USING MACHINE LEARNING TO PREDICT POINTS PER GAME IN THE NBA

TECHNICAL REPORT

Dhruv Verma
Student
The Cathedral and John Connon School
Mumbai, India - 400005

ABSTRACT

Every year billions of dollars are lost on uninformed prop bets. This paper proposes a model to predict points per game in the NBA, which can help mitigate these losses using high-accuracy mathematical predictions. A novel method using a multi-layer perceptron built on individual and team statistics is described in this paper. The proposed model has an R^2 of nearly 0.8, outperforming most publicly available models for the same, with the major difference being feature engineering and the implementation of a neural network. The proposed model is therefore suitable for further academic and practical research and application, both public and private.

Keywords NBA, Points Per Game (PPG), Traditional Statistics, Advanced Statistics, Linear Regression, Multi-layer Perceptron

1 Introduction

The National Basketball Association (NBA) is a professional basketball league. There are 30 teams in the NBA, each team playing 82 games over a season. The teams are split up into two conferences (East and West) based on their location. Following the regular season, the 8 teams with the most wins qualifies for the Playoffs, where a knockout style best of 7 series format is adopted within the conferences. The winners of each conference will then play the each other for the NBA championship in the finals. This paper will only deal with the regular season, due to a more consistent data set, since different teams qualify for the playoffs in different years.

There are multiple statisticians present at each NBA game, who track and tally raw statistics. Some important raw statistics include points, assists, rebounds, steals, blocks, plus-minus and shooting percentage. The raw statistics may be used to formulate advanced statistics to better encapsulate a player's performance. These will be discussed in depth later in this paper.

This paper intends to demonstrate the feasibility and validity of predicting future statistics in the NBA using a combination of past data by using machine learning. While there are multiple defensive stats out there, their lack of objectivity makes them unsuitable for a high-accuracy model. This paper will demonstrate the validity of its central claim by predicting the total production of the team's offence - both due to its relatively objective nature and an abundance of raw statistics to work with. The project will entail collecting and engineering the features, building the prediction models and discussing its results.

2 Motivation

Being incredibly passionate about basketball and computer science myself, I was able to merge to interests of mine and use insights from playing the sport since I was 8 years old in this paper, most notably in recognising the fact that teammates and surroundings significantly influence player performance as well, something that has not been addressed in most publicly available models.

This project is justified from an economic and social perspective as well. The sports betting industry has grown significantly in the past few years, now averaging over 5 billion USD per year in total bets placed in the United States[1]. The NBA in particular, under the helm of Commissioner Adam Silver, was one of the first major sports leagues in the USA to publicly support betting and considers regulated sports betting as a valid source of income. A large amount of the bets placed are on simple over/under bets, where a bettor predicts whether a player will achieve over or under a set value for a certain statistic at determined odds. The most prolific and extensively used statistic is unsurprisingly points per game as the main goal of the sport is to score points. While I recognise the ethical implications of betting, it is a harsh reality that we have to accept. Billions of dollars a year are lost on uninformed bets every year and the numbers are only increasing. A publicly available accurate prediction model as proposed in this paper may help mitigate these losses and strengthen the hands of the common people, thus improving their odds against the betting houses, therefore reducing the damaging effects of uninformed bets on society.

Apart from formal betting, Fantasy sports are also a major industry with the number of players nearing 60 million per estimates according to FSGA [2019]. Fantasy basketball is built upon large prediction machines and algorithms, and most applications provide their own forecasts of player performance to convince users to add or cut them from their team. The winner of a fantasy game is decided on a composite of statistics, with the main component being points scored. Furthermore, the methodology used in this paper could be applied to lower levels of the game such as at the high school or college levels, enabling teams and coaches with lesser funding to evaluate and predict their players performance better.

Therefore, it is clear that there is a large market for the proposed model in this paper, and this paper aims to develop a model that satisfies and improves on current needs.

3 Application

3.1 Data Description

The data used is composed of a combination of raw statistics i.e. unaltered values from NBA statisticians and advanced statistics, which are some mathematical combination of raw statistics. A majority of statistics were player stats or data for individual players, however some team statistics were taken into account as well. This paper tries to stay away from all-in-one statistics to encapsulate total output since they are unreliable and redundant as models have no limit on inputs. Therefore, it does not make sense to essentially feed in pre-determined weights since the proposed models will successfully calculate the weights themselves. Player Efficiency Rating (PER) developed by John Hollinger is the only all-in-one statistic used in this paper. This is because its effectiveness in such models has already been demonstrated by Weiner [2021], therefore it was apt to include in the model.

