Instituto Tecnológico Autónomo de México

APLICACIÓN DE MÉTODOS DE REGRESIÓN LINEAL REGULARIZADA PARA EVALUAR EL VALOR DE MERCADO DE JUGADORES Y PREDECIR RESULTADOS EN LA NBA

Tesis

que para obtener el título de

Licenciado en Matemáticas Aplicadas

Presenta

Pablo López Landeros

Asesor:

Dr. Abdolnasser Sadeghkhani

Ciudad de México 2021

≪Con fundamento en los artículos 21 y 27 de la Ley Federal del Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada "**Aplicación de métodos de regresión lineal regularizada para evaluar el valor de mercado de jugadores y predecir resultados en la NBA**", otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la Biblioteca Raúl Bailléres Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una contraprestación.≫

Pablo López Landeros

Fecha

Firma

To the game of basketball,
because it came with so much more.

Aknowledgments

I would like to thank several people who made my undergraduate journey possible. First of all, my family. I want to thank my mother Claudia, for her unconditional love and for believing in me even in times when I doubted myself. My father Javier, for sparking a love for science and mathematics which prevails to this day. My brother Alonso, for being my life-long companion, and for never failing to lighten tough situations with his presence.

My non-blood family: Carlos, Omar, José Luis, Emiliano and Gustavo. For their love and friendship throughout the last fifteen years. Whether on or off a basketball court, the experiences that we have shared together make up some of the fondest memories of my life. Thank you for your wholehearted support whether at my best or at my worst. I am truly grateful to be bounded to all of you by something much stronger than blood, by choice.

My advisor Nasser Sadeghkhani. For believing in my project and for helping me develop research in a young field in which not many professors offered to assist.

I also want to thank professors Edith Mireya Vargas, Bárbara Carrillo and Rodrigo Zepeda for their valuable recommendations and revisions to this document.

My coach, Allan Clua. For all of his teachings and guidance transcended beyond a basketball court, making me a better player, a kinder person and above all, a better human being. Our interactions are something that I will always treasure, as they never fail to provide words of encouragement, direction when I am lost and uplifting spirits in trying times. Thank you for being such an extraordinary role model.

Lastly, I want to thank my colleagues Hugo Delgado, Francisco Huerta, Eric Bazaldua, Alejandro de Anda, Thomas Bladt, Luis Ares de Parga and especially Fernando Stein. Not only for being the first ones to lend their affability when I was a complete outsider to a new school. But also for accompanying me throughout all my academic endeavors in the last three years. What started with superficial coursework conversations evolved into treasured friendships without which, our difficult college experience would have been impossible.

Resumen

El baloncesto es un deporte dinámico en el que dos equipos de cinco jugadores cada uno intentan anotar la mayor cantidad de puntos y recibir la menor cantidad posible. Aunque en todo momento hay cinco jugadores por equipo en cancha, en un partido de NBA puede haber hasta 13 jugadores activos por bando. Así, el baloncesto puede ser analizado como una interacción entre múltiples unidades de 5 personas asociando a cada unidad una contribución total de puntos llamada *plus-minus*. Dicha cantidad se obtiene sumando los puntos que anota cada unidad mientras está en cancha y restándole los puntos que recibe.

Cada uno de los 30 equipos en la NBA juega 82 juegos cada temporada. Definimos N_i como el número de distintas combinaciones de 5 jugadores que cada equipo i usó durante la temporada. Luego entonces para cada equipo, construimos una matriz de tamaño $82 \times N_i$ que indica la cantidad de segundos que cada unidad de 5 personas jugó durante un partido.

Sea $\boldsymbol{X}$ la matriz de $82 \times N_i$, arriba mencionada y sea Y un vector de tamaño 82×1 con el margen de derrota o victoria para cada juego. El objetivo entonces es utilizar diversos métodos de regresión lineal regularizada para encontrar $\boldsymbol{\beta}$ tal que $\boldsymbol{X\beta} = Y$. De tal manera que, $\boldsymbol{\beta}$

tendrá en cada entrada un *plus-minus* asociado a cada unidad de 5 jugadores.

Utilizando como base el trabajo realizado por el Dr. Huang en 2018: *Cadenas de Markov y sus aplicaciones en la ciencia de datos*[9], se ajustó una regresión ridge, una regresión lasso y una regresión de red elástica a datos de la temporada 2014-2015 de la NBA para posteriormente evaluar el valor predictivo de dichos modelos utilizando datos de postemporada para los equipos que llegaron a las finales.

Nuestro mejor modelo predice correctamente el desenlace de 15 de los 21 (exactitud de 72 %) juegos para los Golden State Warriors y 13 de 20 juegos (exactitud de 65 %) en el caso de los Cleveland Cavaliers.

Como punto final, proponemos una métrica llamada CWOR (Contributed Wins Over Replacement) para explorar las posibles aplicaciones *a-posteriori* del modelo como la regulación de carga de trabajo para ciertos jugadores o la asignación de valor al momento de proponer transferencias.

Todos los datos se obtuvieron mediante la librería *nbastatR* y la metodología se implementó con los lenguajes de programación R y Python.

Summary

Basketball is a dynamic sport in which two teams of five players each try to score as much points as possible while trying to receive the least possible amount. Although, there are five players per team on the court, an NBA roster consists of thirteen active players which can be substituted in and out of the game. As such, Basketball can be modeled as an interaction between multiple 5-man units each with an associated net point contribution called *plus-minus*. Such quantity is computed by calculating how many points each 5-man unit outscores their opponent while on the court.

Since each of the 30 NBA teams play 82 games each season, for each team i, we define N_i as the number of different 5-man units the team used in a season. Let $\boldsymbol{X}$ be the $82 \times N_i$ matrix for each team that indicates the number of seconds each 5-person unit plays in each game. Let Y be the 82×1 vector giving the margin of victory or defeat for each game.

Then, the objective is to find $\boldsymbol{\beta}$ such that $\boldsymbol{X}\boldsymbol{\beta} = Y$. Thus, $\boldsymbol{\beta}$ will contain a plus-minus rate for every 5-person unit. Because $N_i > 82$ for all teams, regularized regression models were adjusted using 2014-2015 NBA season play by play data.

Furthermore, we use playoffs data for the Golden State Warriors and

the Cleveland Cavaliers to assess the models' predictive value. Our best model correctly predicts 15 out of 21 playoff games (72% accuracy) for Golden State and 13 out of 20 (65% accuracy) in Cleveland's case.

We also discuss some applications for the proposed models such as player workload management and its ability to predict game outcomes. Lastly, we evaluate our best model's ability to aid in trade evaluation scenarios.

The required data was obtained through the *nbastatR* library and computed using R and Python programming languages.

Contents

1 Preliminaries **1**

Preliminaries **1**

1.1 Mathematical Preliminaries 1

1.1.1 Statistical Learning 1

1.1.2 Linear Models and Least Squares 4

1.1.3 Markov Chains 6

1.2 Basketball Preliminaries 7

1.2.1 Advanced Statistics 9

2 Previous Work and Literature Review **12**

Previous Work and Literature Review **12**

2.1 Background 12

2.2 Linear Weights for evaluating NBA players 14

2.2.1 NBA efficiency metric 14

2.2.2 Game score ratings 15

2.2.3 Wins produced 16

2.2.4 Modelling basketball as a Markov process 17

2.2.5 Probabilistic Graphical Models 20

2.2.6 Continuous Time Markov Chains for Simulating Playing Times 22
2.2.7 Using shrinkage methods for building scoring models 23
2.2.8 Player tracking data and EPV 24

3 Scoring Models Using Regularization Methods 26
3.1 Research Objectives 26
3.2 Methodology 27
3.2.1 Regularization Methods 31
3.2.2 The need for penalized regression 35
3.3 Data Collection 36
3.3.1 Extracting lineups from play-by-play data 37
3.3.2 Extracting playing time and plus–minus from play-by-play data 40
3.4 Models 44
3.4.1 Assesing elastic net's predictive value 48
3.5 Proposed metric: Contributed Wins Over Replacement . 53
3.5.1 Ersan Ilyasova trade 54
3.5.2 Tiago Splitter trade 55
3.5.3 Problems with CWOR: Stephen Curry and LeBron James 56
3.6 Further Applications: Workload Management 57

4 Conclusion 59

Conclusion 59
4.1 Results and Model Limitations 59
4.2 Further Work 61

A Tables **63**

Bibliography **70**

List of Tables

1.1 20 highest plus–minus recorded by a player in a game throughout the 2014-2015 season 11

2.1 Weights assigned to different statistics, relative to points. 17

3.1 Example of $\boldsymbol{X}$ matrix. 30
3.2 Example of Y vector. 30
3.3 5-man units used by each team during the 2014-15 regular season. 36
3.4 Players who started in a quarter but did not participate in relevant in-game events during the 2014-2015 season. 39
3.5 Sample table of how our data looked after computing playing time and net points for the first Golden State Warriors game. 41
3.6 Difference (in minutes) between our computed playing times and the scraped data from Basketball Reference for the 18 lineups with more recorded on-court playing time. 42
3.7 Top rows for the generated chronological per game lineup usage series for the Dallas Mavericks. 43

3.8 Comparison between predicted and actual outcomes for each of the Golden State Warriors' playoff games. The model correctly predicts the winner in 72.72 % of the games. 50
3.9 Predictions for each of the Cleveland Cavaliers' playoff games. The model accurately predicts the outcome of a game with 65% of accuracy. 51
3.10 Predictions for the Atlanta Hawks' playoff games. Accuracy for this experiment is 43.75% as it correctly predicts 7 out of 16 outcomes. 52
3.11 Projected performance for the San Antonio Spurs with and without Tiago Splitter in their lineup. 55

A.1 Sample rows of our recorded substitution data. 63
A.2 Substitution data. 64
A.3 Rows from raw play-by-play data. 65
A.4 Predicted margin of victory or defeat with and without Ersan Ilyasova in The Milwaukee Bucks' lineup. 66

List of Figures

1.1 Example of a player substitution. 8
1.2 Example of a final box score from a Cleveland Cavaliers vs Golden State Warriors game. 9

2.1 Possession Graphical Model for sequence of events in a game. 21
2.2 EPV for two different possesions 25

3.1 Flowchart of our work process 28
3.2 Predicted point differential (y) vs Actual point differential (x) for the Cleveland Cavaliers. 45
3.3 Results for in-sample predictions using an elastic net model for each team. 47

Chapter 1

Preliminaries

This chapter, introduces some preliminaries that will be used throughout this thesis. The first section is devoted to the main statistical and mathematical notions while the second section, briefly introduces important basketball concepts that readers should be familiar with to have a better understanding of this topic.

1.1 Mathematical Preliminaries

1.1.1 Statistical Learning

Statistical Learning plays a key role in many areas of science, finances and industry. As the name suggests, *Statistical Learning* is a framework for examining and extracting information from large datasets. This framework provides a way to predict an *output variable* that has some degree of dependancy on a given set of *input variables*. For each case, the goal of applying statistical learning is to use the inputs to predict the values of the outputs in each observation. This particular exercise is called *supervised learning*. The other main branch within statistical

learning is *unsupervised learning*.

1.1.1.1 Supervised Learning

With supervised learning, we are concerned with prediciting the values of one or more response variables $Y = (Y_1, \ldots, Y_m)^T$ for a given set of input or predictor variables $X = (X_1, \ldots, X_N)^T$. Furthermore, we denote the parameters for our *i-th* observation from our dataset as $x_i = (x_{i1}, \ldots, x_{ip})^T$ with response measurement y_i. Then, the predictions are made based on the pairs $(x_1, y_1) \ldots (x_N, y_N)$ all for which we know the output response.

This is why supervised learning is also known as "learning with a teacher". Under this metaphor, the student (our trained model) presents an answer $\hat{y}_i$ for each x_i in the training sample and the "teacher" provides either the correct answer or an error associated with the student's answer. The difference between the real output value and the student's answer $\hat{y}$ is characterized by a *loss function* $L(y, \hat{y})$. One of the simplest loss functions is $L(y, \hat{y}) = (y - \hat{y})^2$ known as squared error loss function.

Suppose we have an outcome measurement, usually quantitative (price of a stock), that we wish to predict from a set of features (company performance measures and economic data). We call this set of features and responses the *training set*.

Using this data, we can build a prediction model, or *learner* which will enable us to predict the outcome variable for new unseen objects. To validate the model's ability to accurately predict an outcome, we use 30% of our training set (for which we have the actual outcome values) and we treat it as unseen data.

If we want to adopt a more mathematical approach to defining supervised learning, we suppose that (X, Y) are random variables

represented by a joint probability distribution $f_{X,Y}(x, y)$. Where Y is an independent variable which we want to predict and X is the set of independent features from which we wish to make a prediction. As such, supervised learning can be defined as a density estimation problem where we want to determine the properties of the conditional distribution $f_{Y|X}(y|x)$. In most cases, we are most concerned with estimating the location parameters μ that minimize the error and the loss function at each x.

We can formally define μ as:

$$\mu(x) = argmin_{\theta} E^{Y|X} L(y, \theta).$$

Conditioning on one variable and applying Bayes Theorem, we can then find the joint density as:

$$f_{X,Y}(x, y) = f_{Y|X}(y|x) f_X(x),$$

where $f_X(x)$ is the marginal density of X values alone. This density will be of little to no interest in supervised learning problems. Since often Y is typically of dimension 1, and only the location parameter $\mu(x)$ is of interest, the problem is greatly simplified.

1.1.1.2 Unsupervised Learning

The other main branch of Statistical learning theory is *unsupervised learning*. Extending our previous metaphor, unsupervised learning would be the equivalent of "learning without a teacher". In this case, we have a set of N observations $(x_1, x_2, \ldots, x_N)^T$ from a random p vector X having joint density $f(X)$. The main goal of unsupervised learning problems is inferring the properties of this probability density function without the help of a supervisor or "teacher" that provides a

right or wrong answer. One main difference with supervised learning is the fact that X has a much higher dimension in unsupervised learning problems and thus, the properties of interest for our target density function are much more complex than location parameters. This caveat is somewhat reduced by the fact that X represents all of the variables in consideration and thus, one is not concerned with inferring how the properties of $f(X)$ change when we condition on another set of variables.

With unsupervised learning, we also use functions to establish a measure of success or lack thereof and judge the adequacy of different models over our problem of interest. Lack of success is directly measured by expected loss over the joint distribution $f_{X,Y}(x, y)$. Contrary to supervised learning, we cannot measure success quantitatively. It is difficult to assess the validity of the inferences drawn from unsupervised learning algorithms. For this reason, in most cases heuristic arguments are used for judging the quality of the results.

1.1.2 Linear Models and Least Squares

Within supervised learning, linear models have been one of the most powerful, yet simple prediction tools for the last 40 years. These type of models make huge assumptions about the data and yield stable (low variance) but possibly innacurate (high biase) predictions.

