ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

DEPARTMENT OF STATISTICAL SCIENCES

Second Cycle Degree in Statistical Sciences

EVALUATING SHOOTING PERFORMANCE IN THE BASKETBALL COURT: AN APPROACH WITH MACHINE LEARNING

Presented by:

Gianluca Piromallo

885001

Supervisor:

Prof. Stefania Mignani

Co-supervisor:

Prof. Marica Manisera

Prof. Paola Zuccolotto

III SESSION

ACADEMIC YEAR 2019/2020

Abstract. The evaluation of shooting performance and scoring probability of teams and players in different areas of the court is a relevant theme in basketball analytics. In this thesis, it is reproduced on different data an analysis taken from the paper *Basketball spatial performance indicators and graphs* by Paola Zuccolotto, Marica Manisera and Marco Sandri, that uses classification trees for obtaining a partition of the court in rectangles, which are maximally different with respect to shooting performances. Then, it is proposed a method based on random forests and gradient boosting for predicting shots results and scoring probabilities from every possible point of the court. The obtained results permit to create a unique court map for each player or team with free-shaped areas homogeneous with respect to predicted scoring probability within them.

Keywords: basketball analytics, classification trees, random forests, gradient boosting, performance analysis

TABLE OF CONTENTS

1. INTRODUCTION

In the last few years, sports analytics have become a popular theme in the scientific community as reflected by the large number of publications about it. The main reason consists in the availability of several datasets, which permit to data scientists to extract insightful information and to find answers to complex problems. This thesis is focused on basketball. Generally, the main available datasets for this sport are the following: box scores, which contain basic descriptive statistics about players and teams that face each other in a match; play-by-play data, which describe in details relevant events happened during a match; tracking data, which are detected by technological devices, such as GPS sensors, in order to gather information about players' movement.

The first fundamental contributions in scientific literature about basketball analytics are the book *Basketball On Paper: Rules And Tools For Performance Analysis* (Oliver, 2004) and the paper *A Starting Point for Analyzing Basketball Statistics* (Kubatko et al., 2007). They introduce the concepts of pace and possession, in order to make teams' performances comparable and they exploit them to define the so-called Four Factors, which are the main indicators for performance analysis. Then, several statistical researches were conducted in basketball aiming to analyse a huge variety of aspects of the game. Considering some examples of frequently analysed topics, it is possible to perform analyses about the prediction of the outcome of a game or a tournament (West 2008; Loeffelholz, Bednar, and Bauer 2009; Brown et al. 2010; Gupta 2015; Lopez and Matthews 2015; Ruiz and Perez-Cruz 2015; Yuan et al. 2015; Manner 2016; Vračar et al., 2016), to analyse characteristics of players in order to define advanced roles (Alagappan, 2012; Bianchi et al., 2017), to investigate the aspects that differentiate successful and unsuccessful teams (Koh et al., 2011, 2012; García et al., 2013), to examine optimal game strategies (Annis, 2006; Zhang et al., 2013; Skinner and Goldman, 2017), to detect how players' movement and passes network impact on games' results and teams' performances basing on tracking data (Passos et al., 2011; Lamas et al., 2011; Piette et al., 2011; Fewell et al., 2012; Travassos et al., 2012; Shortridge et al., 2014; Ante et al., 2014; Clemente et al., 2015; Gudmundsson and Horton, 2016; Metulini et al., 2017a,b; Bornn et al., 2017; Miller and Bornn, 2017; Metulini et al., 2018; Wu and Bornn, 2018) and to analyse players' shooting performance (Fearnhead and Taylor, 2011; Gabel and Redner, 2012; Schwarz, 2012; Özmen, 2012; Avugos et al., 2013; Page et al., 2013; Erčulj and Štrumbelj, 2015; Cervone et al., 2016; Deshpande and Jensen, 2016; Passos et al., 2016; Franks et al., 2016; Engelmann, 2017; Zuccolotto et al., 2018, Manisera, Sandri and Zuccolotto, 2019).

Between October 2019 and March 2020, I had the opportunity to do an internship at Big & Open Data Innovation Laboratory of University of Brescia, where the research project BDsports (Big Data analytics in sports) is developed, and I got in touch with basketball analytics by studying the book which can be considered the milestone of this thesis: *Basketball Data Science with applications in R* (Manisera and Zuccolotto, 2020). The realisation of my master thesis is my contribution to the project BDsports.

This final dissertation deals with the theme of the evaluation of the shooting performance. The aim consists in investigating how the shooting performance of a player or a team varies in different areas of the court for underlining shooting patterns and favourite spots. In order to achieve this goal, I propose an approach with machine learning algorithms: classification trees, random forests and gradient boosting are used for the prediction of shot outcome and scoring probability given the position on the court from which the shot is attempted. The obtained results are exploited to produce graphical tools where the court is partitioned in areas, which are homogeneous with respect to shooting performance within them. The proposed tools can be very useful for the coaching staff to define game strategies, to develop specific training programmes and to compare players in scouting.

This thesis is outlined in the following way: chapter 2 contains the methodological description of classification and regression trees, random forests and gradient boosting; in chapter 3, the empirical analyses, obtained results and graphical representations are presented; in chapter 4, the most important results are summarized and a possibility for future research is proposed.

2. METHODOLOGY

In chapter 2, the machine learning algorithms applied in the empirical analyses are described from a methodological point of view. The following algorithms are illustrated: classification and regression trees, random forests and gradient boosting.

2.1 CLASSIFICATION AND REGRESSION TREES

Classification and regression trees (CART, Breiman et al., 1984) are considered as one of the most popular data mining algorithms. The aim of CART is predicting the value of a dependent variable Y given a set of predictors $X_1, \dots, X_p$ by recursively applying binary partitions to the feature space. Therefore, this space is subdivided in p-dimensional hyper-rectangles (regions) and, successively, a simple model, such as mode or average, is fitted to each region. Classification or regression trees are used according to the nature of variable Y, in particular: if Y is a quantitative variable then regression trees are used, otherwise, when Y is categorical, it is the case of classification trees.

More precisely, technical details of the building process of CART are defined as follows. First of all, consider regression trees. Suppose that N sample units are observed, then for each sample observation, data consist in (x_i, y_i) for $i = 1, \dots, N$, where $x_i = (x_{i1}, \dots, x_{ip})$. The CART algorithm produces a partition of the predictors' space into M regions $(R_1, \dots, R_M)$ and the response variable Y is modelled as a constant c_m in each region. Therefore, the resultant tree can be expressed in additive way as the sum of the values of the constants in each region as shown by the following formula:

$$f(x) = \sum_{m=1}^{M} c_m \mathbf{1}\,(x \in R_m)$$

where $\mathbf{1}(\cdot)$ is an indicator function that takes value 1 if $x \in R_m$, otherwise it is equal to 0.

At the first step of the algorithm, all observations are located in the root node, consequently this node is characterized by strong heterogeneity. The goal is to define a rule that separates the observations into two nodes such that the homogeneity within the nodes is maximized. The CART algorithm splits a predictor at all the possible split points and the sample is binary partitioned at each possible split point, then the reduction of heterogeneity is evaluated for every possible binary partition. Therefore, let s be the split point and X_j be the splitting variable and

define the pair of half-planes $R_1(j,s) = \{X \mid X_j \leq s\}$ and $R_2(j,s) = \{X \mid X_j > s\}$. The idea is to seek to the split point s and the splitting variable X_j that solve the following minimum problem:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

Given any choice of j and s, the inner minimization is solved by $\hat{c}_1 = ave(y_i \mid x_i \in R_1(j,s))$ and $\hat{c}_2 = ave(y_i \mid x_i \in R_2(j,s))$, where $ave(\cdot)$ indicates the average. Then, the previous minimum problem resorts to:

$$\min_{j,s}\left[\sum_{x_i\in R_1(j,s)}(y_i-\hat{c}_1)^2+\sum_{x_i\in R_2(j,s)}(y_i-\hat{c}_2)^2\right]$$

Therefore, it is feasible to determine the best pair (j,s) which characterizes the best possible binary partition and each sample observation is assigned to one of the two resulting regions. This procedure can be repeated until every node contains only one observation, which is the case of maximum homogeneity, however this would be clearly meaningless and there would be a problem of overfitting. Then, the tree size has to be treated as a tuning parameter which represents model complexity and the usual strategy is to determine a minimum number of observations that a node should contain (node size), in order to stop the splitting process when it is reached.

