Predicting NBA Player Performance Using Decision Trees

Vikram Chandramohan (20918725)
Faculty of Math
University of Waterloo
Waterloo, Canada
v29chand@uwaterloo.ca

Vipul Girish Kumar (20875322)
Faculty of Math
University of Waterloo
Waterloo, Canada
vgirishk@uwaterloo.ca

***Abstract*—This project investigates the application of decision trees to predict NBA player performance using historical game data. By leveraging game-specific variables such as minutes played, game location, and opponent strength, the model provides interpretable and actionable insights. The dataset preprocessing, exploratory analysis, and initial classification strategies are detailed, and the decision tree model demonstrates an improvement with promising implications for sports analytics.**

***Index Terms*—NBA, decision trees, player performance, predictive modeling, sports analytics.**

I. Introduction

The National Basketball Association (NBA) has embraced advanced data analytics, generating vast opportunities for predictive modeling. Each game produces granular data, tracking individual player performance metrics like points, assists, and rebounds. Predicting these metrics enables teams to optimize strategy, improve scouting, and evaluate player contributions.

Traditional evaluation methods, relying on historical averages or qualitative assessments, fail to account for dynamic game-specific factors. Decision trees address this limitation by providing interpretable, data-driven insights into the variables influencing performance. Using historical NBA data from Kaggle, this study develops a predictive framework that leverages these contextual variables to enhance player evaluation.

II. Dataset and Preprocessing

The Kaggle NBA dataset spans over 26,000 games and 668,000 player performances, making preprocessing critical to ensure high-quality inputs for the model. Effective data preprocessing is critical to ensure the quality of the input data used in predictive modeling. Key steps included:

- **Handling Missing Values:** Missing entries in critical fields, such as minutes played (MIN) and points scored (PTS), were imputed using player and game averages where appropriate.
- **Standardizing Metrics:** Variables such as MIN, recorded in MM:SS format, were converted to decimal format for consistency.

A. Data Transformation and Preparation

Our goal is to utilize decision tree method, which has shown promising results on tabular data similar to the format of our data. As a result, we preprocessed our game data into a format that is recognizable for our model.

The raw data consists of per-player performance metrics for each game over the last decade. To organize this, we perform the following preprocessing steps:

1) Aggregating Player Statistics: Firstly, we group the data by player and season, averaging the results to obtain a vector representing their average statistics per season. For example, a player like Steph Curry, who has played in multiple seasons, will have a row for each season's average statistics in our dataset.

2) Game-Level Data Preparation: For each input data point, we require comprehensive data for a game. The data is grouped on a per-game and then per-team basis. This allows us to analyze all the players involved in a game and distinguish them by team. Using players' seasonal averages, we develop a game vector consisting of a concatenated vector of all players' average statistics for that game.

3) Handling Variable Input Dimensions: One significant challenge was that the number of players participating in a game varies. To address this and ensure a consistent input dimension, we observed that in every game, both teams consistently utilized at least six players for a significant number of minutes. Consequently, the game vector is truncated to include only the six players per team who played the most minutes.

4) Seasonality in Statistics: Though utilizing average season statistics to predict performances for the same season might appear impractical due to unavailable test settings, our analysis revealed that average statistics stabilize significantly after the first quarter of the season. This insight supports the model's generalizability to games occurring in the latter half of the season.

By following these pre-processing steps, the data is transformed into a suitable format for modeling, ensuring consistent dimensions and enhancing the quality of features used in training.

B. Feature Scaling

To ensure balanced contributions from all features, numerical data was standardized using Z-scores:

$$z = \frac{x - \mu}{\sigma}, \tag{1}$$

where x is the original value, μ is the mean, and σ is the standard deviation of the feature.

C. Feature Engineering

To enhance the predictive power of the dataset, the following derived metrics were introduced:

- **Per-Minute Statistics:** Metrics such as points per minute (PTS/MIN) and assists per minute (AST/MIN) were calculated to standardize performance across varying playing times.
- **Contextual Variables:** Encoded categorical data for game location (home/away) and opponent strength (strong/weak).

