Determining the NBA MVP Outcome Using Machine Learning Models

***Abstract*—The NBA's MVP award is one of the most prestigious honors presented to the top-performing players of the regular season. It is annually recognized as a mark of distinction in a player's career, highlighting his impact and dominance in the game. The recognition is highly sought after but also extremely unpredictable, as numerous factors impact the selection of an individual. By developing an MVP prediction model, analysts can delve deeper into the potential impact on a player's career. Furthermore, teams can strategically leverage MVP projections for marketing campaigns, enhancing engagement and sponsorship opportunities. Fans can utilize these forecasts for informed wagering decisions and are brought into the excitement of predicting sports outcomes. This study uses different machine learning models, training them on seasonal data to determine the candidate most likely to get MVP and which algorithm works best. This study considered the statistical averages of 400 players over 16 seasons, implementing key criteria such as Minutes Played, Field Goal Percentage, Three-Point Percentage, Two-Point Percentage, Assists, Steals, Blocks, and Points. The models tested include SVM, KNN, Random Forest, AdaBoost, and Gradient Boosting, each containing unique ways of handling inputs and optimizing performance metrics. By evaluating a player's performance across multiple seasons, the models can recognize trends and patterns contributing to forecasting standout performances. These models also offer valuable insight to media, enthusiasts, and sports bettors. As the intersection of sports and technology grows, predictions illustrate the power of data usage to transform the analysis of athletic achievements.

I. Introduction

The NBA is the most highly respected league in professional basketball, as it attracts audiences globally with its fast-paced games and competitive spirit. For fans, the excitement and passion for the sport bring diverse communities together. The organization's star players often become cultural icons, admired for their drive to become the best they can be and lead their team to the championship. The league's impact on people's lives is further signified by its prestigious awards, which celebrate the players' hard work and inspire teens to be motivated to get better. As each season ends, individuals are presented with honors such as the Sixth Man of the Year Award, which recognizes the league's top performer off the bench, and the Defensive Player of the Year, highlighting one's defensive presence on the court. These awards not only honor individual excellence but also serve as motivation for aspiring athletes.

The MVP (Most Valuable Player) award is given in various sports leagues and competitions to recognize the player who has made the most significant impact on their team's performance throughout the season or a specific series of games. Typically, MVP awards involve voting by a select group of individuals who deeply understand the sport. This group often includes sports journalists, broadcasters, and coaches. Voters consider a range of factors when deciding on the MVP. Statistics such as points scored, assists, points, and other basketball averages are crucial metrics. The player's contribution to their team's success is also often measured by their influence on wins and losses. Other qualities include leadership on and off the field, sportsmanship, and their role within the team dynamic. Depending on the sport, the MVP award might focus on performance throughout the regular season, playoffs, or both. Playoff MVP awards specifically recognize exceptional performance during the postseason. Each sport and league has its nuances in how MVPs are selected, but the principles of recognizing outstanding individual achievement and contribution to team success remain the same.

Deciding the NBA MVP (Most Valuable Player) award involves several challenges due to the subjective nature of evaluating individual performance and its impact on team success. MVP voting can be personalized, as it consists of interpreting various statistical measures and intangible qualities like leadership and impact on team dynamics. While statistics such as points, rebounds, assists, and shooting percentages provide a quantitative basis for evaluation, their interpretation can vary. Some voters may prioritize traditional stats, while others may look at advanced metrics like PER (Player Efficiency Rating), Win Shares, or Net Rating. Comparing players from different positions (e.g., guards, forwards, centers) adds complexity. Each position has different responsibilities and impacts the game differently, making direct statistical comparisons challenging. A player's availability throughout the season due to injuries or load management can impact their MVP candidacy. Voters may penalize players who miss significant time, even if their performance when on the court was exceptional. Despite these challenges, The NBA continues to refine its criteria and voting process to ensure the award maintains its consistency in selection.

