Proceedings of the CONF-MLA 2026 Symposium: Explainable Computing, Modeling & Data Science in Complex Systems
DOI: 10.54254/2755-2721/2026.GF37309

Enhancing Basketball Result Prediction Performance Through Innovative Feature Engineering

Yulun Chen

Shanghai Experimental School, Shanghai, China
yulun_chen18@163.com

Abstract. Sports result prediction is a long-standing topic. Yet most existing studies rely heavily on team-scale data and overlook individual players, leaving abstract qualities like chemistry unexplored. This study emphasizes feature engineering to holistically capture each team's power, style, pace, and chemistry. Since conventional stats are insufficient for modern basketball, novel features are constructed manually based on NBA metrics and in-depth knowledge of the game—including TS%, DNP quantification, weighted standard deviation, winning streaks, and starter-bench depth—via an aggregation step, and game-lag was applied to prevent data leakage. This rich feature engineering successfully integrated players' differentiated performances into machine learning, enabling the model to capture the team's true level in a nuanced way. XGBoost, Random Forest, and CART were selected as the prediction model. The results revealed that these new features significantly boost prediction accuracy. The analysis of feature importance also underscores the significant role these features play.

Keywords: basketball result prediction, machine learning, feature engineering, NBA advanced metrics

1. Introduction

Sports result prediction has long been a prevalent topic, not only because the outcome of a game is the center of attention, but also because predictions can bring commercial value and offer strategic direction. Current studies have demonstrated diversified approaches to augment prediction, yet using only team-scale data, ignoring the significance of individual players. This leaves abstract team qualities—such as chemistry—unexplored.

In this essay, feature engineering was mainly focused on to comprehensively showcase the power, style, pace and chemical reaction of every team. As only conventional indices such as points, rebounds and assists are no longer capable of representing the complexity of modern basketball, new features widely accepted in the field of NBA but not appeared in the dataset had to be manually constructed. These novel features can reflect the distinct contributions that each individual makes to the team, though they can't be simply captured by computer program since they comprise complex calculation instead of basic arithmetic. Multiple advanced data metrics were involved to conclude the implication between features. To be specific, the aggregation step was introduced to ensure the features were properly aligned, and a variety of indicators—such as TS%, DNP quantification,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GF37309

weighted standard deviation, winning streaks, and starter-bench depth—were brought in to enrich the overall feature pool and give the model a complete view of the team. At the same time, game-lag was applied to prevent data leakage. By utilizing XGB, RF and CART, the results were successfully derived, and it was noticed that the new features bring a significant boost to the predictions. Moreover, the results were compared in both breadth and depth, and the relevant information within the feature importance was examined.

2. Literature review

Machine learning has revolutionized sports by enabling data-driven predictions, financial assessments, and injury prevention, shifting the field from empirical analysis to precise modeling. For instance, Omar Farghaly et al. used multiple models to improve injury prediction accuracy [1]. Yunlong Guo found that a player's offensive contribution is the key determinant of salary [2]. Ziqi Xu et al. adopted a hybrid model with novel feature selection to achieve a good salary fit [3].

In NBA research, game outcome prediction is the most popular topic. Jingru Wang et al. compared different feature sets for playoff prediction and found that pre-processed features outperform raw count data [4]. Chenjie Cao scraped six seasons of data and built a fully automated cloud-based prediction pipeline [5]. Cem Osken et al. identified player types via clustering and fed the cluster memberships into an artificial neural network, achieving better accuracy than human experts [6]. Yongjun Li et al. combined regression analysis with DEA models to propose the first ex-ante team performance prediction method based on DEA [7]. Bojan Georgievski et al. used a multilayer perceptron to simulate the effect of performance changes on actual points, confirming that machine learning can reliably quantify such impacts [8].

Feature engineering plays a significant role in enhancing model accuracy. Vitor S. Freitas applied a genetic algorithm to select relevant features and used inverse distance weighting and adaptive weights to emphasize recent games [9]. Chi-Jie Lu et al. combined adaptive weighted features with machine learning to predict NBA scores, showing that these engineered features markedly boost performance [10].

Similar approaches are also effective in other leagues. Tzai Lampis et al. used ensemble learning on four European basketball leagues, constructing features from historical statistics and rating systems, and found that higher league competitiveness leads to greater prediction difficulty [11].

