Database scacchistico - Data Management

Alberto Gadda, Paolo Guerini Rocco, Letterino Sauro

CdLM Data Science, Università degli studi di Milano Bicocca

Abstract

Il gioco degli scacchi è uno dei più antichi giochi da tavolo sopravvissuti fino al giorno d'oggi, con una user base in costante espansione di oltre di mezzo miliardo di giocatori in tutto il mondo [2]. Negli ultimi anni gli scacchi hanno espanso ulteriormente la loro audience attraverso il gioco online, affermandosi quindi anche come una tipologia di e-sport con tornei, sponsor, piattaforme social e fan base dedicati. Questa metamorfosi digitale ha aperto le frontiere ad una datificazione senza precedenti in termini di tempestività, costo e capillarità dell'informazione. L'elaborato qui presente si pone l'obiettivo di descrivere l'aspetto più competitivo di questa realtà, tracciando informazioni storicizzate relative ai migliori giocatori, le partite da essi giocate e le loro abitudini di gioco. Per svolgere questo compito sono stati raccolti dati provenienti dalle due piattaforme leader del settore, "Lichess" e "Chess.com", attraverso l'utilizzo delle loro API ufficiali e di scraper dedicati. La raccolta e l'enrichment sono stati svolti utilizzando il linguaggio di scripting Python e lo storing utilizzando come DBMS Sqlite, in quanto adatto all'immagazzinamento di dati strutturati.

Introduzione

Il gioco degli scacchi è uno dei più antichi giochi da tavolo sopravvissuti fino al giorno d'oggi, con una user base in costante espansione di oltre di mezzo miliardo di giocatori in tutto il mondo [2].
Affonda le proprie radici nell'India del IV secolo a.C., dove si giocava un gioco dalle caratteristiche sorprendentemente simili chiamato "Chaturanga" (che in Sanscrito significa "quadripartito") [4]. Il gioco originale consiste in una simulazione di una battaglia giocata a turni su una griglia di dimensioni 8x8, con due eserciti schierati ai lati opposti. Ogni esercito è composto da un numero fisso di pedine che si possono muovere secondo regole specifiche dettate dal loro tipo [1]. Siccome la disposizione iniziale dei pezzi e le regole sono comuni ad entrambi i giocatori, il Chaturanga si fonda esclusivamente sulla strategia e non ammette alcuna forma di aleatorietà.
Questa forma di scacchi primordiali si è evoluta nel tempo in numerose varianti, di cui la più diffusa sono gli odierni scacchi. La struttura del gioco è rimasta tendenzialmente simile nel tempo eccetto per alcune differenze nella colorazione della griglia, nelle regole di movimento dei diversi pezzi e nelle condizioni di vittoria.
Nell'Europa del XV secolo d.C. gli scacchi hanno ricevuto ampio interesse dai membri della nobiltà e dell'alta borghesia, divenendo implicitamente uno sport dedicato ai ceti più alti della società, un simbolo di ricchezza. Al giorno d'oggi questo connotato si è perso, anche se sono rimaste vigenti in ambiente competitivo numerose norme di *bon ton*, retaggio dell'ambiente altamente formale in cui gli scacchi sono prosperati.
Negli ultimi secoli gli scacchi sono diventati progressivamente più popolari presso tutte le fasce della società. In aggiunta, negli ultimi anni hanno espanso ulteriormente la loro accessibilità e audience attraverso il gioco online, affermandosi quindi anche come una tipologia di e-sport con tornei, sponsor, piattaforme social e fan base dedicati.
La dimostrazione del successo degli scacchi come e-sport è resa evidente da un articolo pubblicato nel 2020 dalla rivista "Forbes", contenente la top 25 degli atleti di e-sport più pagati: Magnus Carlsen (attuale campione del mondo di scacchi in formato classico) è primo con $510K, il 33% in più del secondo classificato. Altri due scacchisti di fama internazionale occupano il settimo e dodicesimo posto, rispettivamente Hikaru Nakamura ($324K) e Wesley So ($246K) [3].

Questa metamorfosi digitale ha aperto le frontiere ad una datificazione senza precedenti in termini di tempestività, costo e capillarità dell'informazione: sono ora disponibili gratuitamente dati relativi a decine di milioni di partite al giorno, giocate a qualsiasi livello.
Questo sarebbe stato impensabile anche solo un decennio fa, dove le partite giocate erano in proporzione molto poche, sporadiche e di cui le mosse erano trascritte manualmente in archivi fisici solo per le partite più rilevanti.
Alla luce di queste nuove opportunità, l'elaborato qui presente si pone l'obiettivo di descrivere l'aspetto più competitivo di questa realtà, tracciando informazioni storicizzate relative ai migliori giocatori, le partite da essi giocate e le loro abitudini di gioco.