Creating an MVP predicting model in sports like the NBA involves several key steps and considerations to effectively predict which player will likely win the award based on various factors. Gathering comprehensive data on player statistics, team performance metrics, and other relevant factors across multiple seasons is crucial. This includes traditional stats (points, rebounds, assists) and advanced metrics (PER, Win Shares, usage rate). Models also need to consider contextual factors such as team standings, strength of schedule, injuries, and lineup changes. These factors can influence both individual player performance and team success.

II. Related Work

Choosing a model to predict NBA MVPs involves several challenges and considerations due to the complexity and variability of the factors influencing MVP voting. Models need to account for dynamic features and changes in player form throughout the season. Balancing model complexity is important to avoid overfitting, where the model fits the training data too closely but performs poorly on new data, or underfitting, where the model fails to capture important patterns. For this reason, ensuring the model generalizes well to new seasons and diverse player profiles is a constant challenge.

A similar analysis that delved into predicting the MVP award in baseball focused on ten different features and statistics that would help determine a player's WAR (wins above replacement). Using the random forest regressor algorithm, it determined the best factor for projecting the winner. Random forests are highly effective for large datasets and complex classification tasks. They operate by constructing multiple decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random Forests are robust against overfitting due to the ensemble learning method that combines predictions from multiple trees. This makes Random Forests a popular choice across various domains where accurate predictions are crucial.

In another paper, researchers used the KNN algorithm to predict the NBA's MVP. To evaluate NBA MVP voting, they considered a range of important variables and relied on established databases for their data. They used Basketball Reference for comprehensive historical player data and incorporated the 'Raptor Metric' from FiveThirtyEight for advanced analytics. Although they initially explored the NBA-API, it provided unclear data, so they opted to use historical databases for a more reliable source of information. They determined that if a single model had to be used, using the KNN model works best for guessing the NBA MVP winner or other predictions. K-Nearest Neighbors (KNN) is well-suited for small to medium-sized datasets with relatively low dimensionality. It operates by memorizing the training dataset and predicting the class of new instances based on their proximity to the most similar instances in the training data. KNN is particularly useful when labeled data is scarce or expensive, as it does not require a training phase beyond storing the data points themselves. However, its prediction time can be slower compared to other algorithms, especially as the dataset size grows.

Analysts applied SVM to observe patterns in the MVP's choice logic and verify the most important statistics in the choice. They wanted to create a model that could predict the 2022-23 regular season MVP in the NBA before the official result was released. Since the objective was to predict the MVP for the 2022–23 season, data from the 2006–07 season through the 2021–22 season were used for the study, covering a total of 16 seasons. The study found that SVM performed better in the most recent season, while other models, such as Gradient Boosting had better performance in the older seasons. SVM was used because it excels in smaller, complex datasets with higher dimensions. SVMs are effective in non-linear classification tasks and are memory efficient during the prediction phase as they use only a subset of training points (support vectors). However, SVMs can be less suitable for very large datasets due to their computational complexity and memory requirements. SVMs are preferred in scenarios where finding the optimal decision boundary and maximizing the margin between classes are critical.

Other similar studies include a comparison of machine learning models to best predict game attendance in Major League Baseball and the use of machine learning to predict future performances of quarterbacks. The first study explores the prediction of the attendance of Major League Baseball games using six commonly used regression models in machine learning. Focusing on a single team helps mitigate the influence of varying fan tendencies between teams. Performance metrics such as the mean absolute percentage error and the root mean square error indicate that the random forest and gradient boost models provide the most accurate attendance predictions. Implementing these models can help baseball teams optimize staff management, event planning, and marketing strategies before games. Random Forest was the top-performing model, as the study used a large dataset, which is similar size to the one trained in this study. The study also trained multiple models at once using the same data, which is a similar process that helps pick the best algorithms. The main objective of the second paper was to produce more accurate statistics, which improved as more seasons were added. Specifically, statistics from three consecutive seasons for the quarterbacks were compiled, trained, and tested on a model. Linear regression was utilized to predict quarterback statistics based on their previous season's performance.

