Modeling Realistic Placement Probabilities in Speedcubing Using Kernel Density Estimation (KDE)

Ryan Saito
Saint John's High School
Shrewsbury, MA 01545
saitor26@stjohnshigh.org

Abstract

This paper offers a simulation-based framework for predicting placements in official World Cube Association (WCA) competitions. Currently, the WCA uses psych-sheet rankings, which only present a competitor's best average of five (Ao5) and singular solve. Our goal is to create a simulation framework that uses recent performance to estimate realistic placement probabilities. We construct kernel density estimates (Gaussian KDEs with an adaptive bandwidth based on the coefficient of variation) for each competitor using their most recent twenty-five official solves, and optionally csTimer practice solves. Then, we use percentile sampling and bootstrap Ao5s to make synthetic solves across 100-1000 tournament iterations.

We also created an open-source app that shows each competitor's probability of advancing, their expected rank, and a KDE distribution with 95% confidence and prediction intervals. We tested this tool at the Saint John's Warm Up 2025, where it correctly predicted the podium of the competition with 80% accuracy (12 out of 15 places) and had 33% of the predictions exactly correct (5 out of 15 places). At the Rubik's World Championships in 2025, the tool also correctly predicted the entire podium for the 3x3 event. Practically, the tool lets competitors and organizers set expectations, evaluate consistencies, and make decisions about training, seeding, and round cutoffs based on the data. It is also accessible to spectators and enthusiasts.

1 Introduction

The world of speedcubing competitions gained popularity back in 2003 and has produced a wide range of statistics from its competitions. The World Cube Association (WCA), the official worldwide body responsible for maintaining detailed records of all sanctioned Rubik's Cube competitions, hosts competitions globally (World Cube Association et al., 2025).In an official WCA competition, competitors compete in structured rounds, trying to advance to subsequent rounds. Each round has five solves with a unique scramble that the competitors solve in the fastest time possible. A judge supervises the solves, and the competitor uses a regulated timer to record their times. After the five solves, the fastest and slowest times are removed, and the remaining three solves are averaged. This average is represented as an "Ao5" and is used as the basis for ranking the round. This process ensures legitimacy while also differentiating the times between competitors. Speedcubing is all about precision, where results can be determined by milliseconds. On the WCA competition page, competitors often refer to the psych sheet, a list that orders them by their fastest Ao5 times. While this system produces a simple ranking, it does not necessarily capture the true performance of a competitor. Variability between solves, outliers, or limited sample sizes can misrepresent a competitor's ranking. Instead of relying on a single best Ao5, we use KDE to model each competitor's full distribution of solve times and simulate Ao5 times with the variance that KDE provides. This makes our rankings reflect the consistencies of a competitor's performance, not how fast they were at one point.

Similarly, sports analytics has been widely used recently in understanding player or team performance. This raises the question of determining the true performance of a competitor, where we used predictive analytics and KDE to address this issue. The WCA has a database that includes a competitor's history in their solves. Users can also input their own solves at home, with the help of csTimer, a professional timing program designed for Rubik's Cube speedsolvers. Using this process, we produce a prediction based on a competitor's simulated data. This prediction helps us understand where competitors stand relative to each other, and allows us to analyze whether a competitor ranks higher or lower than they should.

1.1 Background

In sports analytics, researchers use various methods to predict the outcomes for competitive sports, such as probabilistic modeling and simulation techniques. De Angelis and Fontana (2024) explored the Elo rating of Chess and applied it to tennis through forecasting the winners of major tournaments. Their method uses this Elo rating to dynamically model a player's performance and evaluate the probability that each player will win the tournament. This method also has its faults since Elo rating doesn't capture a player's behavior because the rating itself is static. Similar to my approach, however,

their model yields probability distributions for final placements, which can be applied to analyze speedcubers.

Data scientists and statisticians have given speedcubing little recognition and research, leaving potential opportunities for statistical analysis in a sport that is determined by milliseconds. This project addresses that gap by applying advanced statistical techniques, such as KDE and bootstrapping, to model and predict competition results. By creating an open-source tool to accommodate the research, we broadened the accessibility of speedcubing analytics, giving competitors, organizers, and enthusiasts the ability to examine performance trends. This tool also provides competitors with realistic placement probabilities based on recent performance rather than all-time data. As in professional sports, where analytics can benefit a team's strategy and preparation heading into the next game, this clarity encourages data-driven training in speedcubing, which can potentially raise overall performance standards.