Given a matrix of inputs $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^T$ where each $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ is a vector of feature measurements for the ith case, we predict Y via the model:

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} x_{ij}\hat{\beta}_j, \forall i \in \{1, \ldots, n\} \tag{1.1}$$

where $\hat{\beta}_0$ is the intercept and also known as the *bias* in supervised

learning. Equation 1.1 can be written as:

$$\hat{Y} = \boldsymbol{X}\hat{\boldsymbol{\beta}} \tag{1.2}$$

where $\boldsymbol{X}_{n\times(p+1)}$ is a matrix whose first column is the column $\mathbf{1} = (1, 1, \ldots, 1)^T$. In equation 1.2, $\hat{Y}$ is a vector of length n and $\hat{\boldsymbol{\beta}}$ is a $(p+1) \times 1$ vector of coefficients.

The most popular method to fit a linear model is the method of **least squares**. In this approach, we choose $\boldsymbol{\beta}$ which minimizes the residual sum of squares, given by:

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

which leads to the least squares estimator in :

$$\hat{\boldsymbol{\beta}} = argmin||Y - \boldsymbol{X}\boldsymbol{\beta}||_2^2$$

Because $RSS(\boldsymbol{\beta})$ is a quadratic function of the parameters, its minimum always exists but it may not be unique. The solution is easier to write if we use matrix notation:

$$RSS(\boldsymbol{\beta}) = (Y - \boldsymbol{X}\boldsymbol{\beta})^T(Y - \boldsymbol{X}\boldsymbol{\beta}), \tag{1.3}$$

where $\boldsymbol{X}$ is an $n \times (p+1)$ matrix where each row is an input vector, and Y is a n-vector of the outputs in the training set. Differentiating Equation 1.3 with respect to $\boldsymbol{\beta}$ we get $p+1$ equations known as **the normal equations**:

$$\boldsymbol{X}^T(Y - \boldsymbol{X}\boldsymbol{\beta}) = 0.$$

Solving for $\boldsymbol{\beta}$, we get the unique solution:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T Y \tag{1.4}$$

provided that $\boldsymbol{X}^T\boldsymbol{X}$ is nonsingular and the fitted value at the *i-th* input is $\hat{y}_i = \hat{y}(X_i) = X_i^T\hat{\boldsymbol{\beta}}$ for $i = 1, \ldots, n$. Note that the entire fitted surface is caracterized entirely by the $p+1$ parameters $\hat{\boldsymbol{\beta}}$.

1.1.3 Markov Chains

A stochastic process with discrete states S and discrete parameter space $\{X_n; n = 0, 1, 2, \dots\}$ is called a *Markov Chain* if:

$$P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j | X_n = i).$$

This is to say, the probability of advancing to the next state is defined only by the current state of the system. In this case, we can define a *transition probability* as the probability to transition as

$$P_{ij} := P(X_{n+1} = j | X_n = i) \, \forall n \in \mathbb{N},$$

A *transition matrix* for a given Markov chain with states $S = \{1, 2, ..., k\}$ is a square matrix of size $k \times k$ and is built as:

$$T = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1j} & \dots P_{1k} \\ P_{21} & P_{22} & \dots & P_{2j} & \dots P_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{k1} & P_{k2} & \dots & P_{kj} & \dots P_{kk} \end{bmatrix}$$

where
$\sum_{j=1}^{k} P_{ij} = 1$ and $P_{ij} :=$ Probability of moving from state i to state j.

1.2 Basketball Preliminaries

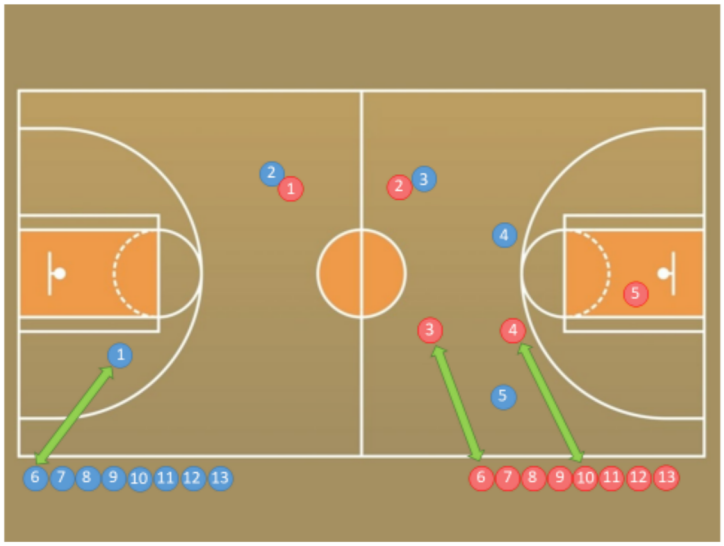
Basketball is a dynamic sport in which two teams of 5 players each, compete in a rectangular court with the primary objective of scoring more points (by shooting a ball through the opponent's hoop) while preventing the opposing team to score points on their own hoop.

1. *Substitutions*

   Even though only 5 players per team may be on the court at a given moment during a basketball game, a team has 13 active players on its roster. Throughout the game, any one of the 13 players can be substituted out of the game and return to play at a later time. As such, each team has $\binom{13}{5}$ combinations of different players that can be on court at the same time. The study of basketball as the interaction between different 5-player units is a key component in our work. Figure 1.1 shows an example of a player substitution.

Figure 1.1

Example of a player substitution.



2. *Possession*

   In basketball, a team is in possession when a player is in control of the basketball. This can be achieved by holding, dribbling or passing the ball. A team possession ends when the defensive team gains possession or there is a field goal attempt.

3. *Box Score Statistic*

   In basketball, a box score is a detailed summary of the results from a game. A basketball box score features a detailed breakdown of team and player statistics, such as minutes played, total points, field goal percentage, three-point shot percentage, rebounds, free throw percentage, assists, steals, and blocked shots. See Figure 1.2.

   From these statistics, one can get a glance at the team's performance in a game. Moreover, many of the first attempts at

dissecting the game from an analytical perspective, relied heavily on applying statistical methods to box scores.

Figure 1.2

Example of a final box score from a Cleveland Cavaliers vs Golden State Warriors game.

Cleveland Cavaliers

PLAYER	MIN	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	STL	BLK	TO	PF	PTS	+/-
LeBron James F	45:46	18	38	47.4	2	8	25.0	6	10	60.0	0	8	8	6	0	0	4	2	44	-3
Tristan Thompson F	47:09	1	4	25.0	0	0	0.0	0	0	0.0	6	9	15	1	1	1	2	3	2	-6
Timofey Mozgov C	33:12	5	10	50.0	0	0	0.0	6	8	75.0	3	4	7	2	0	1	1	1	16	3
Iman Shumpert G	34:27	2	6	33.3	2	4	50.0	0	0	0.0	1	1	2	0	4	1	2	2	6	-7
Kyrie Irving G	43:37	10	22	45.5	2	8	25.0	1	1	100	2	5	7	6	4	2	1	5	23	5
JR Smith	34:21	3	13	23.1	3	10	30.0	0	0	0.0	0	4	4	0	0	0	0	2	9	-8
James Jones	17:05	0	1	0.0	0	1	0.0	0	0	0.0	0	1	1	1	1	0	1	4	0	-11
Matthew Dellavedova	9:23	0	0	0.0	0	0	0.0	0	0	0.0	1	0	1	3	0	0	0	1	0	-13
Joe Harris	DNP - Coach's Decision																			
Brendan Haywood	DNP - Coach's Decision																			
Shawn Marion	DNP - Coach's Decision																			
Mike Miller	DNP - Coach's Decision																			
Kendrick Perkins	DNP - Coach's Decision																			
TOTALS		39	94	41.5	9	31	29.0	13	19	68.4	13	32	45	19	10	5	11	20	100	-8

Source: *Cavaliers @ Warriors*, 2016 (https://www.nba.com/game/cle-vs-gsw-0041400401/box-score)

1.2.1 Advanced Statistics

Data analytics in sports refers to analyzing games, as well as players and team's performance, in an objective manner through statistical evidence.

The main framework for advanced statistics in basketball consists of deriving new metrics from individual or team level box score statistics. The aim of these created quantities is to provide insights which may not be grasped by intuition or simple box score statistics. One of these omnipresent metrics is the *plus–minus*.

[Plus–Minus] The Plus–Minus metric will be one of the core advanced basketball statistics and a key tenet in our work. This metric originated in ice hockey as the number of goals by which a team outscores their opponent when a given player is on the ice. Translated to basketball, a player's *plus–minus* is defined by how many points a team outscores their opponent while said player is on the court.

One important thing to notice is that we can extend the plus-minus statistic to a group of any 2, 3, 4 or 5 player unit. Moreover, if we sum every 5-player unit's plus–minus statistic obtained across an entire game, we get the total amount of points each team scored in a game. Table 1.1 shows the 20 highest plus–minus recorded by a player in a game throughout the 2014-2015 season.

Table 1.1
20 highest plus–minus recorded by a player in a game throughout the 2014-2015 season

Player	Team	Plus Minus	Seconds Played
Dion Waiters	Cleveland Cavaliers	45	1614
Harrison Barnes	Golden State Warriors	45	1785
Chris Paul	Los Angeles Clippers	44	1874
Klay Thompson	Golden State Warriors	44	1693
Draymond Green	Golden State Warriors	43	1669
Timofey Mozgov	Cleveland Cavaliers	42	1736
Damian Lillard	Portland Trailblaizers	40	1566
Jrue Holiday	New Orleans Pelicans	40	1660
Rajon Rondo	Boston Celtics	40	1889
Darren Collison	Sacramento Kings	39	1894
Tyler Zeller	Boston Celtics	39	1868
Klay Thompson	Golden State Warriors	38	1753
Klay Thompson	Golden State Warriors	38	1969
Tristan Thompson	Cleveland Cavaliers	38	1446
Ben McLemore	Sacramento Kings	37	2146
Boris Diaw	San Antonio Spurs	37	1420
D.J. Augustin	Detroit Pistons	37	2118
DeAndre Jordan	Los Angeles Clippers	37	1818
Jamal Crawford	Los Angeles Clippers	37	1240

Source: Data obtained through nbastatR library.

Chapter 2

Previous Work and Literature Review

Within the world of data analysis, basketball analytics has attracted significant interest in recent years. Different machine learning and deep learning methods have been applied to box score results to try to predict game outcomes and margins of victory in the NBA. Although, the box score analysis and indivual performance assessement have made significant strides, few efforts have been dedicated to analyzing basketball as the interaction between different 5 person units.

2.1 Background

In 1985 Bill James stated in his *Baseball Abstract* that although he was a statistician, he did not pay attention to statistics in the stock market, the weather, the gross national product, the crime rate, the circulation of magazines or even world hunger. He only cared about baseball statistics. Why? Because he argued that *unlike the statistics*

in other areas [baseball statistics] have acquired the power of language. (James, 1985).

Seventeen years later, in his book *Moneyball: The Art of Winning an Unfair Game*, author Michael Lewis shared the story of how General Manager Billy Beane of the Oakland Athletics revolutionized baseball by using statistics and data analysis to find value in players that were overlooked by the rest of the league. As a result, the Oakland Athletics achieved what was at the time, the longest winning streak in the American League history and went on to play the American Series Division Championship. Beane's success story ignited a paradigm shift in baseball that led teams to use statistics and gather as much game data as possible in order to gain an edge over their opponents. This movement not only stayed in baseball but carried over to other major sports. Basketball was no exception to this.

Although basketball analytics does not have a pioneer like baseball's Bill James or is yet to have its revolutionary Moneyball moment, data analysis has made its way into the NBA and produced a vast amount of changes in the game of basketball. The most notorious example is the decline in usage of the mid range shot and its substitution for the three point shot. In 2017, Dan Kopk, data editor for the online portal *Business of Sport* argued that over the last decade, the NBA has embraced the use of statistics in a way that may even surpass that of Major League Baseball.

In this section, we explore some of the existing works that have been done on basketball data analysis and review some of these models in the literature.

2.2 Linear Weights for evaluating NBA players

One of the earliest methods for measuring player ability within basketball is through the use of *Linear Weight Methods*. This method multiplies each box score statistic by a weight and equates the weighted sum of player statistics as a measure of a player's ability (Winston 2015, 195). Some of the first well known weighting schemes used to rate NBA players are:

- NBA efficiency metric
- Game Score Ratings
- Wins Produced

2.2.1 NBA efficiency metric

Created by Dave Heeren and considered the first attempt at developing an advanced stat in basketball. This metric is computed as:

$$\begin{aligned}\text{efficiency per game} =&(\text{points}) + (\text{rebounds}) \\ &+ (\text{assists}) + (\text{steals}) \\ &+ (\text{blocked shots}) - (\text{turnovers}) \\ &- (\text{missed FG }) - (\text{missed FT}).\end{aligned}$$

where FG : field goals, FT : free throws.

What this basic form does is add all the positive contributions a player makes to his team and substract the negative ones. This results problematic when evaluating different positions within the game of

basketball as it assumes that all positive contributions are worth 1 value point and all the negative ones are worth -1 value point.

For example, point guards in the NBA are expected to facilitate scoring opportunities for their teammates and distribute the ball around the floor while limiting the amount of times their team loses the ball without scoring points. Thus, when evaluating point guards, *assists* would increase our value much more than *points* or *rebounds* would. Moreover, *turnovers* will have a higher negative impact than *missed FG*. A useful correction would be to apply multiplying factors to the original formula in order to reflect more accurately a point guard's true value.

2.2.2 Game score ratings

In an attempt to improve the drawbacks of the NBA efficiency ratings, John Hollinger (et.al 1995) created the *Game Score* formula by assigning different multiplying factors to each statistic in order to rank player performances during a game. Game Score is computed as:

$$\begin{aligned}\text{game score} =& 1.0(\text{PTS}) + 0.4(\text{FGM}) - 0.7(\text{FGA}) \\ & - 0.4(\text{FTA-FTM}) + 0.7(\text{AST}) + 0.7(\text{OREB}) \\ & + 0.3(\text{DREB}) + 1.0(\text{STL}) + 0.7(\text{BLK}) \\ & - 0.4(\text{PF}) - 1.0(\text{TO}).\end{aligned}$$

where FGM : field goals made, FGA : field goals attempted, FTA : free throws attempted, FTM : free throws made, $OREB$: offensive rebound, $DREB$: defensive rebounds, STL : steals, AST : assists, BLK : blocks, PF : personal fouls, and TO : turnovers.

Although this represented an improvement over the original NBA efficiency rating, Berri [1995] noticed something disquieting about the Game Score metric:

> *A player shooting over 20.4 % on three-pointers will increase Game Score by taking more shots. A player shooting over 29.2 % on two pointers will also increase his Game Score by taking more shots. (This implies that the worst shooter in the league would help his team by taking more shots).* *(Winston 2015,199)*

Moreover, he noticed a 0.95 correlation between Hollinger's game score rating and the NBA efficiency ratings and hence the game score is no more effective at evaluating players and teams than the NBA efficiency rating.

2.2.3 Wins produced

Wins produced is a metric developed in 1999 by Economics Professor David Berri. This method is based on the regression analysis of box score statistics in order to determine the total number of wins (or losses) for which a player is responsible. Thus, by adding up all of a team's players Wins produced we get a very accurate estimate of a teams' actual number of wins in a season. This is of particular interest as it relates individual player performance to team performance. Much like the results we seek in this dissertation.

