University of Michigan - Ann Arbor Date: 6th December, 2023

SI 670: Applied Machine Learning

Final Project Report

on

Future All-Stars: A Data-Driven Approach to NBA Player Predictions

Contributors:

Group 5

Tzu-Yu Peng (typeng@umich.edu)

Te-Hsiu Tsai (tedtsai@umich.edu)

Divyam Sharma (divyams@umich.edu)

Contents:

1. Introduction

After 2015, the NBA entered into a new broadcasting deal, significantly boosting the financial fortunes of many NBA teams and leading to a considerable increase in the salary cap compared to previous years. This financial windfall has empowered players to negotiate higher salaries, setting new benchmarks in the league. Noteworthy examples include Nikola Jokic, who secured an annual average contract exceeding fifty million with the Denver Nuggets in 2022, and Jaylen Brown, who inked a contract with the Boston Celtics in 2023, boasting an annual average surpassing sixty million.

The impact of this financial shift extends beyond individual player contracts. The draft has become a pivotal element in the NBA, serving as a primary avenue for teams to discover and secure foundational players. Consequently, the ability to identify and sign contracts with the right players, especially those destined to become future All-Stars, has become a critical strategic imperative for teams.

Furthermore, the timing of contract negotiations plays a crucial role in securing valuable talent. Securing a contract with a potential star player immediately after the expiration of their initial contract proves advantageous. During this window, teams can leverage relatively lower contract terms to ensure the commitment of a promising player who is likely to attain All-Star status in the future. This strategic approach allows teams to optimize their roster composition while managing budget constraints effectively.

2. Related Work

We aim to predict a player's probability of becoming an All-Star in the future, using a unique dataset from the Oklahoma City Thunder team. This approach differs from common predictions of team performance or next year's All-Star lineup. Our dataset features distinctive metadata-style characteristics, setting it apart from typical internet datasets. We have referenced the most closely related work on next-year All-Star predictions from Google Scholar as an example here.

3. Dataset

We used datasets from Oklahoma City Thunders. These datasets involved players_stats.csv, Team_stats.csv, and Awards_Data.csv.

Team_stats.csv : This dataset has stats for each team in every season such as the team’s offensive rating, defensive rating, net rating, No. of Wins and Losses in the season

Awards_data.csv : This dataset has awards data for each player in every season. The award of interest for us which we will be predicting for the players will be whether they will make it to ‘All Star Game’.

Player_stats.csv : This dataset has a variety of stats and measures for each player in the team's every season such as minutes played, field goals %, Number of assists, steals, blocks, and turnovers

a) Preprocessing

Columns Dropped: We strategically eliminated several columns, including "player," "draftyear," "season," "nbateamid," and "team," each for distinct reasons.

- **Player**: Containing player names, was excluded for clarity and brevity.
- **Draftyear**: This column was deemed non-contributory for future predictions, as it lacked representation for upcoming years, leading to its omission.
- **Season**: This column underwent modification, transformed into the year during which the player actively participated, enhancing the dataset's interpretability.
- **Team and nbateamid:** These two columns were driven by the rationale that team information may not significantly contribute to a player's likelihood of becoming an All-Star.

Consequently, these columns were dropped to streamline the dataset and focus on more impactful features.

Data Join: We enriched the dataset by combining players' awards data with their corresponding statistics. This integration aimed to provide a comprehensive view, aligning individual achievements with performance metrics.

Missing Value Imputation: Addressing missing values in the "draftpick" column was pivotal. A thorough investigation revealed that these gaps predominantly stemmed from undrafted players. To rectify this, we opted for imputation, assigning a value of 100, considering that the maximum pick typically hovers around 60 each year.

Data Aggregation: Recognizing that players may have multiple entries in a given year due to trades or team changes, we executed a data aggregation strategy. This involved consolidating player statistics through methods like aggregation or weighted averaging, fostering a more holistic representation.

Data Transformation: Focusing on the initial four years of a player's career, aligned with the standard duration of a first-round pick's rookie contract, we streamlined the dataset. The transformation into a wide format ensured clarity, with each statistical metric boasting four distinct columns to encapsulate the four-year trajectory. To account for players with careers shorter than four years, we tactfully handled these instances by assigning zero values, maintaining the dataset's structural integrity.

4. ML Approach

This is a classification problem, where we are trying to classify a NBA drafted player as a potential ALL-Star ever in the future just with their initial 4 years of performance in the data. We label All-Star as 1 and Non-All-Star as 0 for our reference.

