAM": ruolo = new SimpleStringProperty("centrocampista"); break;
153                 case "LCM": ruolo = new SimpleStringProperty("centrocampista"); break;
154                 case "RCM": ruolo = new SimpleStringProperty("centrocampista"); break;
155                 case "RM": ruolo = new SimpleStringProperty("centrocampista"); break;
156                 case "LM": ruolo = new SimpleStringProperty("centrocampista"); break;
157                 case "RW": ruolo = new SimpleStringProperty("attaccante"); break;
158                 case "LW": ruolo = new SimpleStringProperty("attaccante"); break;
159                 case "CF": ruolo = new SimpleStringProperty("attaccante"); break;
160                 case "RF": ruolo = new SimpleStringProperty("attaccante"); break;
161                 case "LF": ruolo = new SimpleStringProperty("attaccante"); break;
162                 case "LS": ruolo = new SimpleStringProperty("attaccante"); break;
163                 case "RS": ruolo = new SimpleStringProperty("attaccante"); break;
164                 case "ST": ruolo = new SimpleStringProperty("attaccante"); break;
165             }
166         }
---
```

Infine, ci sono dei vincoli sull'inserimento dei parametri massimi, sia per avere tempi di esecuzione accettabili sia per rendere le analisi realistiche. Il budget non può superare il 15% del valore totale della Rosa ed il numero di calciatori ceduti non può essere più di 1/4 del numero totale di giocatori della Rosa.

5. DIAGRAMMA DELLE CLASSI

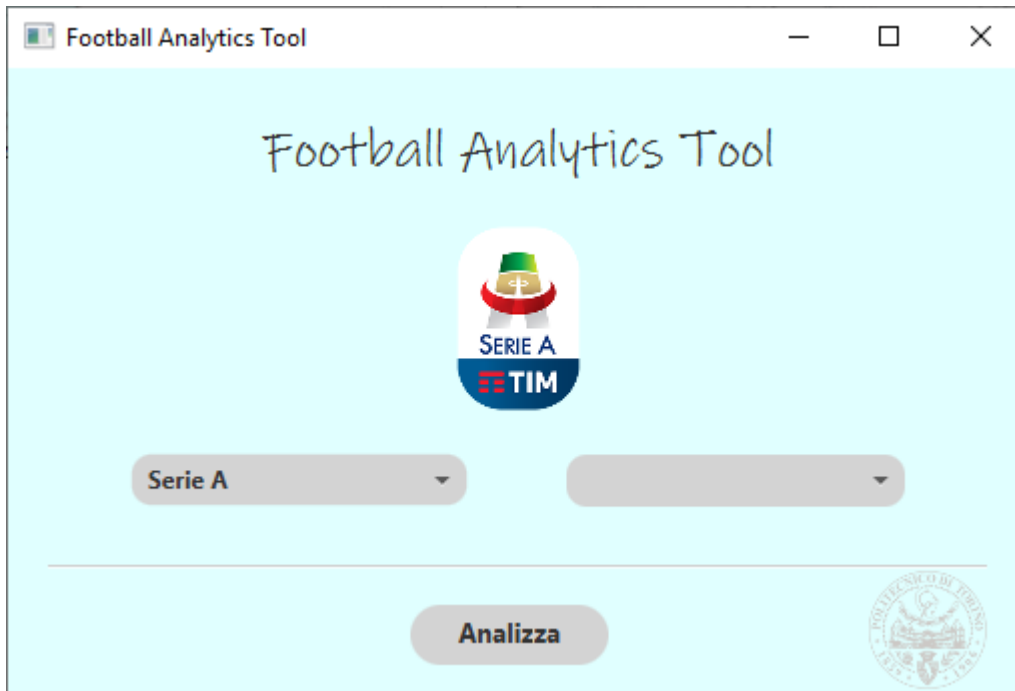
Diagramma delle classi principali (quelle più legate alla parte algoritmica):



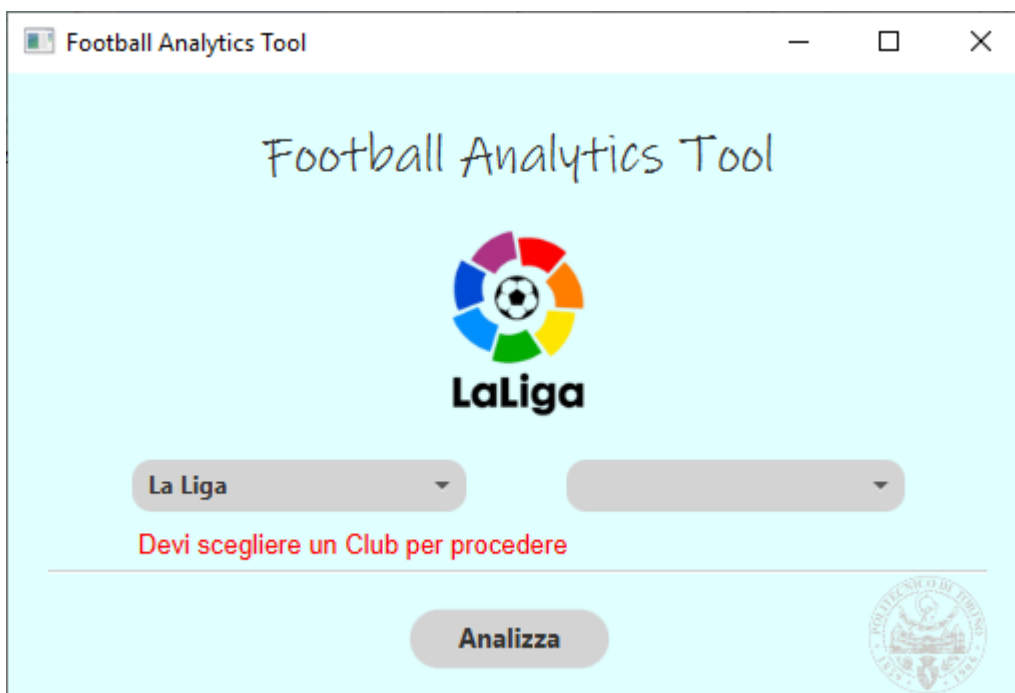
Data la grandezza dell'immagine, nella cartella Documenti è presente il file per permettere l'eventuale zoom.

6. FUNZIONAMENTO DELL'APPLICAZIONE

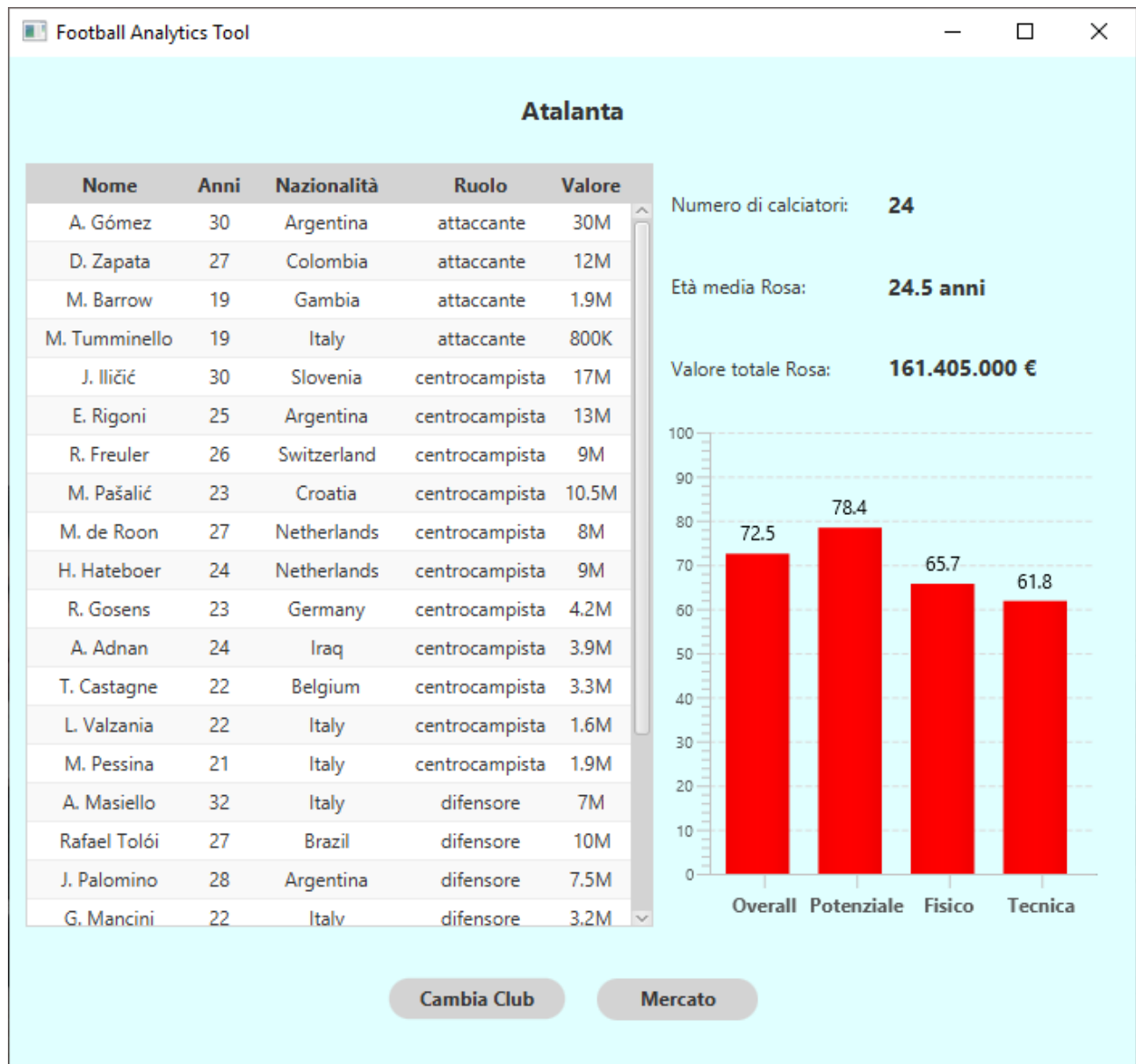
All'avvio dell'applicazione si apre la finestra Home: l'utente può scegliere uno tra i 4 principali campionati europei dal menù a tendina a sinistra e poi da quello di destra il club interessato.



Il logo cambia simultaneamente alla scelta del campionato ed è presente un controllo per la selezione del club.



Cliccando sul bottone Analizza si aprirà la seguente finestra in cui è presente una tabella con l'elenco dei calciatori che compongono la rosa e sulla destra sono mostrate alcune statistiche globali sulla rosa. Nel grafico a barre sono mostrati i 4 indici su cui poi è possibile lavorare nella fase di mercato.



Il bottone Cambia Club permette di tornare alla Home per scegliere un altro club, proseguendo sul bottone Mercato invece si apre una nuova finestra in cui è possibile fare analisi di calcio-mercato.

In questa sezione è richiesto l'inserimento dei parametri da parte dell'utente: nella casella del Budget bisogna inserire una quantità in euro come nel formato esposto nella colonna Valore della tabella; è possibile specificare i calciatori da vendere cliccando sulle righe dei calciatori e sono presenti due slider per permettere all'utente di decidere quale rilevanza dare agli indici che si intendono massimizzare.



Cliccando il bottone Cancella si possono ripristinare i parametri in input.

Cliccando il bottone Calcola, dopo un controllo sui parametri in input massimi e su eventuali errori di immissione, la finestra si aggiorna e genera un risultato come di seguito mostrato.