Table 2.1 shows some relevant forms of linear player evaluation methods derived from modifying the relative weights assigned to each statistic in the NBA efficiency rating formula.

Table 2.1

Weights assigned to different statistics, relative to points.

Statistic	Manley Credits	Hoopstat Grade	Steele Value	Hereen Tendex	Belloti Points Created	Clearbaut Quality Points	Mays Magic Metric	Scheller TPR	Hollinger PER	Berri Indiv. Wins
PTS	1	1	1	1	1	1	1	1	1	1
AST	1	1.39	1.25	1	1.08	0.63	0.98	0.9	0.79	0.92
OREB	1	1.18	1	1	0.92	0.63	0.71	0.75	0.85	3.82
DREB	1	0.69	1	1	0.92	0.63	0.71	0.75	0.35	1.71
STL	1	1.39	1.25	1	0.92	0.63	1.09	1.8	1.2	2.44
BLK	1	1.94	1	1	0.92	0.63	0.87	1.1	0.85	0.86
Missed FG	-1	-0.83	-1	-1	-0.92	-0.63	-0.71	-1	-0.85	-1.38
Missed FT	-1	0	-0.5	-1	-0.92	-0.24	-0.55	0.9	-0.45	-0.79
TOV	-1	-1.11	-1.25	-1	-0.92	-0.63	-1.09	-1.8	-1.2	-2.77
PF	0	0	-0.5	-1	-0.46	0	0	-0.6	-0.41	0.46

Source: *Basketball on Paper*, by Dean Oliver, 2011, p. 83, Potomac Books Inc.

2.2.4 Modelling basketball as a Markov process

Shirley (2007) considered a series of transitions between discrete states to model the basketball game using a Markov chain model. Recall that Markov chain models specify the probability distribution of the next state is determined only by the present state. This idea bears enough resemblance with the progression of a basketball game and any other sport for which we can define a series of discrete states. In Shirley's model, each of the 30 discrete states in a basketball game is defined in terms of:

1. which team has possesion: A or B

2. how the team gained possesion: Restarting the action with an inbound pass (i), steal or non-whistle turnover (s), offensive (o) or defensive rebound (d), and going to the free throw line after a shooting/bonus/technical foul (f).

3. the number of points scored on the previous possesion (0, 1, 2 or 3). [1]

Thus, this leads to a discrete set of 40 states:

$$\{A, B\} \times \{i, s, o, d, f\} \times \{0, 1, 2, 3\},$$

in terms of which we can define any sequence in a basketball game.

To clarify, we break down a sequence from the 2013-2014 NBA finals: Team A scores a 2pt shot after team B missed a 2 pt shot, Team B then inbounds and misses a 3pt shot, Team B recovers the offensive rebound and makes a 3pt shot. Using the defined states, this sequence of plays is represented as:

$$A_d 0 \to B_i 2 \to B_o 0 \to A_i 3.$$

Note that 10 of the 40 defined states are not possible in a game. For example, if team A obtains possesion by stealing the ball from the opponent, no points could have been scored by team B on the previous possesion. Or if team A scored 3 points on their possession, it is impossible for team B to have obtained a defensive rebound. This eliminates 10 states and leaves us with 30 possible states with which any sequence of events in a basketball game can be represented.

Shirley's objective was to accurately simulate basketball games using Markov chains in order to compute quantities of interest such as in-game win probability, expected points in a possesion or change in win probability as a function of the number of possesions left in a game.

Using play-by-play data from 2252 NBA matches, he fitted a model for each team and inferred the transition probability matrices through

[1] Although four point plays are possible within a game, the author decided to exclude them from the model due to their rarity.

summary statistics. Once adjusted, he showed that the simulations based on such models were good at estimating a team's win probability against an average opponent.

Shirley concluded that when simulating a whole season, the model quite accurately represents a team's winning percentage ($R^2 = 0.935$). Nevertheless, his model's forecasting value was not explored at all, as all of the results were based on in-sample data. The author also discusses the possibility of improving his results by using in-match data to fit the models (as opposed to summary statistics).

In addition to this, the author defines the use of Markov chains as an excellent compromise between simplicity and complexity. They capture all of the relevant and rare events that occur during a game while being simple enough to fit and interpret.

Štrumbelj and Vračar (2012) elaborated on Shirley's work and improved his approach by incorporating relative team strengths into Shirley's original model. Thus, this improved model simulated games between two teams as opposed to the original model in which the simulations consisted of each team playing a standard average opponent.

The calculation of transition probabilities was also altered in this new approach. While Shirley used classic box score statistics to infer each team's transition probabilities from one state to another, Štrumbelj and Vračar used effective field goal percentage, turnover ratio, offensive rebound ratio, defensive rebound ratio, free throw factor, opponent's effective field goal percentage, opponent's turnover ratio and opponent's free throw factor.

These statistics are being used in the multinomial logistic regression to compute each team's transition matrices.

After fitting the models and simulating the same season as Shirley

did, this improved model obtained better estimates on teams' actual win percentage for both in-sample data ($R^2 = 0.91$) and out-of-sample data ($R^2 = 0.85$).

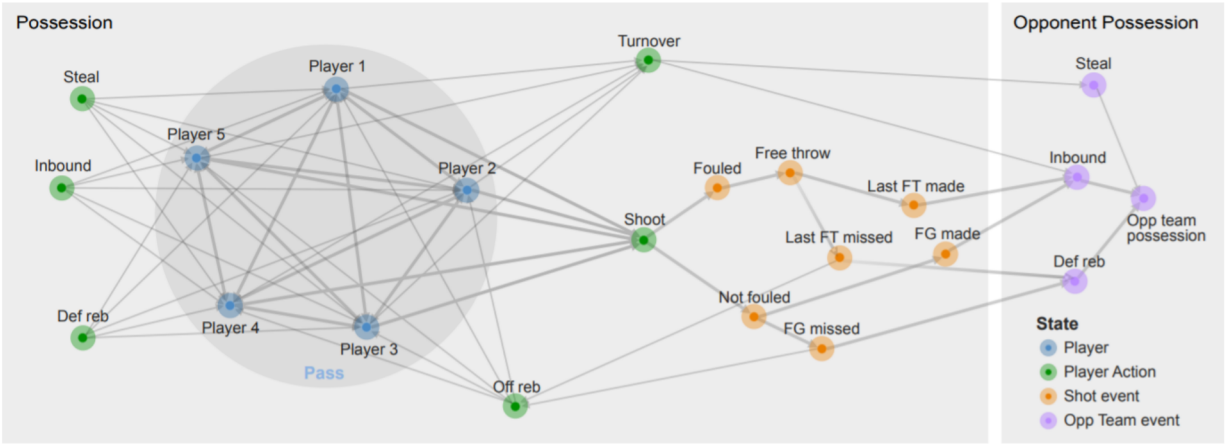
Even though their results were better, the authors recognize that their models overestimate weaker team's winning percentage much like Shirley's original model. This problem arises because both models use homogenous Markov chains. This means that the transition probabilities between states do not change as the match progresses. This assumption is invalid in most sports and especially in basketball where a team's decision making process is severely affected by the time left in a game.

2.2.5 Probabilistic Graphical Models

Although Shirley, Štrumbelj and Vračar's work showed some promising results, their results treat each team as a unit and thus, fail to consider how game outcomes are affected by lineup dynamics. Most approaches to basketball forecasting ignored the fact that ball movement dynamics are highly sensitive to the on-court lineups. In an attempt to bridge the gap between individual player performance and team level performance, Oh, et.al (2015) proposed a probabilistic graphical model for simulating basketball matches.

Figure 2.1

Possession Graphical Model for sequence of events in a game.



Source: "Graphical Model for Basketball Match Simulation" by Oh, M.-H., Keshri, S., and Iyengar, 2015, *MIT Sports Analytics Conference*.

This model still simulates each game as a series of transitions between discrete states but instead of using Markov chains, they used graph theory. By viewing match progression as a graph, the authors defined each node as a network structure of players on the court and the edges of the network as all the possible actions in the flow of the game. The model is subsequently calibrated using play-by-play data from the 2013-2014 season from which the authors establish the conditional probability of moving from each one of the nodes on the network to another.

Their results confirm that changes in both teams' lineups significantly affect match progression. Using 70 % of the 2013-2014 season games as their training data and the remaining 30% as their test set data, their model obtained good estimates for each team's true win percentage, $R^2 = 0.92$ for in sample data and $R^2 = 0.87$ for out of sample data respectively.

2.2.6 Continuous Time Markov Chains for Simulating Playing Times

Huang (2018) proposed a two-part model based on scoring rates which made use of *continuous time Markov chains* (CTMC) in order to simulate the NBA games. His models correctly predicted 12 out of 15 playoff series outcomes even though they suffered from absorbent states.

The first part of the model consisted in using a continuous time markov chain in order to accurately represent each 5-man unit's playing time for every team in the NBA.

Let N_i be the number of different lineups of team i, a transition matrix of size $N_i \times N_i$ is built for every team. By assuming substitutions in basketball are independent and exponentially distributed, each (j, k)-th entry of the i-th transition matrix corresponds to the maximum likelihood estimator for a CTMC matrix. These quantities are computed through play-by-play data and give us a good estimate of the number of transitions from lineup j to lineup k.

By using Monte Carlo simulation and following standard algorithms for sampling from a continuous-time Markov chain, Huang simulates the total number of substitutions for every NBA team during the 2014-2015 NBA season.

Using results for 29 of the 30 teams in the league, the correlation between the simulated and true number of substitutions is 0.8634. For the Boston Celtics, simulations predict 4627.57 substitutions in one season, while the actual number of made substitutions was 1792. The author points out that this is a consequence of assuming substitutions are exponentially distributed and aknowledges that there may be better distributions than the exponential to model the time played by

one lineup before substituting it for another.

Once playing times for each unit were accurately simulated, he considered building a model for how each 5-person unit contributes to the overall score of the game by assigning a scoring rate to each lineup N_i and multiplying the associated scoring rate by the time spent in each state. Summing these point differentials across 48 minute games gives the final score a team would achieve in a game.

2.2.7 Using shrinkage methods for building scoring models

Moreover, Huang (2018) applied ridge regression in conjuction with the plus–minus statistic for each team's 5-player lineup units in order to build a scoring model for the 2014-2015 NBA season. This model takes an $82 \times N_i$ matrix as input where N_i is the number of unique lineups each team i used throughout the season. He defines this matrix as $\boldsymbol{X}$. Subsequently, he defines Y as an 82×1 vector that contains the margin of victory or defeat for each game. The objective then was to find $\boldsymbol{\beta}$ such that $\boldsymbol{X\beta} = Y$. Thus, $\boldsymbol{\beta}$ will contain a plus-minus rate for every 5-person unit.

It is worth noting that because $N_i > 82$ for all teams, the linear system is undetermined. This is the reason why shrinkage methods are required for building these models.

Another remark worth noting is the fact that while the usual penalty for ridge regression is $||\boldsymbol{\beta}||_2^2$, when the author uses this for adjusting the models, the performance is worse than the original CTMC scoring rate model. Therefore, Huang modifies the penalty parameter to $||\boldsymbol{\beta} - \boldsymbol{\beta_0}||_2^2$, where $\boldsymbol{\beta_0}$ is the vector of estimated plus-minus obtained for each lineup in the earlier scoring model.

The model is adjusted using the regular season as training data and the playoffs as test data. In the end, it correctly predicted 12 out of 15 playoff games and obtained an overall accuracy score of 0.682.

2.2.8 Player tracking data and EPV

Even though the statistical prediction methods have been of utmost importance, the availability of optical player tracking data has driven most of the research in basketball analytics lately. Models are centered around player tracking data and are particularly helpful when identifying individual player tendencies and decision making ability.

One of the most promising result in this area is the *expected possession value* (EPV) metric developed by Cervone, et. al (2014). This quantity derives from a stochastic process model which uses optical player tracking data to estimate how many points an offense will score given the spatial configuration of players and the ball at a certain moment. Every dribble, pass, on or off ball movement triggers a change in the number of points the offense is expected to end the possesion with.

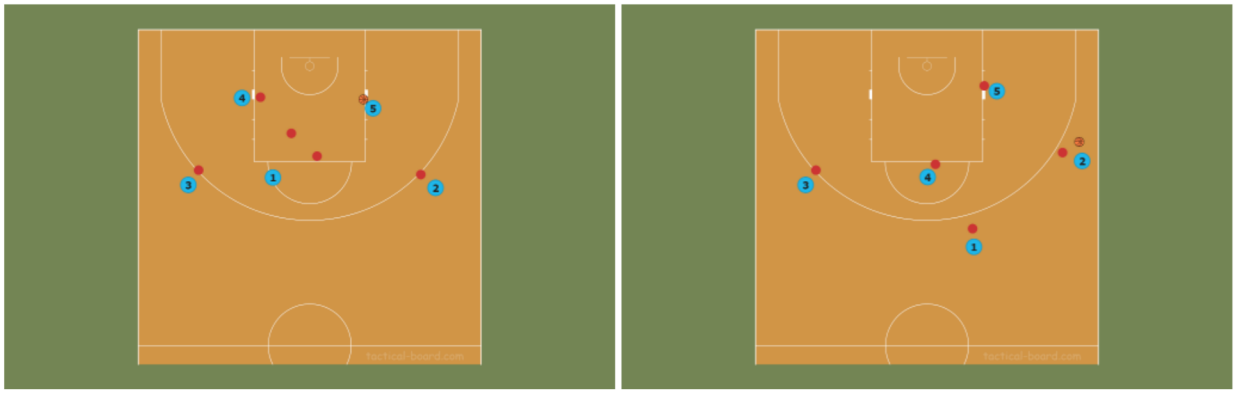
For example Figure 2.2a illustrates an offense at a stage with an $EPV \approx 2$. Since player number 5 has possesion of the ball with a clear lane to the basket and no opposing player to prevent him from scoring. As such, we would expect this possesion to end in a 2-point basket.

Similarly Figure 2.2b portrays an offense where every attacking player is tightly covered by an opponent and the ball is rather far from the rim. Hence, this offense is in a stage where we would expect the attacking team to score less than one or two points ($EPV < 2$).

Cervone et al (2014) also compared the EPV for multiple players in the NBA. Different players were put in similar or identical situations in order to better understand how and why EPV changes.

Figure 2.2

EPV for two different possesions



(a) Possesion with $EPV \approx 2$ (b) Possesion with $EPV \leq 2$

One of the drawbacks of the EPV is it does not take into account the value of interactions between specific teammates. As effective as this model is at evaluating individual player performance and decision making, it still lacks accountability for capturing a team's ability to perform as a unit.

In particular, this model does not consider *substitutions, lineup usage and time played by each lineup.* Given that basketball consists of two 5-player units trying to outscore each other, more hollistic approaches are still required as a complement to these models if an effcient forecast is to be made.

Chapter 3

Scoring Models Using Regularization Methods

In this section we will focus on ridge, lasso and elastic net regressions to explore the lineup usage data results. The performance of our proposed models will be evaluated. These models are being used to address specific player usage caveats in different trade evaluation scenarios.