At a theoretical level, Jeff Heaton empirically studied how different types of engineered features suit different machine learning models. Automated feature engineering is emerging as a key trend [12]. Udayan Khurana et al. proposed a reinforcement-learning framework that turns feature engineering into a graph-search problem [13]. Fatemeh Nargesian et al. introduced Learning Feature Engineering, using historical data to train classifiers that evaluate new datasets [14]. Aditi Gulati et al. developed a two-level hierarchical approach with genetic algorithms to achieve comparable or superior performance to traditional selection methods [15]. Finally, Paraskevas Koukaras et al. wrote a comprehensive review that provides a context-aware taxonomy and actionable decision rules, balancing basic cleaning and advanced techniques while highlighting trade-offs [16].

3. Methodology

3.1. Data collection

The data were obtained from Nathan Lauga's 'NBA games data' on Kaggle.com. It contains all NBA games from 2004 season to December 2020. Two files from the data he collected were used for

DOI: 10.54254/2755-2721/2026.GF37309

training and preprocessing. The first file describes one game on a team scale, where each row represents a single game and includes the full-game statistics for home team and away team. In this file, six fundamental features are extracted for each team: PTS, FG_PCT, FG3_PCT, FT_PCT, AST and REB. These features can only roughly build up a game. The second file is the player box score, which adopts a micro perspective focused on individual player statistics. In each game, every player has a row of data describing his performance in that match. This means that a single game can consist of up to 30 rows of data. Each player is described by 19 features, including traditional counts, shooting percentages, and plus-minus. This is the dataset on which the feature engineering was implemented.

3.2. Data cleaning

Preseason and all-star games were excluded from the dataset after an in-depth study of NBA rules. Preseason games aim at fine-tuning the lineup and adjusting the competitive form, while all-star games, with a performative nature, are not aligned with match-level playing intensity. In fact, both of them can't reflect player's true power and would influence the accuracy of the training. After the deletion, there remained 24877 games in total, including regular seasons, play-in tournaments and playoffs. Although keeping play-in and playoffs would lead to inequality in game counts, the benefits overtake the drawbacks.

The usability of the data is also crucial. The acquired dataset had serious errors in recording the playing time. Every flawed row was examined one by one and specific solution was addressed: games with negative playing time records were directly removed; seconds that exceeded 60 were carried over into minutes; and all mm:ss formats were converted into a unified floating-point representation to ensure the model could properly interpret the data.

3.3. Data aggregation

The advantage of player box score lies in that it expresses the extent of an individual's contribution to the entire team. Compared to team data, looking at individual player data allows me to see the dispersion and other potential metrics (which we'll discuss later), giving us a finer view of how well a basketball team works together. However, the number of players on each side may vary from game to game. If player-level data were used directly for prediction, the feature dimensions on both sides would be inconsistent. Also, individual player data alone cannot reflect the degree of team cohesion.

To address this, a data aggregation step was introduced to ensure that each sample strictly corresponds to a single prediction target. In other words, each game will only have one row of data with same metrics for home and away team. The aggregated features were categorized into summation features and ratio features.

① Summation, as is implied by the name, means directly adding together the data in one team. This directly represents the team's fundamental output.

$$X_i^t = \sum X_i^p \tag{1}$$

In this paper, 't' refers to team and 'p' refers to individual player. 'i' is the serial number of features. The summation here goes through the roster for the team in this match. Summation step was conducted to features that can be added, including PTS, OREB, REB, DREB, REB, AST, STL, BLK, TO and PF.

DOI: 10.54254/2755-2721/2026.GF37309

② With regard to ratio features, I took the ratio of two selected features, aiming to calculate the overall efficiency on a team scale.

$$X_i^t = \frac{\sum X_j^p}{\sum X_k^p} \tag{2}$$

The correspondence of i, j and k are shown in Table 1.

Table 1. Variable names

i	j	k
FG_PCT	FGM	FGA
FG3_PCT	FG3M	FG3A
FT_PCT	FTM	FTA
AST_PCT	AST	FGM

Although metrics such as "per-36-minute efficiency" are popular on basketball data websites, efficiency computed using total team minutes is merely a linear scaling of the corresponding feature and conveys no additional information, since the denominator is fixed at 240 minutes per game. Therefore, using the total team minutes played is not considered. However, calculating the sum of minutes played by a certain group of players can be quite effective, since the denominator varies across players. Therefore, the scoring efficiency for starters and bench players was computed separately. This directly serves as a measure of the team's bench depth. Now new aggregations can be added to the data, as shown in Table 2.

