The Battle for Possessions: Supervised Machine Learning Models for NBA Rebounding Prop Betting

David Maemoto [1]

[1]Department of Computer Science, Stanford

Introduction

At the core of NBA basketball productivity is the creation of possessions, a feature heavily influenced by rebounding on both offense & defense, which drives opportunities, points, and ultimately, team victories.

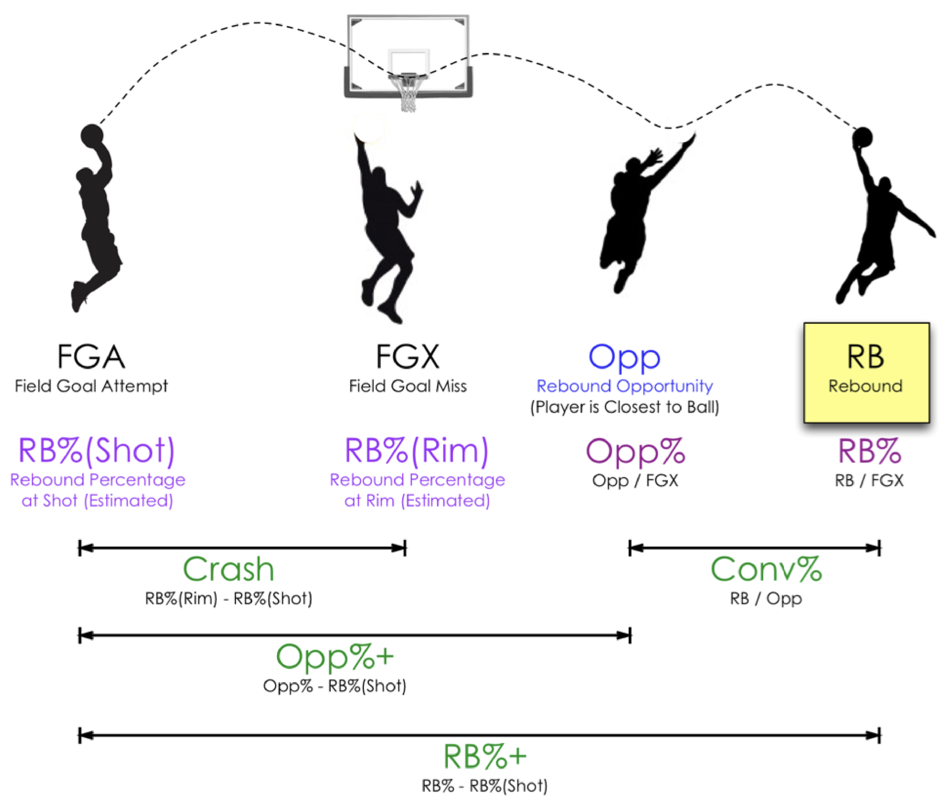

Figure 1. The central idea and various factors that affect rebounding in basketball [1]

I built a machine learning framework to predict individual player rebounding performance, leveraging data from the past four NBA seasons (2017–2021) sourced from Kaggle to create an input vector of 19 features of a combination of each player's physical attributes, each player's average/projected statistics, and team and opposing team statistics for all NBA games. I utilized ML algorithms such as **SVR**, **regression with Huber Loss**, and **extreme gradient boosting** to generate predictions, and then combined their outputs in a **two-layer neural network** to optimize rebounding forecasts and output a singular continuous value corresponding to the predicted number of rebounds that a specific player will have in any given game. This approach achieved a prediction accuracy of **56.93%** and delivered a profit of of **25.96 betting units** during 31 days of testing on the 2024–2025 NBA season.

Data

This project utilizes 2 publicly available datasets from Kaggle [2, 3] to train the ML models, covering four NBA seasons (2017–2021) and spanning **30,760 total rows of data**. The data includes player-level and team-level statistics, where rows correspond to individual player performances for each player that played at least 15 minutes per game for every game, and columns capture various attributes like player performance, team metrics, and opponent statistics. Examples are labeled with ground truth as the actual number of rebounds recorded per game, enabling supervised learning where I employed a **repeated 5-fold cross-validation** to mitigate the risk of overfitting and provide more stable performance estimates.

