Predicting NCAA Tournament Outcomes with Feature Engineering and Data-Driven Modeling

Omri Tzur
omritzur@mail.tau.ac.il

Tim Kushmaro
timk@mail.tau.ac.il

Ben Cohen
benc3@mail.tau.ac.il

github.com/BenC6116/March_Madness_2024

May 7, 2025

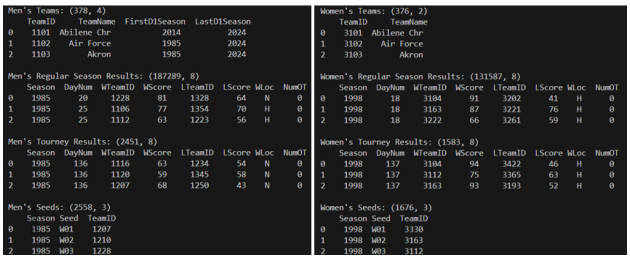
1 Introduction

In the March Machine Learning Mania 2024 competition, we predicted outcomes of the NCAA men's and women's basketball tournaments using only historical data. Our pipeline - from exploratory analysis and feature engineering to model training with rolling window cross-validation-focused on uncovering performance signals beyond basic seedings or win–loss records. By creating comparative features such as scoring margins, Elo ratings, and GLM-based team quality, we built gender-combined models that produced strong probabilistic predictions. This report details our end-to-end approach for generating accurate, generalizable forecasts.

2 Exploratory Data Analysis

2.1 Data Sources and Selection

We used NCAA datasets including regular-season and tournament results, team identifiers, and seedings. These covered both men's and women's teams over multiple seasons and served as a consistent base for modeling.



Men's Teams: (378, 4)

	TeamID	TeamName	FirstDISeason	LastDISeason
0	1101	Abilene Chr	2014	2024
1	1102	Air Force	1985	2024
2	1103	Akron	1985	2024

Men's Regular Season Results: (187289, 8)

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
0	1985	20	1228	81	1328	64	N	0
1	1985	25	1106	77	1354	70	H	0
2	1985	25	1112	63	1223	56	H	0

Men's Tourney Results: (2451, 8)

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
0	1985	136	1116	63	1234	54	N	0
1	1985	136	1120	59	1345	58	N	0
2	1985	136	1207	68	1250	43	N	0

Men's Seeds: (2558, 3)

	Season	Seed	TeamID
0	1985	W01	1207
1	1985	W02	1210
2	1985	W03	1228

Women's Teams: (376, 2)

	TeamID	TeamName
0	3101	Abilene Chr
1	3102	Air Force
2	3103	Akron

Women's Regular Season Results: (131587, 8)

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
0	1998	18	3104	91	3202	41	H	0
1	1998	18	3163	87	3221	76	H	0
2	1998	18	3222	66	3261	59	H	0

Women's Tourney Results: (1583, 8)

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
0	1998	137	3104	94	3422	46	H	0
1	1998	137	3112	75	3365	63	H	0
2	1998	137	3163	93	3193	52	H	0

Women's Seeds: (1676, 3)

	Season	Seed	TeamID
0	1998	W01	3330
1	1998	W02	3163
2	1998	W03	3112

Figure 1: Data basic description

2.2 Data Preparation

Key steps included:

- Parsing textual seeds (e.g., "W16a") into numeric form.
- Validating completeness of essential fields (scores, IDs, seeds).
- Aggregating team-level season stats: win percentage and scoring margin.
- Keeping men's and women's data separate just for the initial data analysis.

2.3 Visual Summary of Tournament Team Strengths

Tournament teams consistently showed higher win rates and margins than non-qualified teams, confirming their value as baseline performance indicators.

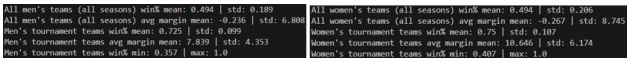


Figure 2: Teams differences

2.4 Seed Relationship and Correlations

We further examined how seed rankings relate to team performance.

- Lower seeds (i.e., better ranked) correlate with stronger season performance.
- More expressive features were needed to encode team strength and matchup quality.

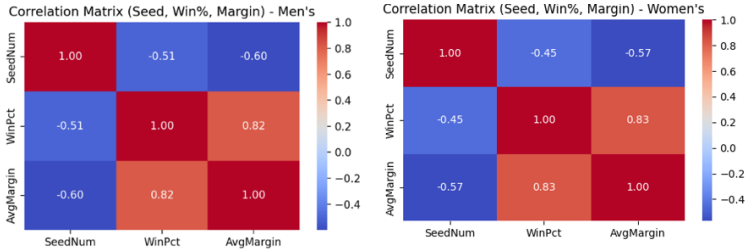


Figure 3: Correlation matrix showing relationship between seeding and team performance metrics.

