Predicting salaries of NBA players based on their in-game statistics

1. Introduction

NBA player salaries have risen rapidly, with both superstars and role players receiving larger contracts. This motivated us to analyze how closely a player's statistical performance predicts their salary. We hypothesized that greater ball-handling responsibilities would directly drive higher compensation. In real game situations, this likely translates to higher scoring and playmaking metrics.

Using NBA seasons from 2010 to 2025, we apply supervised learning to predict salaries. While our focus is on performance metrics, we acknowledge that factors like team budgets, contract structures, and organizational strategies also influence pay.

In section 2 we formulate our problem and explain the nature of the dataset. The 3rd section discusses the methods: data preparation, how and why we chose the models, the different feature sets, our loss functions, and lastly the model validations. Section 4 sums it all up by covering the performance of our models, comparing them and concluding the preferred choice from the models.

2. Problem Formulation

The goal is to test whether NBA player salaries can be predicted trustworthily using individual performance statistics. NBA teams typically consist of a few stars and several role players, creating predictable salary patterns. Although anomalies like fast-developing rookies or players with single standout seasons add noise, a carefully chosen set of statistics should still enable reasonably accurate salary predictions. Notably, the model’s purpose is to capture the main patterns between statistical performance and salary.

Our dataset, with almost 10 000 entries and 30 columns, includes different statistical metrics and common information about the players, such as name, team, position and season. It was taken from Kaggle, which is the world's largest data science community [1].

Features are mainly represented by real numbers that can get continuous values, often between 0 and some upper limit (e.g., points per game ranges from 0 to 35). Some of the other features are categorical, such as team (e.g., BOS, PHI). More explanation and details can be found in the appendix. The label is also continuous, and determines the player's salary in US dollars (USD) for a specific season ranging from thousands to over 50 million for superstars.

Now we can formulate that the machine learning problem in question is a regression problem. We already know the label of each feature vector so the type of learning is supervised learning.

3. Methods

3.1 Cleaning data and data preparation

Since the annual salary pool of the NBA has increased rapidly [2], we only consider three seasons (2022-23, 2023-24 and 2024-25). In addition, since the minimum salary of a rookie was 1.157 million US dollars [3], players under that amount were excluded. Some players attempted or made no shots, which led to undefined (nan) shooting percentages. Because these players still can be valuable, e.g. defensively, their shooting percentages were initialized to zero and the players were kept in the dataset.

These changes led to our final dataset being 1191 entries (players).

3.2 Model choice

Linear Regression is our first choice of model for predicting NBA player salaries. It is suitable because the label, salary, is a continuous variable. In addition, the label can be approximated by a weighted linear combination of the statistics of the player. The coefficients of the variables offer clear metrics on how much each feature, a statistic, contributes to the salary.

Our next choice is Random Forest (RF). Compared to Linear Regression, RF is suitable to consider nonlinear relationships between player statistics and salaries. For example, the salary might not have a linear relationship with outliers, e.g. salaries of superstars. In addition, predictive accuracy of RF models doesn't suffer from correlation between the features, which can lead to more optimal feature selection. Overfitting is also controlled with averaging the errors of the decision trees that don't correlate with each other thus smoothening overall error of the model. This is more suitable in nonlinear data, but can be somewhat of an overkill in linear data.

In results, we will choose between our nonlinear and linear model based on the error.

3.3 Feature Selection

NBA salaries are largely driven by highlights, and since in the last decade three-point shooting has become the key part of a team's offense, points per game (PTS) and some three-point shooting metric are key candidates for features. Still, defense remains vital for team success, so a feature depicting defensive impact should also be included.

To finalize our decisions, we used numbers and statistics. We visualized which features would be suitable metrics by making a correlation matrix (see Appendix). After excluding metrics like Team, Year (season) etc., the matrix ended up being 21 different feature candidates.

