Predicting How Many Points an NBA Player Will Average Per Game Based on Average Minutes Played Per Game

Nicholas Klapatch
Erick Garcia-Vargas
Ngoc Ta

CSCI-4465: Machine Learning, Fall 2021
December 8, 2021

Abstract

In this study, we sought to find a predictable relationship in National Basketball Association (NBA) statistics using machine learning techniques. While investigating individual player statistics, we found an interesting relationship between the average number of minutes that a player played and the average number of points that player scored per game in a single season. We aimed to predict how many points a player will average per game based on minutes played per game using different supervised learning models. From these models, the better performing one would be selected.

The NBA is evolving based on statistics each season. Finding important relationships between statistics can elevate the level of play in the game, especially when it comes to scoring. High scoring games are among the most entertaining to watch. It is even more entertaining when one player scores an abnormally high number of points, like when Kobe Bryant scored 81 points in a single game on January 22, 2006 [1]. No player has come close to this individual score since. The statistics used in this project are from a publicly accessible page from Basketball-Reference [2]. This dataset contains per-game averages for each player from the 2020-21 NBA season. By predicting how many points a player will average per game based on minutes played, opposing teams can create better strategies, fans can choose which games they would prefer to watch, and NBA franchise owners can adjust prices for game attendance.

Each member of the team contributed to the project. In team meetings, we determined the particular project that we were going to work on, implemented the programs, and worked on the report and presentation. Individually between team meetings, each team member contributed to each phase of the project, particularly in validating the program and editing the report and presentation.

Introduction

In a single NBA game, very many statistics are measured. Coaches, players, fans, and team owners are all deeply interested in how well players perform. Some statistics are more popular than others, and some have strong relationships that have unexpected characteristics. Over the course of a season, when these statistics are averaged, they tell a lot about how players perform. Particularly, per-game average minutes played compared to average points scored. Intuitively, the more minutes a player plays, the more points that player will score. The relationship, however, is more interesting than that. As displayed in Figure 1, this relationship seems to grow parabolically rather than linearly.

Is this relationship indeed more parabolic than linear? Perhaps it is. Players who play the most minutes in games are typically the most skilled players on their rosters, who their coaches trust to attempt to score the most. These qualities are not captured in the statistics used, but they could be a great reason for observing this trend. That idea, however, is not the focus of this study (that can become its own study based on the findings from this one). To uncover the true trend is the focus. To do this, we compared two different linear regression fits: one that fit a linear function, and another that fit a quadratic function to the same data points. Of these two fits, one must allow for more accurate predictions, and thus be a better model to select.

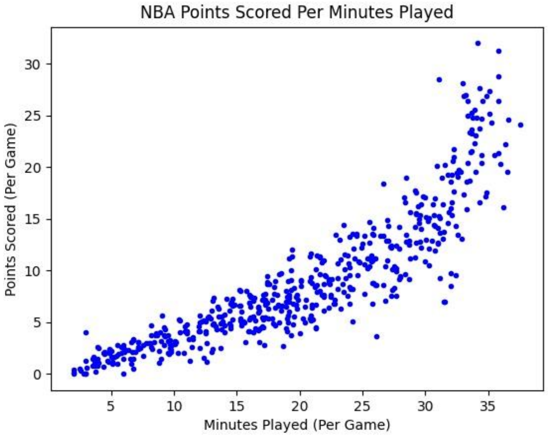

Figure 1

Methodology

Since this is a supervised learning project, training and validation sets needed to be created. To do this, random data points were placed into each set, at an 80-20% split using the Python Random module. Of the 540 total data points (one for each player), 432 went to the training set, and the remaining 108 went to the validation set. At runtime, these sets were randomly generated, so they differ each time the program is run. For both fits, the same sets were used.

To implement linear regression, we used the batch gradient descent technique, which was covered in the CSCI-4465 lecture on September 27, 2021. The update formula used is shown in Figure 2, and was also discussed in the same lecture. This update rule incrementally optimizes the coefficient value for each feature, until the coefficient does not change after an iteration. This means the minimum for the cost function was reached, and no better values can fit the data better. Batch gradient descent considers the entire training set in each iteration, so it is not as fast as using other methods, like stochastic gradient descent, or normal equations.

