THE OHIO STATE UNIVERSITY

Author: R. Price

December 12, 2024

Abstract

Sports gambling has been growing increasingly prevalent over the past decade and shows no sign of slowing down soon. Nearly 63 million people across the USA and Canada participated in a fantasy sports league last year. Within the sports industry, and more specifically, the sports gambling industry, there is an incredible amount of data available to analyze to give one a potential edge. This project, therefore, aims to use said data to create a hypertuned RandomForestRegression model that is trained on past NBA players' metrics and how they progressed through successive seasons, and can predict current players' performances in the upcoming season with a certain level of accuracy. The model utilized ten different datasets and was trained on both inactive and active NBA players from 1974 to 2024, and then was used to predict metrics for the 2025 season. The model will then find the optimized ten-player lineup that will maximize fantasy basketball points, while following certain constraints. While the optimization model will be customizable, it will follow the standard NBA Fantasy rules, including lineup quantity, salary cap, and given point system, as a default. The model will input ten different data files encompassing over 50 years of NBA data and return the ten-person optimized fantasy team lineup.

1 Introduction

Fantasy sports have been steadily growing in popularity in the United States since their conception. Analysts across the world have attempted to create models that will predict game outcomes, total points scored, individual player performances, and thousands of other predictions. There is no limit to the number of metrics that predictive models created by these sports analysts have attempted to project. There is potential monetary gain for those who are able to successfully utilize the data provided to make an educated guess when it comes to gambling, as opposed to a gut-instinct approach.

A model with greater success than pure guesswork would hold great significance to many people who participate in fantasy sports, not only in the United States but across the world. There are thousands of metrics and near-infinite data that can be gathered from the sport of basketball, which is why so many analysts have attempted to use this data to make educated guesses on upcoming season outcomes. Likewise, this model would also have gambling applications. While there are a great many factors that affect a player/team's performance for a given game/season, we can use a few of the most prominent and influential factors to train our model.

Our algorithm receives a variety of player statistics (including age, height, weight, draft round pick, seasons of experience, position, and team), along with actual season metrics per game (games played, minutes played, rebounds, assists, steals, blocks, turnovers, as well as all-attempts/successful-attempts for: overall shooting, 3-point goals, 2-point goals, and free throws) as input. This input was found across ten different datasets, which were all imported into the main Jupyter notebook, cleaned, and combined into one large dataset. Each player from every season since 1974 is utilized.

Our model processes this dataset through a hypertuned RandomForestRegression model. Both a linear regression model and a time-series forecasting model were tested with this dataset, as these were common models found in related projects, but it was determined that a RandomForestRegression model has greater accuracy, especially when hypertuned. This model was trained on our full dataset. It first grouped each unique player's data together chronologically and then used the previous five seasons of data to predict the metrics from the following season, which could be verified for accuracy. It was determined that this model accounted for around 75 percent of the variation in the predicted metrics.

After the model was trained, a new dataset was created that only included active players and predicted their metrics for the coming 2025 season using the previous five seasons. Assuming the predictions are accurate, the model then optimizes these predictions to maximize the total fantasy basketball points a combination of ten players would earn, while following the given constraints. In short, our algorithm receives individual NBA players' metrics across different seasons as input. It processes this through a hypertuned RandomForestRegression model to predict active NBA players' metrics for the coming 2025 season, and then optimizes these predictions to determine the ideal ten-person fantasy basketball lineup, given a variety of constraints.

2 Related Work

Many people, both professional and novice, have attempted to create their own predictive models for fantasy teams and gambling applications. The ten datasets used were found on a website called Kaggle, where one can download the data freely and also upload notebooks showing how one used the data in their projects. Sarpeco was one such individual, who utilized the 'NBA Database'[7] dataset and used said dataset to predict the salaries of different players based on their performance metrics and combine/draft statistics[4]. The gathering of the data and the techniques used to clean said datasets was very informative and helpful when it came to cleaning my own. Sarpeco would often average the non-null values of a row/column when there was a null value within the set. However, Sarpeco did not prioritize marking up his notebook while cleaning the data, which made it difficult to understand what techniques he was using at times. When it came to predicting player salaries, a simple linear regression model was used. I do, however, believe a more complex model would be more appropriate for the large and non-linear data that is being inputted.