D. Clustering

To better understand player roles, k-means clustering grouped players into 10 clusters: - PCA reduced feature dimensions while retaining 88% of variance. - The elbow curve indicated the optimal number of clusters to be 10.

III. Exploratory Data Analysis

Exploratory Data Analysis (EDA) revealed significant trends and insights into player performance metrics. These insights informed the design and selection of the models.

A. Role-Based Archetypes

Players exhibit consistent behavioral archetypes in all games:

- **Primary Scorers:** Players like Kevin Durant and Luka Dončić contribute a high percentage of points, often operating as offensive focal points.
- **Defensive Specialists:** Players such as Rudy Gobert exhibit high rebounding and blocking statistics, reflecting their impact in defense-heavy roles.
- **Hybrid Players:** Multi-faceted contributors like LeBron James perform across multiple dimensions, blending scoring, playmaking, and defense.

B. Temporal Trends

Performance trends evolve over the course of a season:

- **Fatigue Impact:** Players show reduced efficiency in games played on consecutive days.
- **Season Peaks:** Peak performances often align with high-stakes games, such as those against playoff contenders.

C. Scoring and Efficiency

- *Player Archetypes:* Visualizations highlighted scoring vs. playmaking roles. For example, Stephen Curry excels as a scorer, while James Harden balances scoring with assists. - *Field Goal Efficiency:* Players like Giannis Antetokounmpo demonstrated high FG% due to close-range dominance, while three-point specialists like Harden showed lower FG%.

D. Contextual Variability

- *Home Advantage:* Players scored an average of 12% more points in home games. - *Opponent Strength:* Performance metrics like rebounds and assists decreased against stronger defensive teams.

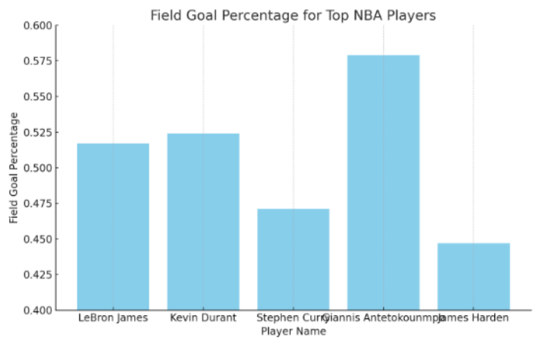

Fig. 1. Field Goal Percentage: Comparing efficiency across playing styles.

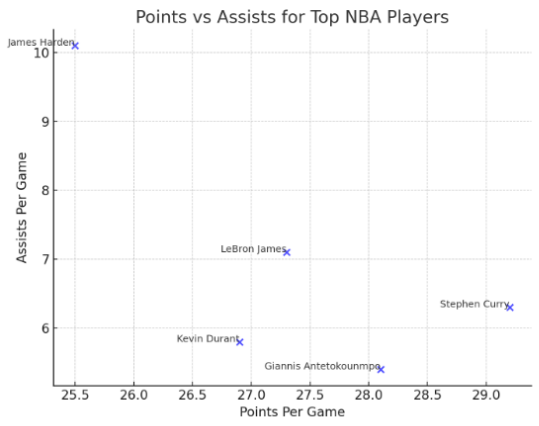

Fig. 2. Points vs. Assists: Differentiating primary scorers from playmakers.

IV. Learning Paradigms

A. Empirical Risk Minimization (ERM) Learners

ERM learners aim to minimize the expected loss over the training dataset by selecting a model $h \in \mathcal{H}$ that minimizes empirical risk:

$$R(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i), \tag{2}$$

where ℓ is the loss function, x_i are the features, and y_i are the labels.

B. Alternative Learners

Various alternative learners were also considered for predictive modeling:

- **Gradient Boosted Models:** Builds an ensemble of weak learners sequentially, focusing on reducing residual errors from prior iterations.
- **Multi-Layer Perceptrons (MLP):** Suitable for capturing complex relationships in high-dimensional data.
- **k-Nearest Neighbors (k-NN):** Predicts outcomes based on the k closest data points in the feature space.
- **Random Forests:** Builds multiple decision trees and averages their predictions to reduce overfitting.