Based on this analysis, we constructed multiple feature subsets to be evaluated during the model validation phase. The first subset included points per game (PTS), three-point attempts (3PA), and defensive rebounds (DRB), which showed strong correlations with salary while maintaining relatively low inter-correlation. A second subset expanded this core by incorporating additional indicators of player role and impact, including minutes played (MP), turnovers (TOV), assists (AST), total rebounds (TRB), steals (STL), blocks (BLK), and age. A third subset focused on scoring-related metrics by combining PTS, 3PA, DRB, STL, MP, and field goals made (FG). Finally, a fourth subset consisted of all 21 performance-related features identified in the correlation matrix.

3.4 Loss functions

At first, MSE was considered as a loss function because it penalizes large errors more, which is desirable when a few superstars earn salaries tens of millions higher than the average.

However, MSE can increase drastically because the values are squared. For that reason, and to capture the original units of our data, we also calculate Root Mean Squared Error (RMSE). Even though MSE nicely penalizes large errors, we chose RMSE, since it shows error clearly in dollars and is smoother than MSE while still preserving sensitivity to large deviations in predictions.

We also use an option for our error, Coefficient of Determination (R^2), since it measures clearly how much of the variance in salaries our model predicts. R^2 gets values from 0 to 1, with 0 meaning that the variance isn't explained at all, and 1 that the model explains all of the variance in the data.

3.5 Model validation

We first split the dataset into a training set (80%) and a held-out test set (20%). For model validation we use k-fold cross-validation (using only the training set), where the dataset is split into k = 5 equally sized folds. The model is trained on four of them and tested on the one fold not included. This process is repeated five times, so that each fold serves as the validation data. The final performance is the average of all folds. This is used for both of the models.

Even though a single 80/20 split could have been a suitable choice, our dataset is relatively small, and therefore cross-validation leads to more stable results and a more precise model. In addition, we chose not to use two seasons as training data and one season as test data, since the salary caps in the NBA increase by several percentage points each year [2]. That would have led to a far more inaccurate model.

4. Results

4.1 Linear Regression

The first model implemented for predicting NBA player salaries was a Linear Regression (LR) model. Figure 1 presents the cross-validated performance of the LR model across four different feature subsets. These results were used to assess the impact of feature selection on model performance and to provide a baseline for comparison with the Random Forest (RF) model.

Model performance was evaluated using Root Mean Squared Error (RMSE) and the coefficient of determination (R^2). To obtain robust performance estimates, we employed a 5-fold cross-validation strategy applied exclusively to the training set, following the 80/20 train–test split described in Section 3.5. This procedure produced five RMSE and R^2 values for each feature subset, which were then averaged to obtain the results shown in Figure 1.

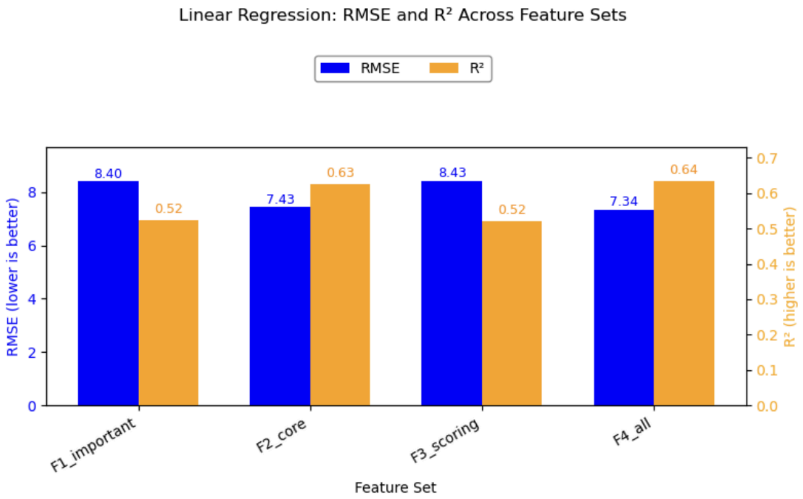

Figure [1]. Linear Regression performance metrics

The extended feature subsets (Models 2 and 4) achieved approximately $1 million lower RMSE compared to the baseline model, indicating improved predictive performance as additional performance-related features were incorporated. Among the tested configurations, the best LR model achieved an RMSE of approximately $7.34 million and an R^2 of 0.64.