The models were built around individual players both due to an abundance of individual statistics and increased portability and scalability. This is because the NBA is in a volatile state. Trades and changes of teams are frequent. Therefore, a model that predicts player statistics can easily be made to predict team statistics by simply summing up the numbers predicted for each player on a given team at that time.

A total of 15 variables were used in the dataset $\mathbf{X}$. The data was scraped from NBAStuffer.com [2011]. To train the models, data from the 2010-11 NBA regular seasons was collected and compiled. Thus, the dimensions of $\mathbf{X}$ is 52692×15, where 52692 is the sum of the total number of players for each season from 2010-11 to 2019-20. The dataset was then separated randomly into a training and testing set (67% training and 33% testing).

The predicted variable $\mathbf{Y}$ is the points per game of the subsequent year, i.e. data from 2020 is used to predict PPG in 2021. Therefore the corresponding $\mathbf{Y}$ array in the code coincides with the previous years data (it is shifted back by a year).

3.1.1 Raw Statistics

Raw statistics are statistics that are directly recorded by statistician via observation or technology (they are not processed mathematically in any way). Before delving further into the variables, 2 important statistics must be defined.

Points - In the NBA points can be awarded to either team in 3 different ways. A made basket or field goal is worth 2 points if it lies within the 3 point line and 3 points if it lies outside. The 3 point arc is a line at distance of 23.75 feet from the basket in front of it and 22 feet away from the basket in the corners. If a player violates the rules the opposing team members may receive a chance to shoot one or multiple free throws worth 1 point each from 15 feet in front of the basket with no defender in front of them.

[1]Source: `https://www.legalsportsbetting.com/how-much-money-do-americans-bet-on-sports/`

Assists - As per the NBA Rulebook, an assist is defined as "The last pass to a teammate that leads directly to a field goal and the player receiving the pass must move immediately toward the basket in a scoring motion." It is important to note that assists are subjective and left to the discretion of the statistician at a game.

A summary of all raw statistics used and their definitions can be found in Table 1. A glossary of all terms used henceforth is found in Section 6.

Table 1: Raw Statistics

Name of Variable	Definition
AGE	The age of an individual player as of the start of the regular season. Age was used as an input as player performance tend to regress as they grow older.
GP	The total number of games played by an individual player in the regular season. Incorporating this effectively disqualifies small sample sizes.
PPG	The average points scored by an individual player per game in the previous regular season. This is trivially relevant for predicting the PPG of the next season
APG	The average number of assists by an individual player per game in the regular season. This demonstrates how often a player passes the ball as opposed to scoring.
$\mathrm{PTS_T}$	The average points scored by the player's team in the regular season. This statistic partially accounts for the pace of play.

3.1.2 Advanced Statistics

Advanced statistics are weighted combinations of raw statistics which try to provide a greater insight on performance. The main advanced statistics used for the models are the following. The formulation of the relevant statistics are courtesy of Forman [2021].

1. Efficiency Metrics
   Both metrics were used to account for shooting efficiency and cover for each others biases. A more efficient player will be encouraged to shoot more by coaches, therefore tends to correlate with better scorers.
   (a) True Shooting Percentage ($TS\%$ or TS%)
   True shooting percentage effectively measures a players efficiency in scoring by effectively calculating points scored per shot.
   $$TS\% = \frac{PTS \times 100}{2 \times (FGA + (0.44 \times FTA))}$$
   (b) Effective Field Goal Percentage ($eFG\%$)
   Effective field goal percentage serves as an alternative to traditional field goal percentage ($FG\% = \frac{PTS}{FGA}$). It provides a greater weightage to 3 Point Efficiency as they result in more points when made.
   $$eFG\% = \frac{FGM + (0.5 \times 3PM)}{FGA}$$
2. Usage Rate ($USG\%$)
   Usage Rate is an estimate of how many possessions a player uses while on the floor
   $$USG\% = \frac{FGA + (0.44 \times FTA) + TOV \times \frac{MP_T}{5}}{MP \times FGA_T + (0.44 \times FTA_T) + TOV_T}$$
3. Assist Percentage ($AST\%$)
   Assist Percentage is an estimate of field goals a player assisted on while they were on the floor. This serves as a metric to demonstrate how frequently players will pass the ball as opposed to shoot it.
   $$AST\% = \frac{APG}{\frac{MP}{\frac{MP_T}{5}} \times FGA_T - FG}$$

4. PACE

   Pace is an estimate for the number of possessions per 48 minutes played by a team. The more possessions played means more opportunities to score, therefore it is relevant in the prediction.

$$PACE = \frac{POS_T + POS_O}{2 \times MP_T/5}$$

5. Unadjusted Player Efficiency Rating ($uPER$)

   Player Efficiency Rating was one of the first advanced statistics used in the NBA. It encapsulates player performance into a single number. While it does have flaws when used on its own, the validity of its usage has been demonstrated by Weiner [2021] as mentioned above.

