Caso di Studio:
Influenza delle Aperture negli Scacchi

Corso di Data Mining
Università degli Studi di Bari

Gianmarco Rutigliano 747002

July 21, 2023

Contents

1 Introduzione

Il gioco degli scacchi è considerato da molti il gioco più antico del mondo. Le sue origini esatte sono ancora oggetto di dibattito tra gli storici, ma si ritiene che abbia avuto origine in India intorno al VI secolo d.C. Nel X secolo gli scacchi arrivarono in Europa, portati dai commercianti arabi e durante il Rinascimento gli scacchi divennero popolari tra la nobiltà e furono adottati come un simbolo di intelligenza strategica. Il gioco si sviluppò ulteriormente con la creazione di aperture studiate e di tattiche elaborate.

Con l'avvento dell'era digitale, l'introduzione dei computer e dei programmi di scacchi ha permesso analisi sempre più approfondite e ha consentito di giocare contro avversari virtuali sempre più forti. La nascita di internet ha poi permesso ai giocatori di tutto il mondo di sfidarsi in tempo reale attraverso piattaforme di gioco online.

Uno degli aspetti più affascinanti dello studio degli scacchi è la sua insita complessità. Sebbene le regole siano semplici e discretamente esigue in numero, la semplice presenza di 32 pezzi distribuiti su 64 caselle da vita ad un numero inimmaginabile di possibili configurazioni della scacchiera, che supera facilmente il numero di atomi nell'universo. Escluso quindi a priori che sia possibile elencare forzatamente ogni configurazione, la formulazione di strategie e l'applicazione di esse all'interno di programmi capaci di simulare un giocatore di scacchi necessita di individuare elementi specifici e limitati nella loro complessità che siano in grado di predire l'esito di una partita senza calcolare ogni possibilità. Questo specifico caso di studio si concentrerà sull'analisi dell'apertura come uno di suddetti elementi.

Il concetto di **apertura** è un elemento basilare dello studio strategico degli scacchi. Con apertura si intende una serie di mosse compiute all'inizio della partita e che tendono a variare più o meno sensibilmente la configurazione della scacchiera. Ogni giocatore applica una precisa filosofia di gioco nella scelta dell'apertura e questo comporta che diverse aperture vengano studiate per portare a stati successivi del gioco profondamente diversi.

La gran parte delle aperture che nel tempo hanno ricevuto attenzione dai giocatori sono state classificate nel sistema ECO (Encyclopaedia of Chess Openings). La codifica prevede un codice alfanumerico di 3 simboli: una lettera dalla A alla E, che indica quale tra i cinque volumi che compongono

l'Enciclopedia contiene l'apertura in esame, e due cifre, di valore molto variabile, che indicano nello specifico la posizione dell'apertura nel volume di riferimento.[1]

L'obiettivo del presente caso di studio è quindi di valutare la rilevanza dell'apertura nell'andamento complessivo della partita e più nello specifico con che livello di precisione questa permetta di prevedere l'esito di una partita. Questa classificazione si servirà di dati aggiuntivi quali la differenza di abilità tra i giocatori (calcolata con diverse possibili statistiche di punteggio, quali l'ELO[2] o il GLICKO[3]) e la categoria di partita secondo il tempo concesso ai giocatori. A questo scopo verranno sviluppati diversi modelli predittivi per lo status di vittoria del giocatore bianco, di vittoria del giocatore nero o di pareggio con cui si conclude la partita.

Lo sviluppo del modello segue le fasi del CRISP-DM, il processo standardizzato di approccio alla Knowledge Discovery:

1. Business Understanding;
2. Data Understanding;
3. Data Preparation;
4. Modeling;
5. Evaluation;
6. Deployment;

2 Business Understanding

2.1 Obiettivi di Business

Nella letteratura non sono state individuate pubblicazioni rilevanti in questo ambito. Il task esplorativo considera il migliore di vari algoritmi predittivi per stabilire se le aperture siano un elemento sufficientemente significativo per prevedere l'esito della partita: per questa ragione non vengono stabiliti a priori dei valori previsti per la precisione e l'accuracy dei modelli. In base ai risultati, verranno ipotizzate delle conclusioni a riguardo della validità delle aperture come indice di predizione.