2.5 Extended Stat Insights

We analyzed game-level box score statistics to better understand what drives outcomes. Each stat is briefly explained below for a general audience:

- Assist Difference (WAst LAst): Reflects team coordination and playmaking ability. Strong positive values aligned with winning.
- Turnover Difference (WTO LTO): Indicates ball control. Fewer turnovers often meant fewer wasted possessions and stronger game management.
- Rebound Difference: Captures physical dominance and second-chance scoring potential.
- Free Throw Attempt Difference (WFTA LFTA): Proxy for drawing fouls-teams attacking the basket more tended to win.
- 3-Point % Advantage: Measures long-range shooting accuracy. Useful in specific matchups, though less consistent overall.
- Offensive Efficiency (Points per Possession): Summarizes how effectively a team converts opportunities. One of the strongest signals.
- Foul Differential: Often correlated with losing, as teams behind tend to foul more in late-game situations.

  Visualizations available at via GitHub repo, in the eda notebook.

3 Feature Engineering

Guided by the findings from EDA, we constructed a matchup-level dataset in which each row represents a tournament game between two teams. Our feature engineering process was split into two parts: basic comparative indicators and advanced performance metrics. We implemented all of this in a combined preprocessing.py module pipeline, which all of the different models used.

3.1 Core Comparative Features

We began by computing regular-season statistics for each team, including:

- Win percentage (`WinPct`)
- Average scoring margin (`AvgMargin`)
- Numeric seed value (`SeedNum`)

For each tournament matchup, we then constructed the following comparative features:

- **SeedDiff** = Team2Seed - Team1Seed
- **WinPctDiff** = Team1WinPct - Team2WinPct
- **AvgMarginDiff** = Team1AvgMargin - Team2AvgMargin

These relative features are simple yet effective indicators of expected performance differences. They normalize away team identities and absolute values, enabling generalization across seasons and matchups.
To address team order bias (Team1 vs. Team2), we applied a **symmetrization strategy** – duplicating each game with reversed team roles and negated feature values. This doubles the training data and prevents the model from overfitting to input ordering.

3.2 Advanced Performance Metrics: Elo and GLM-Based Quality

To enrich our models with deeper insights into team performance, we incorporated two advanced, data-driven features: **Elo ratings** and a **GLM-based quality score**.

Elo Ratings. We implemented a custom Elo rating system using historical regular-season games. This method updates each team's rating after every game based on the outcome and the opponent's strength. The Elo rating offers a dynamic, cumulative strength measure that adjusts over time.

For each tournament matchup, we calculated:

- **Team1Elo**, **Team2Elo** - Elo ratings prior to the tournament
- **EloDiff** = Team1Elo - Team2Elo

We observed that higher Elo values and larger positive differences correlated strongly with tournament wins. This is demonstrated in our distribution plots, where winning teams show a consistently higher average Elo.

GLM-Based Team Quality. We estimated each team's "quality" by fitting a Gaussian Generalized Linear Model (GLM) on point differentials:

$$\text{PointDiff} \sim C(\text{Team1ID}) + C(\text{Team2ID})$$

From this model, we extracted each team's coefficient, debiased for opponent strength, This yielded:

- **Team1Quality**, **Team2Quality** - GLM-predicted team strengths, **QualityDiff** = Team1Quality - Team2Quality

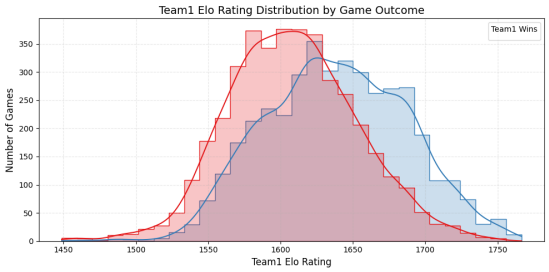


Figure 4: Team1 Elo Rating Distribution by Game Outcome

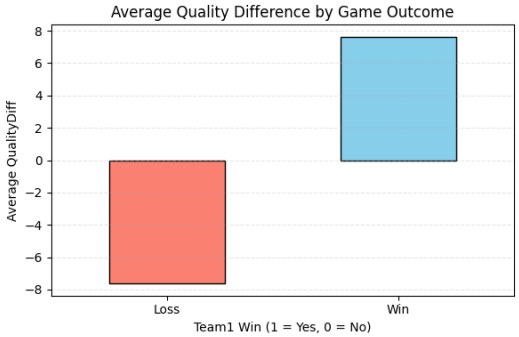


Figure 5: Average Quality Difference by Game Outcome (GLM)

These advanced features - both dynamic (Elo) and statistical (GLM) - added significant predictive signal and were retained throughout all modeling phases, including the final ensemble.

