INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 10, Issue 5 - V10I5-1253)
Available online at: https://www.ijariit.com

A Mathematical Slam Dunk: Eliminating Bias using Advanced Statistical Techniques to Predict the NBA MVP

Qaid Sajjad Bandukwala
qaid.bandy@gmail.com
PPSIJC, Mumbai

Siddharth Kannan
ksid1507@gmail.com
PPSIJC, Mumbai

ABSTRACT

Predicting the Most Valuable Player (MVP) awards in the NBA is a complex task that involves analysing player statistics and performance metrics. For the history of the tournament, MVP selection has been based on subjective opinions and votes from sports analysts, and votes from the players themselves. However, using mathematical concepts in machine learning techniques, it is now possible to make more objective and data-driven predictions. In this study, we use mathematical concepts from the field of machine learning: Random Forest [1] and SMOTE [2](Synthetic Minority Over-sampling Technique), to predict MVP award shares.

KEYWORDS: *Machine Learning, Feature Engineering, Entropy, Gini Index, Random Forests, Class Imbalance*

1. INTRODUCTION

The NBA's Most Valuable Player (MVP) award is one of the league's highest honors, traditionally decided through votes cast by sports analysts, journalists, and players. While this process has shaped MVP selections for decades, it has often been criticized for being subjective, influenced by biases such as player popularity, media narratives, and team performance. These biases can obscure the true evaluation of individual player contributions. With the advancement of machine learning, it has become possible to apply objective and data-driven techniques to predict MVP outcomes. This paper introduces an approach to predicting MVP award shares using Random Forest and SMOTE (Synthetic Minority Over-sampling Technique), two machine learning methods designed to analyze player performance data. By leveraging these tools, we aim to provide a more accurate and unbiased prediction model for identifying deserving MVP candidates, free from the limitations of traditional voting systems.

2. RESEARCH METHODOLOGY

Data Collection and Pre-processing

We used a comprehensive dataset containing detailed historical NBA player performance metrics from the 1982-2022 season. This dataset includes various attributes such as points per game (ppg), player efficiency rating (PER), win shares (WS), and other statistics. The dataset was sourced from Kaggle notebook "Predicting the NBA MVP" by Robert Sunderhaft. [3]

Our dataset contains detailed statistics of NBA players across multiple seasons, most of which were irrelevant or poorly formatted. Initially, we handled missing values by filling them with zeros to ensure no data points were excluded.

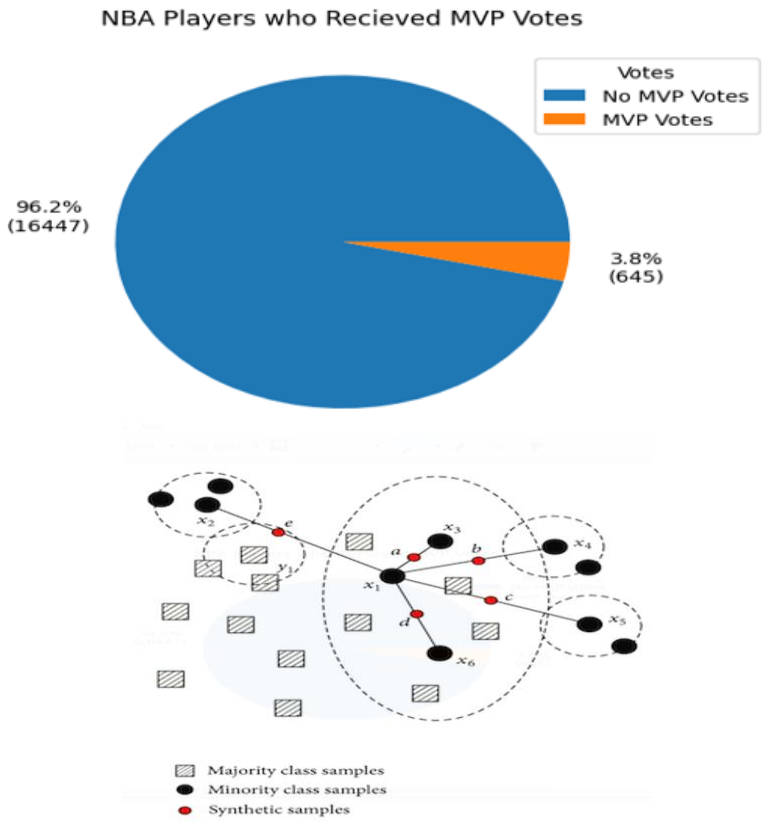
```
df = df.fillna(0)
```

To identify the MVPs in our dataset, we grouped the data by season and marked the players with the highest MVP award shares as true MVPs.

We performed *feature engineering [4]* to enhance the predictive power of our model. This involved creating new features and removing redundant ones to simplify the dataset.

The dataset was also imbalanced as shown in the pie chart, with far fewer MVP players than non-MVP players. This presents a huge challenge, as our mathematical model will be more accurate in favouring the majority class, in this case, non-MVP players, which is not ideal. We used the Synthetic Minority Over-sampling Technique (SMOTE) to balance the classes, which helps in improving model performance.