1.1.1 Kernel Density Estimation

We relied on KDE for this project because it take a small sample of data and creates a density distribution, which in this case is the simulations of a competitor's times (De Angelis and Fontana, 2024). Specifically, we applied Gaussian kernels in the kernel-density estimate, which helped for a unimodal distribution. Since a competitor's times will be most populated at a certain time, such as the mean of their recent times, this process creates a unimodal distribution, which suits the use of Gaussian KDE. The KDE function is represented by:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

where the Gaussian Kernel is

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

- h: the bandwidth (smoothing parameter)
- n: total number of observations
- $x_1, x_2, ..., x_n$: observed data points (solve times)

In addition, Sawczuk et al. (2022) applied KDE to the sport of rugby, particularly analyzing the spatial trends of attacking possessions. They analyzed and understood these trends, which helped them create tactical strategies. This approach was similar to ours, as they applied KDE to predict an outcome from their dataset.

1.1.2 Bootstrapping

Bootstrapping is a statistical resampling technique that enables the estimation of sampling distributions, confidence intervals, and model variability. In essence, bootstrapping involves estimating statistics using an existing dataset, which in this case is the official Rubik's Cube competition solves. We used bootstrapping because it addresses the gap in determining the certainty of a competitor's placement (STINE, 1989). With this, we generate empirical sampling distributions for the placement probabilities by repeatedly resampling the original data. In addition, bootstrapping computes confidence intervals and assesses the validity of simulation-based forecasts.

Bootstrapping has also been widely used in predictive modeling to reduce overfitting and improve model stability. Austin and Tu (2004) acknowledged the use of automated model selection methods, but concluded that overfitting and confounding variables were often found. Thus, they proposed combining bootstrap resampling and automated variable selection methods to develop parsimonious regression models and identify predictors that consistently appeared across resampled datasets. In their case study, they confirmed that bootstrapping improves model discrimination and stability compared to standard selection methods. Elaborating on their findings, we applied bootstrapping to competitive speedcubing to generate accurate performance predictions from variable and noisy solve time data.

Mooney (1996) applied bootstrapping in certain political science scenarios, explaining the importance of avoiding assumptions by only using the data given. In political science, Mooney explains that bootstrapping should be applied because researchers often face small samples and complex statistics. Since bootstrapping is a non-parametric method, he compared bootstrapping to other parametric methods and concluded that bootstrapping often gives wider, more realistic confidence intervals than parametric methods. Using this knowledge for speedcubing, bootstrapping illustrates the variability between competitors' solves more realistically. As well, Olsson et al. (2021) implemented bootstrapping in forecasting state-level elections, and found that Bayesian boostrapping had a lower error than traditional election forecasting methods, proving that bootstrapping is a more accurate way to estimate results.

2 Materials and Methods

2.1 World Cube Association Database

We used data provided by the WCA, which was helpful since it included detailed statistics for each competitor, such as their best single solve times and average solve times across multiple events. Through a rigorous validation process, the WCA can mark every solve official. During a competition,

an official timing device captures a competitor's time, and an overseeing judge verifies its validity before recording the time on the scorecard. Once an average is completed, a delegate double-checks it during and after a competition to identify and rectify any errors. After the delegate's confirmation, the WCA Results Team (WRT) verifies and publishes the competitor's new times on the WCA website.

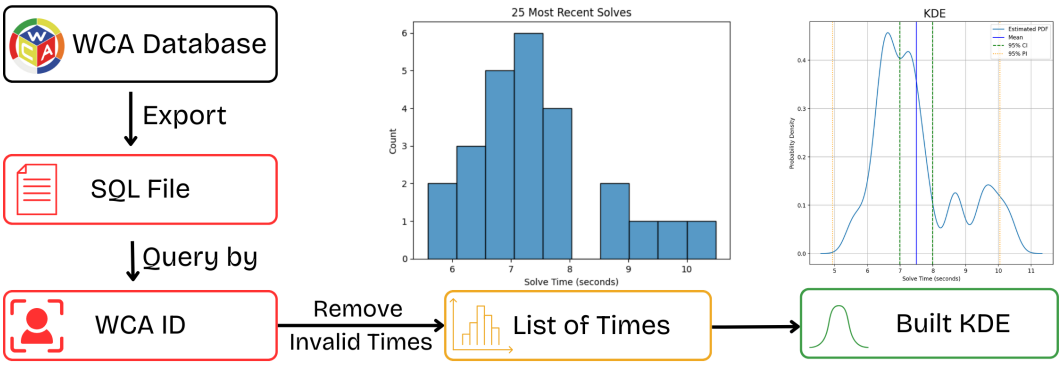

Figure 1: Process for extracting a list of solve times for a competitor using the WCA database.