In the realm of machine learning, the choice of model often hinges on the size and complexity of the dataset. For small datasets, where simplicity and fewer parameters are advantageous, algorithms such as logistic regression, support vector machines (SVM), and decision trees are commonly employed. Logistic Regression is straightforward and efficient for binary classification tasks, providing interpretable results. SVMs excel in both linear and nonlinear classification scenarios, making them suitable for small to medium-sized datasets. Decision trees offer interpretability and versatility, accommodating numerical and categorical data with ease, making them a practical choice for datasets with moderate features.

On the other hand, larger datasets benefit from more complex algorithms that can capture intricate patterns and relationships. Random Forests and Gradient Boosting Models, such as XGBoost and LightGBM, are robust choices for such scenarios. Random Forests leverage multiple decision trees trained on random subsets of data to mitigate overfitting and achieve robust performance across diverse datasets. Gradient boosting models sequentially build decision trees to correct errors from previous iterations, excelling in handling complex relationships within the data. However, caution is warranted with these models on smaller datasets. XGBoost and LightGBM, for instance, are prone to overfitting when trained on datasets containing fewer than 500 rows, potentially compromising their performance due to the model's complexity outweighing the available training data.

In summary, the selection of a machine learning model depends heavily on dataset size and complexity. For small datasets, simpler models like Logistic Regression and Decision Trees are effective and interpretable, whereas larger datasets benefit from the predictive power and adaptability of Random Forests and Gradient Boosting Models, provided careful attention is given to avoid overfitting when dataset sizes are limited. The choice of model depends on factors such as dataset size, dimensionality, availability of labeled data, and the complexity of the problem. Each model's strengths focus on different aspects of machine learning tasks, ensuring optimal performance based on specific data characteristics and problem requirements.

III. Dataset

The data we work with is numerical, and there are three main datasets used to train the machine learning model: game statistics, physiological data, and MVP voting information. The "treat data" function processes the basketball data to prepare it for analysis, with the steps listed in the following:

1) Drop unnecessary columns from all datasets.
2) Rename stat columns with `_PERGAME`, `_TOTAL`, `_AVANCADO`.
3) Merge per-game, advanced, and total stats on `Player`, `Season`, `Tm`.
4) Map team names to abbreviations and merge with standings.
5) Compute win percentage and merge standings into data.
6) Clean player names and merge MVP data.
7) Keep only first entry per player per season (remove duplicates).
8) Filter players based on stat thresholds and MVP votes.
9) Drop columns `Tm`, `Team`, and `Record`.
10) Return the cleaned dataset.

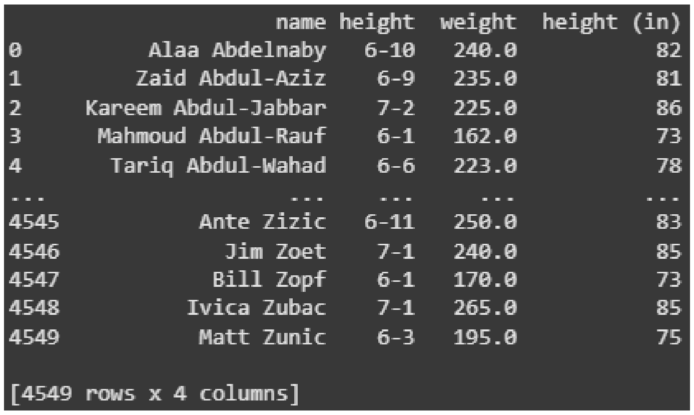
Algorithm 1: Clean and Filter NBA Player Data

```
1:  Input: Datasets per_game, totals, avancados, standings, seasons
2:  Output: Filtered player data dataf
3:  Remove irrelevant columns from each dataset
4:  Rename stat columns with suffixes: _PERGAME, _TOTAL, _AVANCADO
5:  Merge the three datasets on ['Player', 'Season', 'Tm']
6:  Map full team names to abbreviations and join with standings
7:  Calculate win percentage: PCT = Wins / Games
8:  Merge standings into main dataset on ['Tm', 'Season']
9:  Clean player names (remove asterisks and suffixes)
10: Merge MVP data on ['Player', 'Season'], fill missing with 0
11: for each season do
12:     Keep only the first team entry per player
13: end for
14: Filter players based on performance thresholds:
        • Games $> 48$, PPG $> 13.5$, MPG $> 30$
        • Team seed $\leq 16$, AST $> 1$, REB $> 3$
        • FG% $> 0.37$, FGA $> 10$, PER $> 18$
        • Or MVP vote share $> 0$
15: Drop unnecessary columns and return dataf
```