2.2 Obiettivi di Data Mining

Il modello rientra nella branca dei modelli di classificazione, con il compito di stabilire se, forniti in input dati relativi ai giocatori, all'impostazione iniziale della partita e alle prime mosse effettuate, questa si concluderà con una vittoria per il bianco, una vittoria per il nero o con un pareggio.

3 Data Understanding

3.1 Collect Initial Data

Per la realizzazione di questo task è stato individuato un dataset che presenta una mole appropriata di risorse.[4]

Il dataset è stato ottenuto dalla piattaforma di pubblicazione di dataset "Kaggle" e fornisce 20.058 istanze. Queste istanze sono state raccolte dal sito "Lichess.org", una piattaforma online che permette agli utenti di sfidarsi in partite di scacchi in tempo reale con altri utenti. Il sito fornisce una API che permette di acquisire partite effettuate da un particolare utente [5]: in questo modo è possibile raccogliere una grande quantità di dati sulle partite effettuate ogni giorno sul sito. Le partite si collocano temporalmente ai mesi giugno/agosto del 2017, momento della pubblicazione del dataset su Kaggle.

3.2 Describe Data

I dati presentano le seguenti feature:

Feature Categoriche:

- id: *string, len = 8*; indica un valore univoco assegnato a ogni partita;
- rated: *boolean*; indica se la partita sia stata svolta in modalità RATED o meno: le partite svolte in modalità RATED modificano il punteggio dei giocatori in base all'esito delle stesse;
- victory_status: {*mate, resign, draw, outoftime*}; indica il modo in cui si è conclusa la partita: mate indica che è stato raggiunto lo Scacco Matto; resign indica che un giocatore ha deciso di arrendersi; draw indica che la partita si è conclusa con un pareggio; outoftime indica che il giocatore sconfitto ha terminato per primo il tempo a disposizione per effettuare una mossa;
- winner: {*white, black, draw*}; indica chi dei due giocatori abbia vinto la partita, o se questa si sia conclusa con un pareggio; nel presente task, rappresenta la variabile target;
- increment_code: *string, del tipo 'A+B', con A e B numeri interi*; indica il tempo allocato per la partita: il numero A indica il tempo che ogni giocatore ha a disposizione all'inizio della partita (in minuti); il numero

B indica il tempo che ogni giocatore guadagna ogni volta che effettua una mossa (in secondi);

- white_id: *string*; indica il nome che l'utente che ha giocato con i pezzi bianchi ha scelto come proprio identificativo per il sito;
- black_id: *string*; indica il nome che l'utente che ha giocato con i pezzi neri ha scelto come proprio identificativo per il sito;
- moves: *string*; indica le mosse che sono state effettuate durante la partita in notazione algebrica[6];
- opening_eco: *string, del tipo [ABCDE]\d\d*; classifica l'apertura compiuta durante la partita nella codifica ECO;
- opening_name: *string*; indica il nome dell'apertura compiuta durante la partita;

Feature Quantitative:

- created_at: *integer*; indica il momento di avvio della partita in UNIX TIME [7];
- last_move_at: *integer*; indica il momento di conclusione della partita in UNIX TIME;
- turns: *integer*; indica il numero di mosse compiute dai giocatori prima che si concludesse la partita;
- white_rating: *integer*; indica il punteggio del giocatore che ha giocato con i pezzi bianchi[8];
- black_rating: *integer*; indica il punteggio del giocatore che ha giocato con i pezzi neri;
- opening_ply: *integer*; indica il numero di mosse in cui si è svolta la fase di apertura: al concludere di questa mossa la partita entra nel "mediogioco";

3.3 Verify Data Quality

3.3.1 Missing Data

Il dataset non presenta dati mancanti.

3.3.2 Data Coherence

Feature created_at, last_move_at Le feature created_at e last_move_at non presentano tutte la stessa rappresentazione. Sebbene indichino tutte una data in formato UNI, espressa in millisecondi:

- 9287 istanze seguono la notazione scientifica (ad esempio, la feature created_at presenta il valore "1.49118E+12" nella riga 2985);
- 10071 seguono la notazione estesa (ad esempio, la feature created_at presenta il valore "1492279628815" nella riga 13325).