Features

The feature vector includes **19 normalized variables**, combining player, team, and opponent statistics:

- **Raw input features:** Average shot distance, Field goal %, Player position, Usage per game, etc.
- **Derived input features:** Normalized ratios for offensive/defensive ratings, Pace, Game environment

These features all enhance overall interpretability and performance while comprehensively representing the key factors influencing rebounding, ensuring both player and game context are captured effectively.

Example full raw feature vector for 2018, Russell Westbrook, OKC Thunder versus ATL Hawks:

avg_dist_fga	percent_fga_from_x0_3_range	fg_percent_from_x0_3_range	fg_percent	experience	Usage_game	Trb_per_game	Blk_per_game	Fta_per_game
11.3	0.375	0.611	0.449	10.0	36.4	10.1	0.3	7.1

Pos	Offensive_Rating_Ratio	Pace_Ratio	Opp_x3pa_per_game	Defensive_Rating_Ratio	X3par	Oreb_pct	Opp_orb_per_game	Opp_drb_per_game	Home/Away
1	110.7	96.7	30.6	107.2	0.345	27.7	9.6	34.2	1.0

Models

- **Regression w/ Huber Loss:** where $\delta = 3$ (loss function robust to rebounding value outliers) [4]

$$L(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| \le \delta, \\ \delta |y_i - \hat{y}_i| - \frac{1}{2}\delta^2 & \text{if } |y_i - \hat{y}_i| > \delta, \end{cases}$$

- **Support Vector Regression:** s.t. $\epsilon = 0.5$, $C = 1$ & RBF kernel maps input $\rightarrow$ higher-dim space [5]

$$\min_{\mathbf{w}, b, \xi, \xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*),$$

subject to the conditions: $y_i - (\mathbf{w} \cdot \mathbf{x}_i + b) \le \epsilon + \xi_i, \quad (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i \le \epsilon + \xi_i^*, \quad \xi_i, \xi_i^* \ge 0.$

- **eXtreme Gradient Boosting (XGBoost):** [6]

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

where $l(y_i, \hat{y}_i)$ is the loss function (i.e. MSE) and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|\mathbf{w}\|^2$ is the regularization term.

- **Neural Network Ensemble Outputting Rebounds Prediction:**

$$\hat{y} = \sigma(w_1 \cdot \textbf{Regression}_{\textbf{Huber}} + w_2 \cdot \textbf{SVR} + w_3 \cdot \textbf{XGBoost}),$$

where σ is the ReLU activation, w_1, w_2, w_3 are learnable weights s.t. $w_1 + w_2 + w_3 = 1$ which combines model strengths to enhance prediction accuracy utilizing the 3 ML models as a hidden layer.