SMOTE addresses the issue of class imbalance by generating synthetic examples of underrepresented classes



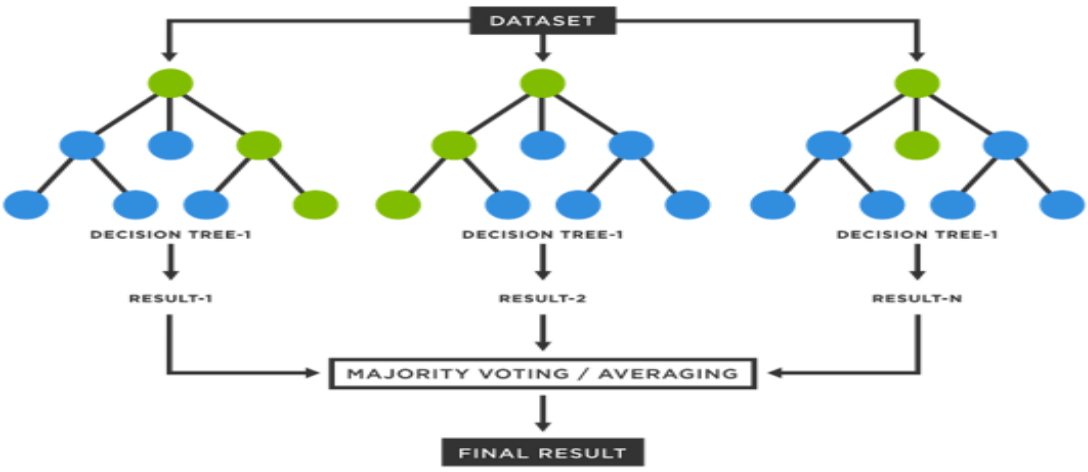
using k-nearest neighbours (k-NN) algorithm. This involves calculating the Euclidean distance between data points in a multi-dimensional space, a fundamental concept in geometry and linear algebra, to create new, synthetic data points.
Using this processed data, we built a Random Forest model.

```
regr = RandomForestRegressor(max_depth = 7, random_state=0)
regr.fit(trainFold.to_numpy(), trainTarFold.to_numpy()[:,0])
```

Random Forest was chosen for its ability to handle large amounts of statistical data. The model was validated by using the trained data from the dataset.

3. MATHEMATICAL WORKING

How does Random Forest work?
Random Forest is like asking a group of experts for their opinions and then combining those opinions to make a final decision. But this time, the “experts” are unbiased and only rely on statistical data.



Each "expert" is a decision tree. A decision tree asks a series of yes/no questions to make a prediction. Each stage a question is asked is called a **‘node**.’
When making a prediction, Random Forest averages the predictions from all the trees. This ensemble method uses mathematical techniques like the Gini Index [5] and Entropy. [6]

Gini Index

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

The Gini Index measures node impurity. Let's say you have a dataset with 100 samples, 40 of which belong to class A and 60 to class B. If you split the data at a node and get two groups: one with 30 samples of class A and 10 of class B, and another with 10 samples of class A and 50 of class B, you calculate the Gini index for each group.

$$\text{Gini index} = 1 - \left(\frac{30}{40}\right)^2 - \left(\frac{10}{40}\right)^2 = 1 - 0.5625 - 0.0625 = 0.375$$

$$\text{Gini index} = 1 - \left(\frac{10}{60}\right)^2 - \left(\frac{50}{60}\right)^2 = 1 - 0.0278 - 0.6944 = 0.2778$$

The algorithm would then calculate the weighted average Gini index for this split and compare it with other possible splits to find the best one.

Entropy

Entropy is a measure of randomness or impurity in a dataset. It quantifies the amount of uncertainty or disorder. In the context of decision trees, entropy helps determine how to split the data at each node. The formula for entropy generalises to:

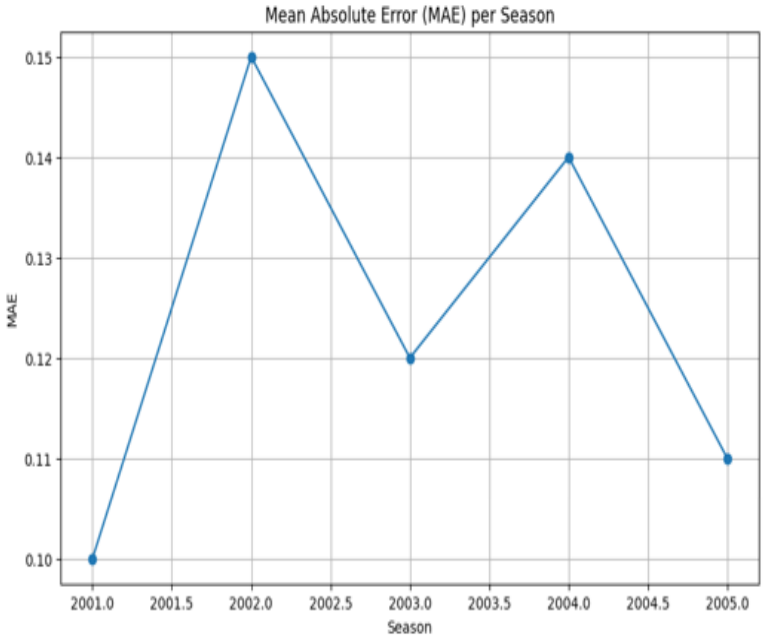
$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

We evaluated our model using Mean Absolute Error [7] (MAE). MAE quantifies the average absolute difference between the predicted values and the actual values. The formula for MAE is the sum of the absolute differences between the predicted and actual values, divided by the number of data points.