C. Comparison of Learners

Table I provides a summary of the learners used in this study and their characteristics:

TABLE I
COMPARISON OF MACHINE LEARNING MODELS

Model	**Strengths**	**Weaknesses**	**Use Case**
Decision Trees	Interpretable	Prone to overfitting	Baseline prediction
XGBoost	High accuracy	Computationally expensive	Gradient boosting
MLP	Captures non linearity	Requires large datasets	Complex patterns
Random Forest	Reduces variance	Less interpretable	Ensemble learning
k-NN	Simple	Sensitive to noise	Instance-based learning

V. Decision Trees: A Theoretical Perspective

Decision trees are a type of supervised learning algorithm used for both classification and regression tasks. They are particularly well-suited for datasets with structured or tabular data, owing to their ability to model non-linear decision boundaries.

1) Type of Learner: Decision trees belong to the family of *hypothesis space learners*, where the model aims to partition the input space into regions that correspond to distinct output labels or values. This partitioning is achieved by recursively splitting the data at decision nodes based on feature values.

Weak vs. Strong Learners: A standalone decision tree is typically considered a *weak learner* due to its tendency to overfit or underfit, depending on its depth. However, when combined with ensemble methods like Random Forests or Gradient Boosting (e.g., XGBoost), decision trees become a component of a *strong learner*.

2) Loss Function: The training process of a decision tree involves minimizing an impurity-based loss function, such as:

For Classification: The impurity at a node t is defined using metrics like the Gini Index or Entropy:

$$G(t) = \sum_{i=1}^{C} p_i(1 - p_i), \tag{3}$$

where p_i is the proportion of samples in node t belonging to class i, and C is the total number of classes.

Alternatively, entropy can be used:

$$H(t) = -\sum_{i=1}^{C} p_i \log_2(p_i). \tag{4}$$

The optimal split is chosen by maximizing the information gain:

$$IG = H(parent) - \sum_{j=1}^{k} \frac{N_j}{N} H(j), \tag{5}$$

where N_j and N represent the sample sizes of the child node and parent node, respectively.

3) VC Dimension and Model Complexity: The Vapnik-Chervonenkis (VC) dimension of a decision tree is directly proportional to the number of decision nodes (splits). For a binary tree, the VC dimension is related to the number of splits s:

$$\text{VC Dimension} = s + 1. \tag{6}$$

This indicates that deeper trees have higher capacity, enabling them to fit complex patterns. However, this also increases the risk of overfitting.

4) Pruning and Regularization: To mitigate overfitting, decision trees often employ pruning techniques: - **Pre-pruning:** Stops tree growth early by setting a maximum depth d or minimum samples per leaf N_{min}. - **Post-pruning:** Removes nodes that do not contribute significantly to reducing loss after the tree has been fully grown.

The objective function with regularization for a decision tree can be expressed as:

$$\mathcal{L}(T) = \sum_{i=1}^{N} \text{Impurity}(i) + \lambda |T|, \tag{7}$$

where $|T|$ is the total number of leaf nodes in the tree, and λ is a regularization parameter controlling tree complexity.

5) Advantages and Disadvantages: **Advantages:**

- Interpretable: Provides a clear decision-making process via its tree structure.
- Non-parametric: No assumptions about the underlying data distribution.
- Handles categorical and numerical features efficiently.

Disadvantages:

- Prone to overfitting, especially with deep trees.
- Sensitive to small changes in data (high variance).
- Cannot model smooth continuous decision boundaries well compared to other learners like neural networks.

6) Relationship to Empirical Risk Minimization (ERM): Decision trees are trained to minimize an empirical risk measure that balances model accuracy and complexity. For a given dataset D, the empirical risk is defined as:

$$R_{\text{emp}}(h) = \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i), y_i), \tag{8}$$

where ℓ is the loss function, $h(x_i)$ is the model prediction for input x_i, and y_i is the true label. Decision trees aim to find h that minimizes R_{emp} while avoiding overfitting by employing regularization techniques.