Table 2. Variable names

i	j	k
Start_eff	PTS_Start	MIN_Start
Bench_eff	PTS_Bench	MIN_Bench

For features with suffix "Start", the starters' stats are taken and the bench players' stats are zeroed out. Same for "Bench".

3.4. Data aggregation

In this part, novel features were generated that describe team chemistry and tactical style—features that cannot be obtained by looking at individual player alone.

① advanced metrics

True Shooting percentage (TS%) is designed to evaluate shooting efficiency. In contrast to the conventional field-goal percentage, TS% is far more reliable, incorporating specific adjustments for three-point shots and free throws. Its core idea is straightforward: it converts every shot attempt into an equivalent two-point shot attempt, thereby reflecting the scoring return per shot attempt. Since opportunities to create scoring chances are inherently limited in a game, seizing those opportunities becomes the key to victory, and TS% serves as a metric that measures the efficiency of converting such opportunities. The calculation is as follows:

DOI: 10.54254/2755-2721/2026.GF37309

$$TS\% = \frac{PTS}{2*(FGA+0.44FTA)} \tag{3}$$

The coefficient 0.44 used in the calculation is an empirical constant derived from fitting a large set of NBA data, representing that each free throw attempt accounts for 0.44 of a shot attempt. The higher TS% is, the more accurate the player shoots.

Consistent with my earlier approach, the TS% for a team was calculated by taking the team-level sum of player values.

$$X^t_{TS\%} = \frac{\sum X^p_{PTS}}{2*(\sum X^p_{FGA}+0.44\sum X^p_{FTA})} \tag{4}$$

Using playing time as the weight, a new feature based on individual TS% was created.

$$X^t_{Weighted_{TS}\%} = \frac{\sum(X^p_{TS\%}*X^p_{MIN})}{240} \tag{5}$$

The 'weighted_TS%' reflects how the coach allocates minutes among players. If the coach gives more playing time to high-efficiency players, this metric will be higher. The magnitude of the value indicates the coach's rationality of decision making. Here '240' refers to the total minutes played. The feature is divided by 240 to ensure a consistent dimension with $\mathrm{X^t_{TS\%}}$.

② dispersion features

These features can distinguish between teams with multiple scoring threats and those that lean heavily on one star player.

The standard deviation for plus-minus (abbreviated as P_M) was first calculated. Plus-minus is a cumulative statistic. Players with longer playing time tend to have larger absolute plus-minus values. To deal with time-dependent effects, the plus-minus was first divided by playing time to achieve normalization, and then the standard deviation was computed on the normalized values:

$$X'^p_{P_M} = \frac{X^p_{P_M}}{X^p_{\mathrm{MIN}}} \quad X^t_{P_M} = \sqrt{\frac{\sum(X^{p'}_{P_M}-\overline{X^{p'}_{P_M}})^2}{N}} \tag{6}$$

'N' denotes the number of players in a team.

Then the weighted standard deviation for PTS (abbreviated as W_PTS) was calculated. The bigger the value is, the more the team is dependent on star players. I chose playing time as the weighting factor for the scores. It magnifies the contribution of long-playing-time players, at the same time small-sample high efficiency will not be overstated. Here's the expression.

$$X^t_{W_{PTS}} = \sqrt{\frac{\sum X^p_{\mathrm{MIN}}*(X^p_{\mathrm{MIN}}-\overline{X^p_{\mathrm{MIN}}})^2}{\sum X^p_{\mathrm{MIN}}}} \tag{7}$$

③ DNP quantification

DNP features are quantifiable metrics representing the team's actual on-court performance deficit for the match. It allows the model to learn that the same absence has fundamentally different

DOI: 10.54254/2755-2721/2026.GF37309

impacts depending on the team's playing style.

First, worthy DNPs needed to be identified. The DNPs of benchwarmers don't have much impact on the team. Instead, players who had a steady playing time (more than 8 minutes per game) in previous games should be focused on. Next, for the meaningful DNP players, the average of their statistics from their last five games played was tracked and extracted as a proxy for their expected performance. The two most straightforward metrics, points and plus-minus, were chosen to construct our features. These player-level values were then summed to obtain the team-level 'DNP_PTS' and 'DNP_PLUS_MINUS' features.