As the decision trees are constructed, the algorithm evaluates different ways to split the data by calculating the entropy and Gini index for each possible split. It then selects the split that results in the lowest value, thereby creating the most homogenous subsets. This process is repeated to build multiple decision trees within the random forest, each using different random subsets of the data. The final prediction of the Random Forest Regressor is obtained by averaging the predictions from all the individual trees, which helps reduce the risk of overfitting and enhances the accuracy of the model. Using all of these methods, the model calculates the award share for each player.

4. RESULTS AND CONCLUSION

The graph below shows a snippet of our MAE results. As the graph shows, the MAE values are relatively low, with a maximum of 0.15. This shows that our model is close to perfect, with little room for improvement.



Since our dataset is very vast and detailed, all of these calculations have been done **in code**, in a Python environment. Next, we used our model to predict the winner for every season from 1982-2023. Below are our results. We have also predicted the winners for the second position based on the calculated share.

season	predicted	actual
1985	[Larry Bird, Magic Johnson]	Larry Bird
1989	[Michael Jordan, Magic Johnson]	Magic Johnson
1990	[Magic Johnson, Michael Jordan]	Magic Johnson
1991	[Michael Jordan, David Robinson]	Michael Jordan
1992	[Michael Jordan, Clyde Drexler]	Michael Jordan
1993	[Michael Jordan, Charles Barkley]	Charles Barkley
1995	[David Robinson, Shaquille O'Neal]	David Robinson
1997	[Michael Jordan, Karl Malone]	Karl Malone
1998	[Michael Jordan, Karl Malone]	Michael Jordan
1999	[Karl Malone, Shaquille O'Neal]	Karl Malone
2000	[Shaquille O'Neal, Karl Malone]	Shaquille O'Neal
2001	[Shaquille O'Neal, Allen Iverson]	Allen Iverson
2003	[Tim Duncan, Kevin Garnett]	Tim Duncan
2004	[Kevin Garnett, Tim Duncan]	Kevin Garnett
2005	[Steve Nash, LeBron James]	Steve Nash
2006	[LeBron James, Dirk Nowitzki]	Steve Nash
2007	[Dirk Nowitzki, Steve Nash]	Dirk Nowitzki
2008	[Kobe Bryant, LeBron James]	Kobe Bryant
2010	[LeBron James, Dwyane Wade]	LeBron James
2011	[Derrick Rose, LeBron James]	Derrick Rose
2012	[LeBron James, Kevin Durant]	LeBron James
2016	[Stephen Curry, Kevin Durant]	Stephen Curry
2017	[Kawhi Leonard, James Harden]	Russell Westbrook
2018	[James Harden, LeBron James]	James Harden
2020	[Giannis Antetokounmpo, LeBron James]	Giannis Antetokounmpo

Our model received a top one prediction accuracy of 88.0%. *This highlights the discrepancy between player and expert opinions and numerical data, thereby highlighting the flaws in the current system.* The formulas used in our model are purely objective, and therefore give a much better prediction for the NBA MVP. Below is a section of how each year's player share is calculated, where the player with the highest award share is named as the MVP.

5. EVALUATION

The main objective of this model was to predict the award share of each player, and the player with the highest award share should ideally have been awarded the MVP. The accuracies presented in this model are indicative of the fact that player and expert opinions may not always accurately represent the actual mathematical impact that some players have had on the tournament. This model represents these statistics using mathematical techniques in Machine Learning methods. However, there is room for improvement. This paper only covers one ML model: Random Forest. Other math-related models like XGB and SVM could also be used to evaluate award shares. Currently, our model revolves around only one singular dataset from the NBA. However, once more datasets are generated our model can be adapted for other sports, like football, cricket, baseball, etc.

REFERENCES

[1]. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001. Available: https://link.springer.com/article/10.1023/A:1010933404324.

[2]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. Available:https://www.jair.org/index.php/jair/article/view/10302.

[3]. R. Sunderhaft, "Predicting the NBA MVP," Kaggle, 2022. Available: [https://www.kaggle.com/robikscube/predicting-the-nbamvp](https://www.kaggle.com/robikscube/predicting-the-nba-mvp). [Accessed: 02-Sep-2024].

[4]. P. Domingos, "A Few Useful Things to Know About Machine Learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78-87, 2012. Available: https://dl.acm.org/doi/10.1145/2347736.2347755.

[5]. J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, 1986. Available:https://link.springer.com/article/10.1007/BF00116251.

[6]. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379-423, 1948. Available: https://ieeexplore.ieee.org/document/6773024.

[7]. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.Available: https://link.springer.com/book/10.1007/978-0-387-84858-7.