

Statystyka w sztucznej inteligencji i analizie danych – projekt

Igor Swat

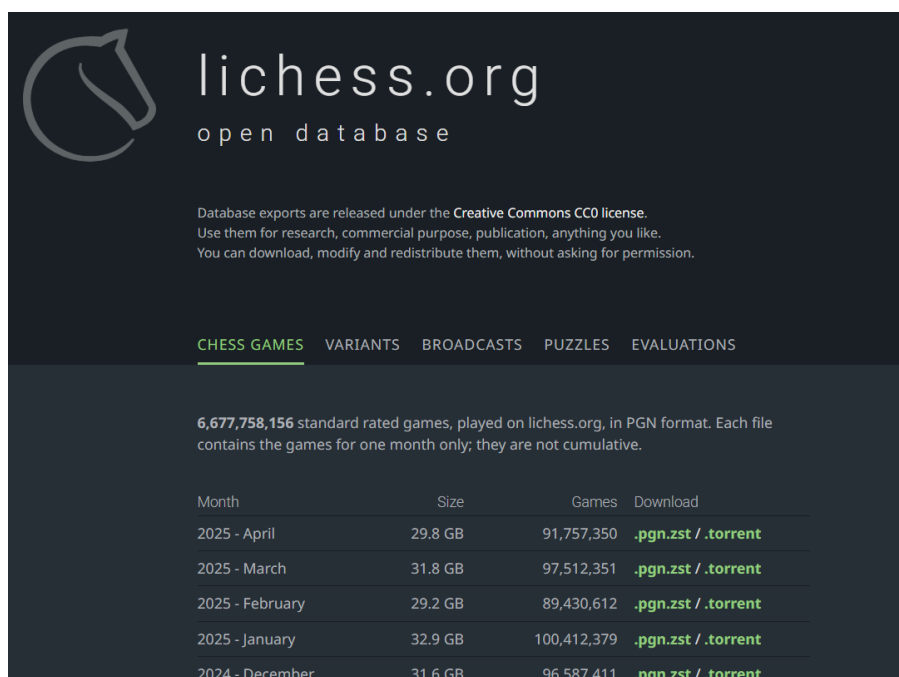
1. Zbiór danych

Tematem przewodnim projektu będzie analiza zależności rankingu gracza od przebiegu partii szachowej (w środowisku on-line), oraz próba **predykcji rankingu** na podstawie kilkunastu wybranych partii danego gracza.

Dla celów tego projektu, stworzony został od podstaw nowy zbiór danych, na podstawie publicznie dostępnej bazy danych partii w serwisie **lichess.org**. Poniżej opisane zostały poszczególne etapy konstruowania zbioru.

1.1 Dostęp do zbioru partii

W celu pozyskania danych do analizy partii szachowych wykorzystano publicznie dostępne archiwum gier z serwisu *Lichess*. Skupiono się na partiach rankingowych rozegranych w marcu bieżącego, co pozwoliło uzyskać dane o objętości około **30 GB**, zawierające ponad **90 milionów** gier. Dane zostały pobrane w formacie **PGN (Portable Game Notation)** z oficjalnej strony *Lichess* (<https://database.lichess.org>), gdzie publikowane są miesięczne zrzuty wszystkich rozegranych partii rankingowych.



| Month | Size | Games | Download |
|-----------------|---------|-------------|---|
| 2025 - April | 29.8 GB | 91,757,350 | .pgn.zst / .torrent |
| 2025 - March | 31.8 GB | 97,512,351 | .pgn.zst / .torrent |
| 2025 - February | 29.2 GB | 89,430,612 | .pgn.zst / .torrent |
| 2025 - January | 32.9 GB | 100,412,379 | .pgn.zst / .torrent |
| 2024 - December | 31.6 GB | 96,587,411 | .pgn.zst / .torrent |

Rys. 1 – baza danych serwisu *Lichess*

1.2 Parser PGN

Aby przetworzyć dane pobrane z serwisu *Lichess*, wymagane jest parsowanie dużej ilości zapisów partii w formacie PGN. Popularne istniejące rozwiązania w tym zakresie, takie jak parser udostępniany przez bibliotekę *python-chess*, są z reguły niewystarczająco wydajne do przetwarzania milionów partii w krótkim czasie (w tym konkretnym przypadku mówimy o około 97 milionach gier).