Non si ha ragione di credere che il momento di inizio e di fine della partita siano utili al presente task, in quanto il risultato della partita non dovrebbe essere influenzato dal momento temporale in cui avviene e un task predittivo avrebbe utilità solo se venisse utilizzato prima che la partita si concluda: per queste ragioni la soluzione proposta per risolvere la discrepanza è di rimuovere completamente le suddette feature.

Nota: Si ritiene doveroso segnalare che la notazione scientifica impedisce nella maggior parte dei casi di ottenere un'informazione appropriata sulla durata della partita, in quanto i dati non presentano abbastanza cifre significative per discriminare accuratamente il momento dell'inizio e quello della fine; preso ad esempio il valore 1.50318E+12 della riga 57, la possibilità di arrotondamento comporta che il valore originale di entrambe le feature ricada in un range di circa 2.5 ore: se i numeri coincidono la partita può essersi svolta in un qualsiasi lasso temporale da 0 a 166 minuti; se non coincidono, il valore massimo raddoppia.

3.3.3 Duplicate Data

Il dataset presenta 945 righe duplicate.

Le righe duplicate indicano partite completamente identiche in ogni loro aspetto, inclusi gli ID. Per evitare di inquinare le operazioni successive, si è deciso di rimuovere tali istanze dal Dataset.

3.4 Explore Data

Di seguito vengono inserite informazioni notevoli riguardanti diverse feature.

Feature	Max	Min	Mode	Median	Mean	StdDev	UniqueVal
turns	349	1	45	55	60.5	33.5	211
white_rating	2700	784	1500	1567	1597.3	290.0	1516
black_rating	2723	789	1500	1563	1590.4	290.4	1521
opening_ply	28	1	3	4	4.8	2.8	23

Table 1: Misure statistiche delle variabili quantitative.

Feature	increment_code	opening_eco
1.	10+0: 7356 occ.	A00: 948 occ.
2.	15+0: 1258 occ	C00: 810 occ
3.	15+15: 821 occ	D00: 701 occ
4.	5+5: 723 occ	B01: 688 occ
5.	5+8: 678 occ	C41: 650 occ
N° valori diversi	400	365

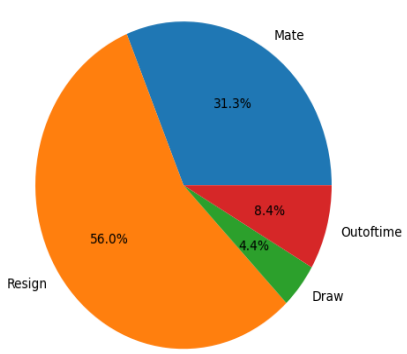
Table 2: Valori più comuni delle variabili categoriche.

Feature	opening_name
1.	Sicilian Defense: 349 occorrenze
2.	Van't Kruijs Opening: 342 occorrenze
3.	Sicilian Defense: Bowdler Attack: 290 occorrenze
4.	French Defense: Knight Variation: 260 occorrenze
5.	Scotch Game: 254 occorrenze
N° valori diversi	1477

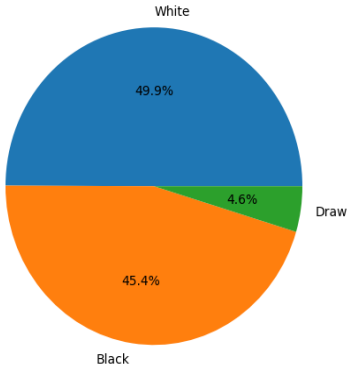
Table 3: Valori più comuni della variabile opening_name.

Feature	victory_status
mate	5974 istanze
resign	10695 istanze
draw	846 istanze
outoftime	1598 istanze
Feature	**winner**
white	9545 istanze
black	8680 istanze
draw	888 istanze

Table 4: Distribuzioni delle variabili victory_status e winner.