Duy Le, similarly, used the dataset '2023-2024 NBA Player Stats'[6] to create his notebook: 'NBA EDA predict points using RandomForestRegressor'[3]. As explained by the title, Le used RandomForestRegressor to predict the points scored in a given game. RandomForestRegressor is a very effective model to use in this situation. Le used a StandardScaler() and OneHotEncoder() to prepare the data for the RandomForestRegressor, which made his code more effective. I will say that there was little clarification as to how effective the model itself was. Le printed the RMSE of the model's prediction, but made no indication of how the model could be used for future applications, or how effective the model is. I believe a RandomForestRegressor would be more effective than a simple linear regression model, and may be a contender for the most effective model.

Using time-series forecasting to build my model was suggested by Dimitre Oliveira, whose notebook titled 'Deep Learning for Time Series Forecasting'[2] gave me a model from which I could base my time-series forecasting model. Dimitre Oliveira has an astute analysis of how to build such a model and why it would be effective in a situation such as when creating a fantasy basketball optimizer. I gained great insight into this foreign topic based on this notebook, which I had initially believed to be too difficult to accomplish (as we had not gone over this technique in class). However, after determining it would be a technique worth trying, it made sense to learn how to use it effectively to make my model as accurate as possible.

The most influential existing model found was by Amir Hossein Mirzaei, titled 'NBA Players Scored Points Prediction'[1]. He did a lovely job of visualizing the data and testing various methods, including linear regression and RandomForestRegressor, to find the most accurate model when it came to predicting the number of points a player will score in a given game. He also does a great job of marking up the notebook and explaining what each code block is attempting to do. I also appreciate how the notebook was visually

appealing and allowed me to clearly see what was being done to manipulate the data. Many of the more interesting graphs I used were created by manipulating his code, and he will be given proper citations below. Significant inspiration was taken from this workbook and how Mirzaei approached the issue: first with thorough visualization, then by using multiple different model types to find the ideal one.

These four notebooks were strong sources of inspiration for myself and this project. It was through using these examples that I was able to create as effective a model as I did.

3 Dataset

A variety of datasets will be utilized to properly train our model. The first of the datasets is titled 'NBA Database'[7], created by Wyatt Walsh. This dataset contains 16 different files, each of which contains different metrics. The files I will be using are titled 'common player info' and 'draft combine stats'. These two files include combine statistics (including height, weight, wingspan, performance metrics, etc.) and draft statistics (including round pick number, team, roster status, etc.). This data file will solely be used for combine/draft statistics, which are great metrics to determine a player's future performance.

The second dataset that will be used is titled 'NBA Players stats since 1950'[5], created by Omri Goldstein. I will be using the file 'playerdata.csv' to obtain individual player's statistics, including position, salary, minutes played, games started, Player Efficiency Rating, points scored, rebounds, steals, assists, blocks, shooting rate, turnover rate, efficiency rate, among many others. These statistics will be crucial to determining how a player's performance will change across seasons. While this dataset is very informative, it only contains players from 1950 to 2017. This would be good to help train the dataset. However, we need current players to accurately create a potential fantasy team lineup.

The six additional datasets come from '2021-2024 NBA Player Stats'[6] (contains both playoff and regular season files for each year, totaling six files), created by Vivo Vinco. This contains the same information as the dataset above but has the past three seasons, so we can use current players to create our optimized fantasy team. We will also be utilizing the dataset 'NBA player stats from 2018 to 2021'[8], created by Sumit Redekar, which, as you can probably guess from the title, will give us the last of the data needed to cover our desired years without breaks.

After cleaning the dataset, we have a total of 26,489 rows and 30 columns. When splitting our train and test data, we will perform a simple train-test split (80/20 split) for our RandomForestRegression model. Preprocessing techniques include one-hot encoding for player names, label encoding for team names, location-mapping for player positions, as well as accounting for any anomalies in the data and adjusting them accordingly. I also had to ensure all datasets were merged correctly without losing too much data and ensure that rows with greater than 20 percent zeros were eliminated while also accounting for any NaN values (either replacing with a 0 or the mean value, depending on the situation). We are left with a dataset that includes all players' season statistics since 1950 (with enough data available to be considered valid). Each row represents a different player from a specific season in ascending order.

