Data Science Lessons From a Predictive Chess Model
(And glimpses into my other work)

Kenneth W. Regan\textsuperscript{1}
University at Buffalo (SUNY)

CSE501, 31 October, 2023

\textsuperscript{1}With grateful acknowledgment to co-authors—including Tamal Biswas now of RKMVERI—and UB’s Center for Computational Research (CCR)
A Predictive Analytic Model

Means that the model:

- Addresses a series of events or decisions, each with possible outcomes $m_1, m_2, \ldots, m_j, \ldots$
- Assigns to each $m_j$ a probability $p_j$.
- Projects risk/reward quantities associated to the outcomes.
- Also assigns confidence intervals for $p_j$ and those quantities.

**Example:** An insurance company may estimate that:

- The probability of a given house having flood damage in a 5-year period is 10% with “95%” confidence that it’s between 5% and 15%.
- This means is that out of 100 homes in similar and independent locations, they expect 10 to be flooded, with 95% confidence of no better than 5 but no worse than 15.
- Homes being close together does not affect the expectation but does widen the confidence interval.

**In my model, the $m_j$ are possible moves in chess positions.**
The model is based on a **utility function / loss function** in a standard way—except for being log-log linear, not log-linear (**why**).

The (dis-)utility comes from (**my heavily scaled version of**) **average centipawn loss** of the played move compared to (**what a powerful chess-playing program thinks is**) the best move.

**No chess knowledge other than the move values is input.**

The (only!) parameters trained against chess **Elo Ratings** are:

- **s** for “sensitivity”—strategic judgment.
- **c** for “consistency” in surviving tactical minefields.
- **h** for “heave” or “Nudge”—obverse to depth of thinking.

Trained on all available in-person classical games in 2010–2019 between players within 10 Elo of a marker 1025, 1050, . . . , 275, 2800, 2825. Wider selection below 1500 and above 2500.
How it Works

- Take \( s, c, h \) from a player’s rating (or “profile”).
- Generate probability \( p_i \) for each legal move \( m_i \).
- Paint \( m_i \) on a 1,000-sided die, \( 1,000p_i \) times.
- Roll the die.
- (Correct after-the-fact for chess decisions not being independent.)

The statistical application then follows by math known since the 1700s. (Example of “Explainable AI” at small cost in power.)

Validate the model on millions of randomized trials involving “Frankenstein Players” to ensure conformance to the standard bell curve at all rating levels.

See: Published papers and articles on Richard J. Lipton’s blog Gödel’s Lost Letter and P=NP which I partner.
Evaluation Criteria For the Model

1. Is it **safe**? That is, do its outputs conform to an expected (normal) distribution over populations that obey the “null hypothesis of fair play”?

2. Is it **sensitive**? And are its positive results clearly pertinent to the desired inferences?

3. How is it calibrated? Are the calibration—as well as positive results—**explainable**?

4. Can it be **cross-validated**? What sanity checks does it provide?

5. Does it model more than what its proximate application demands, so as to be robust against “mission creep”?
Demonstration (may skip)

- I will show data from the full model results, including the ongoing Tata Steel Asian Junior Championships.
- The model is trained to make MM\% (engine move-match) and ASD (scaled average centipawn loss) into unbiased estimators.
- Although the projections on the engine’s second and third moves are moderately out of true, the 4th moves onward agree closely, while projections of various levels of mistakes are in fair agreement.
- In 10–15\% of positions, the model projects an inferior move to be more likely than the engine’s favored move. This yields 2–3 percentage points gain in predicting the played moves, compared to “betting the favorite” move. See this GLL blog article.
- Advancing moves, capture moves, and moves with the knights are played far more often than the model projects.
- Is it better to leave these human tendencies as “theorems” of the model in its minimalist form, or alter projections after-the-fact to match them?
How Well Does It Work?

Internal evidence that it gives \((1 + \epsilon)\) relative error with \(\epsilon \approx 0.04\) for most rating levels. Means it supports betting on chess moves with only 5% “vig” needed to avoid arbitration. (SF11 issue corrected “by hand.”)
Text and Subtext

- **Text:** Despite being *severely underfitted*, the model works checkably well.