Per svolgere questo compito sono stati raccolti e arricchiti dati provenienti dalle due piattaforme leader del settore, "Lichess" e "Chess.com" (nota anche come "ChessDotCom").
Le due piattaforme, per quanto offrano lo stesso servizio, hanno caratteristiche e intenti differenti. La prima infatti è una piattaforma open source, ad-free, incentrata sul gioco in sé e per sé; la seconda è molto più orientata su un modello di business che offre una vasta gamma di servizi, con un focus sulla costruzione di una community, condivisione di articoli, erogazione di canali di streaming ufficiali e sponsorizzazioni. Questo comporta che la user base di ChessDotCom susciti un maggiore fascino presso i giocatori professionisti, che trovano in essa una piattaforma di lancio per la propria carriera. Lichess gode invece di un'utenza

media più amatoriale, di conseguenza venendo etichettata come una piattaforma di minor prestigio.
Siccome la user base online tende a preferire partite rapide, per ogni piattaforma il focus della ricerca sono i migliori 50 giocatori per ciascuna delle tre categorie più veloci: “bullet”, “rapid” e “blitz”. Le tre categorie fanno riferimento alle limitazioni di tempo imposte ad ogni giocatore per compiere le proprie mosse, pena la sconfitta immediata.
Indicativamente il tempo concesso ad ogni giocatore per ciascuna categoria è il seguente:

- **bullet** : fino a 3 minuti.
- **blitz** : da 3 a 10 minuti.
- **rapid** : da 10 minuti a 60 minuti.

Le due piattaforme offrono anche categorie di scacchi più lente o con variazioni rispetto alle regole classiche, ma saranno trascurate in quanto spiccatamente di minor rilevanza e popolarità.

Data Esploration

L’obiettivo che il presente lavoro si pone di raggiungere è quello di produrre un database aggiornato in real-time che storicizzi le performance relative ai migliori giocatori, le loro partite e il loro grado di engagement nei confronti della piattaforma. La definizione di “migliori giocatori” è basata sulla selezione dei soli giocatori che siano stati parte, in almeno un istante temporale (a partire dall’esecuzione dello script), di una delle leaderboard designate. Questo significa che il numero di righe che compongono le tabelle del dataset è in costante espansione. Di conseguenza è doveroso specificare che a seguire verranno riportate le misure di grandezza e qualità riferite all’istante temporale relativo alla stesura di questo documento.

Il database è composto da 8 tabelle, di cui 3 relative a Lichess (*Rating_Lichess*, *Online_Lichess* e *Match_Lichess*), 3 relative alla piattaforma ChessDotCom (*Rating_ChessDotCom*, *Online_ChessDotCom* e *Match_ChessDotCom*) e una comune ad entrambe le piattaforme (*Player*).

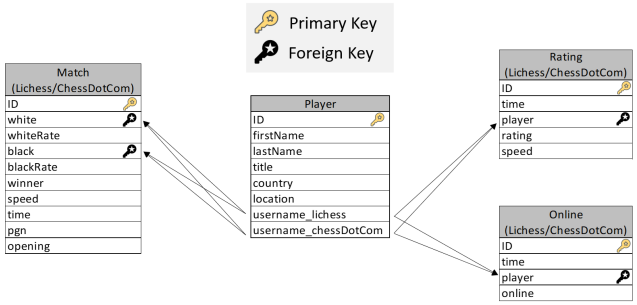

Figura 1. ER Schema

La tabella *Player* è composta da tutti i dati relativi alle informazioni personali dei giocatori, ha 8 colonne (di cui una è l’ID) e 305 righe. 194 dei giocatori presenti provengono dalla piattaforma Lichess e 122 da ChessDotCom. Ciò implica che 11 giocatori possiedono profili riconducibili ad entrambe le piattaforme.

Nome	Descrizione
id	(AUTO INCREMENT PRIMARY KEY)
firstName	nome del giocatore
lastName	cognome del giocatore
title	livello certificazione FIDE[1]
country	nazione d’origine
location	zona d’origine
username_Lichess	username su Lichess (FOREIGN KEY)
username_ChessDotCom	username su ChessDotCom (FOREIGN KEY)

Tabella 1. Player

Le tabelle *Online* storicizzano i dati relativi all’assiduità con cui i giocatori frequentano la piattaforma. Hanno 4 colonne (di cui una è l’ID), con 56058 righe per ChessDotCom e 79494 per Lichess.