To begin visualization, each metric's distribution was displayed in a historgram.

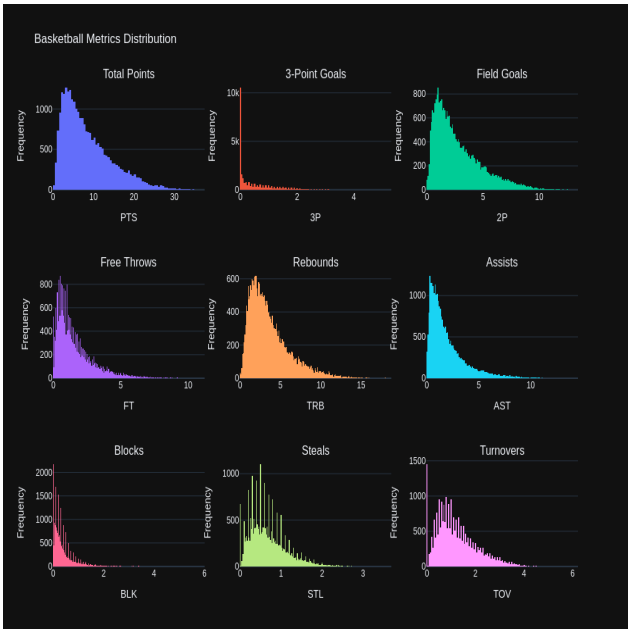

Figure 1: The nine metrics predicted are: total points, 3-points scored, 2-points scored, free-throws scored, rebounds, assists, blocks, steals, turnovers. This figure displays the distribution of each metric per player per game.

The data information was printed, as well as the description of our dataset. From that, a correlation matrix (shown below) was constructed to see how different features influence each other.

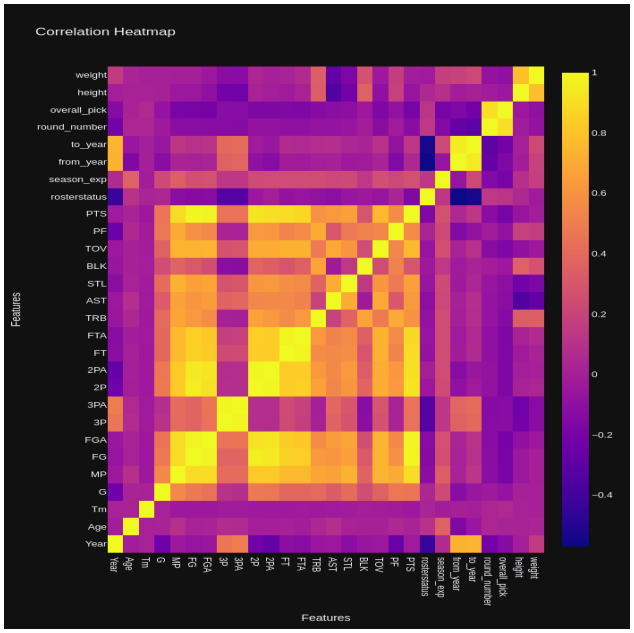

Figure 2: The correlation matrix of our original 28 features within our dataframe.

Other graphs constructed included a radar chart, which selects five different players at random and displays each of the nine statistics being predicted for each player on a single chart to show the similarities and differences between these five randomly selected players. This graph can be found in the Appendix. Along

with that graph, one can also find a graph displaying the ten best defensive players and offensive players, sorted by most blocks and steals (best defensive player) and highest points scored (best offensive player).

Twelve different scatter plots were also constructed to show relationships and correlations between different metrics. These correlations were found using the correlation matrix above.