   The uPER was also mapped to a range of 0-30 for every season and called PER to account for changes in pace of the game every season.

$$\begin{aligned} uPER = \quad & \tfrac{1}{MP} \times [(3PM) + (\tfrac{2}{3} \times AST) + ((2 - factor \times \tfrac{AST_T}{FGM_T}) \times FGM) \\ & +(\tfrac{FT}{2} \times (1 + (1 - \tfrac{AST_T}{FGM_T}) + \tfrac{2}{3} \times \tfrac{AST_T}{FGM_T})) \\ & -(VOP \times TOV) - (VOP \times DRB\% \times (FGA - FGM)) \\ & -(VOP \times 0.44 \times (0.44 + (0.56 \times DRB\%)) \times (FTA - FT)) + (VOP \times DRB\% \times ORB) \\ & +(VOP \times (1 - DRB\%) \times (TRB - ORB)) + (VOP \times STL) + (VOP \times DRB\% \times BLK) \\ & -PF \times (\tfrac{FTM_L}{PF_L} - 0.44 \times (\tfrac{FTA_L}{PF_L} \times VOP)] \end{aligned}$$

Some other less important statistics were also used for certain models. A sample of a total player object containing all the data required for the necessary predictions can be found in Table 3

3.2 Models

In order to predict $\mathbf{Y}$ (player points per game), two models were developed.

3.2.1 Linear Regression

A basic model is linear regression. In linear regression, we make the linearity assumption and the formula takes the following form.

$$h(x) \sim \vec{\beta}\mathbf{X}$$

To derive the optimal solution for linear regression model, we generally start with a loss function $L = \sum_{i=1}^{n} \epsilon_i^2$ where this loss function is called mean square error and ϵ is the difference of true model y and the hypothesized model $h(\vec{\beta}\mathbf{X})$. We can formally write this out as

$$L = \sum_{i=1}^{n} \epsilon_i^2 = \epsilon'\epsilon = (y - \mathbf{X}\vec{\beta})'(y - \mathbf{X}\vec{\beta})$$

The approach to search for optimal weight is to set the partial derivative of the loss function to zero and solve for optimal solution. Hence, we can set

$$\frac{\partial}{\partial \vec{\beta}} L = 0 \Rightarrow \mathbf{X}'\mathbf{X}\vec{\beta} = \mathbf{X}'y$$

which gives us the optimal solution

$$\underbrace{\vec{\beta}}_{\text{optimal}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}y$$

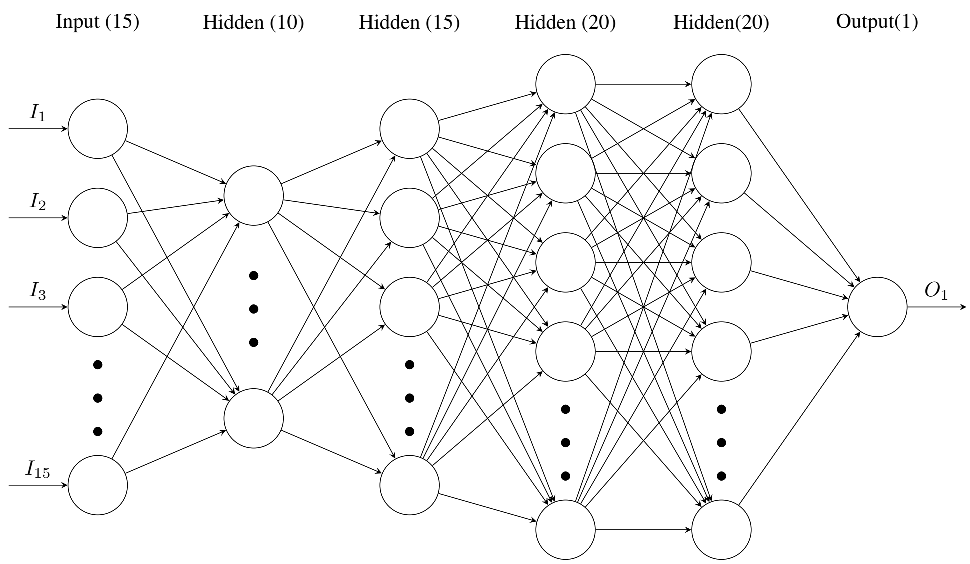
Notice that the equation 3.2.1 is a closed form of the linear coefficients of the linear regression model. This is based on the assumption that $h(x) = \vec{\beta}\mathbf{X}$ and error $\epsilon = y - h(x)$.