E' possibile trovare il VIDEO DIMOSTRATIVO del software al link:

<https://youtu.be/UvMhBUyKIBQ>

7. RISULTATI SPERIMENTALI OTTENUTI

Campionato	Squadra	Calciatori	Budget	Ceduti	Filtrati	Nuovi	Tempo (s)
Bundesliga	Borussia Dortmund	33	60M	1	25	1	0.08
Bundesliga	VfL Wolfsburg	32	27.4M	3	46	1	0.01
Bundesliga	FC Schalke 04	29	13.8M	3	67	3	1.53
Bundesliga	TSG 1899 Hoffenheim	33	19M	5	87	4	39.64
Bundesliga	Eintracht Frankfurt	33	24.67M	7	99	4	139.24
La Liga	Sevilla FC	30	27M	4	82	4	43.12
La Liga	Valencia CF	33	31M	6	66	4	1.49
La Liga	Real Betis	27	4M	2	81	5	2.09
La Liga	Levante UD	32	6M	5	97	5	11.10
La Liga	FC Barcelona	33	90M	2	6	2	0.01
Serie A	Napoli	25	40M	4	42	4	1.58
Serie A	Torino	26	20M	4	182	8	2503
Serie A	Sampdoria	27	0M	5	89	5	5.79
Serie A	Cagliari	24	15M	0	79	3	0.25
Serie A	Fiorentina	24	20M	0	189	5	38.32
Premier League	Leicester City	32	14.6M	5	57	3	1.73
Premier League	Chelsea	33	78M	7	58	5	85.21
Premier League	West Ham United	32	20M	6	106	2	0.51
Premier League	Liverpool	33	0M	3	77	3	2.98
Premier League	Watford	29	18M	2	39	1	0.09

8. VALUTAZIONI E CONCLUSIONI

I tempi di esecuzione sono stati il principale problema da risolvere: mediante un algoritmo che filtra i calciatori del campionato in un sotto-insieme di calciatori adatti alle esigenze richieste dall'utente (andando a valutare quantitativamente le lacune per reparto della Rosa, i parametri dei calciatori ed il loro valore) ed alcuni vincoli sui parametri in input massimi, si è riusciti ad ottenere dei tempi accettabili. Anche se ci sono tutt'ora dei casi molto rari in cui la ricorsione ci mette più minuti, si è scelto di non andare a forzare ulteriormente i vincoli per non perdere di realtà: si tratta di casi difficili da gestire data la quantità e varietà dei parametri in input che sono dinamici, ovvero la scelta dei calciatori da cedere influenza molto il budget che aumenta e i valori medi della rosa che cambiano, quindi la scelta di un calciatore piuttosto che un altro pur mantenendo gli altri parametri invariati può portare a situazioni completamente differenti.

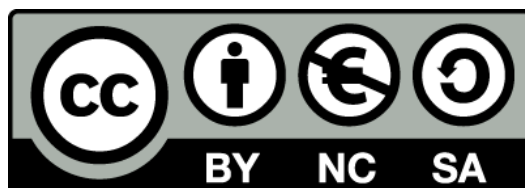
Uno dei punti di forza dell'applicazione è che il suo funzionamento è indipendente dal dataset, quindi può essere aggiornato di anno in anno per avere i parametri aggiornati, fondamentali per poter fare analisi più realistiche possibili.

Nel complesso, si può dire che il funzionamento dell'applicazione è fluido e i risultati ottenuti sono interessanti e alquanto attuali, bisogna tuttavia tener conto che, nel mondo del calcio-mercato reale, andrebbero comunque considerati numerosi altri aspetti: se i parametri dei calciatori possono essere considerati abbastanza affidabili, lo stesso non si può dire per i valori dei giocatori che sono infatti delle stime ma che nella realtà dipendono molto dal momento, dalla volontà del calciatore, del suo agente e dal proprietario del cartellino; un altro fattore importante sono gli stipendi dei giocatori e dello staff che non sono presi in considerazione.

Lo scopo del software è dunque quello di fornire uno strumento, che permette ai dirigenti o stakeholder delle società calcistiche quando devono prendere decisioni, di vedere da un altro punto di vista gli investimenti che si stanno valutando e poter quindi scegliere in modo razionale.

Fonti:

- Link per informazioni sulla struttura organizzativa di una società calcistica:
https://it.wikipedia.org/wiki/Squadra_di_calcio
- Link per informazioni più dettagliate sul problema dello zaino:
https://it.wikipedia.org/wiki/Problema_dello_zaino



Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale.

Copia della licenza consultabile al sito web:

<http://creativecommons.org/licenses/by-nc-sa/4.0/>