Once the players that do not meet the key filtering criteria are removed, the dataset is prepared for deeper analysis, especially focusing on identifying top performers or understanding MVP-related statistics.

The first dataset contains detailed statistics for basketball players, likely from a specific season or a range of seasons. The dataset provides a comprehensive view of player performance metrics, including scoring, shooting efficiency, and defensive statistics. Players like Chris Paul and Devin Booker have high average minutes per game (36.8 and 34.5, respectively), indicating significant roles on their teams. Stats such as steals and blocks provide insight into defensive contributions. For example, David West has a significant number of blocks (38), which indicates a strong defensive presence. The ages of the players range from young talents like Chris Paul (21) to more experienced players like Jimmy Butler (32). This could affect their performance and playing style. Players such as Giannis Antetokounmpo have a high MVP rank and vote share, indicating their significant impact during the season. Statistical data allows for a comparison of players based on key metrics, an analysis of how minutes on the floor affect team success, and an examination of the trends of specific players' performances.

The physiological dataset contains information about basketball players, specifically focusing on their height, weight, and height in inches. The height in the ”Height (in)” column is converted from the ”Height” column. For instance, ”6-10” converts to 82 inches, and ”7-2” converts to 86 inches. This conversion is consistent with the standard formula where height in inches = (feet * 12) + inches. The heights range from 6 feet 1 inch (73 inches) to 7 feet 2 inches (86 inches), and the weights range from 162 pounds to 265 pounds. From this dataset, we can find how height and weight correlate with playing style and performance, and understand the physical characteristics of players on different teams.



```
                        name height  weight  height (in)
0             Alaa Abdelnaby   6-10   240.0           82
1            Zaid Abdul-Aziz    6-9   235.0           81
2       Kareem Abdul-Jabbar    7-2   225.0           86
3         Mahmoud Abdul-Rauf    6-1   162.0           73
4           Tariq Abdul-Wahad    6-6   223.0           78
...                      ...    ...     ...          ...
4545              Ante Zizic   6-11   250.0           83
4546                Jim Zoet    7-1   240.0           85
4547               Bill Zopf    6-1   170.0           73
4548             Ivica Zubac    7-1   265.0           85
4549              Matt Zunic    6-3   195.0           75

[4549 rows x 4 columns]
```

Fig. 1: Snippet of the physiological dataset table with the name, height, and weight of about 4500 players.

The MVP voting dataset focuses on various basketball players, including some basic player statistics and their MVP voting information for a specific season. The metrics it takes in are age, season, MVP votes share, MVP rank, player, weight, and height. This allows us to measure how much support the player receives in MVP voting and the significance of MVP consideration for each individual. Players with a high MVP vote share (like Zaid Abdul-Aziz) are typically considered significant contributors in the league. Several entries have missing values for age, season, MVP votes share, and MVP rank. This might indicate incomplete data or that these players were not active during the season specified. Using this dataset, we can use physical attributes to profile players and recognize how these may impact their careers.

IV. Methods

Once the datasets of players' stats and physiological data of players are established, we categorize them into tables. The "funcmetricas" function is defined, which calculates and records evaluation metrics for a predictive model. The function also returns the root mean squared error (RMSE) between the test and prediction, with a lower value indicating a better fit. Subsequently, the R-squared score is computed, indicating how well the predictions match the actual values.