(a) Distribuzione di victory_status



(b) Distribuzione di winner

Nota: Il conteggio differente del valore draw tra le due feature dipende dal fatto che, per regolamento, se un giocatore termina il tempo concessovi, ma la disposizione dei pezzi è tale che per l'avversario sarebbe impossibile giungere a uno scacco matto con una successione qualsiasi di mosse legali, la partita si conclude in pareggio[9].

Feature increment_code: valori singoli Presi singolarmente, i valori di minuti iniziali e secondi aggiuntivi presentano le seguenti misure statistiche:

- Maximum value: 180; 180;
- Minimum value: 0; 0;

- Mode: 10; 0;
- Median: 10; 0;
- Mean: 13.8; 5.1;
- Standard deviation: 17.0; 13.8;
- Unique Values: 33; 32;

Nota: I valori minimi pari a 0 si spiegano quando il valore dell'altra tempistica diverso da 0: con 0 minuti iniziali e n secondi aggiuntivi, ogni mossa apporrà un tempo aggiuntivo di n secondi e ogni giocatore avrà a disposizione solo il tempo ottenuto in questo modo; con 0 secondi aggiuntivi ed m minuti iniziali, ogni giocatore avrà a disposizione m minuti per giocare la partita indipendentemente dal numero di mosse che compie e dal tempo utilizzato per compierle.

4 Data Preparation

4.1 Construct Data

4.1.1 Combine Data

La feature moves contiene informazioni su tutte le mosse compiute durante la partita. Per lo scopo di questo task è sufficiente che le mosse di apertura vengano utilizzate ed è necessario che ogni mossa oltre l'apertura sia ignorata. A questo scopo, la feature moves verrà modificata sfruttando la feature opening_ply: solo il numero di mosse indicato da questa feature verrano mantenute. In questo modo verrà anche ridotto il carico computazionale e di spazio del dataset.

4.2 Select Data

4.2.1 Feature Selection

Le seguenti feature sono state rimosse dal Dataset.

- Created_at e last_move_at: come valutato in precedenza, queste feature non sono utilizzabili;
- Turns e victory_status: queste feature rappresentano dati che non sono accessibili all'inizio della partita, per questo non è auspicabile utilizzarle nel task;
- Id, white_id e black_id: queste feature non portano conoscenza utile al task.

4.3 Format Data

Per semplificare le operazioni successive, la feature classe "winner" viene posta come ultima nel dataset.

Sono state effettuate le seguenti modifiche alle feature per agevolare il modeling:

- La feature increment_code, originariamente intrerpretata come stringa, è stata convertita in una coppia di variabili numeriche di significato analogo: il primo valore espresso dall'increment code è stato assegnato alla feature "starting_minutes", il secondo alla feature "additional_seconds".

- Le feature moves e opening_name presentano molti spazi e caratteri speciali: questi sono stati rimossi, o convertiti in underscore, per permettere la conversione in file .arff

5 Modeling

Per avere un'appropriata valutazione del task, verranno progettati due modelli di classificazione: un modello C4.5 e un modello Naive Bayes. Siccome il dataset presenta feature categoriche, si è scelto di utilizzare la piattaforma Weka per la costruzione dei modelli.

5.1 C4.5

L'algoritmo C4.5 si basa sul concetto di albero di decisione: l'algoritmo costruisce un albero esaminando le feature e stabilendo su di esse dei test. Questi test consistono in una condizione che permette di costruire delle partizioni del dataset: l'obiettivo è di formare partizioni via via sempre più uniformi nella distribuzione della classe. La scelta della feature avviene tramite la valutazione di metriche di entropia quali l'Information Gain e il Gain Ratio. L'algoritmo presenta anche il vantaggio di essere generalmente spiegabile, in base alla complessità dell'albero costruito.

5.2 Naive Bayes

L'algoritmo Naive Bayes segue il framework dei classificatori bayesiani, i quali basano la loro previsione sulla probabilità di ogni ipotesi considerata in base ai dati osservati. In particolare, il Naive Bayes sfrutta il teorema di Bayes assumendo l'indipendenza condizionale delle feature: il valore predetto dell'attributo target è quello che massimizza il prodotto tra la probabilità $P(v_j)$ del valore stesso e la produttoria delle probabilità condizionali $P(a_i|v_j)$ per ogni valore a_i che la tupla assume negli altri attributi.