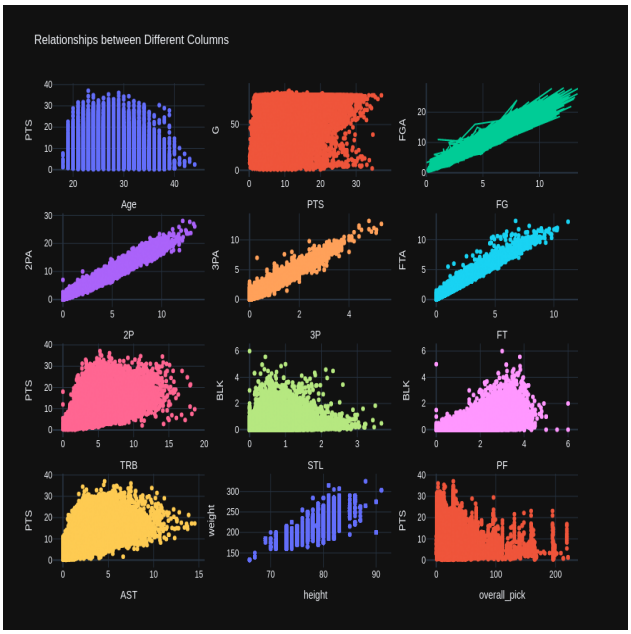

Figure 3: Twelve scatter plots highlighting the relationships between different metrics.

Now that we have visualized our data and the correlations within, we can move on to choosing and hyper-tuning a model.

4 Methods

Three different models were tested: Linear Regression, RandomForestRegression, and time-series forecasting. These were the three most common models found when creating models with goals similar to mine. All three models were used to predict points scored in the following season using the previous season's metrics. Both the mean squared error and the R score (which indicates the proportion of variance in the dependent variable that can be explained by the independent variables) were printed and compared. The model with the highest R2 score was deemed the most effective and then hypertuned using GridSearchCV to predict all nine metrics. The best model was then fit to our new dataset, which was a subset of our original dataframe that includes only currently active players, and used to predict how they would perform in the upcoming season. The three models used are described in depth below.

4.1 Model 1: Linear Regression

A Linear Regression model is the simplest of our three models. It is a supervised learning algorithm used for regression tasks. Its goal is to find the relationship between a dependent variable and one or more independent variables, or features. In this case, around 20 different factors are believed to influence our dependent variables, so multiple linear regression is needed, with the equation shown below:

$$y = b + w_1x_1 + w_2x_2 + ... + w_n * x_n \tag{1}$$

Where y is our predicted value, b is the bias, x_n values are our independent variables/features, and the w_n values correspond to the weights of our independent variables. The model is trained using an optimization algorithm to minimize the loss function, which is the "Mean-Squared Error" (MSE) in our case. Essentially, the model attempts to find a relationship that minimizes the residuals between our predicted and actual values. Below, we can see the residual plot obtained when we tested our data using this method.

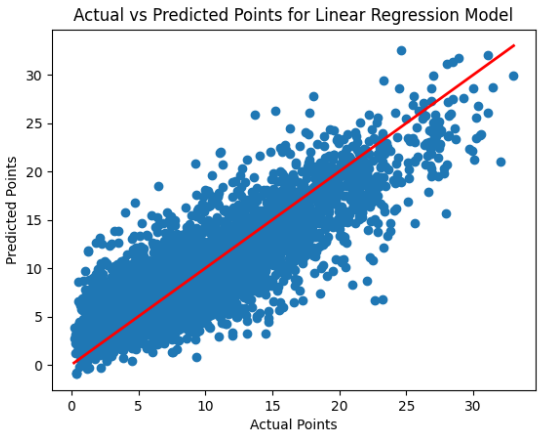

Figure 4: Residual plot obtained using a Linear Regression Model.

4.2 Model 2: RandomForestRegression

A RandomForestRegression model is a collection of decision trees that aim to overcome the two main drawbacks of decision trees: overfitting, sensitivity to the distribution of training data. This model trains a large number of randomized trees (randomized through bootstrapping and using a small selection of the features) and averages the results. The model then passes each desired sample through all of the trees and aggregate the results through voting. Below, we can see the residual plot obtained when we tested our data using this method.

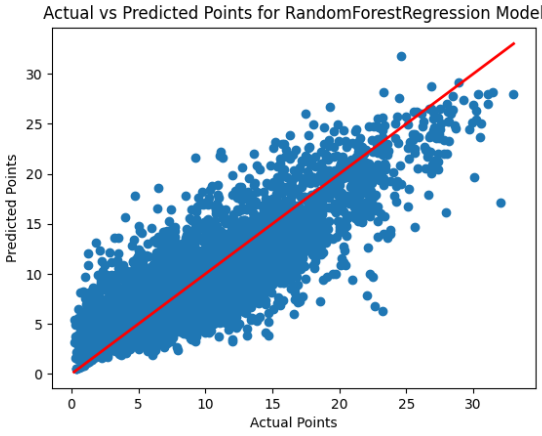

Figure 5: Residual plot obtained using a RandomForestRegression Model.