4 Modeling Approach

To ensure consistency across different models we preformed rolling-window cross-validation over 2011–2021 (excluding 2020), training each model on data up to year t–1 and testing on year t. We computed the average Brier score across these test years our primary metric, since the competition uses Brier score-and applied this procedure uniformly to all models. Finally, we held out 2022 and 2023 as untouched validation sets for tuning and evaluating the final ensemble.

4.1 Baseline Model: Logistic Regression

Following steps 1-3, we began by using Logistic Regression, a widely-used classification method valued for its simplicity and interpretability. As a baseline, LR allowed us to test the predictive signal of our early features while providing a benchmark for more complex models.

Initial Model Setup. Our first implementation included three core features derived from reg-season data:

- **SeedDiff** – the difference in tournament seedings
- **WinPctDiff** – the difference in regular-season win percentage
- **AvgMarginDiff** – the difference in average scoring margins

To eliminate ordering bias, we applied a **symmetrization strategy**: each game was duplicated with flipped team roles and negated feature values. We also normalized all features using MinMax scaling to support stable convergence.

Feature Expansion and Performance Tuning. We began with a broad pool of candidate predictors and used cross-validation (optimizing for Brier score) to identify the most informative subset. Among the contenders, **EloDiff** emerged alongside the three original features to form the optimal four-feature model. This quartet consistently achieved the lowest Brier score of all combinations tested.
Finally, to ensure temporal robustness, we retrained the model using various season cutoffs and found that restricting the data to 2011 onward further improved calibration and reduced noise.

Model Optimization. We used the Optuna framework to tune regularization parameters, testing both L1 and L2 penalties. The best configuration used L2 regularization with a penalty parameter of $C \approx 17.5$. We evaluated performance using log loss, accuracy, and Brier score, with the final tuned model achieving a Brier score of `0.174` on the test set years 2011-2021.

4.2 XGBoost Tournament Predictor

XGBoost is the workhorse of our bracket–forecasting system: a regularised, gradient–boosted tree ensemble that excels on heterogeneous, tabular data. By design it can ingest hundreds of correlated, non-linear signals and still return well-calibrated win probabilities-exactly what a single–elimination tournament demands.

Feature engineering.

- **Season-long box-score differentials** (FGM, 3PT, rebounds, assists, turnovers, *etc.*) averaged separately for offence and defence.
- **Schedule-adjusted team quality** from a Gaussian GLM on point differential (PointDiff $\sim$ Team_1 + Team_2), giving a debiased "true strength" coefficient.
- **Momentum** – win–rate in the final 14days (`Last14WinRate`).
- **Seed** handcrafted indicator.

All team–level features are converted to symmetric match-up variables via simple differences ($\Delta = \text{Team}_1 - \text{Team}_2$). The initial "kitchen-sink" matrix contained $\sim$45 candidate columns.

Data-driven feature reduction. We trained a draft model, ranked variables by mean absolute SHAP value, we took the top 10 most indicative features and applied:

1. **Redundancy pruning** – drop any pair with correlation $|\rho| > 0.85$, keep the higher-ranked term.
2. **Sequential forward selection** – add the feature (pair or triple) that lowers rolling-window and trying different combinations and sophisticated techniques to avoid local minimum.

The loop converged on 13 variables ($\approx 30\%$ of the original set)

Time-aware training protocol. For every season $s \in [2011, 2021]$ we trained on seasons $< s$ and predicted season s ("rolling-window CV"). Using *all* historical seasons only! in this model and in all models.

Hyper-parameter search. An `Optuna` Bayesian sampler explored

`max_depth`	$\{3, 4\}$
`learning_rate`	$\{0.065, 0.05, 0.035, 0.02\}$
`min_child_weight`	$\{30, 40, 50\}$
`gamma`	$\{10, 20\}$
`subsample`	$\{0.35, 0.50\}$
`colsample_bytree`	$\{0.70, 0.80\}$
`reg_alpha`, `reg_lambda`	$\{0, 10^{-4}, 10^{-3}\}$

Optimisation target: mean Brier score over rolling-window folds.