Overall, the LR model provided moderate performance with the best RMSE around $7.34 million and R^2 of 0.64. However, high RMSE suggests that the model struggled to capture elegant nonlinear relationships and was significantly affected by outliers, especially players with salaries below $5 million. Nonetheless, LR provides us a directional estimate of player compensation.

4.2 Random Forest

The second model implemented was a Random Forest (RF) for its nonlinearity. The same experimental protocol used for the Linear Regression model was applied to ensure a fair comparison. Specifically, we employed a 5-fold cross-validation strategy on the training data following an 80/20 train–test split, and evaluated the RF model using the same four predefined feature subsets.

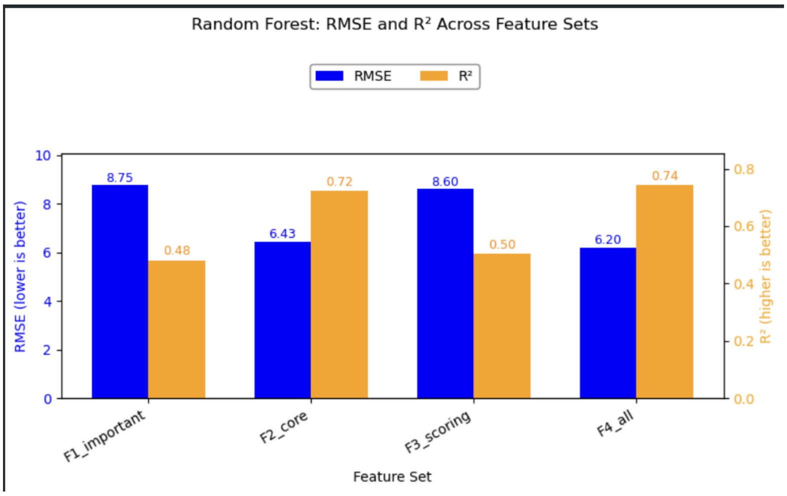

Figure [2]. Random Forest performance metrics

The best RF models outperformed the LR models, achieving lower RMSE and higher R^2 scores. This improvement confirms that RF more effectively captured nonlinear dependencies in the data

4.3 Model Comparison and Conclusion

To compare the predictive performance of the two modeling approaches, we evaluated the best-performing Linear Regression and Random Forest models identified during the cross-validation phase. Model selection was based on the cross-validated results shown in Figures 1 and 2, where the RF model consistently achieved lower RMSE and higher R² values than the LR model across feature subsets. Given the magnitude of these differences, the Random Forest model was selected as the final predictive model.

The selected Random Forest model was then trained on the full training dataset and evaluated once on the held-out test set, which remained completely unseen during feature selection, model tuning, and validation. This final evaluation provides an unbiased estimate of real-world predictive performance. The resulting test-set performance metrics are summarized below.

Model	**Test RMSE ($)**	**Test R^2**
Random Forest	5.69 million	0.74

5. Results

Random Forest successfully captures complex relationships among features, resulting in superior predictive performance. In contrast, Linear Regression is using a simple linear relationship between features and label leading to oversimplifying the underlying patterns in the data. Considering that, player compensation is influenced by a combination of different statistics rather than purely linear dependencies. Overall, the Random Forest model is the preferred choice for practical salary prediction purposes.

Future work could incorporate categorical variables such as contract type or contract year (e.g., rookie deals vs. veteran extensions) to capture structural effects beyond performance statistics. In addition, further tuning of model hyperparameters and experimenting with more advanced nonlinear methods, such as gradient boosting, could improve predictive performance.