4.3 Model 3: Time-Series Forecasting

A time-series forecasting model, in our case, is a predictive modeling technique specifically designed to predict future metrics based on previously observed data points over each season. Our model is built using the Sequential() class, and we add multiple layers: an LSTM layer (a type of Recurrent Neural Network, or RNN) to handle sequences of data, and a Dense layer (with 1 unit, since we are predicting only 1 target). We then compile the model and test it by minimizing the Mean Squared Error (MSE). Below, we can see the residual plot obtained when we tested our data using this method.

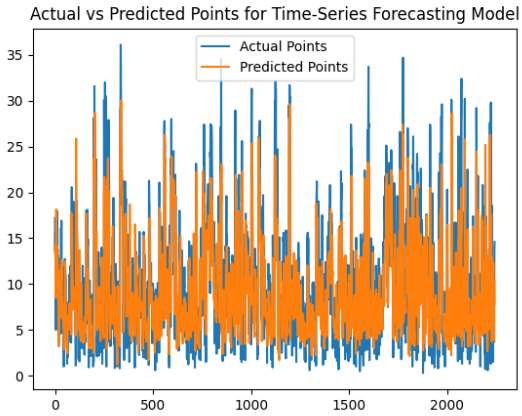

Figure 6: Residual plot obtained using a Time-Series Forecasting Model.

Below is a table containing both the MSE and R2 score of each of our three models.

	Linear Regression	RandomForestRegression	Time-Series Forecasting
MSE	10.058	9.964	12.594
R2	0.732	0.736	0.713

Table 1: MSE and R2 scores for all three models.

5 Results/Discussion

An analysis of the three models reveals that the RandomForestRegression model has the lowest MSE value, as well as the highest R2 (coefficient of determination) value. The R2 score tells us the proportion of variance in the dependent variable that can be explained by the independent variable, so a higher score is better.

Now that we have selected our model, a grid of parameters was constructed, and GridSearchCV was used to determine the ideal combination of parameters that would maximize our R2 score. It was found that the following parameters were optimal for our specific dataset: $nestimators$=200, $maxdepth$=20, $minsamplessplit$=2, $minsamplesleaf$=2, $maxfeatures$='sqrt', $bootstrap$=False. We then fit our data using a model with these parameters.

One drawback of this model, in this specific case, is that it tends to group similar players together, leading to identical predicted metrics for those players. To overcome this limitation, both the target and feature values were standardized using a MinMaxScaler(), which increased the diversity of the metrics and allowed for more specialized predicted fantasy scores.

The primary metrics, as explained above, are MSE (Mean Squared Error) and R2 score. The MSE score represents the average squared error between our predicted metrics and the actual target values, and is given by the equation:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

Our R2-score is the proportion of variance within the target metrics that can we explained by the features.

The residual plots for each of the nine metrics we are testing for are shown below. The plots also contain the MSE and R2 scores for each. As we can see, our model can account for over half of the variance in each metric, up to almost 80 percent for some metrics, which is significantly greater than simple guesswork.