Using the exported SQL file from the WCA, we first extracted the competitor's time. Then, we isolated each competitor's list of times by referencing their own unique WCA ID (e.g., 2018SAIT07) and then took their most recent solves. We used twenty-five solves, the most recent five rounds from competitions, because this number generated realistic performance models and did not overestimate or underestimate one's performance. After sorting the data, we removed any invalid times, such as DNFs and DNSs, because they altered the simulation results when not accounted for. Lastly, we used these solves to build each KDE, which ultimately yielded metrics such as the estimated ranking of the competitors, the probability of making the finals, and confidence and prediction intervals.

2.2 Integrating User-Recorded csTimer Solves

In addition to using official data from the WCA, csTimer is a professional timing program that is often used by speedcubers to track their times at home (Chen and Zhang, 2022). We included this dataset to improve accuracy and address the limitation of the WCA, since competitors may not regularly compete but often practice. As official solves depend on competitors attending a competition, csTimer provides a user-friendly way for speedcubers to submit personal times.

2.3 Kernel Density Estimation Implementation

To make KDE more sensitive to a competitor's variability, an adaptive bandwidth scheme was implemented that took into account the coefficient of variation of a competitor's times:

Using the coefficient of variation helps distinguish consistency between different competitors. For example, if one's coefficient of variation is low, a competitor's KDE will be a narrow graph with a

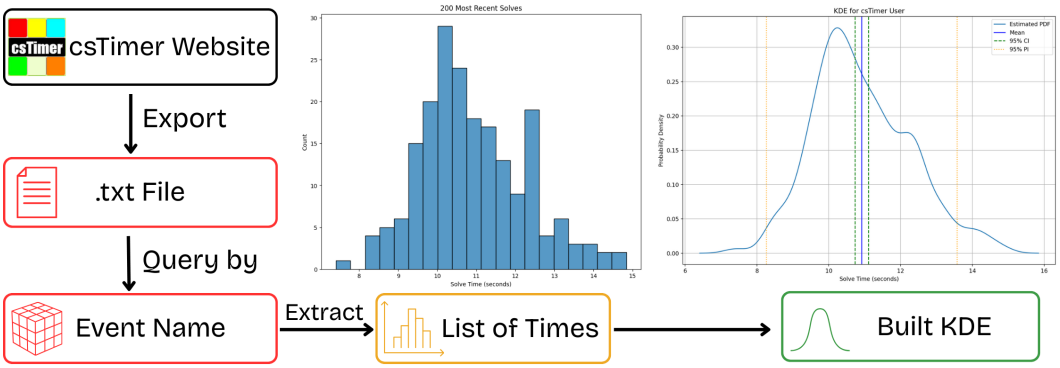

Figure 2: Process for extracting solve times from the csTimer platform. We created a custom function that parsed and extracted event-specific times from a .txt file that can be exported from csTimer.