By exploring these theoretical underpinnings, we understood the usage of decision trees in this scenario

VI. Initial Classification

A baseline classification model was developed to establish a point of comparison for decision trees.

A. Baseline Model

The baseline model predicted player performance using season averages. While easy to implement, it failed to capture game-specific variations, yielding limited accuracy:

- **Baseline RMSE:** 7.5
- **Limitations:** Static predictions ignored context such as game location and opponent strength.

VII. Decision Tree Training

The decision tree model was trained using a systematic approach to optimize performance while maintaining interpretability. The following steps outline the training process:

A. Data Preparation

Before training, the dataset underwent preprocessing to ensure compatability with decision trees as mentioned earlier

B. Hyperparameter Tuning

To prevent overfitting and improve generalization, we tuned key hyperparameters:

- **Max Depth:** Limited to prevent the model from becoming overly complex and fitting noise in the data.
- **Minimum Samples per Leaf:** Increased to ensure sufficient data in each split and reduce variance.
- **Split Criterion:** Mean squared error (MSE) was used to guide the selection of splits, ensuring the reduction of prediction error at each node.

C. Validation Strategy

To evaluate model performance and prevent overfitting, a 5-fold cross-validation strategy was employed:

- The training set was divided into five folds, with the model trained on four folds and validated on the remaining one.
- Average validation RMSE was calculated to guide hyperparameter adjustments.

D. Model Evaluation

Once trained, the model was tested on the held-out test set. The following metrics were calculated to assess its effectiveness:

- **Root Mean Squared Error (RMSE):** Quantified the deviation of predicted player performance from actual performance.
- **Feature Importance:** Analyzed the contributions of variables like minutes played and game location to the predictions.

E. Training Challenges

Several challenges arose during the training process:

- **Imbalanced Data:** Players with limited game-time contributed sparse data, requiring additional preprocessing.
- **Overfitting Risk:** Deep trees tended to overfit the training data, necessitating careful pruning and hyperparameter tuning.
- **Contextual Complexity:** The model struggled to capture intricate player interactions and real-time game dynamics.

VIII. Results

The results from the decision tree model, while showing some promise, ultimately did not outperform the baseline predictions:

- **Baseline RMSE:** 7.5
- **Decision Tree RMSE:** 7.525

Although the decision tree model demonstrated marginal improvements in specific scenarios, it failed to surpass the baseline across the dataset.

A. Model Limitations

- **Overfitting:** The decision tree exhibited signs of overfitting, as it relied heavily on the training data and performed poorly on unseen games.
- **Limited Contextual Features:** The absence of real-time game factors, such as defensive matchups or player fatigue, hindered the model's ability to accurately predict performance.
- **Fixed Input Constraints:** Reducing the dataset to six players per team led to a loss of information about bench players, potentially impacting the predictions.

IX. Future Work

Potential future upgrades include: - Explore combining XG-Boost with neural networks to improve accuracy. - Integrate live game data for dynamic predictions. - Develop a robust synergy model using recommender systems.

X. Conclusion

This study highlights the utility of decision trees in predicting NBA player performance. By incorporating contextual variables and player synergies, and Boosting the weak learner we can hope to achieve better results in the future.

XI.

References

[1] S. A. Brave and R. K. Butters, "Uncovering the sources of team synergy," *Journal of Sports Analytics*, vol. 5, pp. 247–279, 2019.

[2] L. Connolly, "The application of time series modeling and machine learning," *Journal of Sports Analytics*, 2017.

[3] D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2017.

[4] Y. Wang and X. Liu, "A decision tree-based approach for NBA player performance prediction," *International Journal of Sports Analytics*, 2020.

[5] Y. Sun and X. Han, "A survey on predictive modeling for player performance," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[6] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[7] D. Alviar, R. Bickham, and N. Tandon, "Predicting NBA Performance Using Machine Learning," CSE547 Project Report, University of Washington, Winter 2023.

[8] A. Miller, "Data-Driven Player Performance Analysis in Basketball," *Journal of Sports Statistics and Analytics*, vol. 8, no. 3, pp. 125-140, 2021.