The following code defines a function "funcmodelos" that performs the following tasks for a set of machine learning models across multiple seasons of data:

1) Separates Data by Season: For each season in the season's list, the code separates the data into training and testing sets.
2) Prepares Data: It preprocesses the data by scaling the features using StandardScaler.
3) Trains Models: It iterates over a list of models (modelos) and performs hyperparameter tuning using GridSearchCV to find the best parameters for each model.
4) Evaluates Models: It trains the model with the best hyperparameters on the training data, makes predictions on the test data, and records the metrics for evaluation.
5) Records Results: It saves the model and its performance metrics, and aggregates the results for comparison.
6) Saves Parameters: Finally, it saves the best parameters found during hyperparameter tuning to a CSV file.

Algorithm 2: Train and Evaluate MVP Prediction Models

```
1:  Input: Data data, list of seasons, model names
    modelos, test limit n_seasons_to_test
2:  Output: final_results, metricas, best_params
3:  Initialize final_results, metricas, and
    best_params
4:  Set counter i = 1
5:  for each season in seasons do
6:      Split data into training (all other seasons) and test
        (current season)
7:      Separate features X and target y
8:      Standardize features using StandardScaler
9:      for each model name in modelos do
10:         Define hyperparameter grid based on model type
11:         Perform grid search cross-validation on training data
12:         Retrieve best model and store best parameters
13:         Train final model on full training set
14:         Save model to disk
15:         Predict MVP vote shares for test set
16:         Record evaluation metrics using helper function
17:         Add predictions and ranks to results
18:     end for
19:     Append results to final_results
20:     if i == n_seasons_to_test then
21:         break
22:     end if
23:     Increment i
24: end for
25: Save best model parameters to CSV
26: return final_results, metricas, best_params
```

The average performance metrics for each model from a dataframe of metrics are calculated through the "mediametricas" function. Specifically, the average RMSE and R-squared score are computed, allowing for a simplified comparison of model performance across multiple evaluations. The "createrank" function generates the final results, a ranking dataframe that compares the actual and predicted MVP vote shares for players across different seasons and models. It loops over each model in the modelos list, sorts the predicted MVP votes share for the current model in descending order, stores the names of the top players as predicted, and stores the predicted MVP votes share for these top players, rounding it to three decimal places. The created dataframe, "rankfinal," allows insights into the performance of algorithms in ranking players.

The learning algorithms that were used in this dataset are KNN, SVM, Random Forest, AdaBoost, Gradient Boosting, LGBM, and XGBoost. These algorithms were chosen for their diverse approaches to handling classification and regression tasks. The inclusion of these models allowed us to increase our range and diversity of research, enhancing the predictive performance and capturing various patterns in the data. A more detailed description and perspective of each algorithm will be listed below.

K-Nearest Neighbors (KNN)

- **Type:** Classification & Regression
- **Core Idea:** Predicts output by finding the k closest training examples using a distance metric; classifies by majority vote or predicts by averaging.
- **Advantages:** Simple to implement, versatile, fast training.
- **Disadvantages:** Computationally expensive at prediction time, sensitive to distance metrics and scale, memory intensive.

Support Vector Machine (SVM)

- **Type:** Classification & Regression
- **Core Idea:** Finds the optimal hyperplane that maximizes margin between classes in transformed feature space.
- **Advantages:** Effective in high-dimensional space, robust to overfitting with proper regularization.
- **Disadvantages:** Computationally intensive for large datasets, requires kernel tuning, memory usage can be high.

Random Forest

- **Type:** Classification & Regression
- **Core Idea:** Ensemble of decision trees where final prediction is based on majority vote (classification) or average (regression).
- **Advantages:** Reduces overfitting, handles missing data, robust and accurate.
- **Disadvantages:** Less interpretable, slower training, large model size.

AdaBoost (Adaptive Boosting)

- **Type:** Classification
- **Core Idea:** Boosts weak learners by focusing on misclassified examples in a sequential manner.
- **Advantages:** High accuracy, improves weak models, flexible with base learners.
- **Disadvantages:** Sensitive to noisy data and outliers, slower for large datasets.