3.1 Research Objectives

As stated in Section 2.2, we build a model employed by Huang (2018) to explore the predictive value of ridge, lasso and elastic net regression models for 5-player lineup usage data. Nevertheless, his model does not consider other shrinkage methods since there is no reason in promoting sparsity and thus, the ridge is chosen over lasso. *In here, we extend his results using lasso and elastic net models to get some improvement in terms of performance comparing to the ridge*

regression model for predicting the outcome of the NBA games. Our model's ability for assesing player value in trade scenarios will be considered. To do so, questions regarding playing time restrictions will be studied. Particularly our model would prove useful to evaluate how a team's season outcome could be affected if a player had not been in the roster for the season. This can provide the organization with insights that aid in the decision process of whether or not to trade a player.

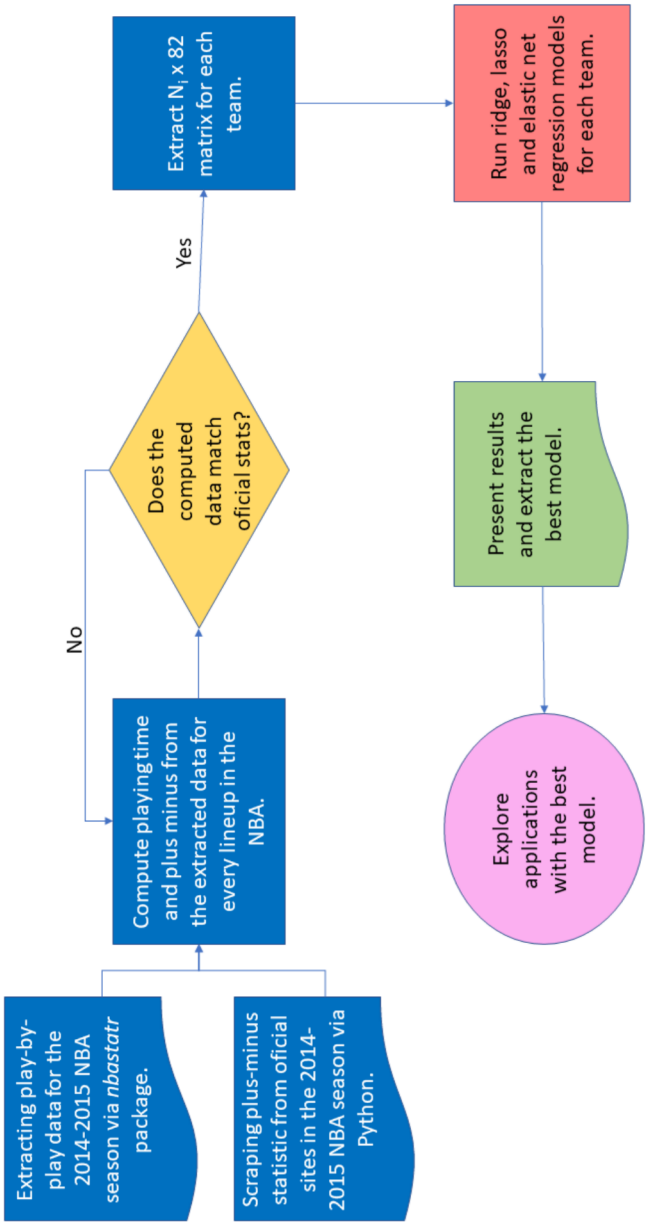
3.2 Methodology

After extracting and processing play-by-play data for the 2014-2015 NBA season, we end up with 30 matrices (1 per each NBA team). Each matrix has $82 \times N_i$ dimensions where N_i is the number of different lineups each team i used throughout the season. For further commodity, this matrices will be refered to as *usage matrices*. Figure 3.1 provides a general flowchart of our work process from the data collection process to the evaluation of our final models.

Although this provides a good summary of our data wrangling process, in section 3.3 we expand upon how the data was mined and prepared in order to transform play-by-play events into the desired matrices.

Figure 3.1

Flowchart of our work process



For each team, the (j, k) entrance of the usage matrix has the playing time for the k-th lineup in the j-th game of the season. Table 3.3 shows the number of different lineups each team used during the season according to our computed matrices.

Furthermore, for each team i, an 82×1 vector was computed. This vector contains the margin by which each team won or lost the game. For instance, the 5th entry of this vector for Philadelphia is -2. Which means that they lost the fifth game of the season by two points. On the other hand, if an entry in this vector is positive, it means the team won the game.

Recall that in linear regression, the $\boldsymbol{X}$ matrix consists of n observations each one with p different variables for which measurement units may differ. In our approach, each observation represents a regular season game for which the features are the amount of seconds each different lineup played during a game and thus, measurement units are the same for all variables.

It is also worth noting the importance of correctly interpretting the coefficients of $\hat{\boldsymbol{\beta}}$ (see page 5) . In the proposed framework, we treat each unique lineup $N_i, i \in \{1, 2, 3, ..., 30\}$ as a variable in an 82 observation matrix. By taking matrix $\boldsymbol{X}$ as team's usage, each observation $(x_{j1}, x_{j2}, \ldots, x_{jp})$ corresponds to the total of seconds each lineup played during a game. We then build our 82×1 vector Y which contains the margin of victory or defeat for each game in a season. By adjusting a penalized regression model for each team, the entries of the obtained $\hat{\boldsymbol{\beta}}$ vector have a tangible interpretation in basketball terms. In our framework, $\hat{\boldsymbol{\beta}}$ is the the vector of associated plus–minus statistics for each 5-player lineup. As such, each entry represents an estimated efficiency metric for a given lineup throughout each game in the season.

Table 3.1

Example of $\boldsymbol{X}$ matrix.

	Lineup 1	Lineup 2	Lineup 3	...	Lineup N_i
Playing time in game 1	864	520	678	...	60
Playing time in game 2	789	985	321	...	205
Playing time in game 3	198	896	123	...	85
...	...	...	...	...	...
Playing time in game 82	987	159	656	...	89

Because when adjusting a ridge regression, we have

$$\hat{\boldsymbol{\beta}}^{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda I)^{-1}\boldsymbol{X}^T Y,$$

where $\lambda \geq 0$ is a parameter which controls the severity of penalization on the coefficients. Recall that in our case, $\boldsymbol{X} \in \mathbb{R}^{82\times N_i}$ and $Y \in \mathbb{R}^{82\times 1}$ thus, our output $\hat{\boldsymbol{\beta}}^{ridge} \in \mathbb{R}^{N_i\times 1}$. Which contains an associated plus–minus statistic for each one of our N_i lineups.

Table 3.2

Example of Y vector.

	Margin of victory or defeat
Game 1	-15
Game 2	-8
Game 3	6
...	...
Game 81	9
Game 82	-10

Furthermore, notice how each entry in our ridge solution $\hat{Y}$ will be an estimation of the final result of a game. Since $\boldsymbol{X}\hat{\boldsymbol{\beta}}^{ridge}$ is multiplying the amount of time each lineup played during a game by

its associated estimated plus–minus statistic. Recall from definition 1.2.1 in the preliminaries section that summing across all the associated plus–minus statistics in a game yields the total amount of points that a team scored in that game. As such, summing across the product (playing time for lineup j)(estimated plus–minus for lineup j) for all lineups that a team used in a game, yields an estimation for the total amount of points a team will score in that specific game.

3.2.1 Regularization Methods

Regularized Regression is a variation of ordinary linear regression where coefficient estimates $\hat{\boldsymbol{\beta}}$ are constraint. The magnitude of the coefficients as well as the error term are penalized by a tuning parameter ($\lambda \geq 0$). This "forces" the coefficients to stay close to zero and avoid big magnitudes. This results useful for mitigating high variance in the coefficients and the presence of multicollinearity within a set of features. When there are many correlated variables, a large positive coefficient on one variable can be canceled by another high but negative coefficient on its correlated counterpart. By imposing magnitude constraints on $\hat{\boldsymbol{\beta}}$ this caveat is alleviated.

Recall that for ordinary linear regression models, we seek

$$\hat{\boldsymbol{\beta}} = argmin||Y - \boldsymbol{X}\boldsymbol{\beta}||_2^2.$$

But for regularized regression models, we add penalty to the coefficients and thus our estimator becomes:

$$\hat{\boldsymbol{\beta}}^{reg} = argmin||Y - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_q. \tag{3.1}$$

Notice how when $\lambda = 0$, a regularized regression is equivalent to linear regression. On the other hand, $\lambda \to \infty \Rightarrow \hat{\boldsymbol{\beta}}^{reg} = 0$ yields that large λ values have a higher shrinking effect on the coefficients of $\hat{\boldsymbol{\beta}}^{reg}$.

Within these models there are two commonly used types of regression methods namely ridge and lasso regressions. The main difference between these two is that ridge regression uses the eucledian norm on the penalty term ($q = 2$ on 3.1) while lasso implements an L_1 regularization on the penalty ($q = 1$ on 3.1) .

3.2.1.1 Ridge Regression

Ridge regression is a way to create parsimonious models when the number of predictor variables in the dataset exceeds the number of observations. Its use is also recommended when our predictor variables have a high presence of multicollinearity. This is to say that ridge regression provides a way to subset variables in a model in order to reduce the variance. We can rewrite the penalized RSS for ridge regression in matrix form as

$$RSS(\lambda) = (Y - \boldsymbol{X}\boldsymbol{\beta})^T(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta},$$

and one can obtain:

$$\hat{\boldsymbol{\beta}}^{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda I)^{-1}\boldsymbol{X}^T Y. \tag{3.2}$$

Notice that because the penalty is defined as $\boldsymbol{\beta}^T\boldsymbol{\beta}$, the solution is still a linear function of Y.

3.2.1.2 Lasso Regression

Lasso regression is another shrinkage method like ridge, but instead of using an L_2 penalty, we use an L_1 penalty on the coefficients. In this regression, coefficients values are shrunk towards a center point, most commonly, the mean. This results particularly useful to mitigate the presence of multicollinearity and to automate variable selection in a

model. This represents the main advantage over ridge models. Lasso regression subsets groups of highly correlated variables and selects one of them. This mitigates multicollinearity problems but promotes sparsity within our models. This means that lasso regression promotes the reduction in dimension of the parameter vector by setting coefficients of the eliminated variables to zero.

The lasso estimate is defined as:

$$\hat{\boldsymbol{\beta}}^{lasso} = argmin_{\boldsymbol{\beta}} \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2, \text{ subject to: } \sum_{j=1}^{p} |\beta_j| \leq t. \tag{3.3}$$

Equation (3.3) can also be written as

$$1/2 \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

This makes evident that as opposed to ridge regression, the solutions will be nonlinear for y_i and thus, we can not have an explicit form for $\hat{\boldsymbol{\beta}}^{lasso}$. Because of this, computing the lasso solutions becomes a quadratic programming problem. However, there are efficient algorithms for computing solutions as λ is varied with the same computational cost as for ridge regression. (Friedman, Tibshirani, Hatstie)[2001].

3.2.1.3 Elastic net regression

Elastic net regression was introduced as a combination of both lasso and ridge methods. Proposed by Zou and Hastie (2004), the elastic net introduces a penalty term that is a linear combination of both lasso (L_1)

and ridge (L_2) penalties

$$\lambda \sum_{j=1}^{p} (\alpha \beta_j^2 + (1 - \alpha)|\beta_j|). \tag{3.4}$$

The paramater α in 3.4 determines the mix of the penalties and is often chosen based on cross validation grid searches. Notice $\alpha = 1$ and $\alpha = 0$, are equivalent to ridge and lasso respectively.

Elastic net subsets and selects variables from highly correlated groups like lasso, but also shrinks together coefficients of correlated variables like ridge regression. Furthermore, when lasso subsets groups of highly correlated variables, it only selects one of them without any care. The authors claim elastic net alleviates this problem and gave the model its name because it is similar to a stretchable fishing net that retains "all the big fish". Even though lasso proofs useful in many cases, Zou and Hastie point out it has some limitations:

- When $p > n$ (the number of variables exceeds the number of observations), lasso subsets at most n variables because of the convex quadratic programming problem. Moreover, lasso solutions are not well defined unless the bound on our coefficients is smaller than certain fixed value.

- If there is a group of highly correlated variables in our dataset, lasso selects only one feature from the group without caring which one.

- For cases where $n > p$, if there are high correlations between predictors, it has been empirically observed that ridge outperforms lasso(Tibshirani, 1996).

3.2.2 The need for penalized regression

As we saw in section 1.1.2, we are looking for $\hat{\boldsymbol{\beta}}$ which minimizes the function RSS in equation (1.3). As established in equation (1.4), solving the normal equations yields that such solution exists if and only if $\boldsymbol{X}^T\boldsymbol{X}$ is a nonsingular matrix. Thus, OLS regression yields maximum likelihood estimators derived from our observed data and relies heavily on the independence of the features. Recall that a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is invertible $\leftrightarrow$ $rank(\boldsymbol{A}) = n$. Thus, when the columns of the $\boldsymbol{X}$ matrix have a high linear dependance, it comes close to becoming singular(non-invertible) because $rank(\boldsymbol{X})$(The maximal number of linearly independent columns of $\boldsymbol{X}$) diminishes.

In equation (3.2) we can clearly see how penalized regression methods address this problem by adding a positive constant to the diagonal of $\boldsymbol{X}^T\boldsymbol{X}$. This makes the problem nonsingular, even if $\boldsymbol{X}^T\boldsymbol{X}$ is not of full rank and contains correlated variables.

In our data, the input matrices $\boldsymbol{X}$ could present multicollinearity issues because playing times for lineups which contain same players may be highly correlated. For example, lineups with starting players like LeBron James may present higher playing times than lineups with non starting forwards like James Jones. Moreover, for all of our models $\boldsymbol{X}$ is of size $82 \times N_i$, where $N_i > 82$ as shown on Table 3.3.

As established in the former section, the solution to these problems is to implement shrinkage methods such as ridge, lasso and elastic net regression. Our prior suspicion is that because for each team, $p >> n$, elastic net models outperform ridge and lasso. Another advantage of implementing regularized regression is the constraint of the coefficients. Because plus–minus statistics are usually not overly high quantities this constraints suit our model particularly well.

Table 3.3
5-man units used by each team during the 2014-15 regular season.

Team	Unique lineups	Team	Unique lineups
ATL	588	MIA	713
BKN	529	MIL	649
BOS	846	MIN	678
CHA	529	NOP	582
CHI	369	NYK	1008
CLE	562	OKC	629
DAL	758	ORL	712
DEN	859	PHI	1042
DET	482	PHX	557
GSW	431	POR	664
HOU	550	SAC	618
IND	556	SAS	747
LAC	518	TOR	327
LAL	476	UTA	603
MEM	564	WAS	528

Source: Data obtained through nbastatR library.

3.3 Data Collection

Even though there is a vast amount of public basketball data available from sources like `stathead.com`, `ESPN.com` and `NBA.com`, most of the information that these sites provide focus on individual and team-level statistics but there is a lack of information regarding the performance of 5-player units. The only site which provides performance statistics on 5-player units is `NBA.com`. Nevertheless, these statistics are provided

only on a season long basis and lineups that played a small number of minutes were excluded. To ammend this lack of data and retrieve the desired information, we had to mine and process NBA play-by-play data.