Nome	Descrizione
id	(AUTO INCREMENT PRIMARY KEY)
player	username del giocatore (FOREIGN KEY)
time	data e ora della rilevazione
online	variabile binaria che riporta se il giocatore era online

Tabella 2. Online

Le tabelle *Match* contengono per ogni giocatore i dati relativi alle partite giocate sulle due piattaforme dal momento della creazione del profilo. Hanno 10 colonne e circa 2 milioni righe.

Nome	Descrizione
ID	ID del match fornito dalla piattaforma[2] (PRIMARY KEY)
white	username del giocatore bianco (FOREIGN KEY)
whiteRate	punteggio del giocatore bianco ad inizio partita
black	username del giocatore nero (FOREIGN KEY)
blackRate	punteggio del giocatore nero ad inizio partita
time	data e ora della partita
pgn	lista delle mosse giocate durante la partita
speed	categoria della partita
winner	colore del vincitore
opening	apertura della partita[3]

Tabella 3. Match

[1] La World Chess Federation (FIDE) assegna diversi titoli basati sulle prestazioni ai giocatori di scacchi. In ordine crescente di importanza sono: Candidate Master (CM), FIDE Master (FM), Master internazionale (IM) e Grandmaster (GM).

[2] L’ID permette di cercare l’analisi tecnica direttamente sulla piattaforma, che offre ad esempio la precisione di ogni mossa e gli errori commessi (l’informazione non era integrabile nella base dati perché assente nell’API e sarebbe stato troppo dispendioso in termini di tempo raccoglierla attraverso lo scraping).

[3] Le prime due mosse di una partita di scacchi sono le più importanti perché ne influenzano enormemente la prosecuzione. Queste coppie di mosse iniziali vengono chiamate "aperture" e assumono nomi diversi a seconda di quali sono le mosse che le compongono.

Le tabelle *Rating* contengono invece la serie storica delle leaderboards nelle diverse categorie. Hanno 5 colonne (di cui una è l'ID) e 12000 righe per entrambe le versioni di Lichess e ChessDotCom.

Nome	Descrizione
id	(AUTO INCREMENT PRIMARY KEY)
time	data e ora della rilevazione
player	username del giocatore (FOREIGN KEY)
rating	punteggio raggiunto
speed	categoria a cui è associato il rating

Tabella 4. Rating

Data Acquisition

Si è deciso di raccogliere i dati relativi ai giocatori che, durante l'esecuzione dello script, sono stati presenti almeno una volta nelle prime 50 posizioni di una delle classifiche nelle categorie bullet, blitz o rapid. La maggior parte dei dati presentati è messa a disposizione dalle API ufficiali dei due siti. I dati ricevuti dalle API, dopo essere stati trasformati in JSON, sono stati sottoposti a varie iterazioni di data cleaning volte a rendere i risultati qualitativamente migliori e più agilmente integrabili tra le due piattaforme.
Un primo ostacolo nella raccolta dati è stato che le API impongono molte limitazioni sul quantitativo di filtri di ricerca ad esse applicabili. Questo ad esempio ha vincolato la definizione di "top players" alla top 50 per categoria, obbligando a scartare l'ipotesi di un filtro invece sul rating. Un secondo impedimento è stato causato dalla funzione "is_online()" della API di ChessDotCom, che essendo stata deprecata ha minato la possibilità di tracciare in real-time lo status dei giocatori. Per fronteggiare questo problema si è fatto ricorso ad uno scraper, essendo questa informazione ancora presente nella interfaccia grafica del sito relativa ai profili giocatore. Siccome tuttavia questo dato è popolato solo dopo l'esecuzione di uno script di JavaScript al momento del caricamento della pagina, è stato necessario l'utilizzo di Selenium per simulare un'istanza di Chrome e consentire così agli script di generare il codice sorgente finale della pagina. Un secondo scraper è stato utilizzato per raccogliere i dati relativi alle aperture delle partite su ChessDotCom, essendo questa importante informazione offerta esclusivamente dall'API di Lichess.

ChessDotCom. La piattaforma ChessDotCom mette a disposizione un'API interrogabile con Python attraverso delle funzioni contenute nella sua apposita libreria "chessdotcom", utilizzabili senza bisogno di autenticazione [5]. Sono state in seguito create delle funzioni custom che incorporano quelle offerte dall'API per semplificarne l'utilizzo. Le funzioni custom impiegate nell'analisi sono illustrate nella Tabella 5.