④ Winning streaks

Winning streak length is a familiar metric. It clearly shows how a team is performing lately—whether they're riding high on a win streak or stuck in a losing streak. It also gives a sense of the team's morale and can partly help forecast when a streak might turn around.

3.5. Correlation test

The aggregated and designed features were briefly analyzed. There are 22 new features created for one team in one game. In other words, a total of 44 features is provided to predict a single game. A Spearman coefficient heat map was plotted. The Spearman correlation coefficient is a measure of the monotonic relationship between two variables. Spearman does not require the data to follow a normal distribution nor assume a linear relationship, making it more suitable for my current dataset.

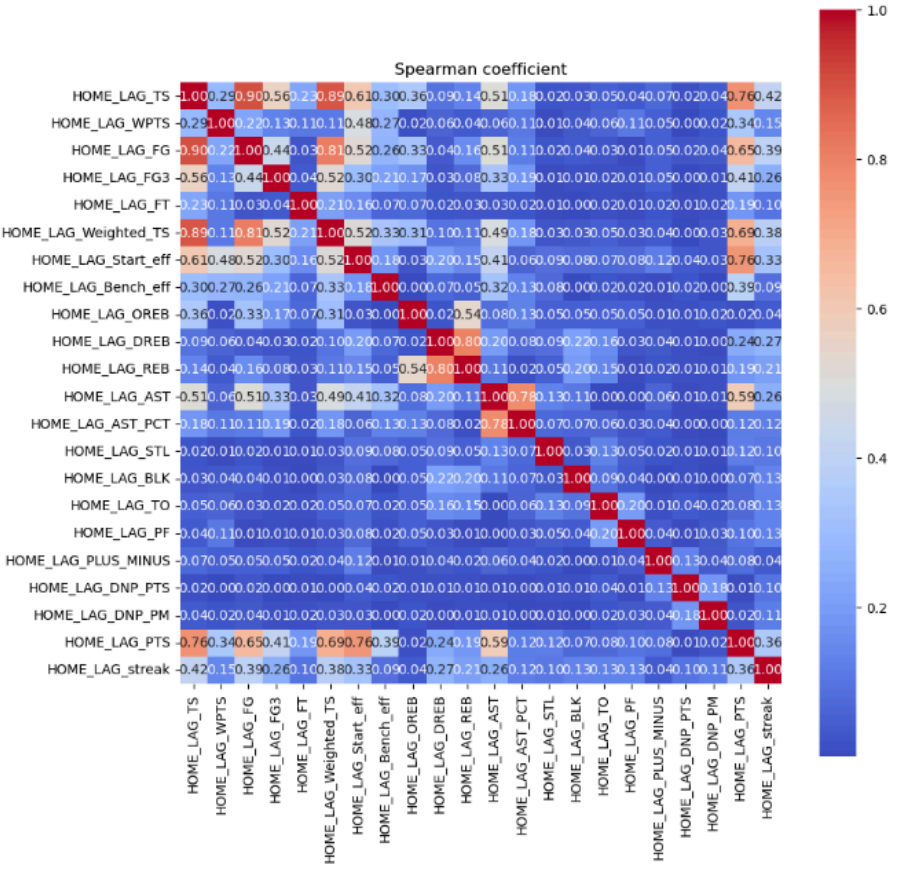

Figure 1. Heat map of the Spearman correlation

DOI: 10.54254/2755-2721/2026.GF37309

From Figure 1 it can be found that no feature exhibits fairly high correlation with one another. This implies that no features have to be dropped, as they already capture the game situation well from diverse perspectives.

3.6. Game-lag

The game data were shifted by one game to prevent data leakage. Apparently, past data need to be used to predict the current game. Without such a shift, the features would include outcomes that have already occurred in the current game.

Shifting by just one game ensures that all feature values come from the most recent state prior to that game, so that the patterns learned by the model are genuinely applicable to predicting the next game. Chi-Jie Lu et al. proposed the inverse-distance weighting (IDW) technique to consider data from up to 6 game-lags [10]. It can effectively boost model performance. However, since the focus of this paper is not on investigating this weighting method, only a one game-lag was still used.

3.7. Baseline models

No matter how intricate the feature engineering is, a range of machine learning models still have to be used to assess the data. After deliberation, CART, XGB and RF were chosen.

CART's advantage lies in its strong interpretability—you can literally draw out the tree structure and see which features influenced the win/loss decisions at key nodes. This makes it especially useful in basketball analysis when you need to explain things clearly to coaches or analysts.