The only public source of play-by-play data that we found was the `nbastatR` package. This package provides multiple wrapper functions for calling APIs to scrape websites such as: HoopsHype, Basketball Insiders, Basketball-Reference, among others.

3.3.1 Extracting lineups from play-by-play data

The dataset for the 2014-2015 season consisted of $562,310$ observations of 40 variables. Each row of the dataset corresponds to an in-game event for which a variety of information is recorded. A particularly useful variable was a *Description* column which contained a text description of what happened in each event. Table A.3 (see Appendix A) shows how some columns of the raw data looked before any processing.

First, we created a column which computes the score margin before and after each play for the home and away team while keeping the quarter, time remaining in the quarter and on court lineups. To specify the score in each play, we use the play description provided and created a column which accounted for the amount of points each team gathered in every event.

Next step comprises finding out the lineups for every team at every moment of each game in order to calculate each unique lineup's plus–minus. This poses a challenge as the data shows every substitution that happens in a game, but it does not show the lineups at the beginning of each quarter. So, if a player is substituted in the resting period between each quarter, this substitution will not appear in the data. To fix this, we figure any players who participated in an event which is described in

our data as them going out of the game but never registering the event of checking back into the game. Another possible scenario which may impact the results are those who play an entire quarter without being substituted. To find out these cases, we filter players who participated in an event of any type in a quarter but none of these events was a substitution.

Lastly, those players who started a quarter but did not participate in any recorded in-game event might cause an issue. There were 12 of these cases and all of them were in overtime or double overtime. This makes sense because these periods are noticeably shorter than a regular quarter. These exceptions had to be manually added to the dataset. Table 3.4 shows all the cases in which players started in a quarter but did not participate in relevant in-game events.

Table 3.4
Players who started in a quarter but did not participate in relevant in-game events during the 2014-2015 season.

Team	Player	Period Number	Location
MIL	Jared Dudley	5	Away
CHI	Mike Dunleavy	6	Home
WAS	Bradley Beal	5	Home
POR	Steve Blake	5	Away
MEM	Courtney Lee	5	Home
POR	Nicolas Batum	5	Away
PHX	Brandon Knight	5	Home
CLE	Matthew Dellavedova	5	Away
HOU	Trevor Ariza	5	Home
PHX	P.J. Tucker	5	Away
NOP	Dante Cunningham	4	Away
BKN	Bojan Bogdanovic	6	Home

Source: Data obtained through nbastatR library.

Now all possible substitutions taking place in a game are available, the next step is to create the columns providing the lineups before and after each substitution. In this regard, we use our data and filter all the substitutions made during a quarter. For all of these events, two columns in the dataset provided information on which player was substituted out and which one was substituted in. So the created dataframe provides the lineups before each substitution, the player going in, the player coming out and the exact moment in the game in which it ocurred. Table A.1 (see Appendix A) shows how our dataset

looked at this point.

Afterwards, one can replace the player coming out with the player coming in and create a new column which contains the lineup after the substitution was made. Finally, we join this final substitution dataframe with our original play-by-play data and thus, we have the lineups before and after each substitution took place. An extra step is required because we need the lineup before and lineup after columns for every event in the dataset, not only for the substitutions. Therefore, the lineup for every event that comes between a substitution will be the last lineup after a substitution type of event took place.

At last, we have the on court lineups for the home and away team for every event in the 2014-2015 NBA season. Table A.2 (see Appendix A) shows how the data looks after processing.

3.3.2 Extracting playing time and plus–minus from play-by-play data

Once we have the on court lineups for each event in the season, we need to extract the plus–minus metric for every 5-player unit and its playing time. Hence, the number of points scored by a team and by its opponent during a lineup stint are required to be computed. Therefore, one needs to extract the score every time a player is substituted into the game. This can be complicated because all the cases in which a scoring oportunity is also a substitution opportunity need to be taken into consideration. In these cases, the methodology that the NBA utilizes to calculate plus–minus was followed. This means that the points scored are assigned to the lineup that is on court at that moment regardless of who promoted the scoring opportunity for any team. Table 3.5 shows how our data looks after

processing.

Table 3.5

Sample table of how our data looked after computing playing time and net points for the first Golden State Warriors game.

Team	Lineup	Lineup Stint	Initial Score Team	Initial Score Opponent	Final Score Team	Final Score Opponent	Initial Time (seconds)	Final Time (seconds)	Total Score Team	Total Score Opponent	Net Score Team	Total Time
GSW	Andrew Bogut, Draymond Green, Harrison Barnes, Klay Thompson, Stephen Curry	0	0	0	12	9	0	360	12	9	3	360
SAC	Ben McLemore, Darren Collison, DeMarcus Cousins, Jason Thompson, Rudy Gay	0	0	0	12	13	0	446	12	13	-1	446
GSW	Draymond Green, Festus Ezeli, Harrison Barnes, Klay Thompson, Stephen Curry	1	12	9	13	11	360	404	1	2	-1	44
SAC	Ben McLemore, Darren Collison, DeMarcus Cousins, Nik Stauskas, Rudy Gay	1	12	13	12	13	446	446	0	0	0	0
GSW	Andre Iguodala, Draymond Green, Festus Ezeli, Klay Thompson, Stephen Curry	2	13	11	13	12	404	446	0	1	-1	42
SAC	Carl Landry, Darren Collison, DeMarcus Cousins, Nik Stauskas, Rudy Gay	2	12	13	14	15	446	502	2	2	0	56
GSW	Andre Iguodala, Festus Ezeli, Klay Thompson, Marreese Speights, Stephen Curry	3	13	12	19	16	446	587	6	4	2	141
SAC	Carl Landry, Darren Collison, DeMarcus Cousins, Derrick Williams, Nik Stauskas	3	14	15	16	19	502	569	2	4	-2	67
GSW	Andre Iguodala, Festus Ezeli, Klay Thompson, Leandro Barbosa, Marreese Speights	4	19	16	23	16	587	632	4	0	4	45
SAC	Carl Landry, Darren Collison, Derrick Williams, Nik Stauskas, Ramon Sessions	4	16	19	16	19	569	569	0	0	0	0
GSW	Andre Iguodala, Festus Ezeli, Justin Holiday, Leandro Barbosa, Marreese Speights	5	23	16	23	18	632	656	0	2	-2	24
SAC	Carl Landry, Derrick Williams, Nik Stauskas, Ramon Sessions, Reggie Evans	5	16	19	29	29	569	937	13	10	3	368
GSW	Andre Iguodala, Jusn Holiday, Leandro Barbosa, Marreese Speights, Ognjen Kuzmic	6	23	18	25	23	656	720	2	5	-3	64
SAC	DeMarcus Cousins, Derrick Williams, Nik Stauskas, Ramon Sessions, Reggie Evans	6	29	29	31	31	937	983	2	2	0	46
GSW	Andre Iguodala, Andrew Bogut, Justin Holiday, Leandro Barbosa, Marreese Speights	7	25	23	27	29	720	899	2	6	-4	179
SAC	DeMarcus Cousins, Nik Stauskas, Ramon Sessions, Reggie Evans, Rudy Gay	7	31	31	31	31	983	983	0	0	0	0
GSW	Andre Iguodala, Andrew Bogut, Justin Holiday, Leandro Barbosa, Stephen Curry	8	27	29	27	29	899	899	0	0	0	0
SAC	DeMarcus Cousins, Jason Thompson, Nik Stauskas, Ramon Sessions, Rudy Gay	8	31	31	31	36	983	1039	0	5	-5	56
GSW	Andre Iguodala, Andrew Bogut, Justin Holiday, Klay Thompson, Stephen Curry	9	27	29	27	29	899	899	0	0	0	0
SAC	Darren Collison, DeMarcus Cousins, Jason Thompson, Ramon Sessions, Rudy Gay	9	31	36	31	36	1039	1039	0	0	0	0
GSW	Andre Iguodala, Andrew Bogut, Draymond Green, Klay Thompson, Stephen Curry	10	27	29	34	31	899	1009	7	2	5	110
SAC	Ben McLemore, Darren Collison, DeMarcus Cousins, Jason Thompson, Rudy Gay	10	31	36	35	37	1039	1160	4	1	3	121

Source: Data obtained through nbastatR library.

To make sure our computed playing times were correct, we scraped playing time for each lineup from *basketballreference.com* and compared to our computed playing times. Table 3.6 displays the difference (in

minutes) for the top 18 lineups with more recorded minutes in a season.

Table 3.6

Difference (in minutes) between our computed playing times and the scraped data from Basketball Reference for the 18 lineups with more recorded on-court playing time.

Rank	Lineup	Team	Computed Playing Time	Scraped Playing Time	Difference
1	Blake Griffin, Chris Paul, DeAndre Jordan, JJ Redick, Matt Barnes	LAC	1217.65	1213.60	4.05
2	Al Horford, DeMarre Carroll, Jeff Teague, Kyle Korver, Paul Millsap	ATL	915.57	913.60	1.97
3	Andrew Bogut, Draymond Green, Harrison Barnes, Klay Thompson, Stephen Curry	GSW	813.25	810.50	2.75
4	Damian Lillard, LaMarcus Aldridge, Nicolas Batum, Robin Lopez, Wesley Matthews	POR	629.63	628.40	1.23
5	Bradley Beal, John Wall, Marcin Gortat, Nene, Paul Pierce	WAS	596.78	594.80	1.98
6	JR Smith, Kevin Love, Kyrie Irving, LeBron James, Timofey Mozgov	CLE	480.83	479.20	1.63
7	Courtney Lee, Marc Gasol, Mike Conley, Tony Allen, Zach Randolph	MEM	460.55	459.20	1.35
8	Chandler Parsons, Dirk Nowitzki, Monta Ellis, Rajon Rondo, Tyson Chandler	DAL	455.08	453.60	1.48
9	Ben McLemore, Darren Collison, DeMarcus Cousins, Jason Thompson, Rudy Gay	SAC	427.45	425.60	1.85
10	Courtney Lee, Jeff Green, Marc Gasol, Mike Conley, Zach Randolph	MEM	426.28	424.90	1.38
11	Arron Afflalo, Kenneth Faried, Timofey Mozgov, Ty Lawson, Wilson Chandler	DEN	415.75	414.90	0.85
12	Alex Len, Eric Bledsoe, Goran Dragic, Markieff Morris, P.J. Tucker	PHX	391.92	391.20	0.72
13	Amir Johnson, DeMar DeRozan, Jonas Valanciunas, Kyle Lowry, Terrence Ross	TOR	390.08	388.60	1.48
14	CJ Miles, David West, George Hill, Roy Hibbert, Solomon Hill	IND	374.45	373.80	0.65
15	Donatas Motiejunas, Dwight Howard, James Harden, Patrick Beverley, Trevor Ariza	HOU	360.73	360.10	0.63
16	Derrick Rose, Jimmy Butler, Joakim Noah, Mike Dunleavy, Pau Gasol	CHI	353.50	353.10	0.40
17	Anthony Davis, Eric Gordon, Omer Asik, Quincy Pondexter, Tyreke Evans	NOP	346.88	345.60	1.28
18	Ersan Ilyasova, Giannis Antetokounmpo, Khris Middleton, Michael Carter-Williams, Zaza Pachulia	MIL	344.03	343.20	0.83

Source: Data obtained through nbastatR library and scraped from *basketballreference.com*.

Seeing that our calculations were validated by the data from Basketball Reference, we used this data for our models.

Recall from section 3.2 that the (i, j) entrance in our covariate matrix $\boldsymbol{X}$ corresponds to the seconds played in game i by lineup j. For building such matrices, we employ two main functions. The first one constructs the $\boldsymbol{X}$ matrix and takes a team slug as an argument. The second one returns an 82×1 vector with the margin of defeat or victory for the given team.

To construct matrix $\boldsymbol{X}$, data are filtered for the provided team from our lineup information data frame. Next, we group by lineup and game id while summing the total playing time and net associated plus–minus.

This output is a chronological table for each unique lineup that a team used per game. Table 3.7 depicts the data for the Dallas Mavericks.

Table 3.7
Top rows for the generated chronological per game lineup usage series for the Dallas Mavericks.

Game Id	Lineup	Time Played (Seconds)	Game Plus Minus	Team	Lineup Id
21400002	Al-Farouq Aminu, Brandan Wright, Devin Harris, Dirk Nowitzki, Jae Crowder	329	0	DAL	3527
21400002	Al-Farouq Aminu, Brandan Wright, Devin Harris, Dirk Nowitzki, Jameer Nelson	0	0	DAL	2139
21400002	Al-Farouq Aminu, Brandan Wright, Dirk Nowitzki, Jae Crowder, Jameer Nelson	0	0	DAL	5872
21400002	Brandan Wright, Chandler Parsons, Devin Harris, Dirk Nowitzki, Monta Ellis	69	1	DAL	2391
21400002	Brandan Wright, Chandler Parsons, Devin Harris, Jameer Nelson, Monta Ellis	209	0	DAL	376
21400002	Brandan Wright, Chandler Parsons, Dirk Nowitzki, Jameer Nelson, Monta Ellis	128	-3	DAL	1849
21400002	Brandan Wright, Chandler Parsons, Jameer Nelson, Monta Ellis, Richard Jefferson	145	4	DAL	13758
21400002	Brandan Wright, Devin Harris, Dirk Nowitzki, Jameer Nelson, Monta Ellis	0	0	DAL	8393
21400002	Brandan Wright, Devin Harris, Jae Crowder, Monta Ellis, Richard Jefferson	143	-2	DAL	15873
21400002	Brandan Wright, Devin Harris, Jameer Nelson, Monta Ellis, Richard Jefferson	0	0	DAL	8400
21400002	Brandan Wright, Dirk Nowitzki, Jae Crowder, Jameer Nelson, Richard Jefferson	0	0	DAL	8402
21400002	Brandan Wright, Dirk Nowitzki, Jameer Nelson, Monta Ellis, Richard Jefferson	61	6	DAL	328
21400002	Chandler Parsons, Devin Harris, Dirk Nowitzki, Monta Ellis, Tyson Chandler	331	3	DAL	83
21400002	Chandler Parsons, Devin Harris, Greg Smith, Jameer Nelson, Monta Ellis	15	-2	DAL	15962
21400002	Chandler Parsons, Devin Harris, Jameer Nelson, Monta Ellis, Tyson Chandler	333	3	DAL	18193
21400002	Chandler Parsons, Dirk Nowitzki, Greg Smith, Jameer Nelson, Monta Ellis	116	3	DAL	5143
21400002	Chandler Parsons, Dirk Nowitzki, Jameer Nelson, Monta Ellis, Richard Jefferson	0	0	DAL	8990
21400002	Chandler Parsons, Dirk Nowitzki, Jameer Nelson, Monta Ellis, Tyson Chandler	548	-7	DAL	20
21400002	Chandler Parsons, Jameer Nelson, Monta Ellis, Richard Jefferson, Tyson Chandler	135	-2	DAL	2438
21400002	Devin Harris, Dirk Nowitzki, Jae Crowder, Richard Jefferson, Tyson Chandler	180	-6	DAL	17211

Source: Data obtained through nbastatR library.