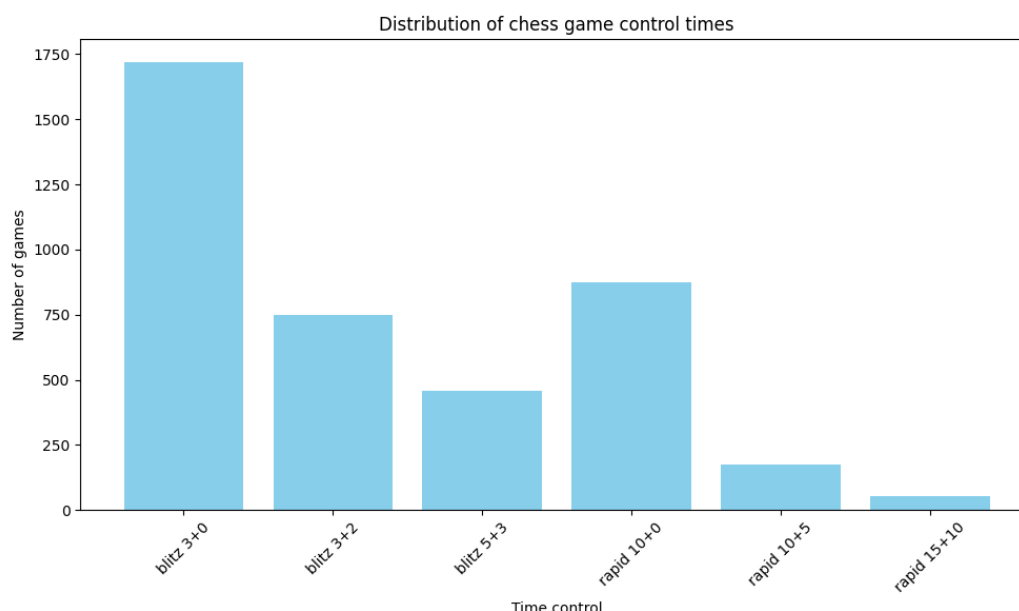
W celu rozwiązania tego problemu stworzona została minimalistyczna, wydajna implementacja parsera w języku C++, z bindingami do Pythona (*pybind11*). Całość rozwiązania znajduje się w repozytorium GitHub (https://github.com/IgorSwat/Elo-Predictions/tree/main/pgn_parser).

1.3 Selekcja i preprocessing

Proces tworzenia zbioru opiera się na następującym schemacie:

1. **Selekcja partii:** określenie cech, jakie musi spełniać partia aby była brana pod uwagę w dalszym przetwarzaniu
2. **Selekcja graczy:** odnalezienie określonej liczby graczy, którzy zegrali odpowiednią liczbę partii spełniających kryteria wynikające z punktu 1.
3. **Analiza i ewaluacja cech:** dla każdego z wybranych graczy obliczamy wartości docelowych cech zbioru, takich jak: ranking, cechy jakości gry, cechy zarządzania czasem

Jako partie docelowe zostały wybrane partie grane tempem szybkim, na 10 minut. Powodem była chęć dobrania jak najbardziej reprezentatywnego, a zarazem wystarczająco popularnego tempa gry.



Rys. 2 – histogram popularności poszczególnych temp gry na przykładowej próbie 10000 partii w serwisie *Lichess*.