The last phase of CART algorithm is the so-called pruning, which is useful for outliers detection. The aim of pruning is to obtain a smaller and less complex tree by collapsing a certain number of non-terminal nodes, in order to have an easier interpretation of the final results. Then, let T_0 be the previously obtained tree and define T as any possible tree obtainable by pruning T_0. The terminal nodes correspond to the regions $R_1, \dots, R_M$ and they are indexed by m. Therefore, define the cost-complexity criterion $C_\alpha(T)$ as follows:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

where $|T|$ is the number of terminal nodes in T, N_m is the number of observation in the m-th terminal node, $\hat{c}_m = \frac{1}{N_m}\sum_{x_i\in R_m} y_i$ is the estimated response in R_m and $Q_m(T) =$

$\sum_{x_i \in R_m}(y_i - \hat{c}_m)^2$ is the impurity measure (squared-error node). α is a tuning parameter that can take only positive values ($\alpha \geq 0$) and represents the trade-off between goodness of fit to the data and tree size: for low values of α the model fits better to the data and the tree size is large, but the tree is more difficult to interpret, while for high values of α the model has a worse fit to data and the tree size is small, but the tree is easier to interpret. Now, for each value of α, find the unique subtree $T_\alpha \subseteq T$ that minimizes the cost-complexity criterion $C_\alpha(T)$ by using weakest link pruning in the following way: collapse the non-terminal node that causes the smallest per-node increase in $\sum_{m=1}^{|T|} N_m Q_m(T)$ and repeat this step until the tree only composed by the root node is produced. In this way, a finite sequence of nested trees is obtained. Finally, $\hat{\alpha}$, which is the optimal value of α, is estimated using K-fold cross-validation and the optimal tree $T_{\hat{\alpha}}$ is obtained.

The only difference between regression trees and classification trees consists in the way of measuring the heterogeneity reduction when splitting nodes in the construction of the trees and also in the pruning phase. More specifically, the impurity measure $Q_m(T)$ previously defined cannot be used for the classification problem, where suitable measures could be the misclassification error, the Gini index or the cross-entropy.

In conclusion, CART has become a very popular data mining algorithm, because it has several advantages and few disadvantages. First of all, it does not need any kind of distributional assumption either for dependent or explanatory variables. Then, these variables can be either quantitative or categorical and CART is also invariant under monotone transformations of independent variables. Moreover, outliers tend not to affect CART, since they are isolated into a single node. Another advantage is that CART effectively deals with high dimensionality problems, because it is able to select few important predictors from a large set of input variables, in order to detect and explain complex interactions in data. In this way, the results are easy to interpret. However, the main weakness of CART is that, since it is not based on a probabilistic model, it is not possible to assign a probability or a confidence interval to the obtained predictions, but the accuracy depends on how well a tree predicted the responses in other situations, similar to the analysed one. Moreover, predictions made by trees usually have high variance. This is due to the instability that generally characterizes classification and regression trees. More precisely, instability means that "small changes in input training samples may cause dramatically large changes in output classification rules" (Li et al., 2002).

2.2 RANDOM FORESTS

Trees are able to capture very complex interactions in data and they are generally low biased, however they are characterized by high variance. This drawback can be faced using random forests (Breiman, 2001), which is an algorithm that aims to reduce the variance in tree-based models by averaging a large collection of de-correlated trees. In fact, Breiman defines random forests as "a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest". The idea is that trees can be treated as random variables, which are identically distributed with pairwise positive correlation. Consequently, given B trees, the mean of their variance is computed in the following way:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

where ρ is the correlation between trees and σ^2 is the variance of each tree. Clearly, as $B \to \infty$, the second term of the sum tends to zero. Therefore, random forests reduces the average variance by reducing the correlation between trees with a small increase in the variance of each tree. The reduction of the pairwise correlation is achieved by randomly selecting a subset of the set of predictors as candidates for splitting when growing random forest trees. If the total number of predictors is small, then it is possible to use linear combination of predictors in order not to have high correlation between trees. Specifically, technical details are described below.

Given a dataset composed by N sample observations, for each observation, values of a response variable Y and a set of predictors $X_1, \dots, X_p$ are recorded. Successively, draw B bootstrap samples of size N. Build one random forest tree T_b $(b = 1, \dots, B)$ for every sample of bootstrapped data by recursively repeating these steps for each terminal node of each tree, until the minimum node size n_{min} is reached:

- Randomly select m predictors from the p predictors as candidates for splitting
- Find the best variable and the best split point among the m predictors
- Binary split of the node

In this way, the output consists in the set of trees $\{T_b; \Theta_b\}_1^B$, where Θ_b is a random vector generated for each tree, which characterizes the trees in terms of splitting variables, split points at each node and terminal-node values. The B random vectors are independent one another and they are identically distributed. Finally, since random forests can be used in both classification and regression framework, the difference between the two approaches consists in how to make

predictions for new data points. In particular, given a new data point x and considering the regression framework, the prediction is made as:

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

If classification is considered, then indicate with $\hat{C}_b$ the class prediction of the b-th random forest tree:

$$\hat{C}_b = majority\ vote\{\hat{C}_b(x)\}_1^B$$

where $majority\ vote$ indicates that the data point x is assigned to the class where it is assigned in the most part of the B random forest trees.

The basic tuning parameters for random forests are the number B of trees to build and the number m of variables candidate for splitting. Usually, when B is around 200 the algorithm tends to stabilize and the prediction performance does not particularly improve. Regarding the number of variables candidates for splitting, m should be generally small. A very simple rule consists in setting $m = \sqrt{p}$, where p is the total number of predictors. In this way, the correlation between trees is reduced and, therefore, also the variance of the average decreases.

Random Forests may be considered as an improvement of CART, because of the variance reduction, which makes the tree to be far more stable and accurate in predictions. Moreover, few tuning is required. In general, random forests algorithm has different strengths: it works with both quantitative and categorical variables, it does not need distributional assumptions of variables, it is able to handle missing values in data and it is robust to outliers. As a drawback, since a large collection of trees has to be grown, random forests are computationally expensive and the interpretation is more difficult than dealing with a single tree as in CART.