Metrics: In this task, we mainly optimize for **Precision** to minimize **False Positive**, use accuracy as well, but mainly focus on Precision. The reason for focusing on Precision is as spending a lot of money on a falsely predicted All-Star player can hit the teams significantly with their cash allocation and resources with no proportionate return. Hence **missing out on a potential-All-Star predicted as not one (FN) seems reasonable** but any **False positive could hurt badly** and hence the focus on **optimizing for Precision**.

Baseline Model: Naive Bayesian, KNN
The two simplest classifiers are our baseline to verify if our analysis is effective.

Model Tuning:

Since the variables involved in our classification tasks are mostly continuous and simply increasing variables, we chose some classification models that are powerful in handling linear classification and imbalanced samples **(e.g. Logistics Regression, AdaBoost)**. Also, we tried a stacked model of logistics regression with tuned models.

Evaluation

We tuned the hyperparameters with all models with **5-fold cross validation**, and mainly observed the **average precision and accuracy** of each model. After selecting the best parameters, we compared the average precision of each model and selected the best one for predicting test data.

5. Result and Discussion

Tuned Parameters Results:

Table 1: Best Hyper-parameters with average Precision and Accuracy

Model	Hyper-Parameters	Average Precision	Average Accuracy
Logistics	penalty = L2, C = 1	0.81	0.96
SVM	degree = 2, C = 5	0.83	0.96
RFM	n_estimators=100, max_depth=4	0.87	0.97
AdaBoost	n_estimators=200, max_depth=4, learning_rate=1	0.82	0.96
XGBoost	n_estimators=200, max_depth=3, eta=0.01	0.8	0.95

Random Forest model with tuned Parameters seems to have the best performance on not only Precision but also accuracy in this process. The next step we’ll use a Logistics Regression-base model to combine multiple models to see if the prediction can be further improved.

Feature Importance:

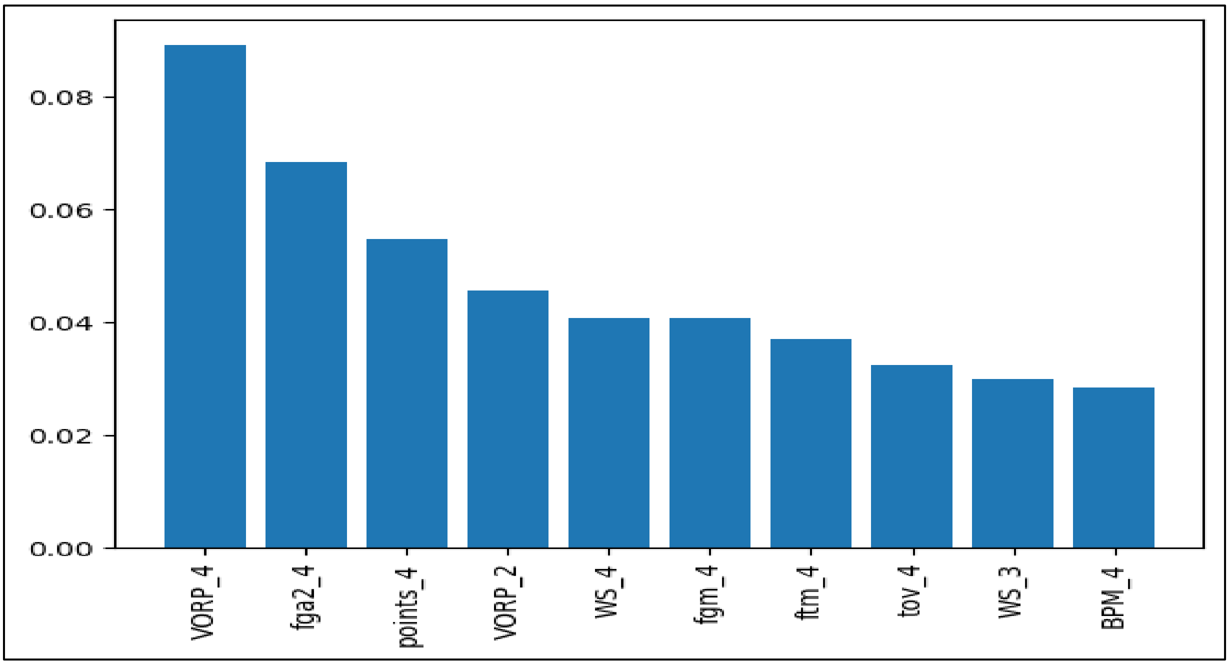

Figure 1. Feature Imporatance

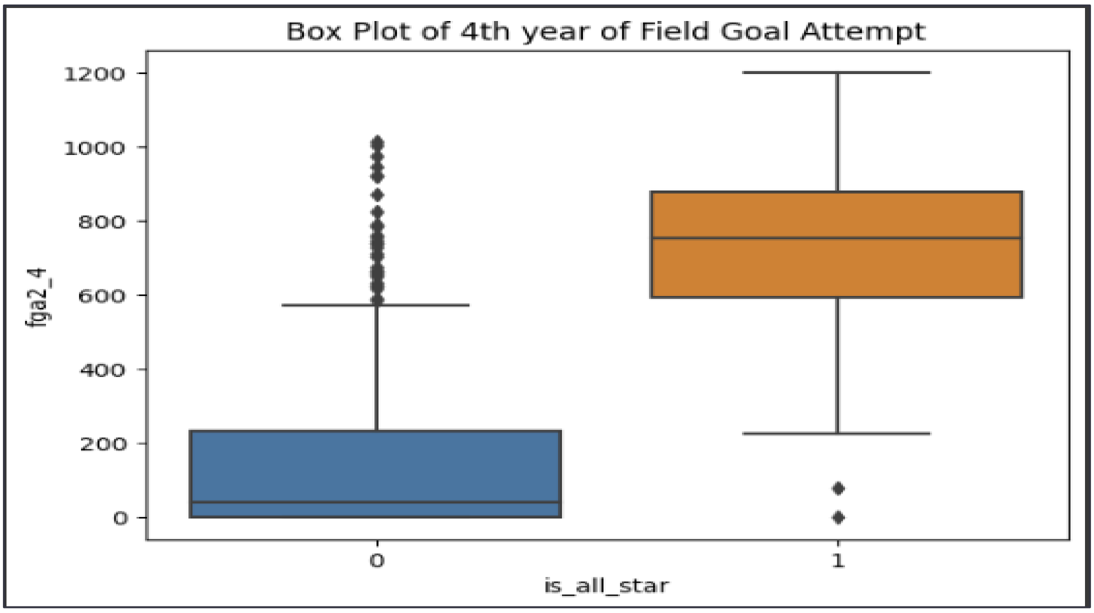

Figure 2: Field Goal Attempt (2 Pt) Between All-Star(1) and non-All-Star(0) Players

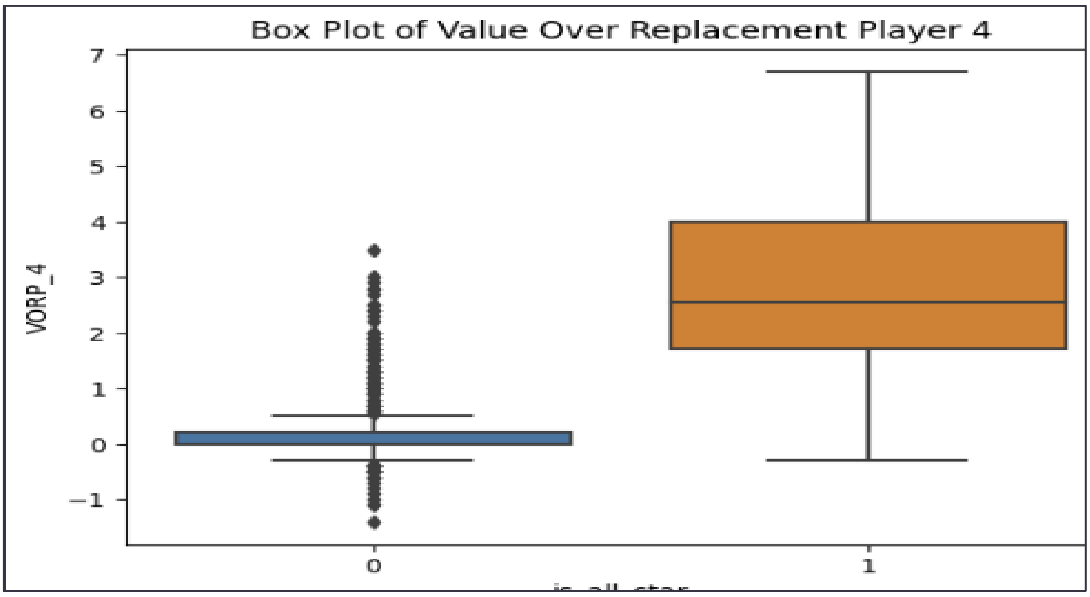

Figure 3: Box Plot of Value over Replacement Player for All-Star(1) and non-All-Star(0) Players