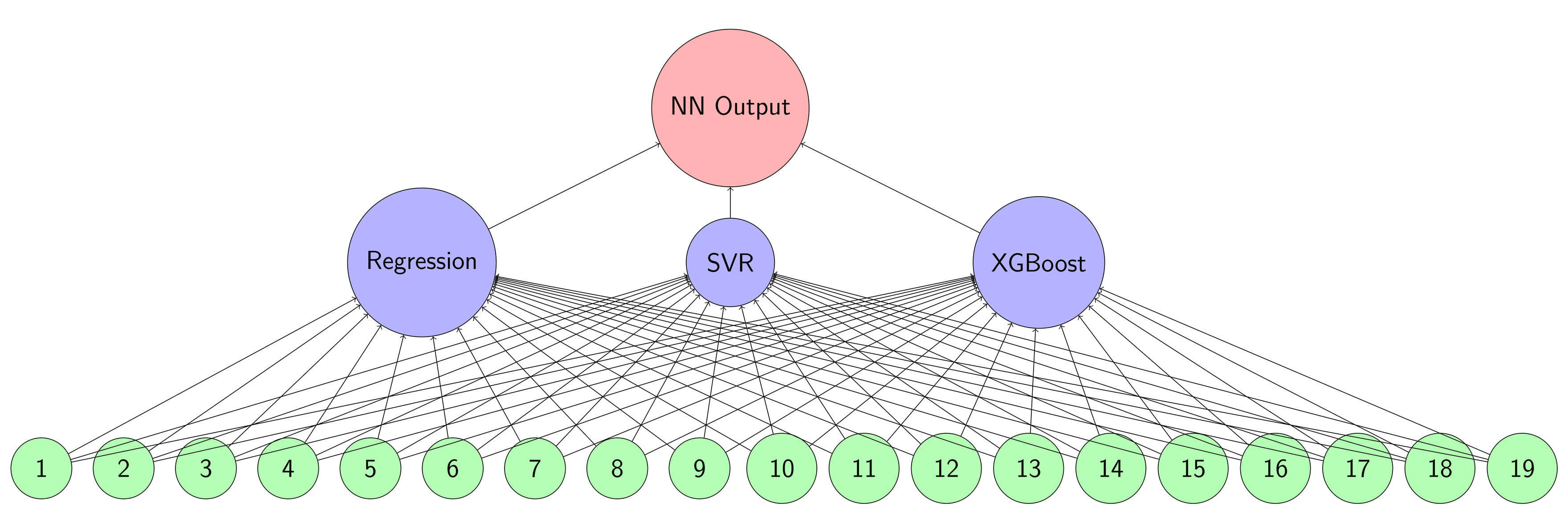

Figure 2. 2-layer Neural network architecture utilized for predicting NBA player rebounds.

Metrics & Evaluation

- **Binary Classification Metric (Accuracy):** Bets are treated as a binary classification problem s.t.:

$$\text{ACCURACY} = \frac{\textbf{Number of correct predictions}}{\textbf{Number of total predictions}} \times 100\%.$$

- **Overall Betting Results:** Net Result is calculated relative to break-even accuracy (52.4%):

$$\textbf{Net Result} = [(\textbf{Model Accuracy} - \textbf{Accuracy}_{\textbf{break-even}}) \times N \times \textbf{bet size}].$$

- **Regression Error Metrics:** Evaluate using Mean Absolute Error and Root Mean Squared Error:

$$\textbf{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|, \quad \textbf{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}.$$

Table 1. Classification Performance Metrics (for edge ≥ 0.13, # bets $\in \{425 - 942\}$)

Model	Accuracy (%)	MAE (train)	RMSE (train)	MAE (test)	RMSE (test)	Betting Results (bet size = $100)
Huber Regression	54.11	2.20	2.86	2.18	2.84	+$726.75
SVR	53.12	2.04	2.70	2.10	2.76	+$588.24
XGBoost	53.08	1.76	2.26	2.16	2.8	+$640.56
NN Ensemble w/ Equal Weights	53.58	1.83	2.12	2.12	2.80	+$676.14
NN Ensemble w/ Optimal Weights	**56.93**	**1.42**	**1.89**	**1.66**	**2.19**	**+$2595.69**

Results in Depth

The figures below reveal that the models perform well for average rebounders (3-6 rpg) but overestimate poor rebounders (0-2 rpg) & underestimate elite rebounders (9+ rpg). Although the true results are skewed towards "under", each model achieves an accuracy >> 52.4%, while highlighting opportunities for improvement through weighted loss functions, oversampling outliers, or adding nuanced features.

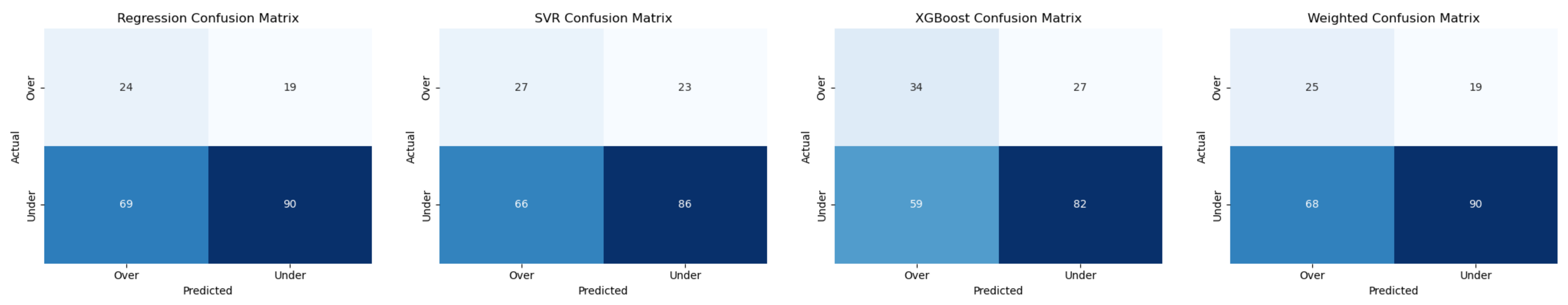

Figure 3. Confusion matrices of the 3 models and the Weighted NN Ensemble.