A linear regression model was used to predict the outcome of the next NBA season first. The linear regression model uses all 15 statistics mentioned above.

3.2.2 Multi Layer Perceptron (MLP) Regressor

An MLP Regressor is a type of neural network used to predict the data in this paper. The final network chosen had dimensions of $15 \times 10 \times 15 \times 20 \times 20 \times 1$, where the first layer is the input variables and final layer is the output. The inner rows are hidden layers which were chosen through trial and error. Various permutations of network architecture were tested with a testing set, and the aforementioned dimensions led to maximal results. To evaluate the effect of incorporating team statistics in the model a control MLP network without team stats was also engineered.

Figure 1: The Network Architecture of the Multi Layer Perceptron Neural Network



Consider a matrix $\mathbf{W}$ with a value (or weight) assigned to a different node (or circle on the diagram). Initially the values of $\mathbf{W}$ are randomised. Assign an input vector $\mathbf{N}$ with the given inputs or statistics in this case. The initial hypothesis will be

$$h(\mathbf{x}) = R(\mathbf{W}^T \cdot \mathbf{N})$$

Where R(x) is an activation function. A rectified linear unit function (ReLu) was used in this model ($R(x) = \max(0, x)$). The ReLu function incorporates a sense of non-linearity into the model by effectively 'folding' the data and allows it to fit the data better as the model will no longer be a simple linear transformation.

The model continually updated its weights or coefficients ($\vec{\beta}$) (where $\vec{\beta}$ is a column vector corresponding to a layer in $\mathbf{W}$) using gradient descent, by minimising a cost function J, where,

$$J = f(\vec{\beta}) = \frac{1}{2m} \sum_{i=0}^{m} (h(\vec{x}) - \vec{y})^2$$

and,

$$\vec{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$$

The values of coefficients are continually updated as follows until a minimum is reached.

$$\vec{\beta} := \vec{\beta} - \alpha \frac{\partial J}{\partial \vec{\beta}} = \vec{\beta} - \alpha \nabla J \cdot \frac{\vec{\beta}}{\left|\vec{\beta}\right|}$$

where α is an arbitrarily small learning rate and $\nabla = \langle \frac{\partial}{\partial \beta_1}, \frac{\partial}{\partial \beta_2}, \frac{\partial}{\partial \beta_3}, \ldots, \frac{\partial}{\partial \beta_n} \rangle$

To optimise the code, a second order Taylor-series approximations of the Cost function was used instead, where the gradient and Hessian matrices (Jacobian of ∇J) were used as analogues for the first and second derivatives respectively. The approximations to these objects were gradually improved using the L-BFGS optimiser (Charles George Broyden and Shanno [1987]).

3.3 Results and Discussion

Both models performed vastly differently. The linear regression model, fail to correctly predict future points per game for a player. The neural network performed vastly better and predicted Player PPG to a much higher degree of accuracy.

To evaluate the models, their R^2 value or coefficients of determination were calculated, where

$$R^2 = 1 - \frac{\text{the residual sum of squares u}}{\text{the total sum of squares v}}$$