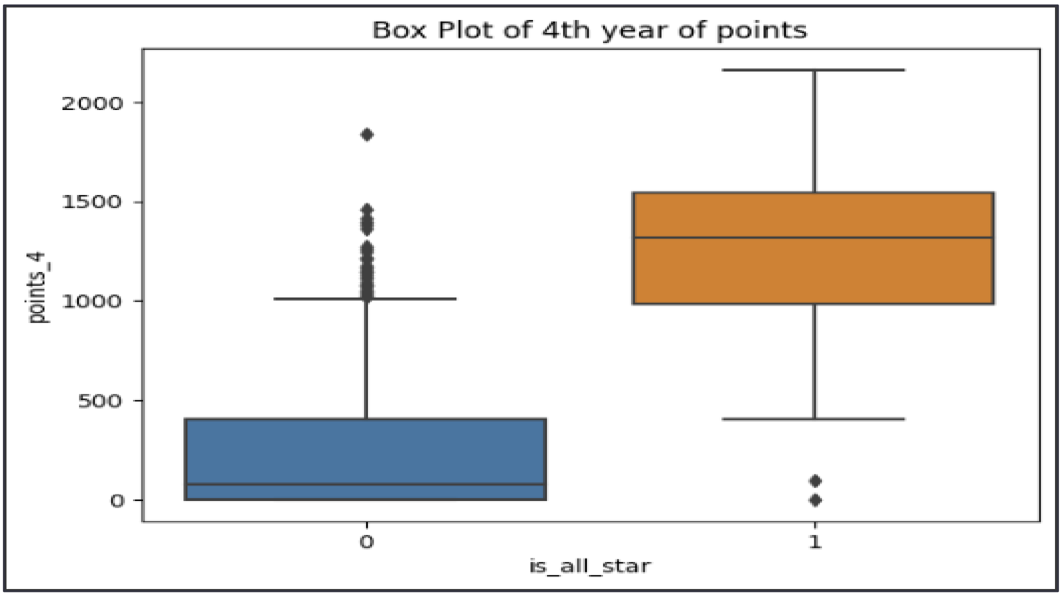

Figure 4: Box Plot of 4th Year of Points for All-Star(1) and non-All-Star(0)

Inferences:

We selected the top 10 features by our tuned Random Forest model. From the feature importance plot, the most important features that came out were VORP_2 and fga2_4 which mean Value Over Replacement that a player brings when he is included in the team over the person that he replaces and fga2 means the field goal attempts that the player made for 2 pointers shoots. The _2 and _4 signify the year’ s data as we are training the model only on the first 4 years of data for the drafted players.

Model Selection:

Table 2: Tuned Models with their Precision, Accuracy and Recall Scores

Model	Precision	Accuracy	Recall
Bayesian	0.28	0.89	0.65
Logistic[1]	0.81	0.96	0.66
KNN[2]	0.62	0.94	0.38
Random Forest[3]	0.87	0.97	0.65
SVM[4]	0.83	0.96	0.5

Stacked Model	**0.90**	**0.97**	**0.67**
AdaBoost	0.82	0.96	0.65
XGBoost	0.8	0.95	0.62

We tried models one by one with Logistic and Bayesian being the baseline. Precision for Bayesian was very poor (0.28), while for Logistic it was 0.78. Then we stacked the models with base estimators as [Logistic Regression, KNN, Random Forest and SVM] and used Logistic Regression again as being the meta/final estimator. The stacked model helped us realize the maximum precision **(90%)** with a good accuracy **(97%).**