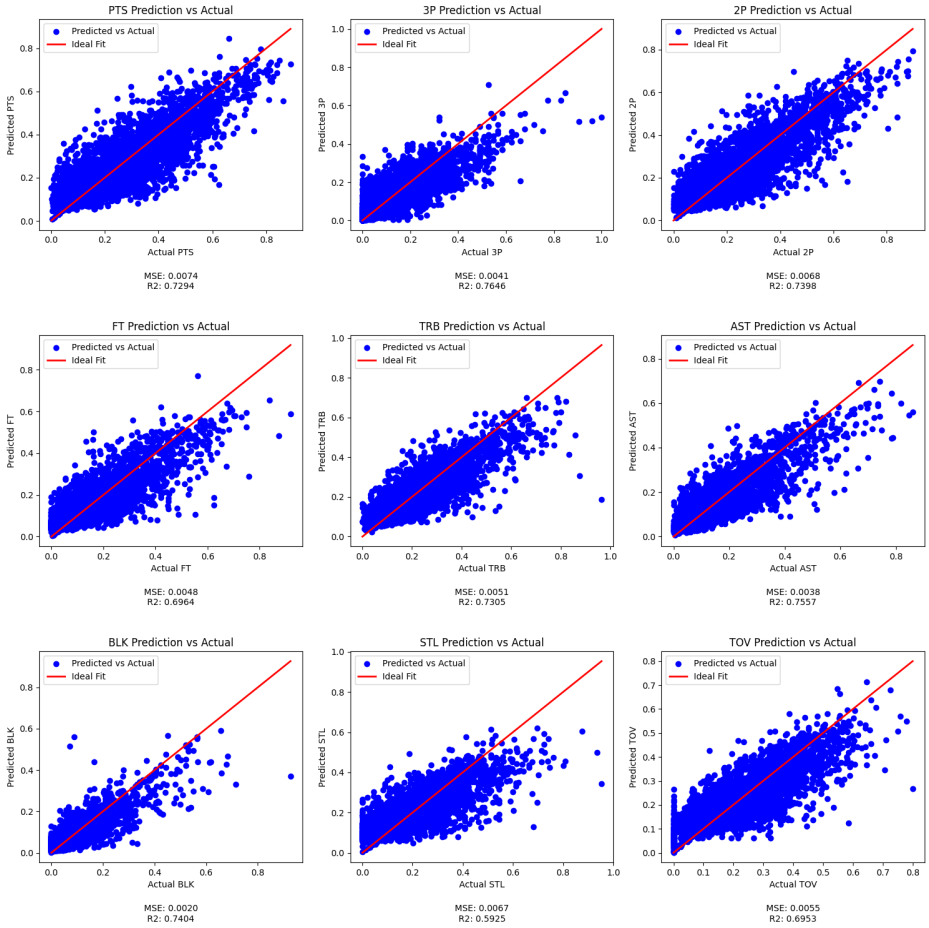

Figure 7: Nine plots highlighting the residuals between our predicted and true values for each of our nine tested metrics, with the MSE and R2 scores recorded at the bottom.

As we can see, our model is not perfect, but it is more effective than pure guesswork. After training the model, we applied it to our desired NBA players and subsequently ran an optimization model to find the ideal ten-player fantasy lineup, considering the following limitations: only ten players per roster, no more than two players from the same team, a requirement of five frontcourt and five backcourt players, and a maximum total salary cap of 100 million USD (based on the salary evaluations available on the NBA fantasy website). Given these constraints, our model predicted the optimized fantasy basketball team. The ten selected players can be found in the graphic below:

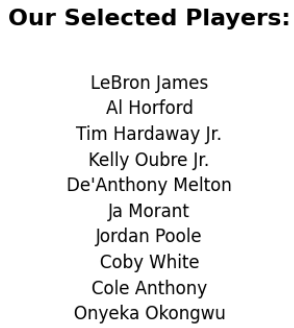

Figure 8: Our ten selected players.

6 Conclusions/Future Work

Utilizing a dataset containing individual NBA players' metrics across over 50 seasons, our hypertuned RandomForestRegression model processed this data and predicted active NBA players' metrics for the upcoming 2025 season. It then optimized these predictions to determine the ideal ten-player fantasy basketball lineup, considering various constraints.

Three different potential models were tested: Linear Regression, RandomForestRegression, and time-series forecasting. Among these, the RandomForestRegression model achieved the highest R2 score, meaning it was able to explain the greatest proportion of variance within the target values. This model likely performed the best because, as random forests are the result of an averaging process, they are less sensitive to outliers and noise. Additionally, by using a smaller subset of features in each randomized decision tree, the model helps assess feature importance more effectively. One notable drawback, however, was that the model tended to group similar players together and assign them the same predicted metrics. This was addressed by standardizing the data before fitting it to the model. In comparison to the Linear Regression model, the RandomForestRegression model is more complex, accounts for greater feature importance, and handles noise much better than the time-series forecasting model.

Looking ahead, it would be interesting to rerun the simulation using different types of time-series forecasting models and train the model exclusively on non-active players. This would require access to additional resources containing more metrics for earlier seasons. Our dataset shrank significantly and excluded any statistics prior to 1974 due to insufficient data. If more time were available, it would be possible to manually research missing statistics and record them individually, which would provide enough data to train the model without using active players.