Gradient Boosting

- **Type:** Classification & Regression
- **Core Idea:** Builds models sequentially to minimize errors using gradients of the loss function.
- **Advantages:** High predictive accuracy, adaptable to many problems.
- **Disadvantages:** Can overfit, requires tuning, slow to train.

LightGBM (Light Gradient Boosting Machine)

- **Type:** Classification & Regression
- **Core Idea:** Uses histogram-based learning and leaf-wise tree growth for fast and efficient boosting.
- **Advantages:** Very fast, scalable, low memory usage.
- **Disadvantages:** Can overfit with deep trees, complex parameter tuning.

XGBoost (Extreme Gradient Boosting)

- **Type:** Classification & Regression
- **Core Idea:** Regularized boosting with advanced pruning and parallel computation.
- **Advantages:** Highly accurate, efficient, handles large datasets well.
- **Disadvantages:** Complex to tune, memory-intensive with many iterations.

V. Results and Discussion

The comparison of MVP share predictions for the 2021-22 NBA season across different models reveals varied performance levels, indicating how well or poorly each model captures the actual MVP shares.

Among the models analyzed, Support Vector Machines (SVM) and Random Forest stand out as two of the best performers. Both models consistently produce predictions that are closer to the actual MVP shares compared to others. SVM strikes a good balance by providing relatively accurate estimates for key players like Jokić and Embiid, while Random Forest shows strong performance overall, particularly with Jokić's MVP share prediction closely matching the real value. Their relative success may be attributed to SVM's ability to handle complex decision boundaries and Random Forest's ensemble approach that effectively captures non-linear relationships and reduces overfitting.

K-Nearest Neighbors (KNN) underestimates the MVP shares for top candidates. For Nikola Jokić, the KNN model predicts a share of 0.7, which is significantly lower than the real share of 0.875. This model struggles to capture the full extent of Jokić's MVP value due to its reliance on local neighborhood data that might not accurately reflect his impact. Similarly, KNN underestimates Joel Embiid's share (0.525 vs. 0.706) and Giannis Antetokounmpo's share (0.324 vs. 0.595), suggesting a general limitation in its ability to account for the significance of high-performing players.

Elastic Net shows predictions that are relatively close to the real values but still exhibit some discrepancies. Elastic Net predicts Jokić's share accurately (0.806), though it falls short of the real value of 0.875. For Embiid, the prediction (0.536) is lower than the actual share of 0.706, indicating a potential underestimation.

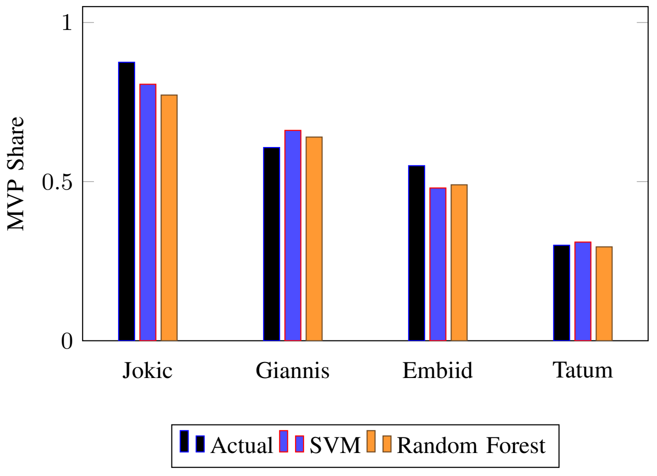


Fig. 2: MVP Share Predictions — 2021–22 (Top 4 Players)

AdaBoost shows a significant tendency to underestimate MVP shares across all players. The model predicts Jokić's share at 0.676, much lower than the actual 0.875, and similarly underestimates Embiid and Giannis with shares of 0.496 and 0.283, respectively. This underperformance might be due to AdaBoost's sequential correction of errors, which might not adequately adjust for the high impact of MVP candidates.