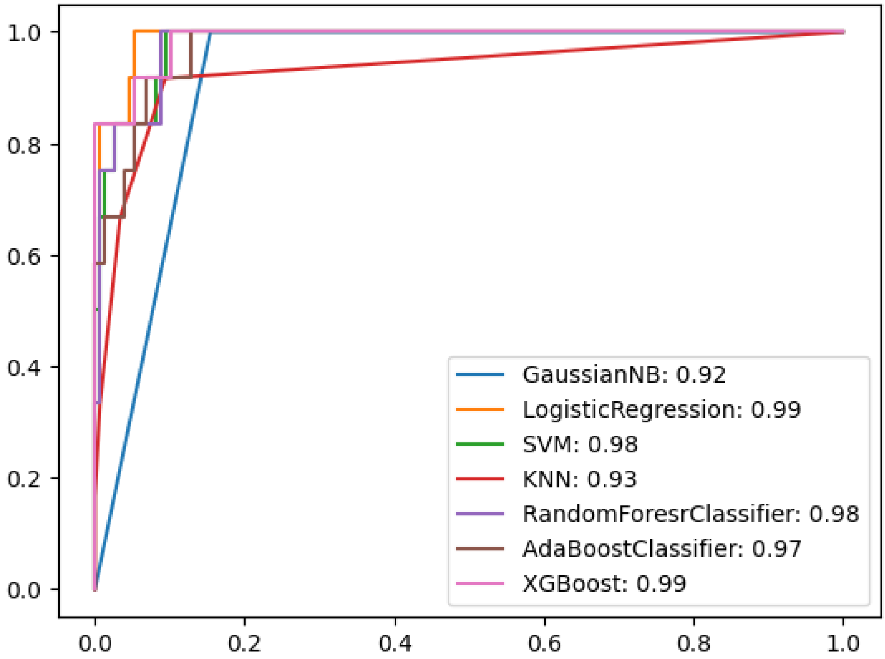

Figure 5: ROC Curve with AUC Score for single Models

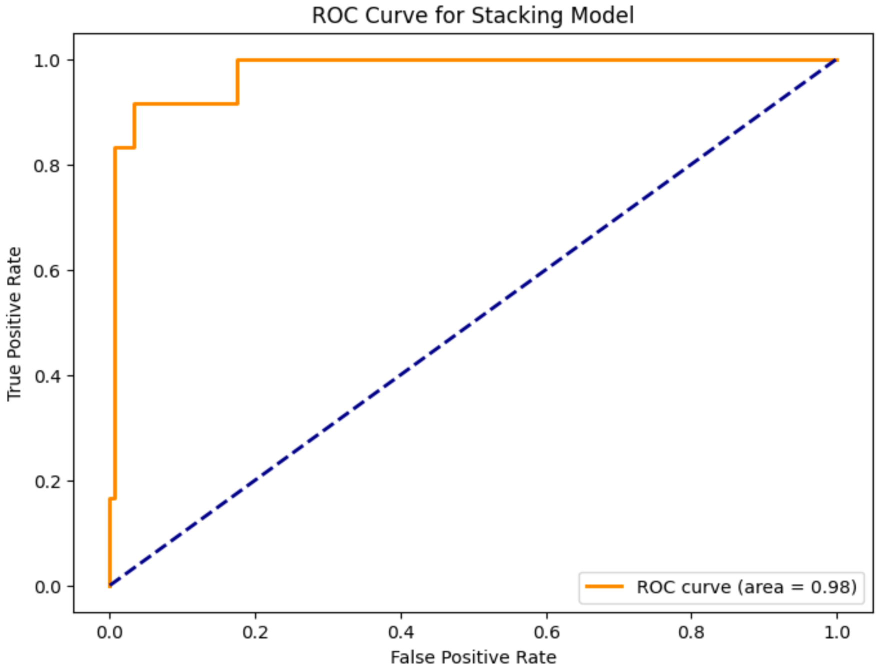

Figure 6: ROC Curve with AUC Score for the Stacked model

For the test data, i.e. for players drafted in 2018, out of 100 players, only 4 players made it to A ll-Star-Players till 2023. Our model, by observing their first 4 years of performance for 2018-20 21 was able to predict with 100% Precision and 75% Recall, by just misclassifying(FN) one Pla yer as Non-All-Star with no False Positives.

Table 3: Confusion Matrix for Predictions on Players Drafted in 2018

	Actual All-Star	Actual Non-All-Star
Predicted All-Star	3 (TP)	0 (FP)
Predicted Non-All-Star	1 (FN)	96 (TN)

2018 Predictions
1. Shai Gilgeous-Alexander
2. Luka Doncic
3. Trae Young
4. Jaren Jackson Jr

The only player misclassified as Non-All-Star (FN), with no False Positives.

6. References:

All our work is original and no open-source code has been used or approach copied in the same context, to the best of our knowledge. We did take help from ChatGPT for coming up for some complex code while preprocessing the data.