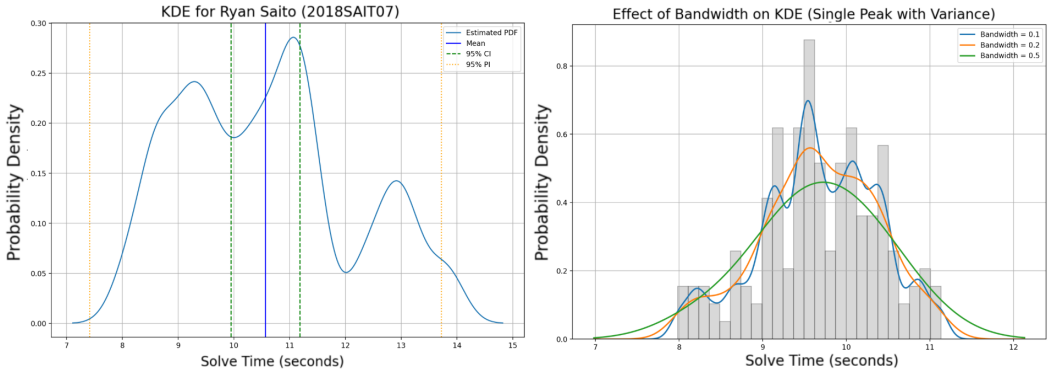

Figure 3: On the left is a KDE of simulated Ao5 times. Solve times were extracted from the WCA, and the KDE curve was constructed from the competitor's most recent 25 official solves. Peaks of a curve represent a strong likelihood, while the width reflects consistency. This visualization of probability illustrates a more nuanced performance prediction in comparison to looking at singular solve times. On the right is a comparison between different KDE bandwidths for a competitor. The histogram is the underlying data distribution with three different bandwidths: 0.1 (blue), 0.2 (orange), and 0.5 (green). The blue line consists of spikes that overfit due to a low sample size. The green line oversmooths the KDE, not showing a competitor's natural variance. The orange line, which is the bandwidth we used, preserves the central peak while capturing a realistic spread in the data.

sharp peak towards the mean, indicating that this competitor is consistent and predictable. In going through the process of bandwidth selection, base_bw was equal to 0.2 because it preserved the peaks and maintained the smoothness of the graph.

Adding confidence intervals and prediction intervals made it simpler to try to predict someone's next solve or next average. A confidence interval is a range where the true average of the competitor is expected, while a prediction interval is a range for the likelihood of a singular solve to occur. As shown in Figure 1, the confidence interval ranged from 9.95 to 11.18 seconds, and the prediction interval ranged from 7.41 to 13.72 seconds.

$$CI = \mu \pm z \cdot \frac{\sigma}{\sqrt{n}}$$

Algorithm 1: Adaptive Kernel Density Estimator (KDE)

procedure DESCRIBESOLVER($data$)
    $mean \leftarrow \text{average}(data)$
    $std \leftarrow \text{stdev}(data)$
    $cv \leftarrow \begin{cases} \dfrac{std}{mean}, & mean > 0 \\ 0, & \text{otherwise} \end{cases}$
    return $(mean, std, cv)$

procedure BUILDADAPTIVEKDE($data$)
    $(mean, std, cv) \leftarrow$ DESCRIBESOLVER($data$)
    $baseBW \leftarrow 0.2$
    $scaledBW \leftarrow baseBW + 0.3 \times cv$
    return $\text{GaussianKDE}(data, bwMethod = scaledBW)$

$$PI = \mu \pm z \cdot \sigma \cdot \sqrt{1 + \frac{1}{n}}$$

For the variables, μ is the sample mean, σ is the sample standard deviation, n is the number of solves in the sample, and z represents the z-score for 95% confidence level

2.4 Bootstrapping Implementation

Rather than resampling observed values directly, we fit a KDE to each competitor's most recent twenty-five solve times. Using this KDE, we generate synthetic values with a percentile sampling approach by transforming it into a cumulative distribution function (CDF):

Algorithm 2: Percentile Sampler from KDE

procedure BUILDPERCENTILESAMPLER($data, kde, percentile$)
    $x \leftarrow \text{linspace}(\min(data) - 1, \max(data) + 1, 1000)$
    $pdf \leftarrow kde(x)$
    $cdf \leftarrow \text{cumtrapz}(pdf, x, initial = 0)$
    $cdf \leftarrow \dfrac{cdf}{cdf[-1]}$ ▷ normalize to $[0, 1]$
    $F^{-1} \leftarrow \text{interp1d}(cdf, x)$
    $value \leftarrow F^{-1}\left(\dfrac{percentile}{100}\right)$
    return $value$

For each competitor's simulated Ao5, we sample five percentiles between zero and one hundred, map them to solve times through the KDE's interpolated CDF, and adjust using Gaussian noise. To account for the variability of a competitor's times, we introduced an outlier value when simulating a tournament:

Algorithm 3: Fast Simulation Tournament

procedure FASTSIMTOURNAMENT($sampler, baseNoise = 0.15, heavyTailChance = 0.05$)
  $p \leftarrow \text{rand}(5) \times 100$ ▷ random percentiles in $[0, 100]$
  $mask \leftarrow (\text{rand}(5) < heavyTailChance)$
  $samples \leftarrow [\, sampler(p_i) \,]_{i=1}^{5}$
  $noise \leftarrow \mathcal{N}(0, baseNoise, 5)$
  $values \leftarrow \begin{cases} \text{Uniform}(10, 16, 5), & mask = \text{true} \\ samples + noise, & mask = \text{false} \end{cases}$
  $sorted \leftarrow \text{sort}(values)$
  $meanVal \leftarrow \text{mean}(sorted[2:4])$ ▷ middle 3 values
  return $\text{round}(meanVal, 2)$

We repeat this simulation process for one hundred to five hundred tournament iterations. In each simulation, competitors compete against each other in a multi-round elimination format. Based on their Ao5 times, competitors advance through subsequent rounds until the simulation decides a winner. For instance, this is how a typical one-hundred-person competition works:

- **Round 1**: All competitors simulated; top 60 advance
- **Round 2**: New simulations; top 20 advance
- **Final**: Simulations rerun to rank final placements

Afterward, we store each simulation to create estimated probabilities of advancement and expected rankings. Fundamentally, this approach uses bootstrapping: we generate the distribution of potential placements by repeatedly sampling from the KDE's distribution.

3 Results

3.1 Case Study: Saint John's Warm Up 2025

To assess the accuracy of this tool, we simulated the Saint John's Warm Up 2025, a one-hundred-person competition that included both veteran and beginner solvers. For each competitor, we constructed a KDE using their most recent twenty-five solves. If a competitor did not have at least twenty-five solves, we used the maximum number of recent times available. With this setup, we generated one hundred simulations of possible average solve times and compared them across all competitors. By aggregating the placement distributions across the simulations, we created the estimated final rankings of that event. Since competitors competed in five separate events, we created five lists of predicted ranks and compared them to actual outcomes. In the end, these were the findings:

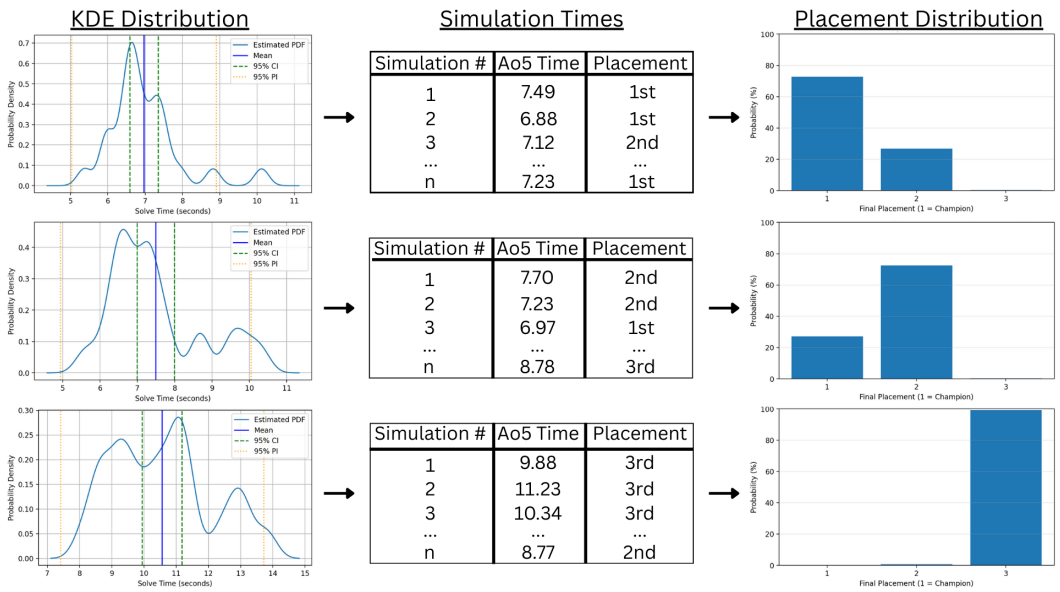

Figure 4: A bootstrapping workflow for prediction competition outcomes in speedcubing. This diagram illustrates the process of converting raw solve times from a KDE distribution to probability estimates. Individual competitor results are resampled with replacement to create simulated Ao5s, which are then compared among competitors to generate the placement predictions. This process is important because it shows both variability and uncertainty that is not revealed in psych sheet rankings.