7 Appendix

I. Radar chart.

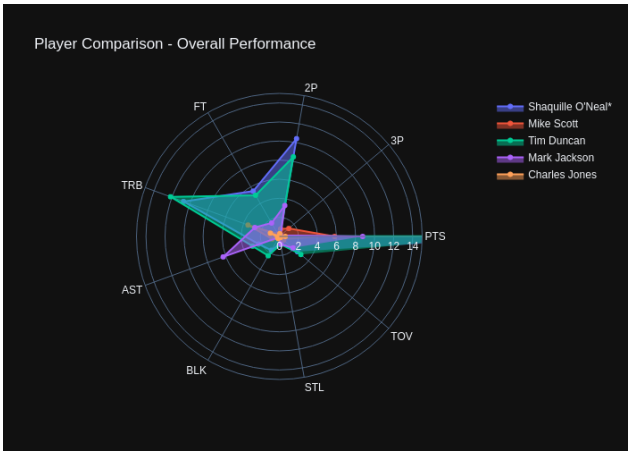

Figure 9: The radar chart of five randomly selected players.

II. Best defensive and offensive players.

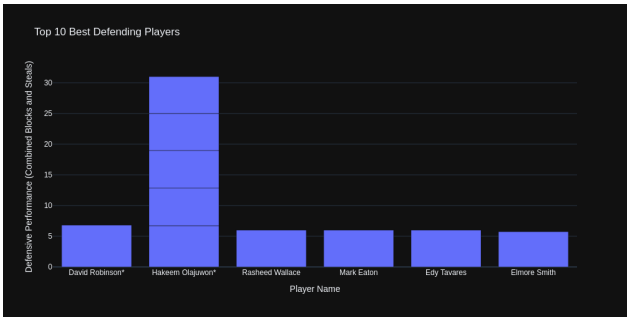

Figure 10: The ten best attacking players.

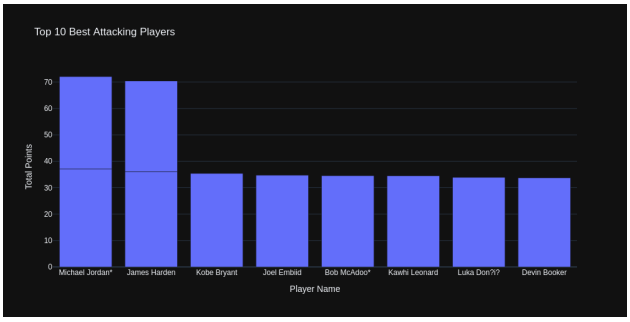

Figure 11: The ten best attacking players.

References

[1] AmirHossein Mirzaei. "NBA players scored points prediction". https://www.kaggle.com/code/amirhosseinmirzaie/nba-players-scored-points-prediction/comments, 2023.

[2] DimitreOliveira. "Deep Learning for Time Series Forecasting". https://www.kaggle.com/code/dimitreoliveira/deep-learning-for-time-series-forecasting, 2018.

[3] Duy Le. "NBA EDA predict point using RandomForestRegressor". https://www.kaggle.com/code/duyle2201/nbaedapredictpointusingrandomforestregressor/notebookModeling-data, 2024.

[4] Sarpeco. "Gathering And Cleaning Data For Salaries". https://www.kaggle.com/code/sarpeco/gathering-and-cleaning-data-for-salaries/notebook, 2021.

[5] Omri Goldstein. "NBA Players stats since 1950". https://www.kaggle.com/datasets/drgilermo/nba-players-stats/data?select=SeasonsStats.csv, 2017.

[6] Vivo Vinco. "2023-2024 NBA Player Stats". https://www.kaggle.com/datasets/vivovinco/2023-2024-nba-player-stats/data, 2024.

[7] Wyatt Walsh. "NBA Database". https://www.kaggle.com/datasets/wyattowalsh/basketball, 2023.

[8] Sumit Redekar. "NBA player stats from 2018 to 2021". https://www.kaggle.com/datasets/sumitredekar/nba-stats-2018-2021, 2021.