Lichess. La piattaforma Lichess fornisce a sua volta una API interrogabile con Python attraverso la apposita libreria "Berserk" [6]. Tramite questa libreria è possibile inizializzare un Client, che dopo aver effettuato un'autenticazione tramite token è in grado di inoltrare richieste all'API del sito. Sfruttando questo strumento sono state create anche in questo caso delle funzioni custom per scaricare i dati di interesse, illustrate nella Tabella 6.

Funzione	Output
getChessDotComProPlayers()	elenco degli username di tutti i giocatori che attualmente sono nella top 50 di almeno una classifica
getChessDotComProfile(username)	dati personali dell'utente (nome, cognome, title, country, location)
getChessDotComPlayerRating(username)	punteggio del giocatore nelle tre categorie
get_chessDotCom(username_list,period)	se period=0 restituisce lo storico di tutte le partite dei giocatori presenti nella lista, altrimenti le partite del giorno corrente di tali giocatori

Tabella 5. ChessDotCom custom functions

Funzione	Output
getLichessProPlayers()	elenco degli username di tutti i giocatori che attualmente sono nella top 50 di almeno una classifica
getLichessProfile(username)	dati personali dell'utente (nome, cognome, title, country, location)
getLichessPlayerRating(username)	punteggio del giocatore nelle tre categorie
get_lichess(username_list,period)	se period=0 restituisce lo storico di tutte le partite dei giocatori presenti nella lista, altrimenti le partite del giorno corrente di tali giocatori

Tabella 6. Lichess custom functions

Data Manipulation

I dati ricevuti dalle API attraverso le funzioni custom sono stati trasformati in JSON per semplificare la ricerca delle chiavi di interesse e per uniformare gli output. Infatti l'API di ChessDotCom restituisce un oggetto di tipo collection, mentre l'API di Lichess un dizionario. Sono seguite operazioni di formatting delle colonne (ad esempio trasformare le date in formato datetime) e di data cleaning (ad esempio mettere in lowercase i nomi di persona ed effettuare lo stripping di eventuali spazi bianchi o caratteri speciali accidentali).
Non essendo supportate da queste API le ricerche basate sulle coppie nome-cognome, ci si è putroppo dovuti limitare a cercare i profili doppi esclusivamente attraverso il match dello username. Questo significa che qualora un giocatore abbia utilizzato due profili con username differenti e non fossero entrambi classificati, non sarebbe possibile risalire al profilo assente nella leaderboard.
Una volta verificato se sulle due piattaforme esiste lo stesso username, si effettua un controllo addizionale per verificare che l'identità fisica del giocatore sia effettivamente corrispondente. Per raggiungere questo scopo è stato usato il seguente approccio: quando lo username corrisponde si controlla che sia identica anche la coppia nome-cognome. Qualora i profili confrontati siano entrambi in classifica si assume questo come sufficiente criterio per decretare l'identità. Qualora invece uno dei due non sia in classifica, viene richiesto per ulteriore sicurezza che anche i campi title e country corrispondano.

Data Storage

La fase di data storage avviene tramite l'utilizzo di query che caricano sul database l'output dalle funzioni che si interfacciano con l'API. In particolare è stata definita una query per ogni tabella, che viene richiamata subito dopo ogni ciclo di data acquisition. La query in oggetto si limita a caricare i nuovi dati sul database verificando però di non violare vincoli di formattazione e di non produrre duplicati dei dati.
Per quanto concerne lo schema da utilizzare nella fase di immagazzinamento dei dati, si è scelto di utilizzare un modello ben comprensibile a tutti gli analisti provenienti anche da background non tecnici, ossia quello relazionale. Il principale vantaggio di un RDBMS è la possibilità di indicizzare rapidamente i records ed effettuare query che filtrano su range di valori, applicazione utile avendo a disposizione dati declinati sull'asse temporale. Inoltre il dato a livello intuitivo si presenta già come strutturato, siccome il numero di attributi d'interesse per questo business case non è di numero variabile o indeterminato (essendo vigente l'ipotesi di mondo chiuso). In aggiunta alla struttura fissa, gli attributi già a disposizione sono quasi sempre non-nulli, massimizzando quindi l'efficienza delle celle generate dal modello. Infine utilizzando gli username come chiavi, è possibile effettuare operazioni di join direttamente con ogni tabella, senza richiedere concatenazioni di join che possano ridurre le performance.
Tra tutti i possibili DBMS che offrono una struttura relazionale, in particolare è stato scelto SQLite. SQLite è un RDBMS gratuito, focalizzato sulla fornitura di un potente database compatibile con SQL. Come suggerisce il nome, questa è una soluzione leggera che può essere eseguita su quasi tutto ciò che supporta C e l'archiviazione di file persistente, compatibile con tutti i linguaggi di programmazione di alto livello più diffusi. La mancanza di un componente server rende SQLite molto più facile da configurare, con processi di popolazione snelli, ma senza sacrificare la resilienza agli errori di archiviazione e agli scenari di memoria insufficiente. Nel caso questo progetto dovesse essere scalato, sarebbe probabilmente necessario trasferirsi su una base di dati che si appoggi ad una componente server, ma questo scenario si verificherebbe dopo un lasso di tempo non d'interesse per la portata di questo progetto.