Where the column *Lineup Id* provides a unique lineup identifying number for every single lineup present in our data. Next, we iterate over each unique lineup and game id to arrange our data into the desired $82 \times N_i$ form.

Extracting the associated Y vector was simpler as we already had the score before and after each in-game event. So the function just filtered the desired team and thus, by grouping together the lineups, we have the points scored and the points received in each game for the

desired team. A simple difference between this quantities results in a 82×1 vector where we have the margin of victory or defeat by which a team won or lost each game in a season.

3.4 Models

To measure the goodness of fit for the proposed models, the ridge, lasso and elastic net regressions for each team using the entire 82 game NBA season are used here.

For lasso and ridge regression we use 10-fold cross validation to search for 25 possible values of λ between $10,000$ and 0.1. For this first run, the data was standardized for both fits. Ridge and lasso regressions yield poor results as the average R^2 for in-sample predictions was 0.088 and 0.0906 for ridge and lasso regression respectively.

Because all of our observations in the $\boldsymbol{X}$ matrix are playing time in seconds for each different lineup, standarization was redundant as all data has the same unit of measurement. Nevertheless, rerunning both ridge and lasso regressions without standardization caused the models to overfit the data.

Using the regular season as training data and the playoffs as out of sample data, in the second experiment average R^2 for in–sample predictions raises to an alarming 0.975365 for ridge regression. Nevertheless, when predicting out of sample data, performance falls drastically. Average λ values were close to 6000 which was an indication that the model was overshrinking the coefficients.

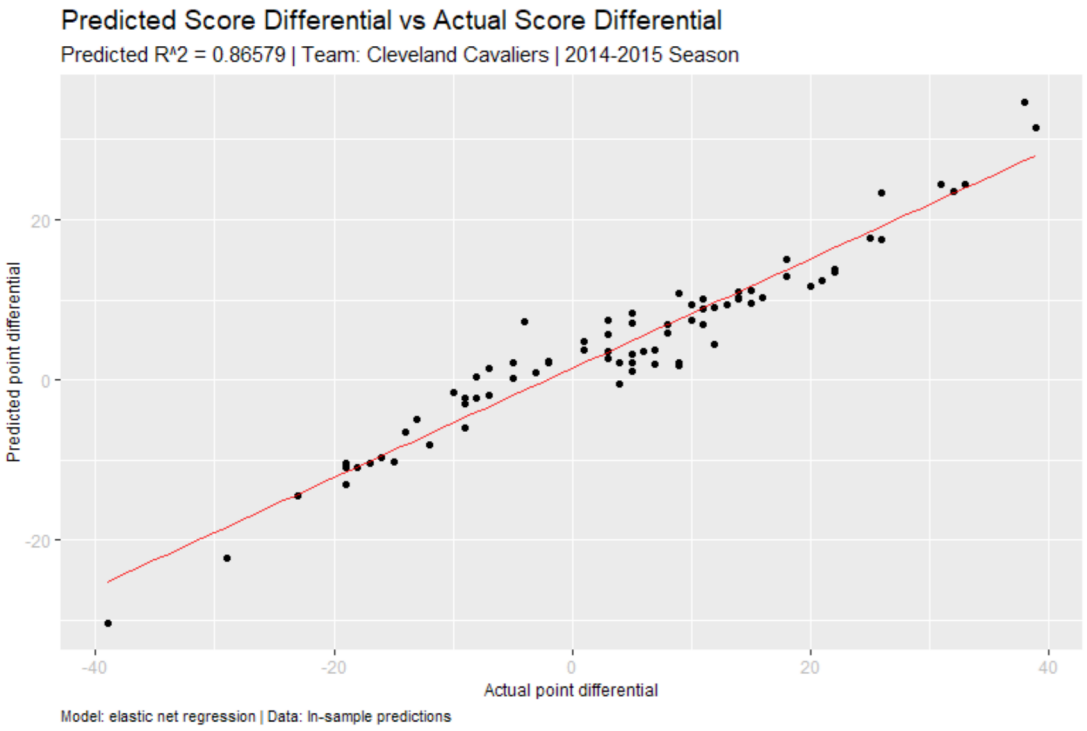
In fact, Lasso regression without standardization also caused the model to overfit. 14/30 models had an in-sample $R^2 \approx 0.99$ while the rest of the models reported $\lambda = 10,000$ and $R^2 = 0$. This confirmed

that both lasso and ridge regression models were poor fits for our data.

For elastic net we apply a 10-fold cross validation with tuneLength parameter equal to 10 to search for the optimal λ. This meant that we specified a grid of 10 equally spaced values of α that the model will use when training. So the possible values of alpha were $(0.1, 0.2, 0.3 \ldots, 0.9)$ and for each one of this values, we searched for 10 values of λ. Figure 3.2 shows the difference between estimated point differential and actual point differential for the Cleveland Cavaliers.

Figure 3.2

Predicted point differential (y) vs Actual point differential (x) for the Cleveland Cavaliers.



In-sample predictions yielded an average R^2 of 0.77835 which is a massive improvement over ridge and lasso models as this does not raise suspicions of over or under fitting like in the previous models. Table 3.3 shows R^2 coefficient for in-sample predictions for every team in the league.

Figure 3.3

Results for in-sample predictions using an elastic net model for each team.

Results for in-sample predictions using an elastic net model

Season: 2014-2015

Team	Alpha	Lambda	R^2
ATL	0.5173496	1.948287938	0.8083240
BKN	0.1179912	3.259495068	0.9456167
BOS	0.5173496	1.948287938	0.7333578
CHA	0.5173496	1.948287938	0.8014265
CHI	0.5173496	1.948287938	0.6691915
CLE	0.8438814	1.120811204	0.8657943
DAL	0.5173496	1.948287938	0.8102889
DEN	0.8438814	1.120811204	0.8392448
DET	0.8438814	1.120811204	0.7702022
GSW	0.5173496	1.948287938	0.6790333
HOU	0.8438814	1.120811204	0.7821134
IND	0.5173496	1.948287938	0.6818513
LAC	0.5173496	1.948287938	0.7534230
LAL	0.1179912	3.259495068	0.8814649
MEM	0.8438814	1.120811204	0.7781305
MIA	0.5173496	1.948287938	0.7088083
MIL	0.1179912	3.259495068	0.9180807
MIN	0.5173496	1.948287938	0.7188933
NOP	0.5173496	1.948287938	0.7642576
NYK	0.2604857	0.001435276	0.9990614
OKC	0.5173496	1.948287938	0.7706275
ORL	0.5173496	1.948287938	0.6688915
PHI	0.5173496	1.948287938	0.7842544
PHX	0.5173496	1.948287938	0.7067130
POR	0.5173496	1.948287938	0.6951917
SAC	0.5173496	1.948287938	0.7624281
SAS	0.5173496	1.948287938	0.7435679
TOR	0.5173496	1.948287938	0.7302092
UTA	0.5173496	1.948287938	0.7444106
WAS	0.8438814	1.120811204	0.8359365

Data: nbastatR

3.4.1 Assesing elastic net's predictive value

To assess whether our model's prediction accuracy is meaningful, two experiments were proposed:

1. Refit the models for each team using 58 games in a season as a train set and 24 games as test set. Although a more logical approach would be to use the playoffs as a test set for out of sample predictions, the amount of data would not be enough to make quality forecasts. Only 16 teams advance to the playoffs in the NBA and some of them play less than 5 games before they are eliminated. This is why we use regular season data divided into train and test sets.

2. Because few teams play significant number of games in the playoffs. An alternative would be to fit a model for the Cleveland Cavaliers and the Golden State Warriors (both of which advanced to the NBA Finals) using their regular season data as a training set and their playoffs games as a test set of data. In this case, both teams played more than twenty games and thus, we have enough data points to attempt a quality prediction.

In the first scenario, although λ values were acceptable, the model severly overfitted the data. This yielded poor results when predicting scoring margins for the last 24 games of the season (out of sample data).

For the second scenario, the models adjusted in the previous section were used as in-sample fits and play-by-play data for the NBA playoffs was mined and processed to use as out-of-sample data. Even though in-sample results were good, the coefficient of determination for out of sample data dropped drastically for both models. Nevertheless, assessing the model´s predictive value just by analyzing

performance metrics is insufficient as we are interested in evaluating a binary result which is whether a team won or lost a game. A fairer way to value the model is to examine its ability to correctly predict the outcome of the game. To do so, we compare the predicted margin of victory or defeat for each game with the actual outcome of the game and see whether or not our model correctly chooses the winner.

For the Golden State Warriors, our model correctly predicts the outcome for 15 out of 21 playoff games. Table 3.8 shows the results of our simulations.

Table 3.8

Comparison between predicted and actual outcomes for each of the Golden State Warriors' playoff games. The model correctly predicts the winner in 72.72 % of the games.

Game	Actual margin	Predicted margin	Correct prediction
vs New Orleans	7	8.622060566	Yes
vs New Orleans	10	10.47796558	Yes
at New Orleans	4	10.47796558	Yes
at New Orleans	11	9.410261189	Yes
vs Memphis	15	10.47796558	Yes
vs Memphis	-7	8.812123127	No
at Memphis	-10	6.749385325	No
at Memphis	17	11.10902905	Yes
vs Memphis	10	7.101823977	Yes
at Memphis	13	8.241935442	Yes
vs Houston	4	8.126382391	Yes
vs Houston	1	9.138531628	Yes
at Houston	35	9.924548121	Yes
at Houston	-13	9.34877036	No
vs Houston	14	10.47796558	Yes
vs Cleveland	8	8.305141458	Yes
vs Cleveland	-2	4.904660462	No
at Cleveland	-5	7.613403375	No
at Cleveland	21	3.581695032	Yes
vs Cleveland	13	3.479191249	Yes
at Cleveland	8	4.587803026	Yes

Source: Data obtained through nbastatR library.

The experiment for the Cleveland Cavaliers yielded an overall accuracy of 65%, correctly predicting 13 out of 20 playoff games.

Table 3.9

Predictions for each of the Cleveland Cavaliers' playoff games. The model accurately predicts the outcome of a game with 65% of accuracy.

Game	Actual margin	Predicted margin	Correct outcome
vs Boston	13	5.8206226	Yes
vs Boston	8	6.270888165	Yes
at Boston	8	9.682654329	Yes
at Boston	8	2.991039265	Yes
vs Chicago	-7	2.149204098	No
vs Chicago	15	5.070787862	Yes
at Chicago	-3	1.693397524	No
at Chicago	2	2.224014741	Yes
vs Chicago	5	2.929070301	Yes
at Chicago	21	11.67080648	Yes
at Atltanta	8	6.953472637	Yes
at Atlanta	12	8.572415049	Yes
vs Atlanta	3	13.76270976	Yes
vs Atlanta	30	-2.643558128	No
at Golden State	-8	1.072757248	No
at Golden State	2	7.966916267	Yes
vs Golden State	5	11.46748196	Yes
vs Golden State	-21	8.472377492	No
at Golden State	-13	12.69705871	No
vs Golden State	-8	6.268008001	No

Source: Data obtained through nbastatR library.

Because of the model's good results in predicting game outcomes, we further evalute the model repeating the experiment for two teams for which we have sufficient data in the playoffs: the Atlanta Hawks and the Houston Rockets. For the latter, the model presented severe overfitting and thus, no further evaluation could be conducted. In Atlanta's case, the model correctly predicts 43.75% of the outcomes.

Table 3.10 summarizes the model's predicted results for the Atlanta Hawks.

Table 3.10

Predictions for the Atlanta Hawks' playoff games. Accuracy for this experiment is 43.75% as it correctly predicts 7 out of 16 outcomes.

Game	Actual margin	Predicted margin	Correct outcome
vs Brooklyn	7	-7.490000227	No
vs Brooklyn	5	8.926737221	Yes
at Brooklyn	-8	5.696965376	No
at Brooklyn	-5	5.696965376	No
vs Brooklyn	10	5.696965376	Yes
at Brooklyn	24	-1.73122041	No
vs Washington	-6	4.911586972	No
vs Washington	16	5.513988041	Yes
at Washington	-2	5.424782679	No
at Washington	5	5.958491235	Yes
vs Washington	1	6.321721596	Yes
at Washington	3	6.076800553	Yes
vs Cleveland	-8	5.696965376	No
vs Cleveland	-12	5.979131807	No
at Cleveland	-3	0.351158676	No
at Cleveland	-30	-15.75666852	Yes

Source: Data obtained through nbastatR library.

It is important to note that lineup usage in the NBA playoffs drastically defers from the regular season. Starting players significantly increase their playing times and bench players are less active. This may be a reason why the coefficient of determination drops so drastically for out of sample predictions and also the cause for overfitting in Houston's case.

3.5 Proposed metric: Contributed Wins Over Replacement

Since elastic net regression provided promising results for in-sample predictions and yielded adequate fits to our data, we further explore possible uses evaluating *a posteriori* results with the model. Applications arise, for example, in player workload management and trade evaluation areas.

Our proposed methodology produces a retrospective measure of individual player contributions to the team's total wins in a season. Nevertheless, our estimates of player contribution are lineup-dependent, making them unsuitable for forecasting future performance since the lineups with which a player performs can vary each season or even each week. The employed methodology can provide insight into whether coaches are efficiently distributing playing time for different players and help understand the extent to which a player's individual performance translates to wins for his team. It can also be used by general managers to determine which players positively impact the team and which ones can be released or traded.

With only a simple modification to our models, we can create projections for how a team's season would have looked without certain player(s) in the organization. This can provide insights as to how many more games would have been won or lost if said player was not in the lineups during the past season. We call this *Contributed Wins Over Replacement*(CWOR), defined as follows

$CWOR =$ Wins with player in lineup $-$ Wins without player in lineup.

Because our model fits using an input matrix in which the number of rows is the unique amount of lineups a team used during the season,

by removing the lineups in which a certain player appears, we redistribute team performance credit (measured by plus–minus) among the remaining lineups. Thus, fitting the model using the remaining data results in a reduced matrix for which we can still fit a regression model.

To exemplify how CWOR can be used, we analyze two of the most relevant trades that ocurred during the 2015-2016 offseason.

3.5.1 Ersan Ilyasova trade

On June 11, 2015 the Milwaukee Bucks traded their starting power forward Ersan Ilyasova to the Detroit Pistons in exchange for forwards Caron Butler and Shawne Williams. During the $2014 - 2015$ season, the Bucks won 41 games. When fitted on regular season data with Ilyasova in the lineup, our model predicts the Bucks win 45 games with $R^2 = 0.918$. Now, if we adjust the covariates matrix $\boldsymbol{X}$ and remove Ersan Ilyasova from the roster and refit the model, the Bucks now win 42 games with $R^2 = 0.66$. This means that Ilyasova's CWOR during the 2014-2015 season was +3.