2.3 GRADIENT BOOSTING

Boosting is one of the most powerful machine learning algorithms introduced in the last twenty years. Like CART and Random Forests, it can be used in both classification and regression frameworks. The key idea in boosting consists in combining a certain number of weak learners, such as trees, in order to obtain a powerful final model. In general, a boosted tree model can be represented as:

$$f_M = \sum_{m=1}^{M} T(x, \Theta_m)$$

where $T(\cdot)$ indicates the tree model, x represents the data points and $\Theta_m = \{R_{jm}, \gamma_{jm}\}_{j=1}^{J}$ is a vector that characterizes each tree in terms of terminal nodes (R_{jm} with $j = 1, \dots, J$ indicating the number of terminal nodes) and constants that the model associates to each terminal node (γ_{jm} with $j = 1, \dots, J$). Now, assume to have N data points of the form (x_i, y_i) such that x are the predictors and y is the response variable. At each iteration, the aim is to find a tree such that the prediction error (thus, a loss function) is minimized. Then, the previous model f_M is obtained by solving in a stagewise manner the following minimum problem:

$$\widehat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} L\big(y_i, f_{m-1}(x_i) + T(x_i, \Theta_m)\big)$$

where $L(\cdot)$ is a loss function. A possible solution can be obtained by numerical optimization via gradient boosting. For each data point at each iteration, define the gradient as:

$$g_{im} = \left[\frac{\partial L\big(y_i, f(x_i)\big)}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

The gradient is defined only on data points in the training set, but the main goal is to generalize the boosted tree model f_M to new data points in order to make predictions. Consequently, at each iteration, the solution consists in fitting a regression tree whose prediction are as close as possible to the negative gradient in the following way:

$$\widetilde{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} \big(-g_{im} - T(x_i, \Theta_m)\big)^2$$

$\widetilde{\Theta}_m$ is an approximation of $\widehat{\Theta}_m$, however they are similar enough to reach the purpose. In this way, the regions R_{jm} and the constants γ_{jm} are obtained. Finally, at each iteration, the constants exploited to find the next constituent tree:

$$f_m = f_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm}\, I\big(x \in R_{jm}\big)$$

In this algorithm, the main differences between classification and regression are in the functional forms of the loss function $L(\cdot)$ and the negative gradient $-g_{im}$. In particular, when

classification is considered, the negative gradient is defined for every possible class and for each observation as: $y_{ik} - p_k(x_i)$, which corresponds to the difference between the observed classification in k-th class and the estimated probability of belonging to k-th class. The probabilities $p_k(x_i)$ are obtained as:

$$p_k(x_i) = \frac{e^{f_k(x_i)}}{\sum_{l=1}^{K} e^{f_l(x_i)}}$$

where $f_k(x_i)$ is the logit transform.

The basic tuning parameters of gradient boosting algorithm are the number of iterations M and the size of the constituent trees. On one hand, increasing the number of iterations will lead to an improvement in the accuracy of predictions in the training set; on the other hand, if the number of iterations becomes too high, then it is possible to have an overfitting problem, which will cause poor prediction performances on new data points. Considering the size of the constituent trees, also growing the trees too deeply could be a cause of overfitting. Usually, the size is fixed to be the same for each constituent tree and it does not vary from an iteration to another. Furthermore, there is another possible regularization strategy: the shrinkage. Specifically, shrinkage can be employed in the last step of each iteration of the algorithm, where the m-th constituent tree is defined:

$$f_m = f_{m-1}(x) + \nu \sum_{j=1}^{J} \gamma_{jm} I\left(x \in R_{jm}\right)$$

ν is a regularization term, which scales the contribution of each tree of a factor $0 < \nu < 1$. Basically, the parameter ν controls the learning rate of the boosting procedure and it is straightforward that the risk of overfitting increases as the value of ν gets close to 1.

In conclusion, boosting became a very popular algorithm in data science, because it has several advantages. First of all, predictions produced by this method are generally very accurate. Moreover, it works well with both quantitative and categorical variable, it does not need the features to be scaled (or any kind of transformation) and no distributional assumptions are needed. However, boosting presents also some disadvantages. Like Random Forests, since it is an ensemble of trees, the interpretation is quite hard and the computational cost could be very high, especially when dealing with a lot of predictors. Furthermore, the parameter tuning for controlling the learning rate of boosting procedure may be computationally expensive.

3. EVALUATING SHOOTING PERFORMANCE IN THE BASKETBALL COURT

This chapter represents the main part of my thesis. All the empirical analyses are descripted in details, including the description of the dataset.

3.1 AIM OF THE ANALYSES

The aim of this thesis is proposing the use of machine learning algorithms, in order to obtain graphical tools for the evaluation of shooting performance of players or teams in the basketball court. The idea is to study how the shooting performance varies in different areas of the court, in order to highlight players' or teams' shooting patterns and favourite spots. In order to achieve this goal, in the first analysis basic tools for basketball analysts are described. Successively, an approach with machine learning is proposed in two deeper analyses. The first one is taken from the paper *Basketball spatial performance indicators and graphs* by Paola Zuccolotto, Marica Manisera and Marco Sandri and the aim is to find a partition of the court induced by classification trees. The second analysis aims to predict shot outcomes and scoring probabilities in every point of the court by applying random forests and gradient boosting and the results are exploited for producing a further graphical tool. Finally, a real life application of the previously proposed tools shows how to use them for the comparison of the shooting performance of three players.

3.2 DATASET

The data source is bigdataball.com, which is a website that provides data science materials about different American sports. In particular, it makes available several kinds of datasets about the main championships of basketball (NBA and WNBA), baseball (MLB), football (NFL) and hockey (NHL).

The dataset used for the analysis contains the so-called play-by-play data of all NBA regular season and playoff games played in 2018-2019. This particular type of data consists in a set of variables that describe all relevant events happened during a match. Specifically, it is composed by the following variables:

- **game_id**: number that identifies a specific match
- **data_set**: name of the dataset (2018-2019 Regular Season or 2019 Playoff)

- **date**: it indicates when the match was played
- **a1, a2, a3, a4, a5**: name of away team's players that are on the court in a specific instant of the match
- **h1, h2, h3, h4, h5**: name of home team's players that are on the court in a specific instant of the match
- **period**: discrete variable that indicates the quarter of the match
- **away score** and **home score**: numbers that indicates the score for respectively away and home teams in a certain instant of the match
- **remaining_time**: time remained to end a period of the match
- **elapsed**: time passed from the beginning of the period
- **play_length**: length of the play in terms of time
- **play_id**: number that identifies a specific play
- **team**: name of the team for which an event happens
- **event_type**: qualitative variable that describes the event
- **assist**: when a shot is scored, it indicates the name of the player who made the assist
- **away** and **home**: when there is a jump ball, it indicates the name of players involved for respectively the away team and the home team
- **block**: when a shot is blocked, it indicates the name of the player who blocked it
- entered and left: when there is a substitution, it indicates the names of players that respectively enter and left the court
- **num**: when there is a free throw, it indicates if it is the first (1) one or the second (2) one and so on
- **opponent**: when there is a foul, it indicates the name of the player that is fouled
- **outof**: total number of free throws shot
- **player**: name of the player that made a play
- **points**: when a basket is scored, it indicates the number of points scored with that basket
- **possession**: after a jump ball, it indicates the name of the player that controlled the ball
- **reason**: description of a foul or a turnover
- **result**: categorical variable that, when there is a shot, takes value "made" or "missed" describing the result of the shot
- **steal**: when there is a turnover, it indicates the name of the opponent player that stole the ball
- **type**: it indicates the type of play

- **shot_distance**: when there is a shot, it indicates the distance from the basket (in feet)
- **original_x** and **original_y**: coordinates of the position from where a shot is taken
- **converted_x** and **converted_y**: see above, but the origin is changed
- **description**: brief description of the event

In order to perform the analysis using the R package BasketballAnalyzeR (Manisera, Sandri and Zuccolotto), it is necessary to manipulate the dataset with the command PbPmanipulation, which adds the following five more variables to the original dataset:

- **periodTime**: number of seconds passed in a period when the event happens
- **totalTime**: number of seconds passed in the whole match when the event happens
- **playlength**: number of seconds that indicates the length of the play in terms of time
- **oppTeam**: when an event happens for a team, it indicates the name of the opponent team

3.3 EMPIRICAL ANALYSES

In this section, different analyses are presented for the evaluation of shooting performance in the basketball court and the production of graphical tools. At the beginning a basic analysis is illustrated and, going ahead in the chapter, the analyses become deeper and more complex under a statistical point of view. In the end, a practical example of how to use the proposed tools in the comparison of players' shooting performance is proposed.