Gradient Boosting and XGBoost provide predictions that are not close to the actual values. Gradient Boosting predicts Jokić's share at 0.629; for Embiid and Giannis, the predictions (0.387 each for both players) reflect a moderate fit but still fall short of the real values. These models might benefit from better handling of feature importance and model parameters to improve prediction accuracy.

VI. Conclusion and Next Steps

The NBA MVP award is a highly prestigious recognition that reflects individual excellence and contributions to team success. Given the subjective nature of MVP voting, predicting the winner involves analyzing various statistics and intangibles, including player performance metrics, team standings, and historical data. Machine learning models are increasingly used to forecast MVP outcomes, employing diverse algorithms such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forests, AdaBoost, Gradient Boosting, LightGBM, and XGBoost.

Previous research has explored different approaches for predicting MVP outcomes. For example, Random Forests and Gradient Boosting models have been effective in handling complex datasets and providing accurate predictions, while KNN and AdaBoost showed limitations in capturing the full impact of MVP candidates. SVM performed variably, with some success in recent seasons but challenges with non-linear relationships in older data. The use of advanced metrics and comprehensive player data has been central to these analyses.

For further research, several steps can enhance the applicability of the models. We can explore additional features such as advanced player metrics, team chemistry, and game situational statistics to enhance the model inputs and potentially improve prediction performance. They can also incorporate real-time data to make predictions more responsive to current season dynamics, such as injuries or changes in player performance. Finally, using interpretability tools in the future can help understand how each model makes predictions. This can provide insights into which features are most influential in MVP voting and enhance transparency.

REFERENCES

[1] T. Maher, “How to Bet on NBA MVP: Odds, Winners and Betting Tips,” *Forbes Betting*, July 17, 2024.

[2] E. Blumenfeld, “Using Machine Learning Driven Models to Predict NBA’s MVP,” *Medium*, Jan. 4, 2022.

[3] G. Pastorello, “Predicting the NBA MVP With Machine Learning,” *Towards Data Science*, Jan. 21, 2025.

[4] GeeksforGeeks, “K-Nearest Neighbor (KNN) Algorithm,” Jan. 29, 2025. [Online]. Available: https://www.geeksforgeeks.org

[5] GeeksforGeeks, “Support Vector Machine (SVM) Algorithm,” Jan. 27, 2025. [Online]. Available: https://www.geeksforgeeks.org

Appendix

TABLE I: MVP Rank and Share Comparison (All Models)

Season	Real	KNN	SVM	RF	Ada	GB	LGBM	XGB
21–22	Jokić (.875)	Jokić (.700)	Jokić (.806)	Jokić (.772)	Jokić (.626)	Jokić (.615)	Giannis (.641)	Giannis (.601)
20–21	Jokić (.961)	Jokić (.687)	Jokić (.728)	Jokić (.704)	Jokić (.663)	Jokić (.584)	Jokić (.584)	Jokić (.584)
19–20	Giannis (.952)	Giannis (.581)	Giannis (.661)	Harden (.548)	Harden (.498)	Giannis (.746)	Giannis (.724)	Giannis (.750)
18–19	Giannis (.932)	Harden (.712)	Giannis (.738)	Harden (.685)	Harden (.615)	Giannis (.766)	Giannis (.760)	Giannis (.778)
17–18	Harden (.955)	LeBron (.591)	Harden (.710)	Harden (.617)	Harden (.831)	Harden (.948)	Harden (.989)	Harden (.976)
16–17	Westbrook (.879)	LeBron (.570)	LeBron (.519)	Westbrook (.561)	Kawhi (.623)	Westbrook (.602)	Kawhi (.592)	Kawhi (.592)
15–16	Curry (1.000)	Curry (.528)	LeBron (.601)	Curry (.752)	Curry (.808)	Curry (.990)	Curry (.988)	Curry (.986)
14–15	Curry (.922)	Harden (.479)	Curry (.489)	Harden (.473)	Curry (.604)	Curry (.629)	Curry (.590)	Curry (.618)
13–14	Durant (.986)	Durant (.661)	Durant (.813)	Durant (.734)	Durant (.775)	Durant (.773)	Durant (.773)	Durant (.773)
12–13	LeBron (.998)	Durant (.711)	LeBron (.882)	LeBron (.795)	LeBron (.854)	Durant (.901)	Durant (.901)	Durant (.901)
11–12	LeBron (.888)	LeBron (.499)	LeBron (.564)	LeBron (.437)	LeBron (.489)	LeBron (.704)	LeBron (.689)	LeBron (.658)
10–11	Rose (.977)	LeBron (.540)	LeBron (.634)	LeBron (.476)	Rose (.600)	LeBron (.523)	LeBron (.523)	LeBron (.540)
09–10	LeBron (.980)	LeBron (.721)	LeBron (.864)	LeBron (.826)	LeBron (.888)	LeBron (1.044)	LeBron (1.044)	LeBron (1.037)
08–09	LeBron (.969)	LeBron (.649)	LeBron (.869)	LeBron (.744)	LeBron (.923)	LeBron (.979)	LeBron (.980)	LeBron (.962)
07–08	Kobe (.873)	LeBron (.502)	LeBron (.482)	LeBron (.603)	LeBron (.568)	LeBron (.506)	LeBron (.506)	LeBron (.506)
06–07	Dirk (.882)	LeBron (.425)	Dirk (.437)	Kobe (.387)	Dirk (.576)	Dirk (.512)	Dirk (.512)	Dirk (.512)