For context, that year the 8th seed in the playoffs (Brooklyn Nets) won 38 games. Theoretically, this means that the Bucks could trade Ilyasova and still achieve the required number of victories to secure a playoff spot. It has to be aknowledged that one of the main assumptions for this framework is that we do not account for the added value that new players may bring to the organization. Nevertheless, in this particular case, our assumptions hold because 15 days after the trade, both players were waved by Milwaukee. This means that Butler and Williams did not play a single game for the Milwaukee Bucks after being signed.

Aside from in game performance, we must consider that there are external factors that affect a team's decision making process when

trading players. For example, payroll management. After the 2014-2015 season, Ilyasova entered the final year of a 5 year, 40 million dollar contract. This means that (on average) the Bucks would have to pay 8 million dollars for those three additional projected wins.

Table A.4 shows a detailed breakdown of the projected season results with and without Ilyasova in the lineup.

3.5.2 Tiago Splitter trade

After missing 20 out of the first 21 games of the 2014–15 season with a back injury, Brazillian player Tiago Splitter was traded from the San Antonio Spurs to the Atlanta Hawks in exchange for draft rights to two second round picks. Because of Splitter's long absence, CWOR yields particularly good results in this case as we have significant non-hypothetical data for how the San Antonio Spurs performed without Splitter in their lineups. This situation improves CWOR's reliability when evaluating players.

During the 2014-2015 season, our model assigns Tiago Splitter a $CWOR$ value of +1. Table 3.11 shows our model's projected performance for the San Antonio Spurs with and without Splitter in their lineups.

Table 3.11
Projected performance for the San Antonio Spurs with and without Tiago Splitter in their lineup.

Predicted results	With Splitter	Without Splitter
Games won	73	72
R^2	0.74	0.72

Source: Data obtained through nbastatR library.

Because Splitter suffered significant back injuries that year and still had two years left on a $ 9,000,000 per year average salary contract, trading him for two draft picks was the right decision according to our CWOR metric. Releasing $9,000,000 in cap space for just one additional win in a season seems to be convenient for the organization. Contrary to the first analyzed case, our assumption of not accounting for the additional value that the incoming players bring to the organization is relevant to the final interpretation.

3.5.3 Problems with CWOR: Stephen Curry and LeBron James

As previously stated, CWOR is a lineup dependant metric and as such, context is required in order to avoid interpretation mistakes. One particular caveat is the fact that our metric underestimates the impact of MVP (Most Valuable Player) caliber players. Because our model associates plus–minus to time spent on court, when excluding star players like Stephen Curry, CWOR calculates plus–minus for all lineups in which Stephen Curry is not present.

In this particular season, the times when Curry was on the bench, was whenever the game was already decided by a large margin. As such, most of these lineups have an incredibly large associated plus minus. For the 2014-2015 season, our model estimates that the Warriors would actually perform better without Stephen Curry in their lineup. When put into context, CWOR's interpretation in this case is mistaken as Steph Curry was named the MVP of the league that year.

A similar situation arises with LeBron James. Because LeBron was such an important part of the Cleveland Cavaliers' roster, when trying to fit a model without James, we get insufficient playing time for the

remaining lineups and thus, our model fails to provide adequate fits.

3.6 Further Applications: Workload Management

While we have simulated the effect of a player not being able to play at all, it should be noted that we can extend CWOR calculation to less extreme scenarios such as

- A player only being able to play a limited number of minutes per game.
- A coach making a decision to use particular lineups more often against a given opponent.

Both of this scenarios make up what in sports is known as *workload management*.

For any high performance sport, it is crucial that players strike a balance between excessive fatigue and lack of work. The role of *workload management* is to reduce the risk of injury and optimize performance. Anyone who follows basketball knows that star players tend to sit out of random games particularly in the latter half of the season. This is more frequent whenever a team's season outcome seems to be defined. The reason for this spontaneous rest days is to avoid injuries and keep players rested and healthy for the playoffs.

It is obvious that this maneuver does not come without a downside, particularly for teams with their season still on the balance. The question that may be arised is should we risk a costly injury to our star players at the expense of trying to get into the post season? Or should the organization manage his/her minutes in order to have

the player well rested but with the inevitable risk of losing a playoffs spot? A similar dilemma arises for players coming back from injuries. How much will a team's results be affected if certain player is not in the lineup for an x amount of games? How many games can an organization rest a player without compromising their results?

Notice how, similar to our trade evaluation methodology in Section 3.5 we can filter specific players from our usage matrix $\boldsymbol{X}$ in order to fit a model that yields the hypothetical performance of a team with said player only playing a limited amount of minutes. This new prediction may provide insights as to whether or not the team's performance will be sufficient to achieve the necessary number of wins in order to reach the post season.

Even though our model could provide insights in this area, further improvement in its predictive value would be necessary as these scenarios require forecasting future results. Because NBA teams play multiple times against each opponent in a single season, a possible workaround for this caveat exists.

Suppose the season is 70 games in with 12 still left to play. An elastic net model very similar to the one used for assigning CWOR for trade evaluation in section 3.5 can be adjusted using 70 games as a complete season. For this adjustment, we would exclude the player's name and thus, we have a projection of how the team's season up to that point would look without said player on the lineup. We could then analyze the team's performance against those 12 teams that are still remaining on the calendar.

Chapter 4

Conclusion

Regardless of the lack of predictive value that was assesed for our model, practical applications still arose from the adequate in-sample predictions that were obtained.

4.1 Results and Model Limitations

In Chapter 3 we evaluated the predictive value of ridge, lasso and elastic net models. We conclude that due to the presence of overfitting and underfitting in ridge and lasso models, these lack any value to further analyze upon.

Elastic net on the other hand, provided adequate in-sample fit with an average $R^2 = 0.77835$. Although accuracy was satisfactory when evaluating the outcome of 41 playoff games, we conclude this model's performance metrics, particularly the coefficient of determination, are not reliable enough when predicting out of sample data. One possible cause for the lack of predictive value in terms of scoring margin in our models may be the extensive difference in lineup usage between the

regular season and the playoffs. Our model is based on playing time for 5-man units and these are highly variable during regular season games.

Nevertheless, our work achieves solid accuracy in predicting game outcomes, which may be encouraging particularly because of the availability and simplicity of the data used to build the experiments. As mentioned in Section 2.2.8 the majority of cutting edge frameworks in basketall analytics rely heavily on player tracking data. Because of the massive infrastructure required to implement this technology, this kind of data is not available for public access and its usage in basketball associations outside of the NBA and major FIBA competitions would be impossible. As such, we feel our model strikes a satisfactory compromise between simplicity and admissible predictive value.

Another caveat that is present for our data is the mid-season transfer window. NBA teams can trade players anytime between the final game of the past season and 10 days after the All-Star Game of the ongoing season. This opens up the possibility that a player who is present in our data for one team, suddenly gets traded and provides data for another team. Such is the case of Iman Shumpert and J.R. Smith. Who were traded on January 5th, 2015 from the Oklahoma City Thunder to the Cleveland Cavaliers where Shumpert even became a starter. The presence of this sudden changes in lineup usage promotes the overfitting that was present in the ridge and lasso models and diminishes the predictive value of the elastic net model.

One important issue that we need to mention is the fact that Huang's ridge regression model used a modified penalty $\boldsymbol{\beta}_0$ which he obtained previously with a Markov chain model. This resulted troublesome for our work as we did not have access to his results and thus, replicating his outcomes with ridge regression was not possible.

Regardless, the suitable in sample predictions that were obtained for regular season data gave place to a new metric: Contributed Wins Over Replacement (CWOR). This proposed metric can provide valuable insights to coaches and general managers as to how many more (or less) games a team can win with a given player in the lineup. An upside to this metric is the simplicity in the data required to compute it. Although in our case, lack of available data made collection and transformation diligent, an organized basketball team would have no problem collecting the necessary lineup usage data required to compute CWOR. It is worth noting that this tool provides a restrospective analysis and as of now, lacks the predictive value to project reliable results.

4.2 Further Work

Although our results were promising, there is a significant area of opportunity in which its predictive value can be improved. Particularly in its performance metrics. A more trustworthy coefficient of determination for out of sample predictions would be the most clear direction in which the model can improve.

Upon review, one of the most significant improvements that can be made to CWOR metric is the inclusion of forecasting value. As stated in Section 3.6, we can slightly work around this issue and use CWOR as a possible tool for a rough evaluation of team strength against certain teams. Nevertheless, the inclusion of a proper forecasting methodology in this framework can answer hypothetical questions within lineup usage such as: "If a particular player is injured, which of the remaining available lineups is the best one to use?" or "If I rest a specific player for 8 games, would my team still be able to achieve the desired number of wins for clinching a playoff spot?" The ability to

accurately answer such questions results of great interest to coaching staffs within the league.

Another major issue that must be adressed is the fact that CWOR severely underestimates the value of superstar players. If this drawback were to be alleviated, we could perform an analysis to try to identify which players, particularly superstars, have a synergetic effect upon their teammates.

We recognize that CWOR presents its caveats, but if we include into the framework all the potential enhancements discussed in this section, we see a great scope of useful applications for this metric to be applied. Whether internally within basketball organizations or as a tool for better result assesement in the betting markets when a player is absent from the lineup.

Appendix A

Tables

Table A.1

Sample rows of our recorded substitution data.

Game Id	Period Number	Time Remaining	Seconds Passed	Team	Player Out	Player In	Event Number	Lineup Before
21400001	1	05:17	403	ORL	Kyle O'Quinn	Aaron Gordon	61	Elfrid Payton, Evan Fournier, Kyle O'Quinn, Nikola Vucevic, Tobias Harris
21400001	1	04:51	429	NOP	Omer Asik	Ryan Anderson	66	Anthony Davis, Eric Gordon, Jrue Holiday, Omer Asik, Tyreke Evans
21400001	2	11:02	778	ORL	Dewayne Dedmon	Nikola Vucevic	144	Aaron Gordon, Ben Gordon, Dewayne Dedmon, Luke Ridnour, Willie Green
21400001	2	08:49	911	NOP	Ryan Anderson	Jrue Holiday	173	Austin Rivers, Jimmer Fredette, Omer Asik, Ryan Anderson, Tyreke Evans
21400001	3	06:44	1756	NOP	Jrue Holiday	Austin Rivers	304	Anthony Davis, Eric Gordon, Jrue Holiday, Omer Asik, Tyreke Evans
21400001	3	05:40	1820	ORL	Kyle O'Quinn	Ben Gordon	317	Elfrid Payton, Evan Fournier, Kyle O'Quinn, Nikola Vucevic, Tobias Harris
21400001	4	11:06	2214	NOP	Anthony Davis	Omer Asik	389	Anthony Davis, Austin Rivers, Jimmer Fredette, John Salmons, Ryan Anderson
21400001	4	06:39	2481	ORL	Ben Gordon	Evan Fournier	431	Aaron Gordon, Ben Gordon, Elfrid Payton, Nikola Vucevic, Tobias Harris
21400002	1	06:22	338	SAS	Danny Green	Manu Ginobili	44	Danny Green, Marco Belinelli, Matt Bonner, Tim Duncan, Tony Parker

Source: Data obtained through nbastatR library.

Table A.2
Substitution data.

Game Id	Home Team	Away Team	Player1 Team	Player2 Team	Player3 Team	Period	Time	Elapsed Seconds	Event Number	Player1 Name	Player2 Name	Player3 Name	Description Home	Description Away	Pts Home	pts Away	Shot Pts Home	Shot Pts Away	Margin Before Home	Margin Before Away	Home Lineup	Away Lineup
21400018	CLE	NYK	CLE	NYK	CLE	1	12:00	0	2	Anderson Varejao	Samuel Dalembert	Kevin Love	Jump Ball Varejao vs. Dalembert: Tip to Love	-	0	0	0	0	0	0	Anderson Varejao, Dion Waiters, Kevin Love, Kyrie Irving, LeBron James	Carmelo Anthony, Iman Shumpert, Quincy Acy, Samuel Dalembert, Shane Larkin
21400018	CLE	NYK	CLE	CLE	-	1	11:47	13	3	Kevin Love	Kyrie Irving	-	Love 1' Layup (2 PTS) (Irving 1 AST)	-	2	0	2	0	0	0	Anderson Varejao, Dion Waiters, Kevin Love, Kyrie Irving, LeBron James	Carmelo Anthony, Iman Shumpert, Quincy Acy, Samuel Dalembert, Shane Larkin
21400018	CLE	NYK	NYK	NYK	-	1	11:31	29	4	Carmelo Anthony	Shane Larkin	-	-	Anthony 16' Jump Shot (2 PTS) (Larkin 1 AST)	2	2	0	2	2	-2	Anderson Varejao, Dion Waiters, Kevin Love, Kyrie Irving, LeBron James	Carmelo Anthony, Iman Shumpert, Quincy Acy, Samuel Dalembert, Shane Larkin
21400018	CLE	NYK	CLE	-	-	1	11:20	40	5	LeBron James	-	-	MISS James 19' Jump Shot	-	2	2	0	0	0	0	Anderson Varejao, Dion Waiters, Kevin Love, Kyrie Irving, LeBron James	Carmelo Anthony, Iman Shumpert, Quincy Acy, Samuel Dalembert, Shane Larkin
21400018	CLE	NYK	CLE	-	-	1	11:19	41	6	Anderson Varejao	-	-	Varejao REBOUND (Off:1 Def:0)	-	2	2	0	0	0	0	Anderson Varejao, Dion Waiters, Kevin Love, Kyrie Irving, LeBron James	Carmelo Anthony, Iman Shumpert, Quincy Acy, Samuel Dalembert, Shane Larkin
21400018	CLE	NYK	CLE	-	-	1	08:40	200	20	Kevin Love	-	-	Love REBOUND (Off:0 Def:2)	-	5	6	0	0	-1	1	Anderson Varejao, Dion Waiters, Kevin Love, Kyrie Irving, LeBron James	Carmelo Anthony, Iman Shumpert, Quincy Acy, Samuel Dalembert, Shane Larkin

Source: Data obtained through nbastatR library.