3.3.1 SPGs AND CART FOR PARTITIONING THE BASKETBALL COURT

This analysis comes from the paper *Basketball spatial performance indicators and graphs* by Paola Zuccolotto, Marica Manisera and Marco Sandri. Focusing the attention on spatial performance graphs (SPGs), the authors start the analysis describing the basic tools for analysts for evaluating the shooting performance of a player or a team: the shot chart with made and missed shots and the density plots. These SPGs are useful to highlight players' favourite spots on the court to shoot from. Successively, another analysis that can be performed consists in splitting the court into predetermined areas and analyse the shooting performances in these zones of the court. The R package BasketballAnalyzeR allows to produce all the previous cited SPGs. The last analysis introduced in the paper aims to overcome the shooting performance analysis in predetermined area of the court identifying, for each player, a unique partition of the

court in rectangles based on player's shooting accuracy. The key idea is to use classification trees in order to find the best partition of the court such that the difference in shooting percentages between rectangles is maximized. Then, the dependent variable Y is a binary variable that describes the outcome of an attempted shot (made or missed) and the predictors are the space coordinates of each attempted shot: x (width) and y (height). Clearly, the sample observations are the attempted shots. In this way, it is possible to produce several data visualizations by representing the rectangles obtained by the partition of the predictors' space induced by the estimated tree and colouring the obtained rectangles according to different indicators. As case study, the authors analysed the shooting performance of Stephen Curry in the regular season games of 2017/2018 NBA championship.

Since the aim of my thesis is to extend the analyses introduced in this paper, as first step, I decided to replicate them using play-by-play data referring to regular season and playoff games of 2018/2019 NBA championship and the analysed player is again Stephen Curry. First of all, consider Curry's shooting percentages and attempted shots, in order to have a rough idea of his shooting performance: 52% for 2-point shots (377/725) and 42.9% for 3-point shots (441/1027). Since scoring 3-point shots is obviously more difficult than scoring 2-point shots, it is possible to say that he performs better from outside the 3-point line, even if the shooting percentage is lower. Expected points are helpful in order to understand this concept. They are defined as the multiplication between the value of the shot (2 or 3 points) and the scoring probability, which can be roughly estimated by the shooting percentage. Then, Curry's expected points are 1.04 when he attempts a 2-point shot and 1.287 when he takes a 3-point shot. The evidence of his better performance in 3-point shots than 2-point shots is confirmed by the most interesting aspect that comes out from this very basic analysis: Curry takes a lot more 3-point shots than 2-point shots. There could be many reasons to justify this: for example, the role and the small size of the player or the evolution of nowadays basketball, where all teams tend to increase the number of 3-point attempts year by year. However, knowing if a shot is taken outside or inside 3-point line is not enough for in depth evaluation of shooting performance: these rough statistics does not give information about the precise position in the basketball court from when the shot is attempted. In order to have more precise idea (but still rough) of players' shooting patterns and favourite spots, it is possible to display the shot chart with made and missed shots and the density plots, as shown respectively in figure 1 and figure 2. Analysing these plots, it is possible to notice that Curry has a long shooting range, since he attempted several shots from nearly 2 metres away from the 3-point line (about 9 metres from the basket) and that he shoots frequently

from outside the 3-point line and from very close to the basket, while the number of shots attempted from middle-distance is definitely lower. He also prefers shooting from the middle than from the corners.

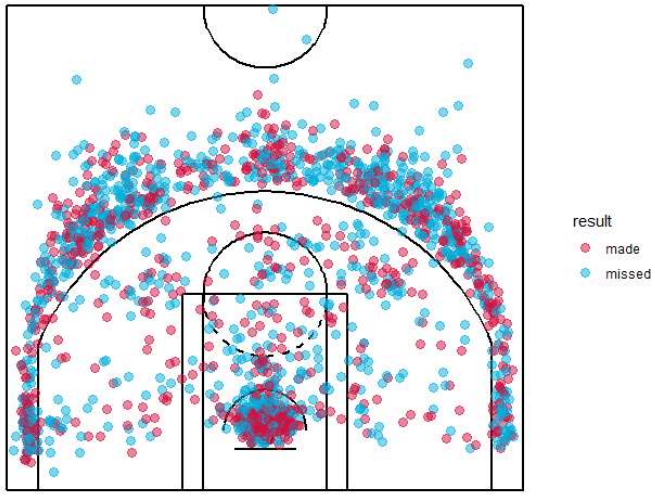

Figure 1: Shot chart with made and missed shots - Stephen Curry, NBA regular season and playoff 2018/2019

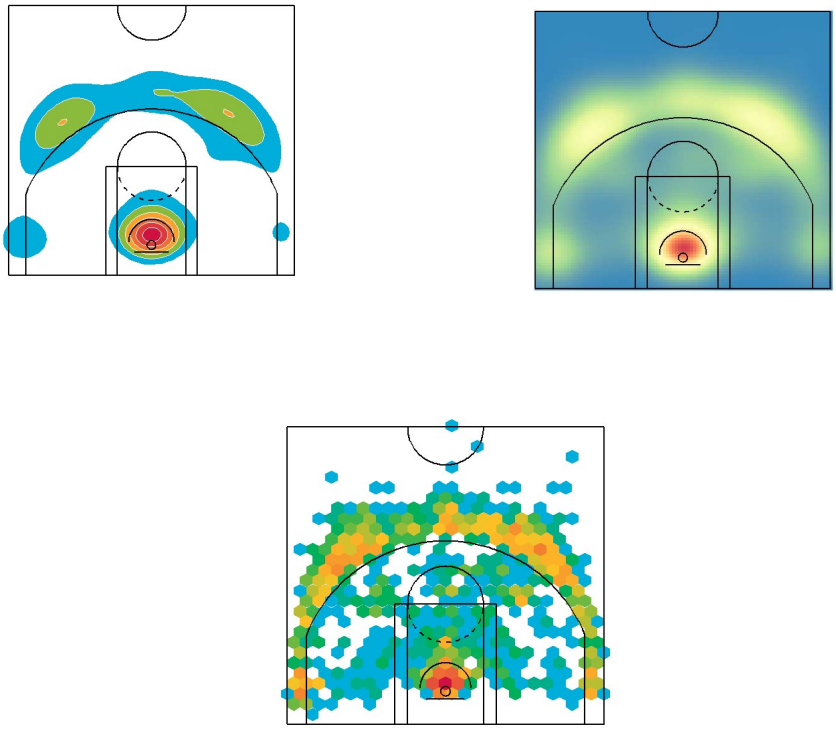

Figure 2: Density plots with polygons (top-left), raster (top-right), hexbins (bottom) - Stephen Curry, NBA regular season and playoff 2018/2019

Successively, the analysis proceeds with the realization of another shot chart where the court is subdivided in predetermined areas, in order to evaluate the player's shooting performance in these zones. The function `shotchart` of `BasketballAnalyzeR` splits the court in four areas according to the distance of the basket: the splitting lines are the 3-point line, a virtual line in between the basket and the 3-point line and the arc of radius 4 feet around the basket (the latter determines the so-called restricted area). Then, this function allows the user to make a further split of the former three zones (restricted area is excluded) into the desired number of sectors. In order to be coherent with the analyses that are described later, I decided to colour each area with respect to the shooting percentage within it. The resultant shot charts are displayed in figure 3. These SPGs show that Curry's shooting percentages are higher in the restricted are (61% with 312 attempted shots) and in the very next area on player's right side (61% with only 18 attempted shots). Considering the 3-point shots, he has better shooting percentages from his right than from his left. A possible explanation of these facts could be that he is a right-handed player. On the contrary, considering middle-distance shots, he performs better from his left, but

the number of attempted shot from middle-distance is small and probably it is not enough to assume that his performance on the right is definitely worse. The most interesting result is shown by the shot chart with 6 sectors: Curry's 3-point shots are more accurate when attempted from the wings (48% from both left and right wings) than from the middle (38% from middle-right and 36% from middle-left).