XGB's strength is its usually top-notch prediction accuracy. It handles feature interactions and missing values automatically, and through gradient boosting, it keeps correcting errors from previous rounds step by step. That makes it great for dealing with lots of combined features. In basketball win/loss predictions, it often catches non-linear boundaries that other models might miss.

Random Forest stands out for being robust and not prone to overfitting. By averaging across many trees, it reduces variance and stays insensitive to outliers and noise. It also gives you a reliable ranking of feature importance, so you can quickly pick out the core driving factors from dozens of features.

3.8. Model application

In this phase, the final prediction of NBA games was carried out with 3 machine learning models, namely CART, XGB and RF. First, all data were standardized to bring them onto the same scale, eliminating weight bias caused by differing units of measurement.

Then, for the preprocessed dataset, 44 features were fed into the model for training, with the winning team as the training target. While for the raw data, 12 features were fed into the model. Home wins were encoded as 1 and away wins as 0.

Accuracy, F1 score, Precision and Recall were selected as the evaluation metrics. They enable a direct, comprehensive, and fair comparison of model performance.

4. Result

After training both the raw dataset and preprocessed dataset, the results were obtained, and their performance was evaluated using the four metrics mentioned earlier. The numerical values are displayed in Table 3.

DOI: 10.54254/2755-2721/2026.GF37309

Table 3. Results manifested by four metrics

	Xgb	RF	CART
raw	Accuracy: 0.586519 F1 score: 0.733220 Precision: 0.590794 Recall: 0.966130	Accuracy: 0.568410 F1 Score: 0.694052 Precision: 0.595156 Recall: 0.832364	Accuracy: 0.584708 F1 score: 0.719793 Precision: 0.596668 Recall: 0.906944
Final	Accuracy: 0.635339 F1 score: 0.743442 Precision: 0.645551 Recall: 0.876328	Accuracy: 0.632815 F1 Score: 0.737105 Precision: 0.648483 Recall: 0.853784	Accuracy: 0.620582 F1 score: 0.749807 Precision: 0.622316 Recall: 0.942995

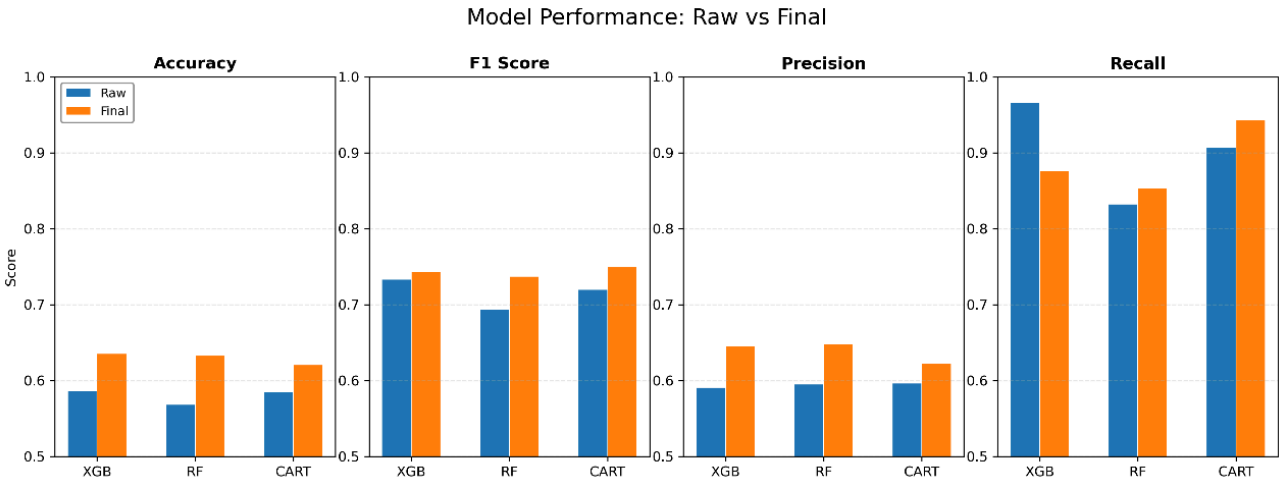

Figure 2. Model performance: raw vs final

From Figure 2 above, the following conclusions were drawn. For the same model and metric, training on preprocessed data significantly outperforms training on raw data. Specifically, the accuracy of the RF model improved by the highest margin of 7 percentage points, with an average increase of 4.9 percentage points across the three models; the F1 score of CART improved by 4 percentage points, with an average increase of 2.9 percentage points; the Precision of XGB improved by 5.48 percentage points, with an average increase of 4.46 percentage points; counterintuitively, the Recall of XGB dropped by 8.98 percentage points, indicating that it sacrificed some recall in exchange for improved precision and accuracy, ultimately achieving a better balance in the F1 score.