References

[1] Ratin, A. (2021). *NBA player stats and salaries (2010–2025)* [Dataset]. Kaggle. https://www.kaggle.com/datasets/ratin21/nba-player-stats-and-salaries-2010-2025

[2] RealGM. *NBA salary cap information*. https://basketball.realgm.com/nba/info/salary_cap

[3] Sports Illustrated. (2024, June 21). *What is the minimum NBA salary in 2024–25?* https://www.si.com/nba/what-minimum-nba-salary-2024-25

Appendix

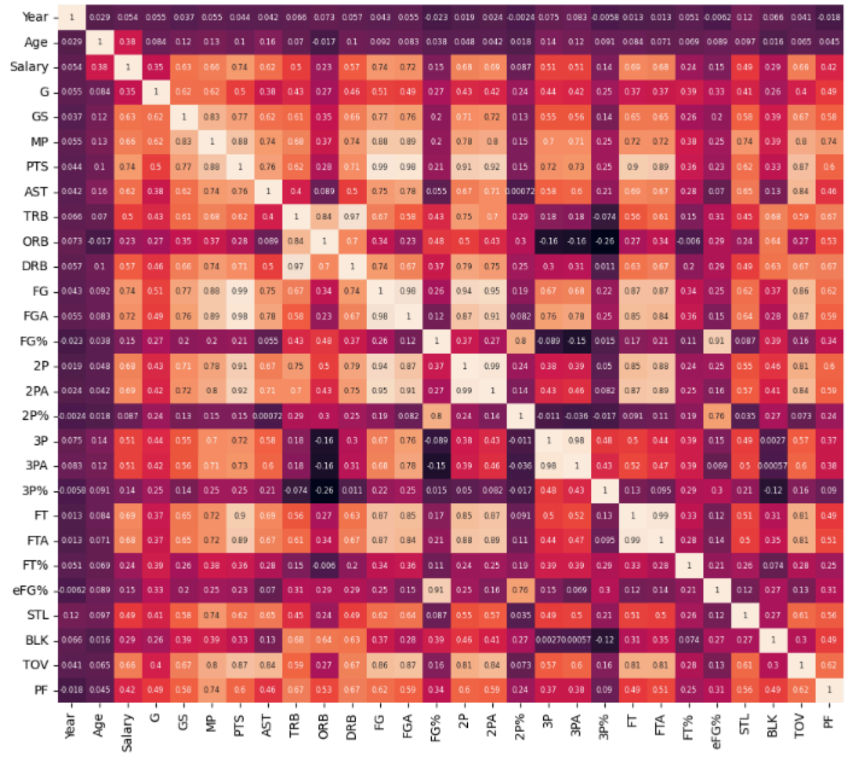

Figure [3]. The correlation matrix, where each entry is the correlation of two features.

#	Variable name	Role	Description	Data type
1	Player	Feature	Player’s name	Categorical
2	Salary	Label	Player’s salary in $	Continuous
3	Year	Feature	Season	Continuous
4	Pos	Feature	Player’s position	Categorical
5	Age	Feature	Player’s age	Continuous
6	Team	Feature	Player’s team	Categorical
7	G	Feature	Amount of games	Continuous
8	GS	Feature	Amount of games started	Continuous
9	MP	Feature	Amount of minutes played	Continuous
10	FG	Feature	Player’s field goals made	Continuous
11	FGA	Feature	Player’s field-goal attempts	Continuous
12	FG%	Feature	Player’s FG%	Continuous

13	3P	Feature	Player’s three-point field goals made	Continuous
14	3PA	Feature	Player’s three-point field goals attempts	Continuous
15	3P%	Feature	Player’s 3P%	Continuous
16	2P	Feature	Player’s two-point field goals made	Continuous
17	2PA	Feature	Player’s two-point field goals attempts	Continuous
18	2P%	Feature	Player’s 2P%	Continuous
19	eFG%	Feature	Effective Field Goal %	Continuous
20	FT	Feature	Player’s free throws made	Continuous
21	FTA	Feature	Player’s free throws attempts	Continuous
22	FT%	Feature	Player’s FT%	Continuous
23	ORB	Feature	Player’s offensive rebounds	Continuous
24	DRB	Feature	Player’s defensive rebounds	Continuous
25	TRB	Feature	Player’s total rebounds	Continuous
26	AST	Feature	Player’s assists	Continuous
27	STL	Feature	Player’s steals	Continuous
28	BLK	Feature	Player’s blocks	Continuous
29	TOV	Feature	Player’s turnovers	Continuous
30	PF	Feature	Player’s personal fouls	Continuous
31	PTS	Feature	Player’s points made	Continuous