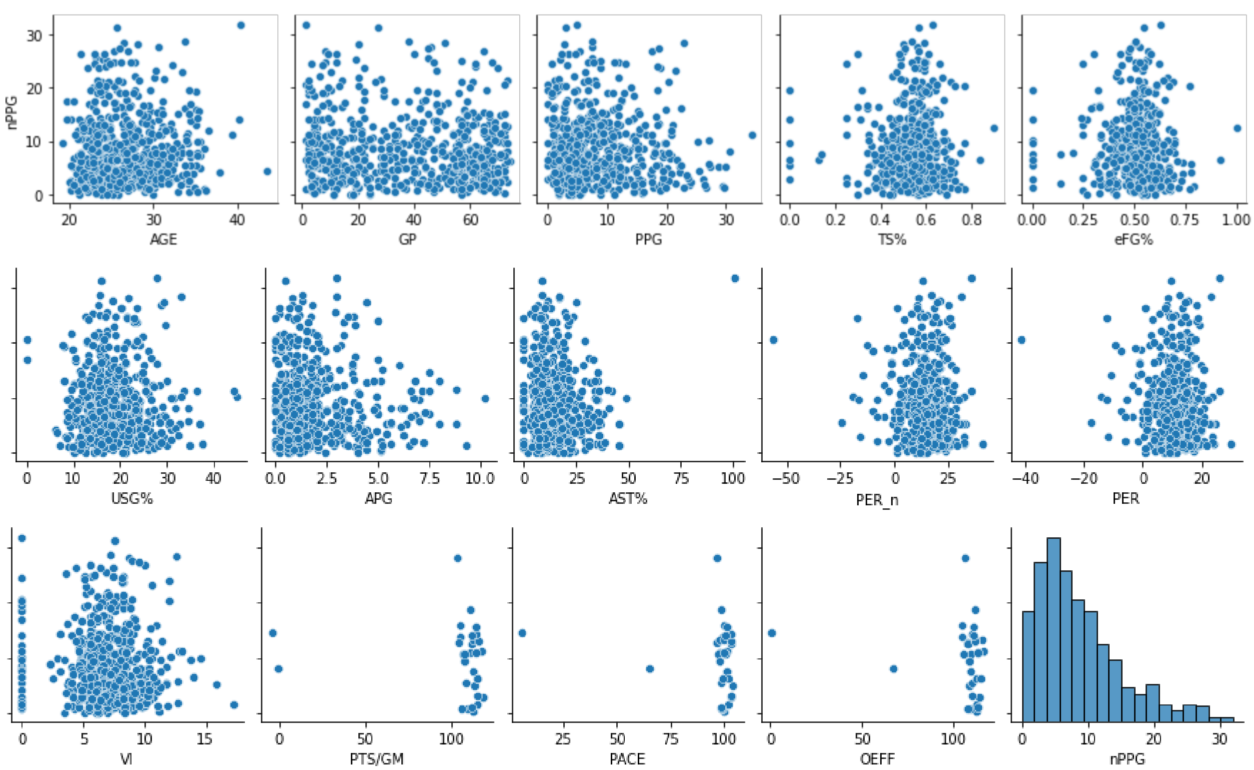
.

$$u = \sum(\mathbf{Y} - h(x))^2 \text{ and } v = \sum(\overline{\mathbf{Y}} - \mathbf{Y})^2$$

$$\therefore R^2 = 1 - \frac{u}{v} = 1 - \frac{\sum(\mathbf{Y} - h(x))^2}{\sum(\overline{\mathbf{Y}} - \mathbf{Y})^2}$$

Before directly exploring the results analysis of the pairplot of data is necessary.

Figure 2: This is the pairplot of the data. The Y axis contains $\mathbf{Y}$, which is the Players points per game (PPG) the next year. The X axis is the different variables in the dataset $\mathbf{X}$, as explained above (Image generated using Hunter [2007])



3.3.1 Linear Model

From the relations in 2 we can more accurately deduce why the linear model failed to perform accurately. For linear regression to successfully work, the following assumptions must be true.

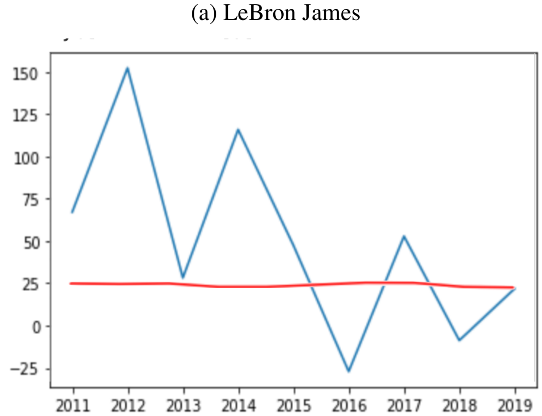
1. Linearity requires the relations between $\mathbf{X}$ and $\mathbf{Y}$ to be linear. We can easily observe that this condition is not satisfied by just looking at the pairplot of the data.
2. Homoscedasticity requires constant variance. Every independent variable is heteroscedastic with the distance from a regression line not remaining constant. For example, the variance increases as PPG increases and decreases as efficiency increases. The team statistics, however, do satisfy homoscedasticity, but are useless for judging individual player performance on their own, therefore the model cannot satisfy homoscedasticity.
3. Independence requires the variables to be independent observations from each other. This is not satisfied by the dataset as each variable may contribute to a change in the other. The points per game of a players teammates and their teams pace for example directly influence the chances a player has to score points. This is due to the fact that more points scored by teammates lead to less opportunities for the player in question to score. Similar qualitative analysis can be used to justify this independence for other combinations of variables.
4. Normality requires $\mathbf{Y}$ to be normally distributed for any fixed value of $\mathbf{X}$. This assumption is partially valid as pre-normalised statistics like Player Efficiency Rating satisfy it by definition. In general more advanced efficiency related statistics like True Shooting and Effective Field Goal Percentage appear to be normalised, however traditional variables like points per game and assists per game appear not to be from observing the pairplot. Therefore, the normality assumption is also partially violated since all variables do not satisfy it.