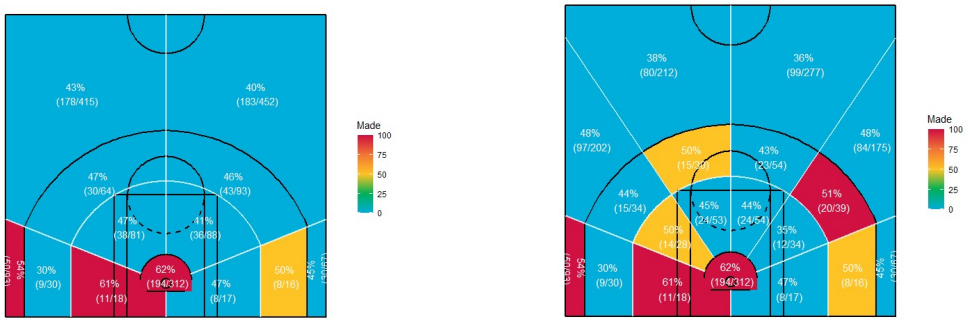

Figure 3: SPGs with 4 sectors (top) and 6 sectors (bottom) and areas coloured according to shooting percentages - Stephen Curry, NBA regular season and playoff 2018/2019

Finally, I propose the partition of the court with classification trees. Before fitting the model to the data, I decided to remove all shots attempted from beyond the half-court line, because they are useless for evaluating the shooting performance. These observations are clearly outliers and they would have affected the model estimation in a bad way. I fit a classification tree with complexity parameter $cp = 0.005$ and minimum number of observations in a node for having a split equal to $minsplit = 300$. This means that a node can be candidate for a split only if it is composed of at least 300 observations and that any split must decrease the overall lack of fit at least of a factor $cp = 0.005$. These parameters permit to save computational time and to improve the easiness of interpretation by pruning the tree. In a preliminary analysis, I tried to fit the tree with $minsplit = 150$. In this way, 13 terminal nodes were obtained, then the court was subdivided in 13 areas and the interpretation was hard: it is not very informative to have several small areas in the evaluation of the shooting performance.

Figure 4 shows the obtained tree. Every node is coloured in green or in blue, according to the frequency of made and missed shots that it contains: if there are more made shots than missed shots, then the node is green otherwise it is blue. Moreover, for each node, the name of the

prevalent category (made or missed), the relative frequency of both made and missed shots, the node number, the percentage of observations contained and the splitting rule are displayed. Considering the latter, observations that satisfy the rule move to the left branch, whereas the others move to the right one.

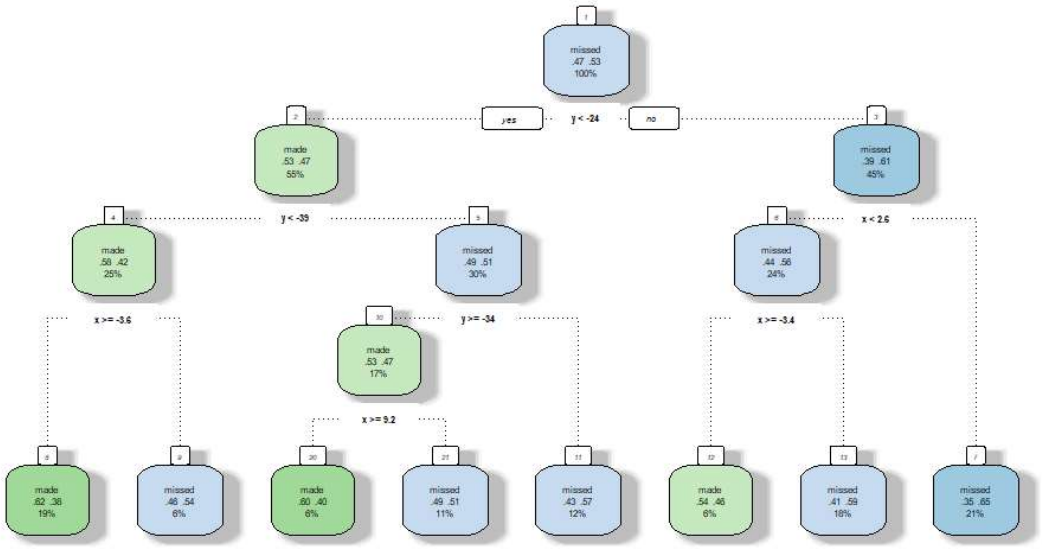

Figure 4: Classification tree - Stephen Curry, NBA regular season and playoff 2018/2019

Starting from the root node containing all the observations, there is the first split for predictor $y = -24$. Consequently, this split creates a virtual horizontal line at 24 feet from the centre of the court and the shots are divided in two groups according if they are attempted above or below this virtual line. It follows that 55% of the shots are attempted below with a shooting percentage of 53% and 45% of the shots are attempted above with a shooting percentage of 39%. Then, the shots above the line split for predictor $x = 2.6$, which means that a vertical virtual line is created at 2.6 feet to the right of the centre of the court and so on, until terminal nodes are obtained. The terminal nodes are represented by rectangles on the court and the difference in shooting percentages between rectangles is maximized, because the algorithm defines groups of observations that are homogenous with respect to the response variable (shot outcome: made or missed). The partition of the court induced by classification tree is visualized in figure 5. Analysing this graph, it is possible notice that the results are quite coherent with the ones obtained by the partition of the court in predetermined areas. First of all, there is again evidence that he performs better from his left when shooting from middle-distance. This is remarked by

the presence of the two rectangles with the highest shooting percentages (62% and 60%). In particular, consider the rectangle coloured in dark red, which ranges from very close to the basket until outside the 3-point line in the player's left corner. It means that the opponent defence has to be very worried in not letting him shoot from beyond the 3-point line in his left corner, because his shooting performance from this position is similar to his performance close to the basket, but he is even more dangerous, since it is a 3-point shot. Another evidence confirmed is the fact that he performs really well when attempting 3-point shots from the wings, but the graph shows also one more spot from where he shoots with confidence (even more than from the wings): the light red rectangle in the middle that ranges from above the free throw line until beyond the 3-point line. This latter zone was not identified by the SPG with predetermined areas and this is a proof of how this analysis can help in catching more details.

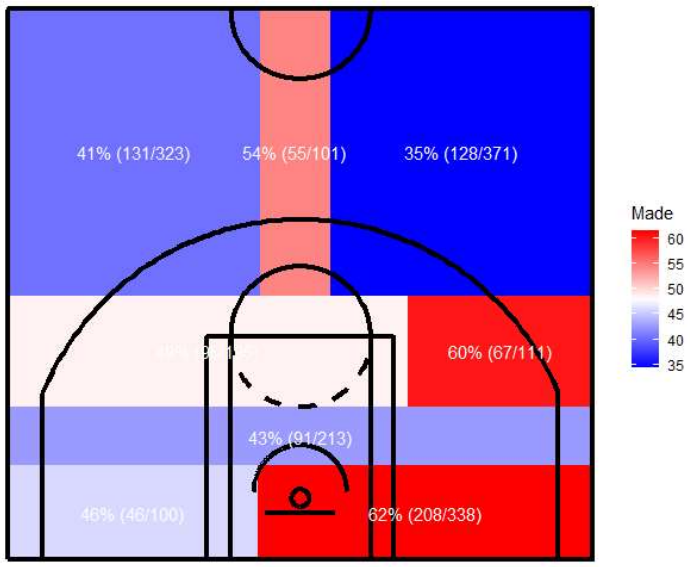

Figure 5: Partition of the court induced by the classification tree, with rectangles coloured according to shooting percentage - Stephen Curry, NBA regular season and playoff 2018/2019