Calibration. Raw probabilities already tracked empirical win rates closely (10-bin reliability curve); nonetheless we fitted a season-aware logistic calibrator. Isotonic and Platt scaling were also tested but did not improve score, so the logistic layer is our default.

Performance snapshot. Across the 2011–2021 test tournaments the final model achieved:

- **Mean log-loss:** 0.524
- **Mean Brier:** 0.176
- **Accuracy:** 0.727

While marginally behind our tuned random forest in raw accuracy, XGBoost provides crucial complementary diversity and the clearest *local* explanations (via SHAP), making it a cornerstone of the ensemble.

4.3 Random Forest Classifier

As a lightweight yet robust alternative, we implemented a Random Forest (RF) classifier. This ensemble of decision trees offers fast training and solid generalization, making it ideal for compact modeling.

Data Source and Feature Set. Unlike XGBoost, the RF model relied only on compact regular-season and tournament data. We engineered a concise feature set:

- **SeedDiff, WinPctDiff, AvgMarginDiff**
- **Team1Games**, **Team2Games**: manual total games played per team
- **Last14WinRate**, **NeutralWinRate**: short-term and location-adjusted performance

All features were symmetrized. Missing values were imputed with column-wise means.

Feature Selection and Optimization. similarly to the XGBoost We adopted the forward-selection procedure. At each iteration, we added the single feature (or, when promising, a top feature pair) that yielded the largest reduction in Brier score under rolling-window cross-validation. Any addition producing a statistically significant improvement was retained. To avoid local minima, our approach was (1) adding the best feature single or pairs as individual candidates, (2) removing the most recently included feature and re-evaluating the top pairs, or (3) introducing the three most indicative features together. This cycle continued until no further feature or pair reduced the Brier score beyond our predefined threshold. Optimized hyperparameters with `Optuna` library.

Performance and Role. Despite using fewer features and lighter data, the RF model performed surprisingly well - with a Brier score of 0.173 and 15× faster training time. While the model initially appeared promising, further evaluation revealed that it did not contribute meaningful value to the ensemble and was therefore excluded from the final combination.

Advantages:

```
Overall average metrics: Accuracy: 0.734, Log Loss: 0.514, Brier: 0.173
```

Figure 6: Random forest results

4.4 Neural Network Attempt (Excluded from Final Ensemble)

To explore deep learning approaches, we implemented a feedforward neural network (NN) with a single hidden layer of 16 units and ReLU activation, trained on the same matchup-level features as our tree-based models (`SeedDiff`, `WinPctDiff`, `AvgMarginDiff`, `EloDiff`, etc.). We applied L2 weight regularization to the hidden layer and normalized all inputs to the [0,1] range using a `MinMaxScaler`. Training used binary cross-entropy loss with the Adam optimizer (learning rate = 0.001).

Limitations Encountered:

- The training set contained under 2000 labeled tournament games-insufficient for stable training of a deep model.
- Performance varied across cross-validation folds (Brier score 0.205 ± 0.015).
- Validation loss often plateaued, indicating limited generalization.
- The NN underperformed compared to logistic regression and random forest in both log loss and calibration, and lacked interpretability.

```
Overall average metrics: Accuracy: 0.672, Log Loss: 0.588, Brier: 0.203

Overall NN CV Metrics:
        accuracy   logloss     brier
season
2019    0.723077  0.497722  0.169328
2021    0.655039  0.617965  0.213158
2022    0.667910  0.599265  0.207259
2023    0.641791  0.638824  0.224214
```

Figure 7: Neural network validation loss and Brier score across folds.

Conclusion: Despite careful tuning and regularization, the NN was not well-suited to this tabular, low-data context. We therefore excluded it from the final ensemble, prioritizing more stable and interpretable models with stronger empirical results.