For the same data and metric, in terms of accuracy, XGB performs best for preprocessed data (63.53%) and raw data (58.65%), demonstrating the power of the XGB model. In terms of F1 score, the three models each have their own strengths, while CART performs best on preprocessed data (75.23%). Regarding Precision, RF shows its excellence on preprocessed data (59.52%) and raw data (64.85%). For Recall, XGB results best on raw data (96.61%).

DOI: 10.54254/2755-2721/2026.GF37309

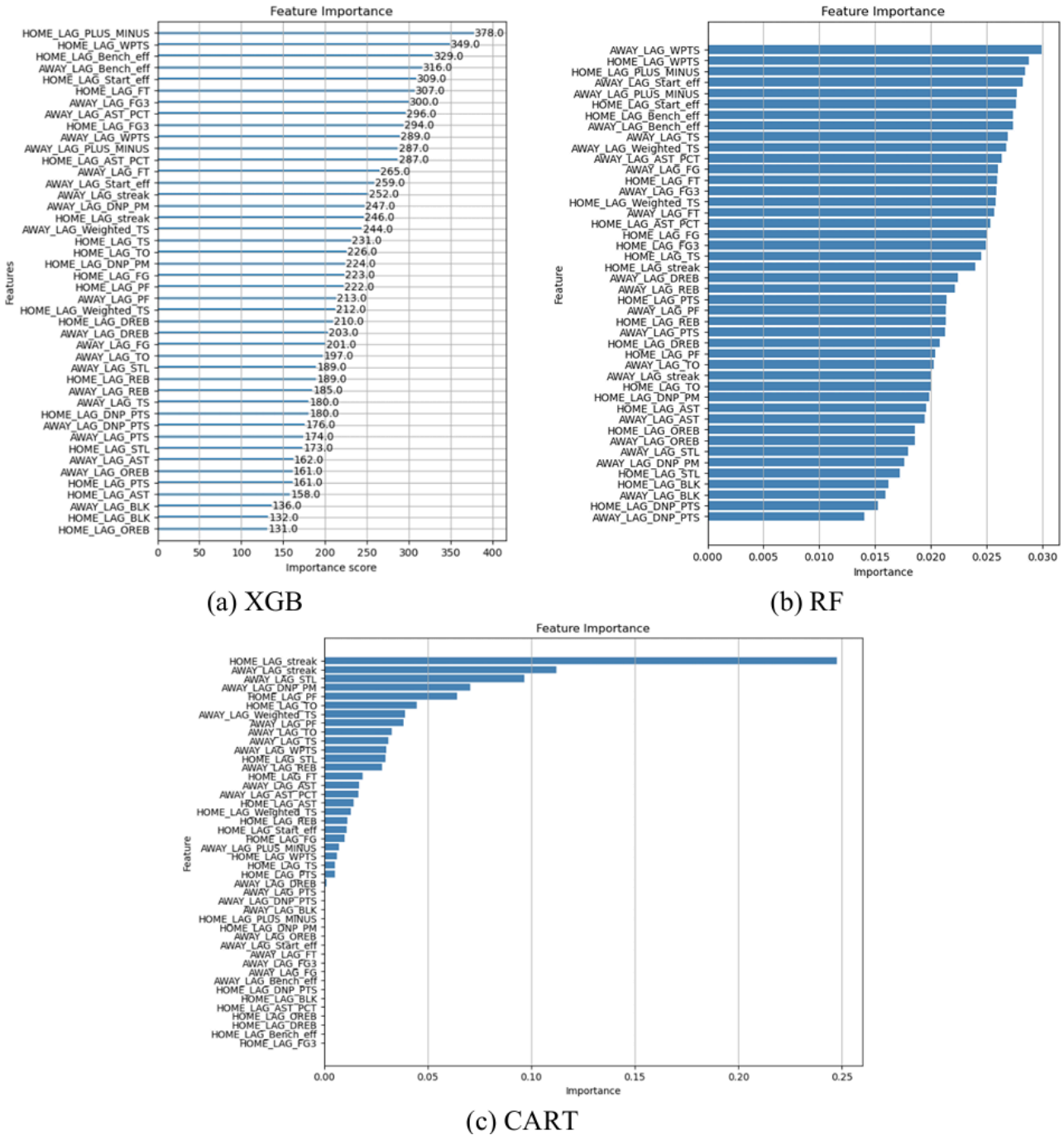

Figure 3. Feature importance of preprocessed data across three models

From Figure 3 it can be found that every feature plays its role. Dividing all the new features into aggregated features and designed features, it was found that among the top ten most important features for the models, XGB and CART have exactly half aggregated and half designed, while for RF, designed features account for 60% of the top ten. This fully shows that the two types of features complement each other.

DOI: 10.54254/2755-2721/2026.GF37309

5. Conclusion

This study employs comprehensive and well-established feature engineering methods to process basketball data, extracting numerous features at the team level that effectively reflect a team's overall strength. Such features are traits that cannot be captured from individual players alone, yet they inherently depend on the aggregation of all individual contributions. Three classification models—CART, XGBoost, and Random Forest—were trained on the processed data and raw data, and the results demonstrate that feature engineering provides a promising enhancement in the performance of machine learning predictions.

This study still has several limitations and room for future improvement. For example, the inverse distance weighting method mentioned earlier could be incorporated to better reflect a team's recent state using multiple consecutive past games, rather than relying on just one game. Meanwhile, play-by-play data could be leveraged in the future to gain a more comprehensive understanding of efficiency per round, and physical nature for each player could be added to further enrich the feature set.

References

[1] Farghaly, O., & Deshpande, P. (2024). Leveraging machine learning to predict National Basketball Association player injuries. In *2024 IEEE International Workshop on Sport, Technology and Research (STAR)* (pp. 216–221). IEEE.