Since the 4 assumptions of linear regression are violated, it is no surprise that the model fails.

The R^2 value achieved was 0.0079 which essentially means that the model would perform the same by completely disregarding input features. Some sample predictions made by the model are shown below. Note that some predictions for a single player were more than the greatest teams of all time and some were negative, both of which are impossible.

Figure 3: The Y axis shows the points per game predicted based on the cumulative statistics of the previous years, and the X axis is the year predicted. The red line is the actual PPG of the player, while the blue line is the PPG predicted by the linear model. We observe that linear regression model does relatively poor jobs for predictions for both players. This is due to the nature of linear regression and that it allows the predictions to by projected onto a real set, which means it can take negative values. The values clearly do not coincide and are even impossible in multiple cases. (Image generated using Hunter [2007])

(a) LeBron James



(b) Stephen Curry

3.3.2 MLP Regressor

As expected, the neural network performed far better than the linear model. The model was able to predict the data to a high degree of accuracy. Similar models built failed to account for team statistics and the role of teammates in the individual stats of a player, therefore appear to have capped out at an R^2 of 0.7, however this paper in incorporating team or surrounding factors has achieved greater accuracy.

1. Without Team Statistics

   A model built without accounting for team statistics was built as a control to demonstrate the change in performance of the model by incorporating them. This model produced an R^2 value of 0.702.
2. With Team Statistics

With incorporating the team statistics mentioned in the data description, the R^2 value shot up to 0.798, showing a 13.75% increase in the coefficient of determination. Therefore the model performed significantly better when team statistics were accounted for. This demonstrates that the inclusion of team statistics and accounting for the player's surroundings and environment are effective method in increasing the validity of the relationship between the data and points per game, therefore increasing the accuracy of the model.

The model was tested on the 2021 season stats and achieved a mean square error of 1.66. This value is significantly lower than most publicly available models. Furthermore, it is important to note that the COVID-19 situation has forced many lesser productive players to step up due to many players being placed under quarantine. This has skewed the actual values, with lower productivity players receiving more minutes than they usually would thus averaging more points than they usually would. When introducing a minimum production filter, the mean squared error drops notably. At a filter of 10 PPG, the error reduces to 1.26 and a filter of 15 PPG reduces this to 0.87. This supports the hypothesis that the higher performing players continue to play as the model predicts, however smaller role players stepping up to major rotation roles due to players being sent into quarantine has inflated their scores explaining a higher mean square error as the filter reduces.

As of January 9th 2022, roughly a third of the NBA season has finished. The model was tested on the data from this season and achieved a mean squared error of 2.4. A smaller sample size and the superior infecitivity of the Omicron Variant have compounded the problems mentioned above, further backing the reasoning behind a portion of the error. At a filter of 10 PPG, the error reduces to 1.7 and a filter of 15 PPG reduces it further to 1.3. (Filters 17 and above reduce the error to less than 1, however the sample size also decreases drastically as observed in figure 4) Note that this error should naturally reduce on its own as the season progresses and players revert to their original offensive load. A recent significant rule change for the 2021-22 season making it illegal for offensive players to draw shooting fouls from initiating unnecessary contact has also led to a league-wide drop in offensive production, which should account for the remaining marginal error.