5.3 Double Approach

Il dataset presenta un notevole sbilanciamento tra i valori possibili della classe winner. Il valore "draw" si verifica sensibilmente di meno rispetto ai valori di "white" e "black". Si propone di sviluppare in parallelo dei modelli a classe ternaria con valori possibili ["white", "black", "draw"] e dei modelli a classe binaria considerando solo i valori ["white", "black"]. Una volta valutati i modelli, i risultati verranno confrontati per verificare se ci siano differenze rilevanti tra di essi nella previsione di "white" e "black".

5.4 Approach I

Questa sezione segue la progettazione dei modelli a classe ternaria.

5.4.1 Model Evaluation

class	TruePos	FalsePos	TPRate	FPRate	Precision	Recall	F-Measure
white	5611	3946	0.588	0.412	0.587	0.588	0.587
black	5143	4343	0.593	0.416	0.542	0.593	0.566
draw	8	62	0.009	0.003	0.114	0.009	0.017
average	--	--	0.563	0.395	0.545	0.563	0.551

Table 5: Misure statistiche del modello C4.5

Il modello C4.5 ha riscontrato una percentuale di istanze classificate correttamente del 56.31% , con un MAE (Mean Absolute Error) pari a 0.3417 e un RMSE (Root Mean Square Error) pari a 0.4351.

class	TruePos	FalsePos	TPRate	FPRate	Precision	Recall	F-Measure
white	5248	3935	0.550	0.411	0.571	0.550	0.560
black	4722	4183	0.544	0.401	0.530	0.544	0.537
draw	90	935	0.101	0.051	0.088	0.101	0.094
average	--	--	0.526	0.390	0.530	0.526	0.528

Table 6: Misure statistiche del modello Naive Bayes

Il modello Naive Bayes ha riscontrato una percentuale di istanze classificate correttamente del 52.63% , con un MAE pari a 0.3385 e un RMSE pari a 0.4503.

5.4.2 Result

Il modello C4.5 presenta un F-score maggiore in riferimento alle classi principali, "white" e "black", e ha un punteggio di corretta classificazione maggiore, mentre il modello Naive Bayes effettua predizioni più equilibrate e predice la classe minoritaria "draw" in molte più istanze portando a molti più TP ma anche molti più FP: il dislivello dato dalle altre classi lo porta in ogni caso a dei valori medi inferiori.

5.5 Approach II

Questa sezione descrive i modelli a classe binaria. Ricordiamo che, una volta rimosse le istanze con "winner" = "draw", il dataset è composto da 11025 istanze.

5.5.1 Model Evaluation

class	TruePos	FalsePos	TPRate	FPRate	Precision	Recall	F-Measure
white	5998	4459	0.628	0.514	0.574	0.628	0.600
black	4221	3547	0.486	0.372	0.543	0.486	0.513
average	--	--	0.561	0.446	0.559	0.561	0.559

Table 7: Misure statistiche del modello C4.5

Il modello C4.5 ha riscontrato una percentuale di istanze classificate correttamente del 56.07% , con un MAE pari a 0.4646 e un RMSE pari a 0.5513.

class	TruePos	FalsePos	TPRate	FPRate	Precision	Recall	F-Measure
white	5553	3724	0.582	0.429	0.599	0.582	0.590
black	4956	3992	0.571	0.418	0.554	0.571	0.562
average	--	--	0.577	0.424	0.577	0.577	0.577

Table 8: Misure statistiche del modello Naive Bayes

Il modello Naive Bayes ha riscontrato una percentuale di istanze classificate correttamente del 57.66% , con un MAE pari a 0.4546 e un RMSE pari a 0.5137.

5.5.2 Result

In questo caso il modello Naive Bayes mostra risultati più promettenti quasi su tutta la linea. Anche qui si concentra di meno sulla classe di maggioranza, portando quindi statistiche peggiori per quanto riguarda "white" ma migliori su "black" al punto da raggiungere risultati medi migliori.