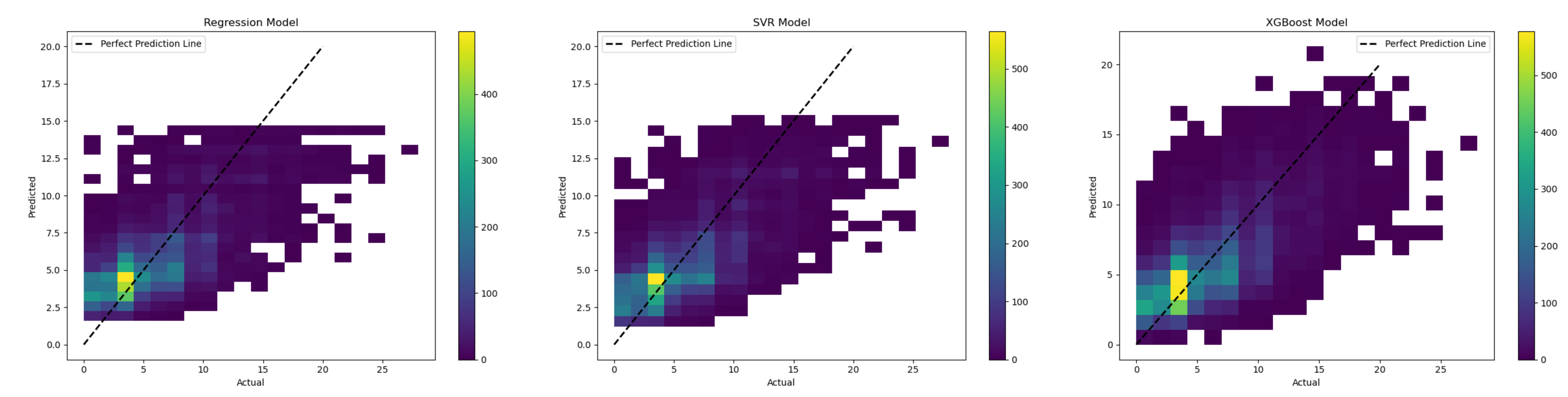

Figure 4. Heatmaps of the 3 Foundational models in their predictive accuracy

Discussion

- **Summary:** The weighted NN Ensemble w/ optimal weights achieved **the highest accuracy (56.93%) & the best betting results (+$2595.69)**, although each model attained +EV results.
- **Interpretations:** Model performance was enhanced due to the neural network's ability to learn an optimal linear combination of predictions, effectively leveraging the robustness of Huber Regression, the margin-based flexibility of SVR & the non-linear complexity of XGBoost, utimately alleviating individual model biases and resulting in a consistent edge in predictive accuracy and betting returns.
- **Expectation:** Consistently profitable model performance was unexpected, but can be attributed to the robust feature engineering, data consistency, and model complexity found within my approach.
- **Bias:** Since the majority of players cluster near the mean rebound range, and the models optimize for the majority class, it would be possible to address these biases through weighted loss functions, oversampling outliers, or adding nuanced features to achieve better accuracy.

Future Work

- **Segmenting the data by player archetype or position** could or exploring advanced ensemble architectures, such as attention-based layers or deeper neural networks, could improve the model's ability to capture position-specific dynamics while improving generalizability.
- **Incorporating richer features**, such as player-tracking metrics, advanced NBA metrics like player efficiency rating, defensive schemes & rivalry/playoff game contexts, could enhance predictive power.

References

[1] R. Masheswaran, Y.-H. Chang, J. Su, S. Kwok, T. Levy, A. Wexler, and N. Hollingsworth, "The three dimensions of rebounding," in *Proceedings of the MIT Sloan Sports Analytics Conference*, 2014.

[2] Justinas, "Nba players data," 2020.

[3] N. Lauga, "Nba games," 2020.

[4] K. Gokcesu and H. Gokcesu, "Generalized huber loss for robust learning and its efficient minimization for a robust statistics," 2021.

[5] D. Basak, S. Pal, and D. Patranabis, "Support vector regression," *Neural Information Processing – Letters and Reviews*, vol. 11, 11 2007.

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 785–794, ACM, Aug. 2016.

davidmaemoto@stanford.edu