Predicting the NBA Most Valuable Player Using Machine Learning

ML Mini Project

Soumil Jain PES1UG23CS587
Suraj Kumar PES1UG23CS617
Section J

Problem Statement

The goal of this project was to build a machine learning model capable of predicting the NBA's Most Valuable Player (MVP) based on historical player statistics. The MVP award is one of the most prestigious honors in basketball, given to the player who contributes most significantly to their team's success. However, MVP selection often involves complex patterns that go beyond raw scoring — including efficiency, team performance, and consistency.

The challenge was to use data-driven methods to identify which statistical factors best predict MVP winners and use them to forecast future results.

Approach

We framed the MVP prediction task as a binary classification problem where each player-season is labeled as either MVP (1) or non-MVP (0). The final workflow included the following major steps

Data Cleaning:

Removed leakage-related columns such as *Share*, *Pts Won*, *Pts Max*, and irrelevant index columns. Missing numerical values were filled using column means, while categorical values were filled using the mode.

Feature Engineering:

Ensured the dataset contained a reliable rebounding metric. When “REB” was not available, it was created using either TRB or ORB + DRB.

Final feature set used for modeling:

G, MP, FG%, 3P%, FT%, REB, AST, STL, BLK, PTS

Train/Validation/Test Split:

The dataset was split chronologically:

- **Training:** Seasons up to 2015
- **Validation:** 2016–2024
- **Test:** 2025

Feature Scaling:

All numerical features were standardized using **StandardScaler**, ensuring equal weight across different stat ranges.

Model Training:

Multiple models were trained and compared, including Logistic Regression, Ridge Classifier, Random Forest, and Naive Bayes. Class imbalance (1 MVP per season) was handled using *class weights* and *dynamic threshold tuning*.

Threshold Optimization & Calibration:

For models producing probability scores, *CalibratedClassifierCV* and threshold sweeping were used to maximize F1-score on the validation set.

Implementation Overview

The project was implemented in Python using the following key libraries:

- **Pandas & NumPy** for data handling
- **Matplotlib & Seaborn** for visualization
- **Scikit-learn** for model training, evaluation, and tuning

The workflow was executed in Jupyter Notebook, enabling modular development, visualization, and reproducibility. Each model's training, tuning, and predictions were logged and compared systematically.

Model Building and Training

Four different machine learning models were trained and evaluated:

1. **Logistic Regression (Baseline + Tuned + Calibrated):**
   Hyperparameters were optimized using RandomizedSearchCV, followed by probability calibration.
   Seasonal MVP accuracy: **44.44%**
2. **Calibrated Ridge Classifier (Best Model):**
   RidgeClassifierCV was used with logarithmic alpha values, and its probabilities were calibrated.
   Achieved the highest seasonal accuracy:
   55.56%
3. **Random Forest Classifier:**
   Trained with balanced subsampling and evaluated with dynamic thresholding.
   Seasonal accuracy: **11.11%**
4. **Naive Bayes (GaussianNB):**
   Very fast but performed poorly on this task.
   Seasonal accuracy: **0%**

Conclusions and Challenges

The final Logistic Regression model proved that MVP prediction is feasible using statistical data, correctly identifying real MVPs and balancing interpretability with accuracy.
Key challenges included data imbalance (only one MVP per season), subjectivity in MVP selection, and limited advanced statistics. Overall, the project successfully demonstrated how machine learning can uncover patterns in sports analytics and predict MVP outcomes with strong reliability.