This was implemented in Python using the NumPy library, primarily for its efficient linear algebra functions. This implementation is suitable for any number of features, although in this project, we used two different fits using just one feature, and one target variable. For visualization, MatPlotLib.pyplot was used.

$$\theta_i \coloneqq \theta_i - \alpha \sum_{j=1}^{m} (h_\theta(x_i) - y_i) x_i^{(j)}$$

$$\alpha = \ learning\ rate$$

$$h_\theta = \theta^T x = \ hypothesis\ function$$

$$x = feature$$

$$y = target\ variable$$

$$\theta = coefficient\ of\ feature$$

$$m = number\ of\ training\ examples$$

Figure 2

When implementing batch gradient descent, the learning rate parameter needed to be tuned so that the algorithm could converge. If the learning rate was too high or too low, the algorithm would either not converge, or take too long to converge. It was determined that the larger the data set, the smaller the learning rate needed to be. In the dataset we used, the training size was 432 elements, and we tuned our learning rate to be 10e-8. This allowed for the algorithm to quickly and accurately converge. Figure 3 displays the linear fit, and Figure 4 displays the quadratic fit to the dataset. It is important to note that the training and validation datasets are randomized at runtime, so they will be different each time the program is run.

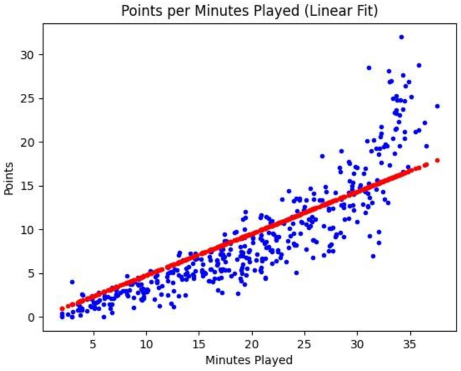

Figure 3

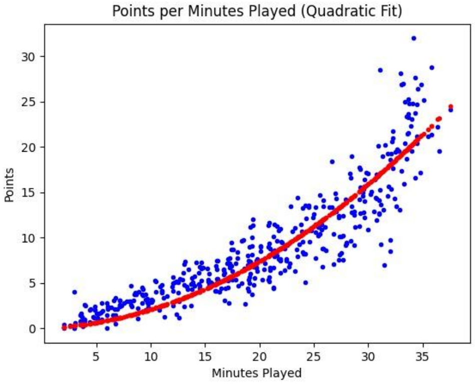

Figure 4

Results

Although the training and validation datasets were randomized at runtime, the quadratic fit consistently fit the data better than the linear fit. The metric used to determine this was the mean squared error, its equation is shown in Figure 5. Figure 6 displays the linear fit applied to the validation set, and Figure 7 displays the quadratic fit applied to the validation set. The mean squared error is significantly less in the quadratic fit. In addition to this metric, when the validation set data points go beyond roughly 30 minutes played, the linear fit is noticeably further away from the data points than the quadratic fit is. This demonstrates that the linear model under fit the data, whereas the quadratic fit adequately covered each data point. Hypothetically, if some NBA players were to play even more time, this disparity would become even greater.

$$Mean\ Squared\ Error = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2$$

$$n = number\ of\ data\ points$$

$$y = observed\ values$$

$$p = predicted\ values$$

Figure 5

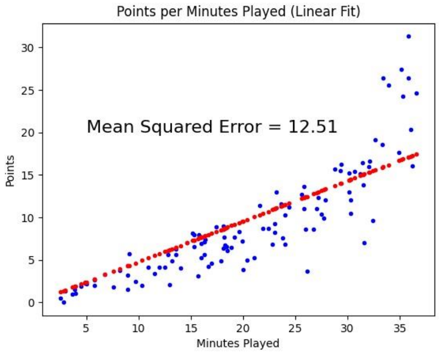

Figure 6

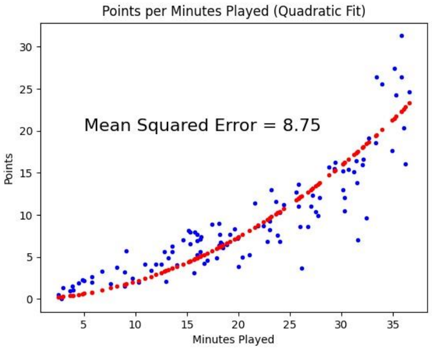

Figure 7

Conclusion

For predicting the amount of points an NBA player will average for a season based on minutes played per game, a quadratic function is better than a linear function. Batch gradient descent can adequately converge and produce a valid result. When implementing this algorithm, it is very important to experiment with learning rates to find one that works for the dataset. This learning rate must allow the algorithm to converge efficiently.

Finding that the relationship between minutes played-per-game and points-per-game is quadratic can also lead to new projects. Particularly, in unsupervised learning. Discovering what makes this trend exist can further help players, coaches, team owners, and fans.

Reference List

[1] “Raptors vs. Lakers - Game Summary - January 22, 2006.” ESPN. https://www.espn.com/nba/game/_/gameId/260122013 (accessed December 9, 2021).

[2] “2020-21 NBA Player Stats: Per Game.” Basketball Reference. https://www.basketball-reference.com/leagues/NBA_2021_per_game.html (accessed December 9, 2021).