Table A.3

Rows from raw play-by-play data.

slug Score	name Player	team Name Player1	slug Team Player1	name Player2	team Name Player2	slug Team Player2	name Player3	team Name Player3	slug Team Player3	slug Team Leading	id Game	number Event	numberEvent MessageType	numberEvent Action Type	numberPeriod	id Person Type1	id Player NBA1	id Team Player1	id Person Type2	id Player NBA2	id Team Player2	id Person Type3	id Player NBA3	id Team Player3	has video	time String WC	time Quarter	description Play Home	description Play Neutral	description Play Visitor	city Team Player1	city Team Player2	city Team Player3	minute Game	time Remaining	minute Remaining Quarter	seconds Remaining Quarter	score Away	score Home	margin Score
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Home	21400001	0	12	0	1	0	0	NA	0	0	NA	0	0	NA	FALSE	08:13 p. m.	12:00	NA	Start of 1st Period (8:13 PM EST)	NA	NA	NA	NA	0.00	48.00	12	0	NA	NA	NA
NA	Anthony Davis	Pelicans	NOP	Nikola Vucevic	Magic	ORL	Jrue Holiday	Pelicans	NOP	Home	21400001	1	10	0	1	4	203076	1610612740	5	202696	1610612753	4	201950	1610612740	TRUE	08:14 p. m.	12:00	Jump Ball Davis vs. Vucevic: Tip to Holiday	NA	NA	New Orleans	Orlando	New Orleans	0.00	48.00	12	0	NA	NA	NA
NA	Anthony Davis	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	2	2	1	1	4	203076	1610612740	0	0	NA	0	0	NA	TRUE	08:14 p. m.	11:43	MISS DAVIS 20' Jump Shot	NA	NA	New Orleans	NA	NA	0.28	47.72	11	43	NA	NA	NA
NA	Evan Fournier	Magic	ORL	NA	NA	NA	NA	NA	NA	Home	21400001	3	4	0	1	5	203095	1610612753	0	0	NA	0	0	NA	TRUE	08:14 p. m.	11:42	NA	NA	Fournier REBOUND (Off:0 Def:1) Vucevic 19' Jump Bank Shot (2 PTS) (Payton 1 AST)	Orlando	NA	NA	0.30	47.70	11	42	NA	NA	NA
2 - 0	Nikola Vucevic	Magic	ORL	Elfrid Payton	Magic	ORL	NA	NA	NA	Away	21400001	4	1	66	1	5	202696	1610612753	5	203901	1610612753	0	0	NA	TRUE	08:14 p. m.	11:31	NA	NA	O'Quinn STEAL (1 STL)	Orlando	Orlando	NA	0.48	47.52	11	31	2	0	-2
NA	Jrue Holiday	Pelicans	NOP	Kyle O'Quinn	Magic	ORL	NA	NA	NA	Home	21400001	5	5	1	1	4	201950	1610612740	5	203124	1610612753	0	0	NA	TRUE	08:14 p. m.	11:18	Holiday Bad Pass Turnover (P1.T1)	NA	Fournier Travelling Turnover (P1.T1)	New Orleans	Orlando	NA	0.70	47.30	11	18	NA	NA	NA
NA	Evan Fournier	Magic	ORL	NA	NA	NA	NA	NA	NA	Home	21400001	6	5	4	1	5	203095	1610612753	0	0	NA	1	0	NA	TRUE	08:14 p. m.	11:18	NA	NA	Orlando	NA	NA	0.70	47.30	11	18	NA	NA	NA
NA	Anthony Davis	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	7	2	1	1	4	203076	1610612740	0	0	NA	0	0	NA	TRUE	08:15 p. m.	11:06	MISS Davis 18' Jump Shot	NA	NA	New Orleans	NA	NA	0.90	47.10	11	6	NA	NA	NA
NA	Nikola Vucevic	Magic	ORL	NA	NA	NA	NA	NA	NA	Home	21400001	8	4	0	1	5	202696	1610612753	0	0	NA	0	0	NA	TRUE	08:15 p. m.	11:05	NA	NA	Vucevic REBOUND (Off:0 Def:1)	Orlando	NA	NA	0.92	47.08	11	5	NA	NA	NA
NA	Elfrid Payton	Magic	ORL	NA	NA	NA	Jrue Holiday	Pelicans	NOP	Home	21400001	9	2	5	1	5	203901	1610612753	0	0	NA	4	201950	1610612740	TRUE	08:15 p. m.	10:54	Holiday BLOCK (1 BLK)	NA	MISS Payton 2' Layup	Orlando	NA	New Orleans	1.10	46.90	10	54	NA	NA	NA
NA	Tyreke Evans	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	10	4	0	1	4	201936	1610612740	0	0	NA	0	0	NA	TRUE	08:15 p. m.	10:52	Evans REBOUND (Off:0 Def:1)	NA	NA	New Orleans	NA	NA	1.13	46.87	10	52	NA	NA	NA
NA	Kyle O'Quinn	Magic	ORL	Tyreke Evans	Pelicans	NOP	NA	NA	NA	Home	21400001	11	6	2	1	5	203124	1610612753	4	201936	1610612740	1	0	NA	TRUE	08:15 p. m.	10:50	NA	NA	O'Quinn S.FOUL (P1.T1)	Orlando	New Orleans	NA	1.17	46.83	10	50	NA	NA	NA
02-ene	Tyreke Evans	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Away	21400001	12	3	11	1	4	201936	1610612740	0	0	NA	0	0	NA	FALSE	08:15 p. m.	10:50	Evans Free Throw 1 of 2 (1 PTS)	NA	NA	New Orleans	NA	NA	1.17	46.83	10	50	2	1	-1
NA	Tyreke Evans	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	14	3	12	1	4	201936	1610612740	0	0	NA	0	0	NA	FALSE	08:16 p. m.	10:50	MISS Evans Free Throw 2 of 2	NA	NA	New Orleans	NA	NA	1.17	46.83	10	50	NA	NA	NA
NA	Omer Asik	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	16	4	0	1	4	201600	1610612740	0	0	NA	0	0	NA	TRUE	08:16 p. m.	10:49	Asik REBOUND (Off:1 Def:0)	NA	NA	New Orleans	NA	NA	1.18	46.82	10	49	NA	NA	NA
NA	Nikola Vucevic	Magic	ORL	Omer Asik	Pelicans	NOP	NA	NA	NA	Home	21400001	17	6	2	1	5	202696	1610612753	4	201600	1610612740	1	0	NA	TRUE	08:16 p. m.	10:46	NA	NA	Vucevic S.FOUL (P1.T2)	Orlando	New Orleans	NA	1.23	46.77	10	46	NA	NA	NA
NA	Omer Asik	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	18	3	11	1	4	201600	1610612740	0	0	NA	0	0	NA	FALSE	08:16 p. m.	10:46	MISS Asik Free Throw 1 of 2	NA	NA	New Orleans	NA	NA	1.23	46.77	10	46	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Home	21400001	19	4	1	1	2	1610612740	NA	0	0	NA	0	0	NA	FALSE	08:16 p. m.	10:46	PELICANS Rebound	NA	NA	NA	NA	NA	1.23	46.77	10	46	NA	NA	NA
NA	Omar Asik	Pelicans	NOP	NA	NA	NA	NA	NA	NA	Home	21400001	20	3	12	1	4	201600	1610612740	0	0	NA	0	0	NA	FALSE	08:17 p. m.	10:46	MISS Asik Free Throw 2 of 2	NA	NA	New Orleans	NA	NA	1.23	46.77	10	46	NA	NA	NA

Source: Data obtained through nbastatR library.

Table A.4
Predicted margin of victory or defeat with and without Ersan Ilyasova in The Milwaukee Bucks' lineup.

Game number	Predicted margin without Ilyasova	Predicted margin with Ilyasova	Actual margin
1	-0.290439814	-1.931067203	-2
2	2.815782923	7.678384847	12
3	-5.344458077	-8.42549467	-11
4	-0.290439814	2.398370085	6
5	-1.62339347	-5.397298295	-9
6	0.140981124	-0.913281276	-3
7	-0.290439814	0.008224005	1
8	-0.290439814	3.398328621	7
9	-6.891190064	-12.01760044	-16
10	0.245045182	3.522486601	7
11	-0.286013437	2.884899946	4
12	0.201945674	2.897874454	4
13	-32.49804364	-37.14706529	-42
14	-5.107652383	-7.566055775	-11
15	-0.290439814	7.923758266	12
16	7.635672172	12.5707067	17
17	6.659209492	11.63208957	16
18	-4.793188738	-10.10996198	-14
19	0.058183726	0.187357068	-3
20	-3.188639173	-2.849501839	-2
21	14.85556078	20.0369582	24

Game number	Predicted margin without Ilyasova	Predicted margin with Ilyasova	Actual margin
22	-13.60307312	-18.29998776	-23
23	-4.415344517	-9.01856181	-13
24	-0.440687107	2.281007259	5
25	-1.580050999	0.233356437	2
26	-0.290439814	-3.766401857	-7
27	-0.290439814	0.585116968	1
28	-2.007487614	-2.814228646	-4
29	-2.421719108	-4.963976161	-7
30	20.74667271	25.70530737	30
31	4.277508999	-0.526696606	-5
32	2.478634675	6.355023045	10
33	6.877566393	12.14814864	16
34	1.397150219	1.398848792	-3
35	5.435378739	9.353449944	13
36	0.284617872	-1.13167975	-6
37	11.42698267	16.21259657	20
38	4.917635503	10.11732611	14
39	0.233306677	-4.269617834	-8
40	5.120251608	8.698239824	16
41	0.820213709	0.517437718	-3
42	1.049957458	1.004947104	-2
43	6.689596279	10.06393211	15
44	1.385710031	-2.096862816	-6
45	3.165617866	4.463469221	7
46	9.503932898	11.92030663	15

Game number	Predicted margin without Ilyasova	Predicted margin with Ilyasova	Actual margin
47	4.279468977	5.733859527	7
48	2.887018655	5.000613001	7
49	1.903994411	4.730208356	8
50	1.681193893	-2.435662197	-6
51	1.49661475	2.295696143	3
52	1.186107834	3.174677521	6
53	2.174515303	4.51349675	8
54	2.431706744	5.332899167	8
55	-1.870858065	-7.695198437	-11
56	-6.680813164	-12.34714258	-16
57	6.410561224	10.34626418	16
58	-0.290439814	-4.503397203	-8
59	-2.727200795	-4.470448672	-7
60	-3.521539682	-8.238680523	-11
61	-0.290439814	-5.701127802	-9
62	-0.290439814	2.446484529	6
63	-6.77898045	-8.474931332	-11
64	1.526965825	3.820596428	6
65	-0.290439814	-3.280934404	-6
66	-3.908452942	-9.538155838	-13
67	-0.290439814	-0.164044732	-1
68	-1.568650138	-6.966917577	-10
69	-0.592817229	-1.655812886	-2
70	-8.906480257	-14.35967638	-18

Game number	Predicted margin without Ilyasova	Predicted margin with Ilyasova	Actual margin
71	-0.290439814	0.141944078	1
72	1.805892688	2.248655083	4
73	-3.847163103	-9.017171545	-13
74	-4.595503248	-9.441532271	-13
75	2.380368421	2.217986942	4
76	2.954844155	7.467130517	9
77	-1.625707815	-2.704154743	-7
78	0.465384012	-1.753023309	-5
79	4.476700923	4.567885443	8
80	13.47851541	18.35308385	23
81	0.928297135	6.416076422	10
82	-0.290439814	-3.012157598	-5

Source: Data obtained through nbastatR library.

Bibliography

Bentes, R. (2020, July 9). *Adding lineups to NBA play-by-play data.* NBA in R. https://nbainrstats.netlify.app/post/adding-lineups-to-nba-play-by-play-data/.

Bhat, H. S., Huang, L.-H., and Rodriguez, S. (2015). Learning stochastic models for basketball substitutions from play-by-play data.*MLSA Workshop at ECML/PKDD 2015.*

Bhattacharyya, S. (2020, September 28).*Ridge and lasso regression: L1 and l2 regularization* . Medium. https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b.

Cervone, D., D'Amour, A., Bornn, L., &; Goldsberry, K. (2016). A multiresolution stochastic process model for predicting basketball possession outcomes. *Journal of the American Statistical Association* ,111(514),585–599. https://doi.org/10.1080/01621459.2016.1141685

Dalpiaz D. (2020). *R for Statistical Learning!.* https://daviddalpiaz.github.io/r4sl/.

Deshpande, S. K., &; Jensen, S. T. (2016). Estimating an NBA Player's impact on his TEAM'S chances of winning. *Journal of Quantitative Analysis in Sports*, 12(2). https://doi.org/10.1515/jqas-2015-0027

Friedberg, S. H., Insel, A. J., &; Spence, L. E. (1997). *Linear algebra.* Prentice Hall.

Hastie, T., Friedman, J., &; Tisbshirani, R. (2017). *The elements of statistical learning: Data mining, inference, and prediction. Stanford University.* Springer. https://web.stanford.edu/ hastie/ElemStatLearn/.

Huang, L.-H. (2018). *Markov Chain Models and Data Science Applications* (thesis). eScholarship. Retrieved from https://escholarship.org/uc/item/6ch8z986

kjetil b halvorsen (https://stats.stackexchange.com/users/11887/kjetil-b-halvorsen), Where does linear regression fit into the bias-variance tradeoff?, URL (version: 2019-03-12): https://stats.stackexchange.com/q/397045

Kopf, D. (2017, October). *Data analytics have made the NBA UNRECOGNIZABLE.* Quartz. https://qz.com/1104922/data-analytics-have-revolutionized-the-nba/.

Lewis, M. (2013). Field of Ignorance. In *Moneyball: The art of winning an unfair game* (pp. 64–97). essay, W.W. Norton.

Linear Models. scikit. (n.d.). http://scikit-learn.org/stable/modules/linearmodel.html.

M, N. (2020, October 26). *Bias and variance in linear models.* Medium. https://towardsdatascience.com/bias-and-variance-in-linear-models-e772546e0c30.

Oh, M.-H., Keshri, S., and Iyengar, G. (2015). Graphical model for basketball match simulation. In *MIT Sloan Sports Analytics Conference*.

Oliver, D. (2011). *Basketball on paper: rules and tools for performance analysis*. Potomac Books, Inc.

Park, M. Y., &; Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4), 659–677. https://doi.org/10.1111/j.1467-9868.2007.00607.x

Tarr, G., &; Müller, S. (n.d.).*Lab 3: Regularization procedures with glmnet* . https://garthtarr.github.io/avfs/lab03.html.

Štrumbelj, E., &; Vračar, P. (2012). Simulating a basketball match with a homogeneous Markov model and forecasting the outcome.*International Journal of Forecasting* , 28(2), 532–542. https://doi.org/10.1016/j.ijforecast.2011.01.004

StatsUser (https://stats.stackexchange.com/users/48756/statsuser), When to use Ridge regression and Lasso regression. What can be achieved while using these techniques rather than the linear regression model, URL (version: 2016-12-15): https://stats.stackexchange.com/q/251708

Shirley, K. (2007). A Markov model for basketball. In *New England Symposium for Statistics in Sports*, Boston, MA

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Winston, W. L. (2009). Linear Weights for Evaluating NBA Players. In *Mathletics how gamblers, managers, and sports enthusiasts use mathematics in baseball, basketball, and football*(pp. 195–202). essay, Princeton University Press.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: A retrospective.*Journal of the Royal Statistical Society: Series B (Statistical Methodology)* , 73(3), 273–282. https://doi.org/10.1111/j.1467-9868.2011.00771.x

Xie, F., &; Xiao, Z. (2020). Consistency of L1 penalized negative binomial regressions. Statistics &; *Probability Letters*, 165, 108816. https://doi.org/10.1016/j.spl.2020.108816

Zhang, H., &; Jia, J. (2017). Elastic-net regularized high-dimensional negative binomial regression: Consistency and weak signal detection. *Statistica Sinica*. https://doi.org/10.5705/ss.202019.0315

Zou, H., &; Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B* (Statistical Methodology), 67(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Aplicación de métodos de regresión lineal regularizada
para evaluar el valor de
mercado de jugadores y
predecir resultados en la NBA
escrito por Pablo López Landeros,
se terminó de imprimir en noviembre de 2021
en los talleres de Impresos Martínez.
República de Cuba 99, colonia Centro Histórico,
Ciudad de México.