Figure 4: A distribution function of predicted PPG in 2022. The function was estimated by using a Gaussian Kernel Density Estimate (KDE) via Virtanen et al. [2020]

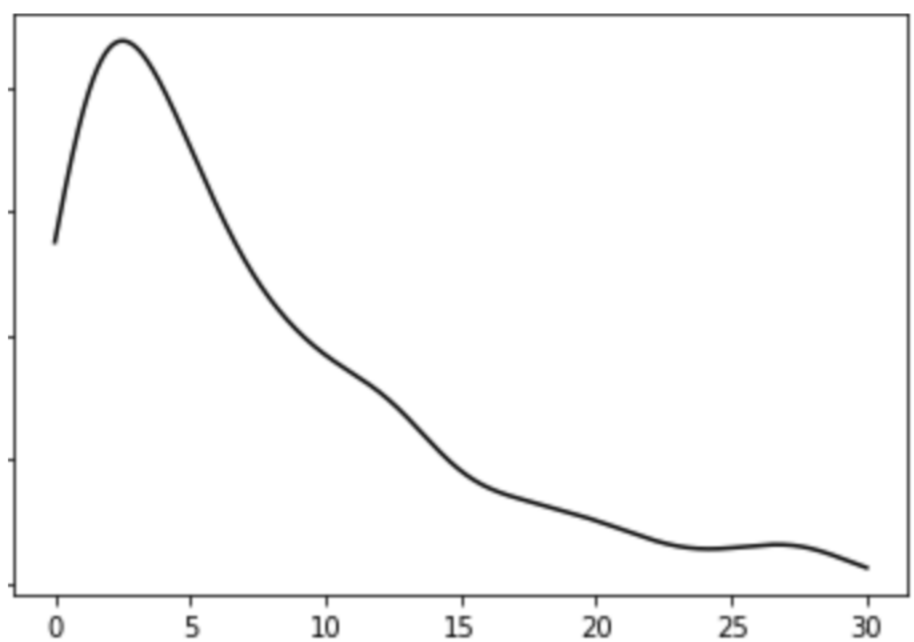
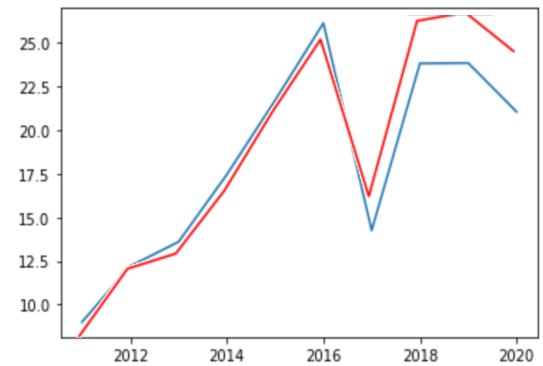


Figure 5: The Y axis shows the points per game predicted based on the cumulative statistics of the previous years, and the X axis is the year predicted. The red line is the actual PPG of the player, while the blue line is the PPG predicted by the neural network. Note that the activation function prevented extreme values as observed with linear regression. (Image generated using Hunter [2007])

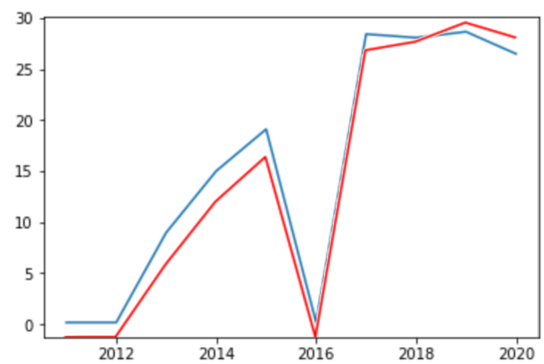
(a) Stephen Curry

(b) Kawhi Leonard

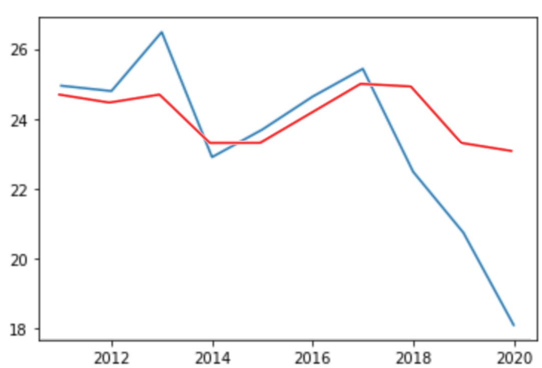


(c) Draymond Green

(d) Giannis Antetokounmpo



(e) LeBron James



(f) Joe Ingles

We can easily observe in Figure 5 that the neural network fits the data much better than the linear model and appears to be able to accurately predict the data. The predicted values coincide fairly well with the actual values. There are no extreme deviations in the values and the general shape or trend predicted appears accurate.Note that the model tends to stress on age as an important factor and can be seen to under-predict player PPG in some cases as they grow older as seen with LeBron James and Stephen Curry. However, players like LeBron and Curry are the exception to the norm and most players are well past their peak production by their early to mid 30s as observed in role players like Joe Ingles and Draymond Green, where the model more accurately fits with reality. This should seem logical as a lot of the data in the pairplot (Figure 2) is normally distributed. Therefore, the model is trained with an inherent bias towards the 'average' NBA player.