Prediction Range	Correct / Total	Accuracy
Exact placement	5 / 15	33.3%
Top 3 (podium)	12 / 15	80%
Top 4	14 / 15	93.3%
Top 5	15 / 15	100%

Table 1: Prediction accuracy of the model compared to actual results across five events (15 podium spots total) for Saint John's Warm Up 2025. There are different placement thresholds, where exact placement signifies the prediction and real outcome to be the same, and other ranges compare the accuracy of the predictions and results from the competition. The high accuracy in predicting the podium and subsequent rankings demonstrates the tool's reliability in forecasting competitions in an official setting.

As a result, these results demonstrate that the simulation tool is especially reliable at identifying top-tier competitors, but exact placements are still difficult to forecast. This finding supports applying the tool in the real competition to help speedcubers and organizers better understand competitive expectations.

3.2 Case Study: Rubik's World Championship 2025

We simulated the Rubik's World Championship to further evaluate the predictive power of this tool. Composed of two thousand competitors and seventeen events, we decided to investigate the main event: the 3x3. Focusing exclusively on this event tests whether the simulation framework can

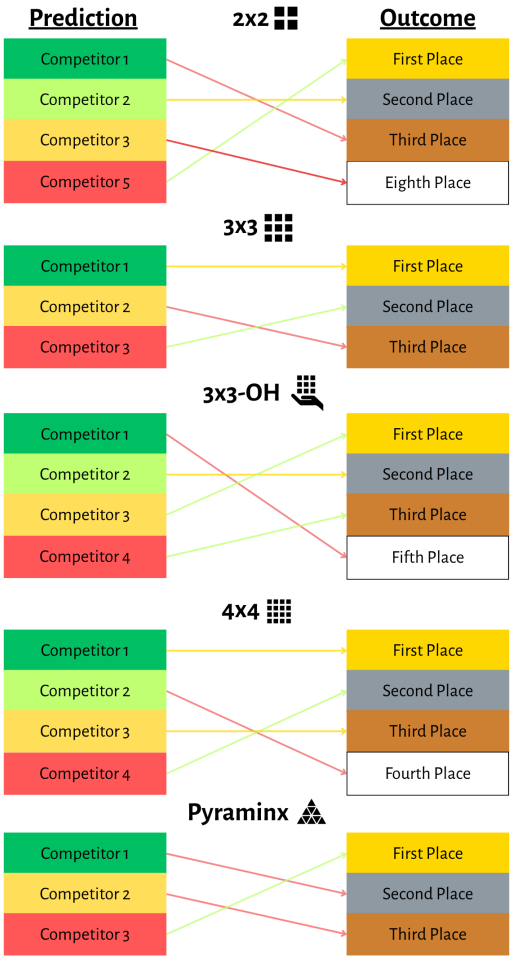

Figure 5: Comparison of predicted versus actual podium placements across five events at Saint John's Warm Up 2025: 2x2, 3x3, 3x3 One-Handed, 4x4, and Pyraminx. The arrows indicate the change from prediction to outcome, where a green arrow shows that the result was better than the prediction, a yellow arrow shows the prediction and result were the same, and a red arrow signifies the prediction was better than the result. This figure highlights the model's accuracy and shows discrepancies.

2x2 Prediction Accuracy
Exact matches: 0 / 3 (0%)
Within Top 3: 2 / 3 (66.7%)
Within Top 5: 3 / 3 (100%)

3x3 Prediction Accuracy
Exact matches: 1 / 3 (33.3%)
Within Top 2: 2 / 3 (66.7%)
Within Top 3: 3 / 3 (100%)

3x3 OH Prediction Accuracy
Exact matches: 1 / 3 (33.3%)
Within Top 3: 2 / 3 (66.7%)
Within Top 4: 3 / 3 (100%)

4x4 Prediction Accuracy
Exact matches: 2 / 3 (66.7%)
Within Top 3: 2 / 3 (66.7%)
Within Top 4: 3 / 3 (100%)

Pyraminx Prediction Accuracy
Exact matches: 0 / 3 (0%)
Within Top 2: 2 / 3 (66.7%)
Within Top 3: 3 / 3 (100%)

Table 2: Prediction accuracy for each event (2x2, 3x3, 3x3 One-Handed, 4x4, and Pyraminx) at Saint John's Warm Up 2025. Accuracy is reported for exact placement, and either Top 2, 3, 4, or 5 thresholds. The results emphasize the consistency in predicting the Top 3-5 placements with high certainty, as predicting the exact ranking was inconsistent.

forecast precise outcomes at the highest level of competition. Each competitor's KDE distribution was modeled using twenty-five solves and generated one thousand simulations to ensure an accurate placement probability estimate.