5.6 Improvement

Confrontando gli F-score delle due classi principali per i due modelli è possibile valutare se ci sia un miglioramento dovuto alla rimozione della classe minoritaria.

$$Improvement_{w,c4.5} = \frac{0.600 - 0.587}{0.600} = 2.1\%$$

$$Improvement_{b,c4.5} = \frac{0.513 - 0.566}{0.513} = -10.3\%$$

$$Improvement_{w,NB} = \frac{0.590 - 0.560}{0.590} = 5.1\%$$

$$Improvement_{b,NB} = \frac{0.562 - 0.537}{0.562} = 4.4\%$$

È possibile notare che rimuovere dal dataset la classe minoritaria comporta un miglioramento netto nel caso del classificatore bayesiano, mentre l'albero di decisione vede un lieve miglioramento rispetto alla classe "white" e un consistente peggioramento rispetto alla classe "black".

6 Conclusions

6.1 Evaluation

In conclusione, il presente task ha cercato di valutare la capacità predittiva delle aperture in una partita di scacchi riguardo al vincitore finale della partita. Tuttavia, dall'analisi dei risultati ottenuti utilizzando sia il classificatore bayesiano sia l'albero di decisione, risulta evidente che le prestazioni predittive di questi metodi sono piuttosto insoddisfacenti. Sebbene la percentuale di classificazioni corrette superi il caso, le misure statistiche indicano che i modelli non offrono risultati sufficientemente affidabili nel lungo periodo. Ciò suggerisce che i classificatori scelti, perlomeno nell'iterazione corrente, potrebbero non essere adatti a prevedere con precisione il vincitore di una partita di scacchi basandosi esclusivamente sulle mosse di apertura.

Si ritiene che il presente task non abbia prodotto il successo predittivo desiderato: tuttavia esso è solo il primo di una serie di ulteriori indagini nel campo dell'analisi degli scacchi. La ricerca del perfezionamento dei modelli predittivi negli scacchi contribuisce non solo al dominio dell'intelligenza artificiale, ma ha anche implicazioni pratiche per i giocatori, gli allenatori e gli appassionati che cercano intuizioni strategiche e opportunità di allenamento.

6.2 Future Improvements

Ci sono molteplici possibilità che possono essere esplorate per migliorare le performance dei modelli.

Il task potrebbe essere specializzato rispetto a diversi ambiti: a diverse fasce di abilità di gioco e di tempistiche adottate le strategie adoperate possono essere molto diverse, rendendo alcune aperture più efficaci. Dei modelli specializzati su questi sottoinsiemi di partite potrebbero portare a risultati differenti.

Un altro punto di miglioramento è senz'altro una codifica migliore di feature categoriche, quali le mosse, i nomi delle aperture e i loro codici ECO: con una gestione più adeguata di queste feature è possibile cogliere similarità che con l'approccio qui utilizzato potrebbero non essere state colte appieno.

Notes

[1]Un elenco completo delle aperture con rispettivo encoding ECO può essere trovato al seguente indirizzo:
https://www.365chess.com/eco.php

[2]https://www.chess.com/terms/elo-rating-chess

[3]https://en.wikipedia.org/wiki/Glicko_rating_system

[4]Il dataset e la documentazione dell'autore possono essere reperiti al seguente indirizzo:
https://www.kaggle.com/datasets/datasnaek/chess

[5]L'API è accessibile al seguente repository:
https://github.com/lichess-org/lila

[6]La notazione algebrica è una codifica che converte ogni mossa in una stringa che indica il pezzo che è stato mosso, la cella che ha raggiunto ed eventuali azioni rilevanti che la mossa comporta (presa, scacco):
https://en.wikipedia.org/wiki/Algebraic_notation_(chess)

[7]Il tempo in Unix time è definito al momento come il numero di secondi passati dalla data di Giovedì 1° Gennaio 1970 alle ore 00:00:00 UTC:
https://en.wikipedia.org/wiki/Unix_time

[8]Il sito Lichess.com utilizza come sistema di calcolo del punteggio il sistema GLICKO:
https://lichess.org/page/rating-systems;

[9]https://lichess.org/forum/game-analysis/draw-by-running-out-of-time