- **Subtext:** Many deployed models *satisfice*—

—designed toward one prime objective but don’t build in cross-checks or invest in the space of neighboring objectives.

**Nonreproducibility**, **Mission Creep**, and **Shifting Sands**. E.g., I do not reproduce the longer conclusions of this study.

- **Going back to my model**, since it is fundamentally incorrect regarding independence, the cross-checks are a vital basis.

- **Build not a Model but a Root System.**
The #1 scientific role I’ve played during the pandemic has been estimating the true skill growth of young players while their official ratings have been frozen. But this has perforce been post-normal science. My “back of the envelope” formula held up over two years with only one small revision for preteens. Larger revision in Oct. 2022 to curtail projections past Elo 2000 level. Would have been more “normal” if comprehensive studies of the career arcs (measured by Elo rating) of young players were to hand. Lack of such studies exposed by the controversy over Hans Niemann’s rise from 2465 Elo to 2700. Show this GLL article including example of Ms. Velpula Sarayu.
The article’s larger subject is a **drastic** proposal by US statistician Jeff Sonas—long used by FIDE—to overhaul chess ratings below Elo 2000—that is, for beginning and amateur players.

(This is on top of things I’ve been telling FIDE about ratings *above* 2000.)

My own work has been “tinged” by this issue.

A natural metric **apart** from both my model and Sonas’s domain cross-validates his observations and arguments.

I will now discuss some other applications that these solid foundations enable.
Hans Niemann: Platform or Plateau?

*Ceci n’est pas un plateau*

*(celui-là, oui)*
The Gender Gap in Chess

- Is clear: with Judit Polgar retired, there are no women in the top 100 by rating.
- Where/when does it begin?
- How should one begin to address this question?
- What data could corroborate a result—or a proposed explanation?
- Picture emerging from recent youth events...?
Computational Complexity

- The study of the time \textit{needed} to solve computational problems, and how much memory and other resources computers require.
- Largely independent of the computer model, beyond a fundamental divide into \textit{serial}, \textit{parallel}, and \textit{quantum}.
- Main technical achievement: the relation of computational problems by \textit{reducibility}.
- Main scientific surprise:

\begin{quote}
The many thousands of computational problems that have been studied in many disciplines, some for centuries, cluster into barely over a dozen equivalence classes under reducibility.
\end{quote}

- The biggest cluster is the class of \textbf{NP-complete} problems.
P=?NP and Worse

- **P**: problems with algorithms that solve them in **polynomial time**:
  
  As the size of the data doubles, the time needed goes up by at most a **linear** factor:  
  \[ t(n) = n^k \implies t(2n) \leq Kt(n), \quad K = 2^k. \]

- **NP**: “Nondeterministic” Polynomial Time: If you know a secret fact or guess a good answer, you can verify and **teach** it to someone in polynomial time.
- Example: Given a Boolean formula \( f \) like

  \[ f = (x_1 \lor \neg x_2) \land ((\neg x_1) \lor x_2 \lor x_3) \land ((\neg x_2) \lor (\neg x_3)), \]

  is there a way to make \( f \) true?

  - Called **Satisfiability** (SAT).
  - Equivalent to \( \neg f \) **not** being a **tautology**.
  - Is NP-complete, so \( \text{NP} = \text{P} \iff \text{SAT} \) belongs to \( \text{P} \).
  - We don’t even know whether SAT can be solved in **linear** time!
Factoring is among a handful of problems in NP not known to be complete or in P.

RSA security depends on it, so many want it to be hard.

But solvable in polynomial time by a quantum computer.

Textbook on quantum algorithms; blog series: Can QCs be Built?

Research on simulating quantum circuits by logic and algebra:

\[ p_1 = p_0 \text{ XOR} (y_0 \& x_1) \quad p_2 = p_1 \text{ XOR} (y_1 \& x_2) \quad v_0 = x_3 \text{ XOR} (y_0 \& y_1) \]
\[ p_3 = p_2 \text{ XOR} (y_2 \& y_0) \quad p_4 = p_3 \text{ XOR} (y_3 \& v_0) \quad v_1 = y_2 \text{ XOR} (y_3 \& y_1) \]