[2] Guo, Y. (2025). Exploring the impact of various factors on the salaries of players in the National Basketball Association. In *Exploring Science Academic Conference Series* (Vol. 8, pp. 55–61).

[3] Xu, Z., & Xu, Z. (2026). Enhanced prediction of NBA players' salaries using hybrid ensemble models and multi-objective optimization techniques based on the 2022–23 season dataset. *Expert Systems with Applications, 295*, 128849.

[4] Wang, J., & Fan, Q. (2021). Application of machine learning on NBA data sets. In *Journal of Physics: Conference Series* (Vol. 1802, No. 3, p. 032036). IOP Publishing.

[5] Georgievski, B., & Vrtagic, S. (2021). Machine learning and the NBA game. *Journal of Physical Education and Sport, 21*(6), 3339–3343.

[6] Osken, C., & Onay, C. (2022). Predicting the winning team in basketball: A novel approach. *Heliyon, 8*(12).

[7] Cao, C. (2012). *Sports data mining technology used in basketball outcome prediction* (Master's thesis). Technological University.

[8] Li, Y., Wang, L., & Li, F. (2021). A data-driven prediction approach for sports team performance and its application to National Basketball Association. *Omega, 98*, 102123.

[9] Freitas, V. S. (2024). Optimizing sports outcome prediction through feature engineering and machine learning.

[10] Lu, C. J., Lee, T. S., Wang, C. C., et al. (2021). Improving sports outcome prediction process using integrating adaptive weighted features and machine learning techniques. *Processes, 9*(9), 1563.

[11] Lampis, T., Ioannis, N., Vasilios, V., et al. (2023). Predictions of European basketball match results with machine learning algorithms. *Journal of Sports Analytics, 9*(2), 171–190.

[12] Khurana, U., Samulowitz, H., & Turaga, D. (2018). Feature engineering for predictive modeling using reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (Volume 32)*(1).

[13] Heaton, J. (2016). An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016* (pp. 1–6). IEEE.

[14] Nargesian, F., Samulowitz, H., Khurana, U., et al. (2017). Learning feature engineering for classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 2529–2535).

[15] Gulati, A., Felahatpisheh, A., & Valderrama, C. E. (2025). Feature engineering through two-level genetic algorithm. *Machine Learning with Applications, 21*, 100696.

[16] Paraskevas, K., & Christos, T. (2025). Data preprocessing and feature engineering for data mining: Techniques, tools, and best practices. *AI, 6*(10), 257.