4.5 Final Ensemble Model

4.5.1 Overview and Methodology

To capitalize on the complementary strengths of our three base learners-Logistic Regression (LR), Random Forest (RF), and XGBoost (XGB)-we form a weighted-average ensemble. We search over the 3-model simplex to directly minimize the hold-out Brier score:

$$P_{\text{ens}} = w_{\text{LR}}\, P_{\text{LR}} + w_{\text{RF}}\, P_{\text{RF}} + w_{\text{XGB}}\, P_{\text{XGB}}, \quad w_{\text{LR}} + w_{\text{RF}} + w_{\text{XGB}} = 1.$$

1. **Grid Search.** We enumerate weights on a 0.02-step grid (8 000 combinations) over $(w_{\text{LR}}, w_{\text{RF}}, w_{\text{XGB}})$.
2. **Selection Criterion.** For each weight combination, we calculate the Brier score on the 2022 and 2023 hold-out sets. The combination with the lowest average error across these years is selected, as these sets are used for the final tuning of ensemble weights aimed at accurately predicting the 2024 brackets.
3. **Final Weights.** The optimal weights found are

   $$(w_{\text{LR}},\, w_{\text{RF}},\, w_{\text{XGB}}) = (0.30,\, 0.00,\, 0.70), \quad \text{Brier Score} = 0.18793.$$

   In effect, RF is dropped (weight = 0), and the ensemble blends LR (30%) with XGB (70%).

4.5.2 Hold-Out Performance

Table 1 compares the Brier scores of each base model and of our final ensemble on the 2024 hold-out tournament:

Table 1: Hold-Out Brier Scores (2024 Tournament)

Model / Method	Weight Vector	Brier Score
Logistic Regression (LR)	-	0.19027
Random Forest (RF)	-	0.19103
XGBoost (XGB)	-	0.18833
Grid-Search Weighted Averaging	**(0.30, 0.00, 0.70)**	**0.18793**

4.5.3 Ensemble Prediction Distribution

Figure 8 plots the ensemble's win-probability histogram on the 2024 hold-out games. Rather than bunching up around 0.50, the predictions are spread almost uniformly across the entire 0–1 range.

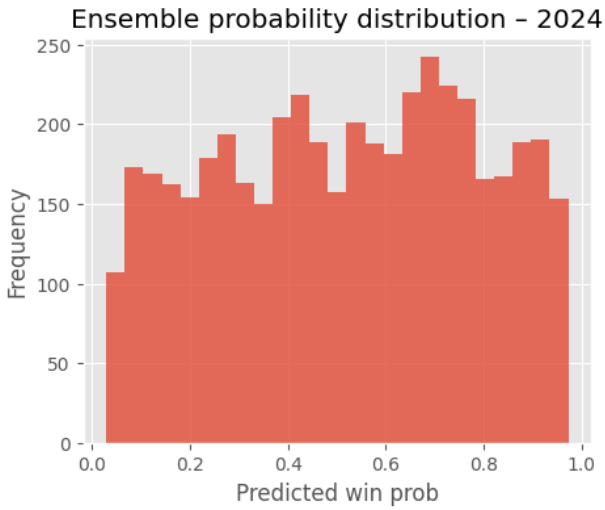


Figure 8: Histogram of ensemble predicted probabilities for the 2024 hold-out tournament.

4.5.4 Conclusions

- The final weighted-average ensemble $(0.30, 0.00, 0.70)$ yields a Brier score of **0.18793**, representing a 0.21% reduction in squared-error loss compared to XGBoost alone (0.18833) and a 1.2% reduction compared to Logistic Regression (0.19027).
- Random forest is effectively excluded (weight = 0), suggesting its signal is already captured by LR and XGB in our feature set or its not indicative enough in comparison. Even unused in voting, RF confirmed our feature choices, flagged season-to-season drift, and highlighted high-variance matchups, boosting confidence in the final model.
- Interpretability vs Complexity trade-off: By weighting XGB at 0.70 and LR at 0.30, we retain a clear "linear + nonlinear" decomposition-each game's brier-score contribution splits neatly into a transparent baseline. Our extensive feature engineering and feature selection paid-off from our XGBoost model. It did so by capturing nonlinear relationships and extracting detailed aspects of the data and providing better results.

5 Final Thoughts

This project set out to answer a deceptively simple question: Can a carefully engineered, data–driven pipeline consistently predict sport tournaments? Our results suggest the answer is a qualified "yes." By blending interpretable linear baselines with tree–based models and coupling them through an ensemble, we produced tremendous tournament win probabilities.

We achieved overall score of 0.0611 by competition metrics which ranked us in the top 150.

After reviewing many top-performing solutions, we found that the main difference was the use of Nate Silver's subscription-based NCAA predictions, which blend power ratings and expert analysis. Some top competitors even skipped machine learning entirely, relying solely on Silver's forecasts to generate brackets. In contrast, we chose to keep our approach fully self-contained, without relying on external datasets.

Verdict: We're aiming high for next year's competition, don't be surprised if you find us leading the leaderboard :)