After comparing the simulated rankings with the official results, the tool predicted the podium with complete accuracy, showing its ability to identify consistent and fast competitors under the high-stakes environment. We used this case study to suggest that KDE-based simulation methods can yield reliable predictions, given the large dataset of speedcubers.

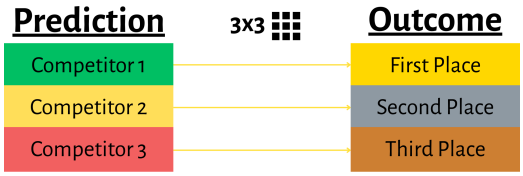

Figure 6: Predicted versus actual podium placements for the 3x3 event at the Rubik's World Championship 2025. The tool correctly identified the champion, runner-up, and third-place finisher. The yellow line shows that the prediction was the same as the result.

3x3 Prediction Accuracy
Exact matches: 3 / 3 (100%)
Within Top 2: 3 / 3 (100%)
Within Top 3: 3 / 3 (100%)

Table 3: Prediction accuracy table for the 3x3 event at the Rubik's World Championship 2025. The tool's predictions were 100% correct, showing its ability to predict competitor performance at the highest level of international competition.

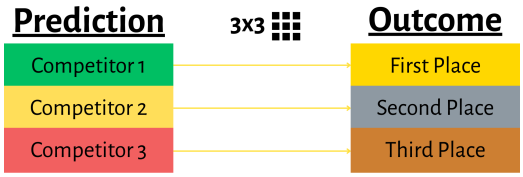

Figure 7: Predicted versus actual podium placements for the 3x3 event at the Rubik's World Championship 2025. The tool correctly identified the champion, runner-up, and third-place finisher. The yellow line shows that the prediction was the same as the result.

3x3 Prediction Accuracy
Exact matches: 3 / 3 (100%)
Within Top 2: 3 / 3 (100%)
Within Top 3: 3 / 3 (100%)

Table 4: Predicted versus actual podium placements for the 3x3 event at the Rubik's World Championship 2025. The tool correctly identified the champion, runner-up, and third-place finisher. The yellow line shows that the prediction was the same as the result.

4 Discussion

4.1 Using the App

Our simulation framework produced detailed probabilistic estimates of competitor performance in speedcubing tournaments. The results demonstrated that psych sheet rankings, based on a competitor's best official average, often fail to capture the true likelihood of success in competition.

We ran the simulations using a KDE-based percentile sampling strategy. For each tournament, we constructed twenty-five-solve KDEs for each competitor, and bootstrapped five synthetic solves to create an Ao5. When creating KDEs from csTimer solves, we used two hundred solves to show consistencies in solving. The metrics computed for each competitor were: mean (μ), standard deviation (σ), the coefficient of variation ($\frac{\sigma}{\mu}$) as a consistency metric, 95% confidence intervals (CI), 95% prediction interval (PI), and simulated estimated rankings.

As well, this open-source tool is now made available for the public, where competitors, organizers, and those interested can use the tool in order to predict a competition or competitor.

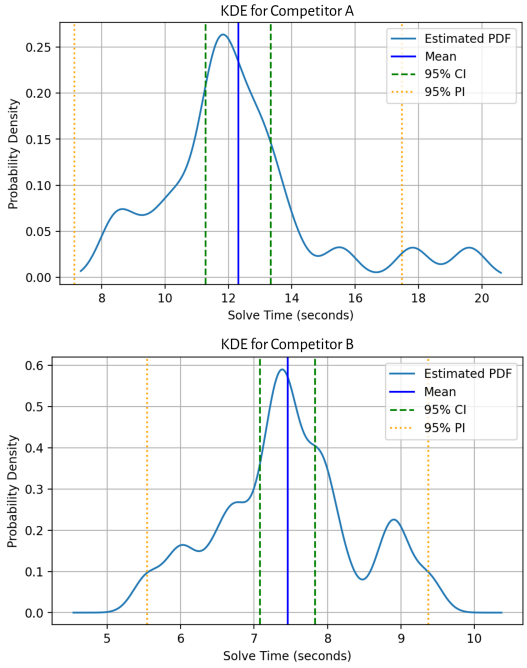

Figure 8: This figure compares the consistencies between competitors, where Competitor A is an inconsistent solver and Competitor B is a professional speedcuber. The horizontal axis of the KDE plot is the range of solve times in seconds, while the vertical axis represents the relative likelihood of a solve time. The difference in the horizontal and vertical axes scales is substantial. Competitor A's highest peak density of around 0.25 and variance of twelve seconds shows high variance. Competitor B's highest peak density of around 0.6 and variance of five seconds reflects lower variability and higher stability across times.