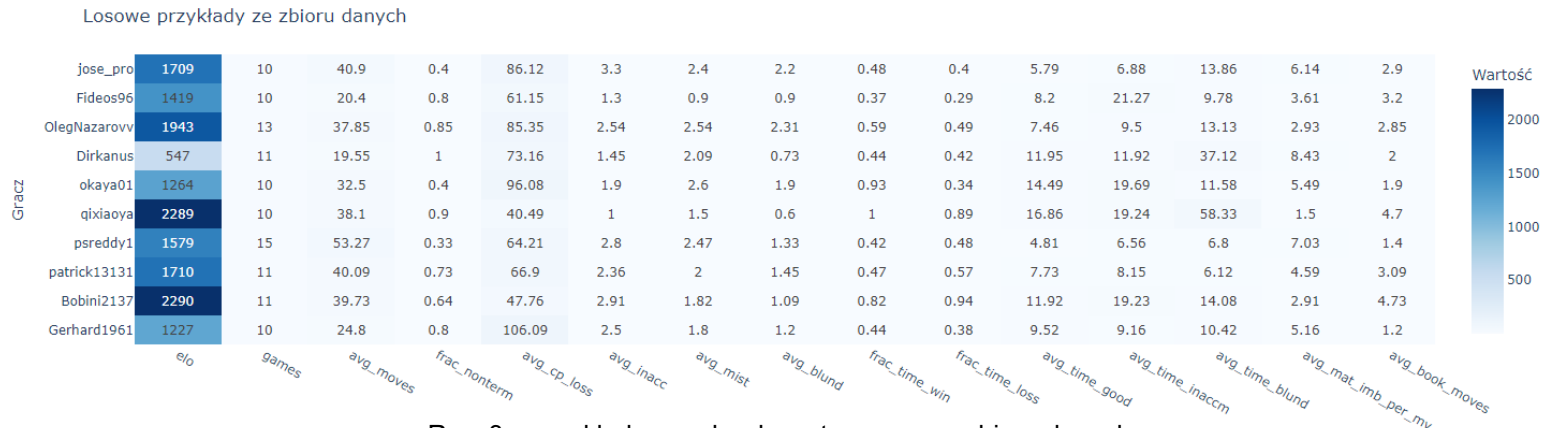
1.4 Finalny kształt zbioru

W utworzonym zbiorze (**data.csv**) każdy rekord odpowiada jednemu graczowi i zawiera szereg cech opisujących jego ogólną siłę gry oraz charakterystyczne zachowania podczas partii. Zmienną docelową w analizie jest **elo**, czyli ranking szachowy danego gracza, który odzwierciedla jego poziom umiejętności. Pozostałe kolumny pełnią rolę cech wejściowych w zadaniu regresji. Wszystkie cechy (poza **name** pełniącym rolę identyfikatora gracza) są **numeryczne**:

- **name** – identyfikator gracza (niewykorzystywany w uczeniu),
- **games** – liczba partii, na podstawie których wyliczono statystyki (nie mniej niż 10),
- **avg_moves** – średnia liczba ruchów na partię,
- **frac_nonterm** – ułamek partii zakończonych przez rezygnację lub obopólną zgodę na remis (wliczając potrójne powtórzenie pozycji) – czyli nie przez mata ani pata
- **avg_cp_loss** – średnia utrata centypunktów na ruch, czyli miara niedokładności względem silnika szachowego (*Stockfish 16*),
- **avg_inacc**, **avg_mist**, **avg_blund** – średnie liczby niedokładności, pomyłek i przeoczeń na partię,
- **frac_time_win**, **frac_time_loss** – średni udział czasu zużywanego przez gracza na ruchach w wygranych i przegranych partiach (w stosunku do pełnego czasu partii, czyli 600 sekund w przypadku partii 10-minutowej),
- **avg_time_good**, **avg_time_inaccm**, **avg_time_blund** – średni czas przeznaczony odpowiednio na dobre ruchy, niedokładności i błędy,
- **avg_mat_imb_per_mv** – średnia nierównowaga materiału (miara chaotyczności partii)
- **avg_book_moves** – średnia liczba ruchów granych zgodnie z teorią debiutową (książką debiutów).

Warto nadmienić, iż ponieważ jest to autorski zbiór, selekcjonowane cechy są absolutnie subiektywnym wyborem i w zależności od potrzeb, możliwa jest ich zmiana lub dodanie zupełnie nowych cech.

Zbiór danych liczy **15 000 przykładów**, z czego **10 000** wykorzystano do trenowania modelu, a **5 000** przeznaczono na jego testowanie. Celem modelu jest przewidywanie wartości ELO na podstawie statystyk stylu gry i jakości decyzji, co pozwala na analizę, które cechy najistotniej wpływają na poziom szachowy gracza.



Rys. 3 – przykładowe rekordy z utworzonego zbioru danych