A summary of the performance of all models can be found in Table 2.

Table 2: Raw Statistics

Model	Performance (R^2)
Linear Model (with team statistics)	0.0079
MLP Regressor (without team statistics)	0.7016
MLP Regressor (with team statistics)	0.7981

3.4 Code

The models were written and data was processed in a python notebook. The SciKit Learn package (Pedregosa et al. [2011]) was used for optimising the code. For reproducibility or to see the program, please find the code and data at `https://github.com/dv145/AI_Research`. Additional statistics and visualisations of the data and model can also be found in the notebook.

4 Conclusion

This paper successfully produced a unique model to predict player PPG, a defining statistic for offensive output in professional basketball. The key reason for the success of the model is the data collected and used to make the prediction. By incorporating team statistics and advanced computed statistics into the the model, as opposed to traditional counting statistics, a higher level of accuracy was achieved. In the future, possible improvements to the model could include accounting for weather, referees, arena or city played in and other external factors. The methodology in this paper could also be extrapolated and reused for other team sports. An attempt to estimate defensive efficiency may also be looked into in the future.

5 Acknowledgements

I would like to thank Yiqiao Yin, who mentored me throughout the project and help me streamline and optimise many aspects of it. We met once a week and worked out the bugs in the code, while also discussing the bigger picture implications of the project. I learnt the mathematics and methodology behind the models themselves from Yiqiao and a Machine Learning Course on Coursera offered by Professor Andrew Ng from Stanford University. I also got a deeper understanding of the algorithms by watching a YouTube series on the same by Grant Sanderson (3 Blue 1 Brown).

6 Glossary

The meanings of all terms used in 3.1 can be found below

$STAT$ indicates a relevant statistic

$STAT_L$ indicates a league average statistic.

$STAT_T$ indicates a total team statistic (sum of the values of each player).

For the calculation of PER, 3 additional statistics were used, namely

$$\text{factor} = \frac{2}{3} - \frac{0.5 \times \frac{AST_L}{FGM_L}}{2 \times \frac{FGM_L}{FTM_L}}$$

$$VOP = \frac{PTS_L}{FGA_L - ORB_L + TOV_L + 0.44 \times FTA_L}$$

$$DRB\% = \frac{TRB_L - ORB_L}{TRB_L}$$

GP	Games Playes	PPG	Points Per Game
APG	Assists Per Game	PTS	Total Points
FGA	Field Goals Attempted	FTA	Free Throws Attempted
FGM	Field Goals Made	3PM	3 Pointers Made
TOV	Turnovers	MP	Minutes Playes
POS	Number of Possessions	STL	Steals
TRB	Total Rebounds	ORB	Offensive Rebounds
BLK	Blocks	PF	Personal Fouls
TS%	True Shooting Percentage	eFG%	Effective Field Goal Percentage
USG%	Usage Rate	AST%	Assist Percentage
PER	Player Efficiency Rating	uPER	Unadjusted PER
VOP	Value of Possession	DRB%	Defensive Rebound Percentage

Table 3: Sample Player Object

NAME	Stephen Curry
TEAM	Gol
AGE	33.17
GP	63.0
PPG	32.0
TS%	0.66
eFG%	0.61
USG%	34.8
APG	5.8
AST%	30.5
PER_n	34.28
PER	25.71
VI	11.8
SEASON	2020
PTS_T	113.7
PACE	102.2
OEFF	111.1

References

FSGA. Industry demographics, fantasy sports and gaming association. https://thefsga.org/industry-demographics/, 2019.

Josh Weiner. Predicting the outcome of nba games with machine learning. https://towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bb768f20, 2021.

NBAStuffer.com. Ugur, serhat. https://www.nbastuffer.com/, 2011.

Sean Forman. Basketball reference. https://www.basketball-reference.com/, 2021.

Donald Goldfarb Charles George Broyden, Roger Fletcher and David Shanno. Practical methods of optimization (2nd ed.). *Practical Methods of Optimization(2nd ed.), New York: John Wiley & Sons, ISBN 978-0-471-91547-8*, 1987.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jaak Uudmae. Cs229 final project: Predicting nba game outcomes. *Stanford University*, 2017.

Eric Scot Jones. Predicting outcomes of nba basketball games. *North Dakota State University of Agriculture and Applied Science*, 2016.