Metric	Competitor A
μ	12.31
σ	2.63
CV	0.214
95% CI	(11.28, 13.34)
95% PI	(7.14, 17.48)

Metric	Competitor B
μ	7.46
σ	0.96
CV	0.129
95% CI	(7.08, 7.83)
95% PI	(5.54, 9.37)

Table 5: This table summarizes the statistics from Figure 6. Competitor B is a more consistent solver than Competitor A, given the lower standard deviation and CV. These variables measure consistencies and relative variability. As well, a smaller difference in PI and CI for Competitor B reflects a higher predictability in performance, which translates into consistency in future solves.

4.2 Limitations

Three major issues threaten the accuracy of the tool's estimated rankings. The first issue is the limited sample size that comes with the competitors. Not all competitors have twenty-five official solves, which may lead to inaccurate KDE distributions; lowering this threshold of twenty-five solves further increases inaccuracy. The second concern is the effect of outlier times on the KDE distribution. When testing out the tool, a competitor that only had one solve far away from the mean produced averages that didn't represent their true performance. This variance ultimately distorted their estimated results, projecting them better or worse than they should have been. Lastly, the tool does not adjust for the time elapsed since a competitor's last official solve. This gap negatively affects the performance distribution because it relies on outdated data. As a result, the KDE may fail to reflect a competitor's performance and produce inaccurate predictions.

5 Conclusion

By analyzing the consistencies of speedcubing competitors using predictive analytics, KDE, and bootstrapping, this study demonstrates that those with the best Ao5 times are not necessarily the best performers in competition; competitors with more stable distributions often outperformed faster, but more variable, solvers. This supports our goal of examining competitors' recent performance to provide a more accurate estimate of their performance in competition by simulating Ao5 outcomes from each solver's KDE and producing placement probabilities. With the help of an open-source, transparent, and replicable tool, we used real competition data to provide competitors, organizers, or spectators with realistic insights into how competitors will perform, surpassing the analysis of psych sheets that account for only the best Ao5 and ignore variance. At the Saint John's Warm Up 2025, we evaluated this tool and demonstrated its ability to predict placements, with an accuracy of 80% in predicting the competitors in the podium.

6 Acknowledgments

I wanted to thank the speedcubing community for making this all happen. When attending competitions, I received terrific feedback from fellow competitors, and I am grateful to them for helping with my research and predictor tool. As well, I appreciate the volunteers who keep the WCA database up and running, as that was the main database I used for my research.

References

Austin, Peter C. and Jack V. Tu. "Bootstrap Methods for Developing Predictive Models." *The American Statistician* 58 (2004): 131–137.

Chen, Shuang and Yue Zhang. 2022 "csTimer.".

De Angelis, Luca and Saverio Fontana. "Monte Carlo meets Wimbledon: Elo-based simulations for predicting tennis tournament winners." (11 2024).

Mooney, Christopher Z. *Bootstrap Statistical Inference*. Sage Publications, 1996.

Olsson, Henrik, Wändi Bruine de Bruin, Mirta Galesic, and Drazen Prelec. 2021 "Combining Survey Questions with a Bayesian Bootstrap Method Improves Election Forecasts.". Preprint.

Sawczuk, Thomas, Anna Palczewska, Ben Jones, and Jan Palczewski. "Use of Kernel Density Estimation to understand the spatial trends of attacking possessions in rugby league." (2022).

STINE, ROBERT. "An Introduction to Bootstrap Methods: Examples and Ideas." *Sociological Methods & Research* 18 (1989): 243–291.

World Cube Association, Abdullah Gulab, Blake Thompson, Dan Smith, Kerrie Jarman, and Nick Silvestri. 2025 “World Cube Association: Official Results and Rankings Database.”. Governing body for official twisty puzzle competitions worldwide.