3.3.2 SHOOTING PERCENTAGE PREDICTION WITH RANDOM FORESTS AND GRADIENT BOOSTING

I extended the previously illustrated analysis proposing the use of random forests and gradient boosting for predicting shooting percentages in all possible positions in the court, in order to produce a further SPG. For both algorithms I use, as in the analysis with CART, the space coordinates of the court as predictors and the outcome of the shot as response variable. In order

to classify each shot as made or missed, boosting and random forests estimate the probability for each observation of belonging to each class of the response variable: if the estimated probability that a shot scores a basket given the space coordinates is greater than 50%, then the shot is assigned to the class "made", otherwise it is assigned to the class "missed". Therefore, the probability that a shot belongs to the class "made" can be interpreted as an estimate of shooting percentage (or scoring probability) from that specific position of the court. It is possible to exploit these results for displaying a court map where each point is coloured according to its estimated shooting percentage. In this way, unlike the court partitioning with classification trees, areas of the court which are homogeneous with respect to shooting percentage are not constrained to be rectangles, but they are free-shaped. Also in this case, the partition of the court is unique for each player (or team) and the areas are not predetermined.

Again, Stephen Curry's shots are analysed as case study. Before fitting the models, the shots attempted from beyond the half court line are removed, in order not to let them affect the estimates in a bad way, and the sample is randomly divided into training and test set for assessing the models' performances. The training set is composed by 1314 observations (75% of the whole sample) and the test set by 438 observations (25% of the whole sample).

First of all, consider random forests. I fit the model on the training set with the following tuning parameters: number of trees equal to $ntree = 500$ and number of variables candidates for splitting equal to $mtry = 1$. The former means that 500 trees are built and averaged in order to make predictions, whereas the latter is set to 1, because the model has only 2 predictors, then the only way to reduce the correlation between trees is to try only one variable at each split. In a preliminary analysis, I fitted the model the model with $ntree = 200$ and $ntree = 1000$. The obtained results were very similar (but slightly worse) than in the presented model, then I decided to set $ntree = 500$, which is the default value of function randomForest of R package randomForest. Successively, the prediction power of the model is tested both on training and test sets and the function randomForest computes also the out-of-bag estimate of the error rate. The prediction accuracy is computed as the ratio between well classified observations and the number of observations in the sample. Results are summarized in table 1.

Training set	Made	Missed
Made	619	17
Missed	3	675
Accuracy	98.5%	

Test set	Made	Missed
Made	96	99
Missed	103	140
Accuracy	53.9%	

Out-of-bag sample	Made	Missed
Made	299	323
Missed	300	392
Accuracy	52.6%	

Table 1: Confusion matrices and accuracy for random forests - Stephen Curry, NBA regular season and playoff 2018/2019

Observing the table, there are two main facts to notice. The first one is the coherence between the accuracy computed in the test set and in the out-of-bag sample, which show very similar values. The second (and most relevant) one is the large difference between the accuracy in training and test set. Since the model shows an outstanding prediction performance in the training set and a poor one in the test set, this is a clear evidence of overfitting. This is reflected also in the court map that will be illustrated later.

Now, consider gradient boosting. I fit the model on the training set and I used the function train of R package caret to optimize the tuning parameters by 10-fold cross-validation. The optimal tuning parameters obtained in this way are the number of iterations equals to $nround = 150$, the size of constituent trees equals to $maxdepth = 1$ and the shrinkage parameter equals to $eta = 0.4$ (it corresponds to ν in the methodological section). The meaning is that 150 trees are summed up in order to minimize the prediction error, each tree admits only one binary split (trees do not grow deeply to prevent overfitting) and the contribution of each tree is scaled of a

factor 0.4 (again, for preventing overfitting). The results regarding confusion matrices and accuracy are shown in table 2.

Training set	Made	Missed
Made	311	197
Missed	311	495
Accuracy	61.3%	

Test set	Made	Missed
Made	83	70
Missed	116	169
Accuracy	57.5%	

Table 2: Confusion matrices and accuracy for gradient boosting - Stephen Curry, NBA regular season and playoff 2018/2019

In this case, the prediction accuracy in training set is only slightly better than in test set. It is an evidence of how good gradient boosting is in preventing overfitting. Moreover, it achieves a prediction accuracy of 57.5% in test set, which is a very good result and it is higher compared to random forests. For these reasons, it is possible to assert that gradient boosting performs better than random forests in this specific task.

As explained before, random forests and gradient boosting produce an estimate of the probability that an attempted shot scores a basket, given the position in the court, for determining the classification of shots as made or missed. I exploited this property to obtain an estimate of the scoring probability in every possible point of the court. Therefore, I created a dataset with new observations, which represent shots from every possible position on the court: the space coordinates x and y are assigned to each sample unit and it is generated one data point for each possible couple of values of x and y. Successively, I used the previous descripted models to compute the scoring probability for each new data point. Finally, I displayed these results in a court map, which I propose as SPG. The shots (data points) are divided in four groups according to their scoring probability: 0% - 30% (low percentage shot), 30% - 45%

(medium-low percentage shot), 45% - 60% (medium-high percentage shot), 60% - 100% (high percentage shot) and each point on the map is coloured according to the group it belongs to. The values of scoring probabilities for determining the groups are chosen according to pragmatic considerations about shooting percentages in basketball. The court map obtained fitting random forests is displayed in figure 6, while figure 7 shows the one obtained fitting gradient boosting.

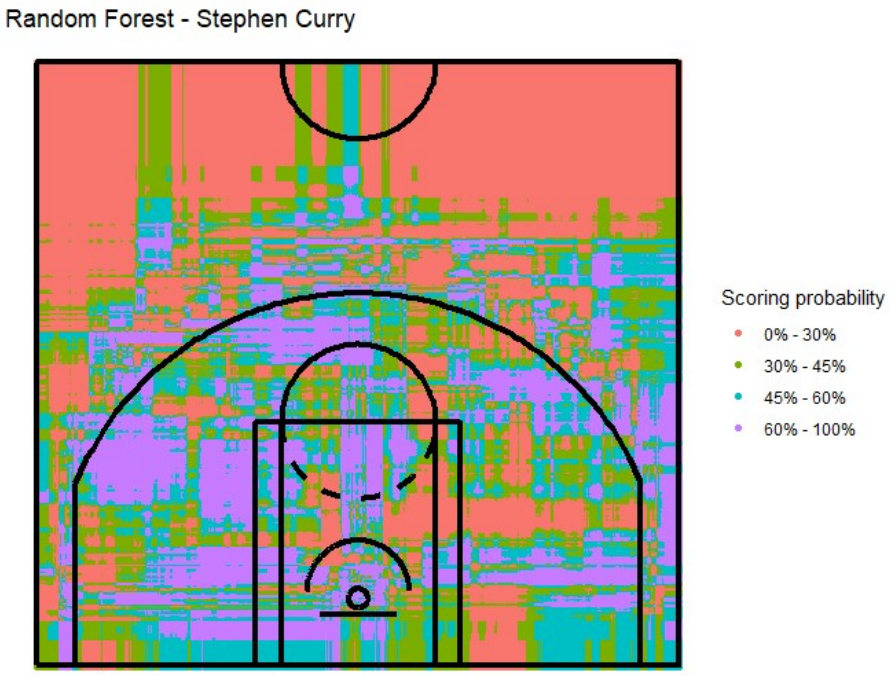

Figure 6: Partition of the court obtained by estimating scoring probabilities with random forests - Stephen Curry, NBA regular season and playoff 2018/2019