Data Quality

Le principali misure impiegate per valutare la qualità del dataset ottenuto sono la currency e la completeness. La prima quantifica il ritardo tra l'accadimento di un fenomeno reale e l'acquisizione del dato. La seconda verifica quanto i dati contengano tutte le informazioni necessarie a descrivere tale fenomeno.
Relativamente alla currency, le tabelle Rating e Player si aggiornano con frequenza oraria, le tabelle Online ogni 10 minuti, mentre quelle dei Match una volta al giorno. Queste tempistiche sono state scelte sulla base del workload richiesto per processarle e della granularità ritenuta utile.
Relativamente alla completeness, le uniche tabelle a presentare missing values sono le tabelle Match e Player. Nelle tabelle Match l'esistenza di missing values è limitata al campo opening (intorno allo 0.1% sul totale). La loro presenza è giustificata dall'impossibilità di identificare il nome dell'apertura in partite conclusesi prima di raggiungere la seconda mossa (presumibilmente per problemi di connessione). Nella tabella Player i dati mancanti per colonna arrivano fino al 60% e sono imputabili esclusivamente all'irreperibilità dei doppi profili, siano essi inesistenti o soltanto non trovati dallo script.
Siccome il database si fonda sull'ipotesi di mondo chiuso rispetto al suo business case, l'unico e oggettivo difetto di qualità rilevato consiste quindi nella porzione di missing values che è causata dalle limitazioni di questa architettura.

Risultati e Possibili Sviluppi Futuri

Il database prodotto risponde alla domanda di business di descrivere l'aspetto più competitivo del mondo scacchistico, tracciando informazioni storicizzate relative ai migliori giocatori, le partite da essi giocate e le loro abitudini di gioco. In particolare, sfruttare due fonti diverse ha permesso di porre il focus sui giocatori in possesso di un profilo per ogni piattaforma.
Il maggiore punto di forza di questa architettura è la possibilità di immagazzinare degli storici di grandi dimensioni e di raccogliere nuovi dati anche in real-time. Complessivamente la varietà di informazioni fornite dalle API ufficiali permette numerosi spunti interessanti per l'applicazione di modelli di clustering e machine learning (ad esempio cluster analysis sullo stile di gioco o forecasting sui punteggi).
Per quanto concerne invece i giocatori con profili su entrambi le piattaforme, era purtroppo atteso che alcune corrispondenze sarebbero state irreperibili utilizzando lo username, a causa dell'impossibilità di effettuare ricerche sulla base della coppia nome-cognome. Qualora le API permettessero questo tipo di interrogazione, il numero di match aumenterebbe indubbiamente. In aggiunta, il meccanismo che è stato applicato per la verifica dell'identità fisica dei giocatori è piuttosto stringente: considerando la poca attenzione dedicata alla compilazione dei dati anagrafici dei profili, è possibile che alcuni match siano andati persi (ad esempio nell'eventualità di un profilo con lo stesso cognome ma nome di persona vuoto, nel dubbio viene scartata l'ipotesi di identità). In breve, a scanso di equivoci, per garantire il match viene richiesto che i due profili siano classificati o che sia stata caricata sulla piattaforma una prova ufficiale del loro titolo FIDE e che questo sia combaciante.
Un ultimo marigine di miglioramento già brevemente menzionato in precedenza è la scelta del RDBMS: la soluzione con SQLite è adatta ad un progetto di scope limitato, ma qualora si volesse scalare le funzionalità dell'architettura proposta, potrebbe essere più congeniale l'utilizzo di un RDMBS che si appoggi ad una componente server.

Bibliografia

[1] H. J. R. Murray, “A History of Chess” (Oxford University Press, 1913)

[2] YouGov, “Chess Redux” (AGON, 2012)

[3] Forbes, “Esports’ Biggest Winners In 2020: Prizes Plummet, Chess Leads Pack” (Forbes, January 2020)

[4] A. Cincotti Iida, Hiroyuki J. Yoshimura, "Refinement and Complexity in the Evolution of Chess" (2007)

[5] https://chesscom.readthedocs.io/en/latest/

[6] https://berserk.readthedocs.io/en/master/usage.htmlby-player