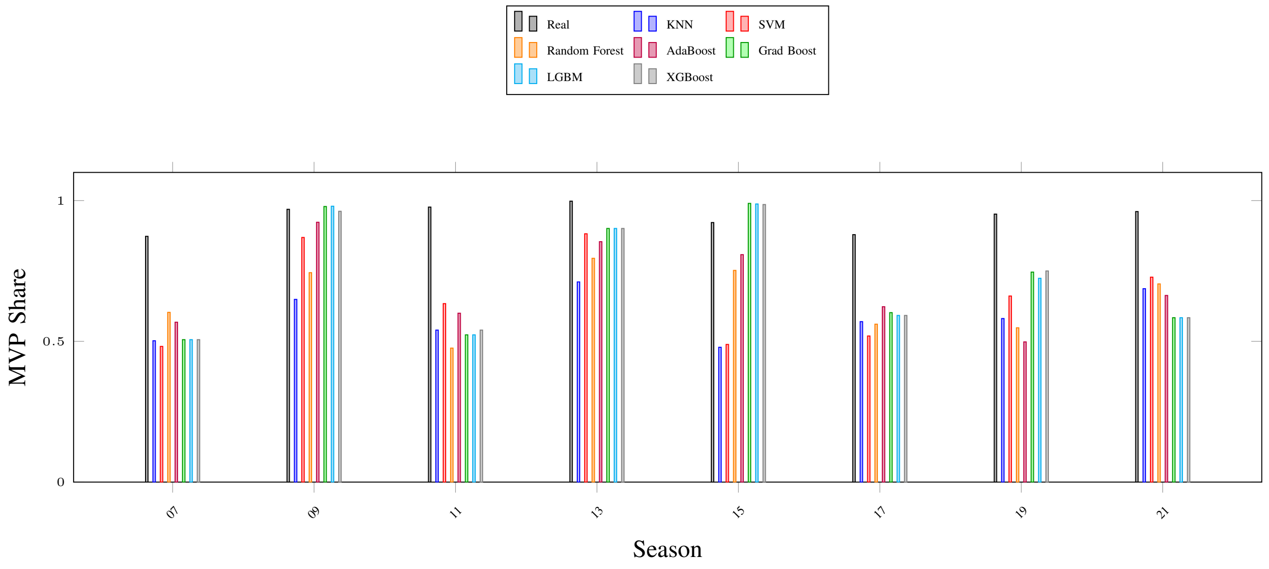


Fig. 3: MVP Share Predictions by All Models vs Real (Selected Seasons)