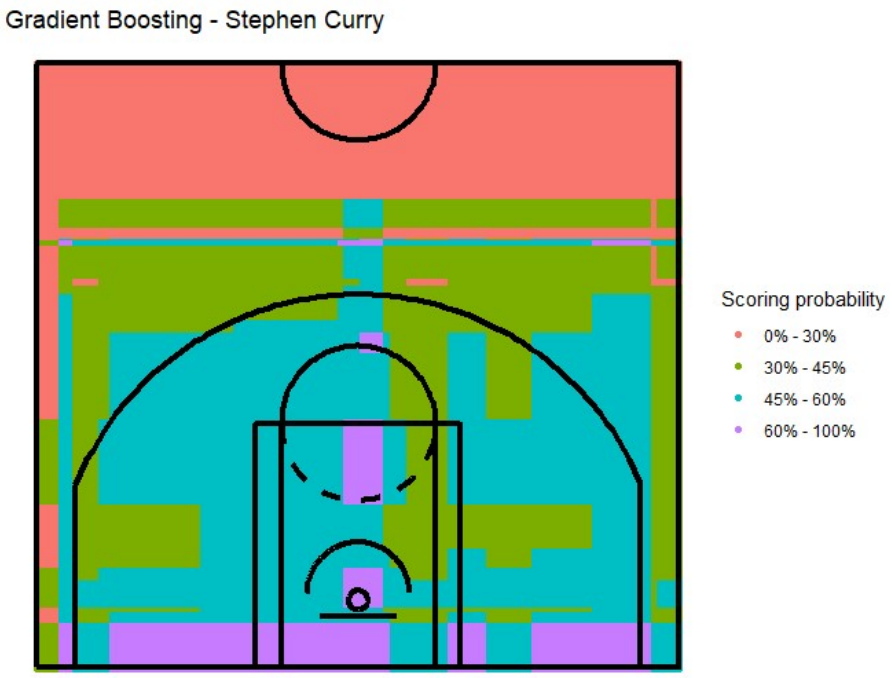

Figure 7: Partition of the court obtained by estimating scoring probabilities with gradient boosting - Stephen Curry, NBA regular season and playoff 2018/2019

Analysing the graphs, it is straightforward to notice the effect of overfitting in random forests: there are several zones of the court with estimated scoring probability over 60%, even at 9 or 10 metres away from the basket and it is obviously unrealistic. Moreover, all the area near the half court line should have low scoring probability (0% - 30%), but there are areas with higher scoring probability estimates. On the contrary, the graph obtained with gradient boosting shows more conservative estimates of the scoring probability: there are few areas characterized by scoring probability over 60% and the zone around the half court is estimated to have scoring probability under 30%. The main problem is the high scoring probability associated to areas very close to the baseline, because it is generally very rare to shoot from there and scoring probability should be low. It is due to the algorithm, which interprets the position as close to high shooting percentage areas such as close to the basket or 3-point shots from the corners. Anyway, both graphs are useful for the evaluation of player's shooting ability and to highlight the favourite spots. In fact, many evidences that came out in the previous analyses are confirmed. The first thing to notice is Curry's good performance in 3-point shots from the wings, from the middle and from the corners, which are coloured in violet and blue in the graph for random forests and in blue in the graph for gradient boosting. His shooting ability from

middle-distance is quite homogeneous from left to right, whereas from inside the painted area he performs better from his right. Furthermore, since the area coloured in red (low percentage shots) ranges from about 10 metres from the basket to the half court line in both SPGs, this is another evidence of Curry's very long shooting range. In conclusion, this SPG can be a very useful tool for evaluating players' shooting ability. However it is important to remember that the algorithms try to predict the outcome of virtual shots basing on existing shots, then it reflects the player's performance in real life, but the presence of outliers (e.g.: shots from near the half court line) or the absence of shots from some positions (e.g.: very close to the baseline) can lead to some errors in the estimation of scoring probabilities.

In conclusion, predicting the outcome of a shot is a very difficult task, because there are many factors that play an important role in influencing the shot result and often, they are not measurable (e.g.: the player is off balance when attempting the shot) or they are not available in play-by-play data (e.g.: time that ball has been in player's hands before shooting). However, the fact that the prediction made basing on the position in the court is over 50% accurate for both random forests and gradient boosting is an evidence that the spatial position is one of the most determinant factors. Furthermore, when the aim is to predict the outcome of future shots in order to produce a court map, the position on the court is the only available information.

3.3.3 COMPARING PLAYERS' SHOOTING PERFORMANCES

The tools proposed in this chapter are useful for evaluating players' shooting performances, in order to define game strategies (designing offensive situations that aim to make players attempt shots from their favourite spots or understanding where the opponent's players can be more dangerous and how to limit them) and specific training programmes for each player. Another possible application is the comparison between players, which is essential in scouting when the coaching staff has to choose the players that have the best fit with the team, in order to sign them. In this section, I focus on the latter and I decided to compare the shooting performances of Stephen Curry, Damian Lillard and Luka Doncic. It is fundamental to remind that the aim of this analysis is not to determine which of the three players is the best, but it is the evaluation and the comparison of them only in terms of shooting performance, that is a small part of basketball. I selected Lillard and Doncic for making a comparison with Curry for many reasons. First of all, they play the same role: playmaker or point guard as traditional role and primary shot creator as advanced role, which means that they often have the ball in their hands and they

are very good in scoring themselves and making their teammates score. Consequently, these three players attempt a large number of shots per game, which permits to have a more effective analysis of their shooting performance. Moreover, Curry and Lillard can be considered very similar players, because of their style of playing, their size (Curry stands 1.91 metres and weighs 86 kilograms, while Lillard stands 1.85 metres and weighs 89 kilograms) and their experience background (2018/2019 NBA season was the tenth in Curry's career and the seventh in Lillard's career). Another interesting common aspect between the two players is that they played in excellent teams in season 2018/2019: Portland Trail Blazers (Lillard's team) finished the season at the second place of Western Conference, while Golden State Warriors (Curry's team) finished at the first place of Western Conference and at second place of all NBA. On the contrary, Luka Doncic has big size compared to players in his role (he stands 2.01 metres and weighs 104 kilograms), 2018/2019 season was his first in NBA (he was born in 1999, whereas Lillard was born in 1990 and Curry in 1988) and his team (Dallas Mavericks) did not qualify to playoff that year. Consequently, there are less data about Doncic (72 games played), with respect to Curry (91 games played) and Lillard (96 games played). Furthermore, Doncic played three years in Europe at Real Madrid before going to NBA, while Curry and Lillard played only in USA.

The first step of the analysis consists in displaying the shot chart with made and missed shots, the density plots and the SPG with the court divided in predetermined areas, in order to identify players' shooting patterns and favourite spots and to remark the differences between Curry, Doncic and Lillard. Figure 8 and Figure 9 show the results respectively for Lillard and Doncic; for Curry see figure 1, figure 2 and figure 3.

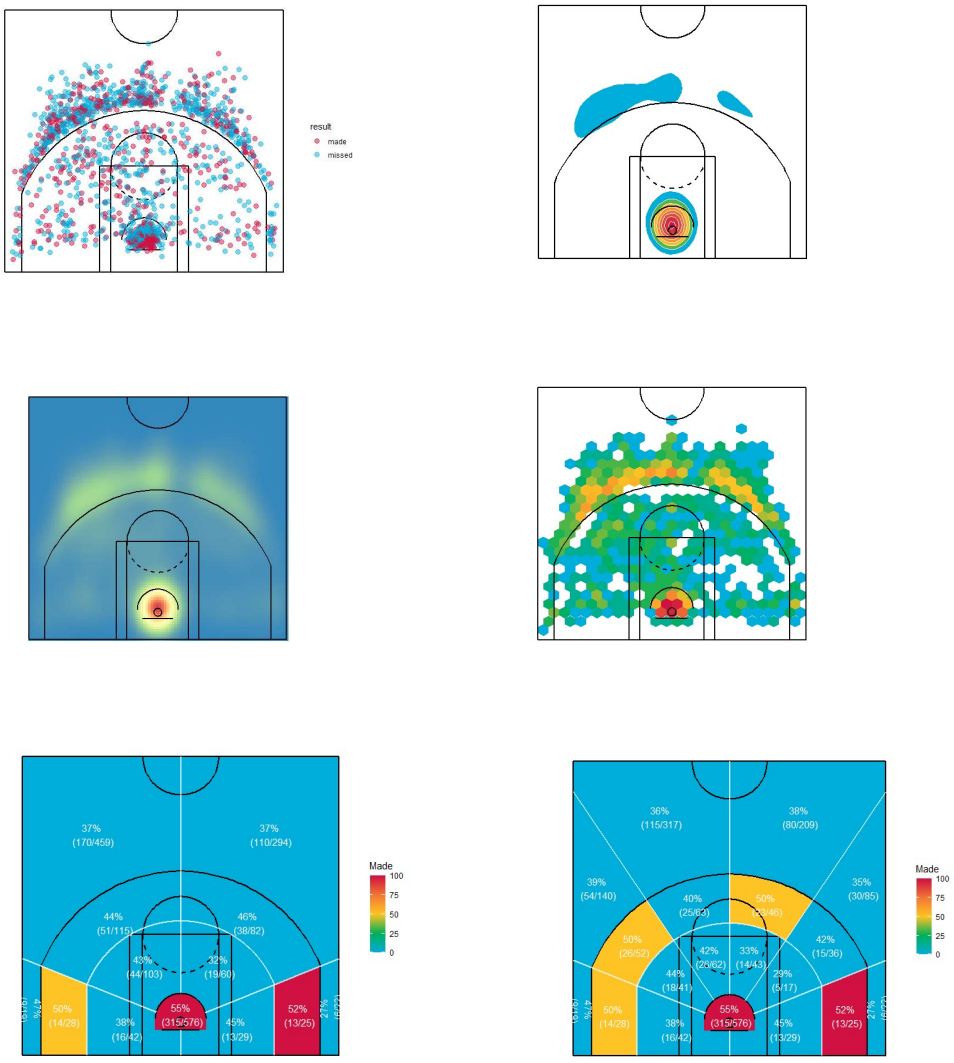

Figure 8: Shot chart with made and missed shots (top-left), density plots with polygons (top-right), raster (middle-left), hexbins (middle-right), SPGs with 4 sectors (bottom-left) and 6 sectors (bottom-right) and areas coloured according to shooting percentages - Damian Lillard, NBA regular season and playoff 2018/2019

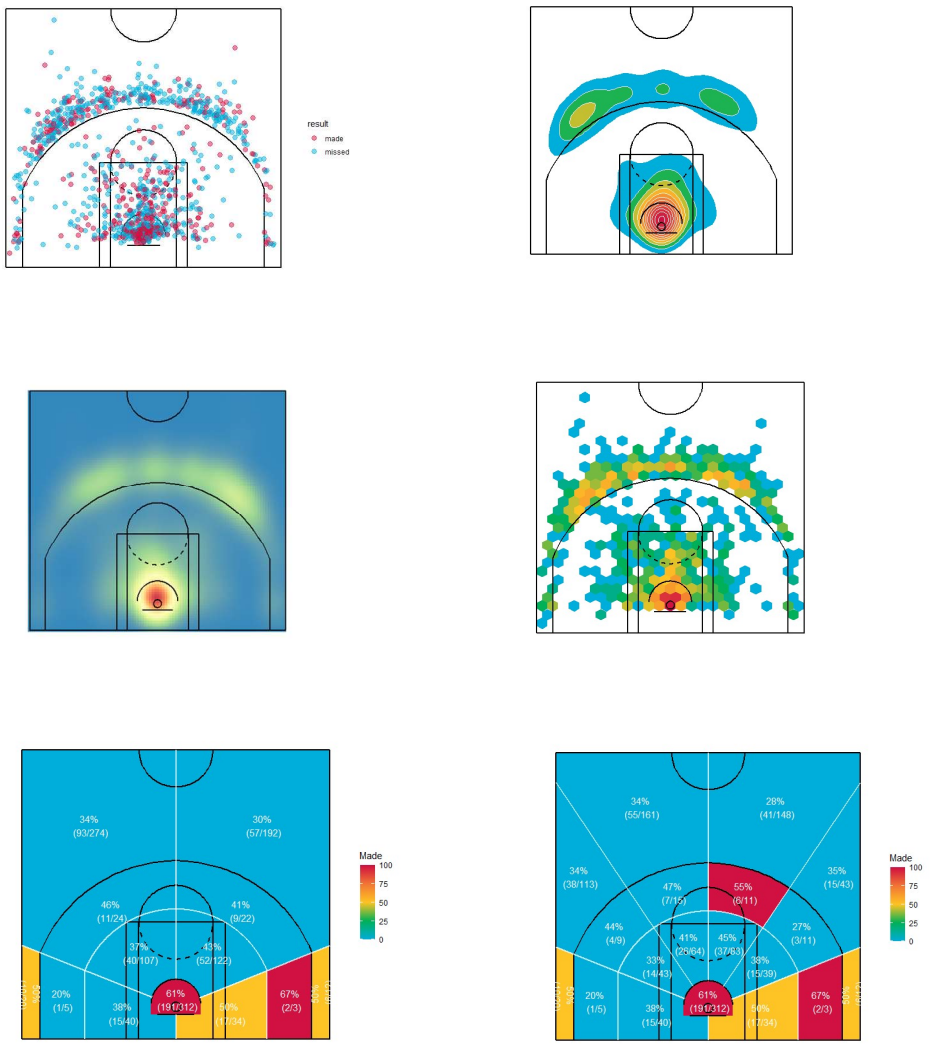

Figure 9: Shot chart with made and missed shots (top-left), density plots with polygons (top-right), raster (middle-left), hexbins (middle-right), SPGs with 4 sectors (bottom-left) and 6 sectors (bottom-right) and areas coloured according to shooting percentages – Luka Doncic, NBA regular season 2018/2019

By this analysis many evidences come out. A common tendency for the three players consists in attempting the most part of their shots from beyond the 3-point line or close to the basket, while shots from middle distance are rarely attempted: this trend is particularly extreme for Luka Doncic. Then, they show similar performances in the restricted area, where Lillard has the worst shooting percentage (55%), but with the highest number of attempts (576). Curry and Lillard frequently shoot from very long distance, in particular the nickname “Logo Lillard” was

assigned to him because of this peculiarity (there is a sticker with the NBA logo attached on the court at beyond 10 metres from the basket). Doncic and Lillard take very few 3-point shots from the corners than from the wings or from the middle, although the former shows a good performance from that position. Lillard shoots more and better from his right than from his left both for 3-point and middle-distance shots, while Curry is more efficient from his right for 3-point shots and from his left from middle-distance. Doncic's shooting performance is quite homogeneous throughout the court, with some peaks such as the area along the baseline on his left (but with few attempted shots) and the restricted area. Looking at the density plots, it is possible to notice the most interesting evidence in this analysis: Doncic attempts shots from inside the so-called painted area (the rectangle that ranges from the free throw line to the baseline) more frequently than Curry and Lillard. A reasonable explanation is the bigger size: often, Doncic has a physical advantage with respect to his direct defender, then he tries to exploit it going closer to the basket when attempting shots. In general, Curry's shooting performance seems to be the best, because he shoots with higher percentages from almost all predetermined areas in the court. Apart from his undiscussed ability (several experts consider him the best shooter in history), there could be also another factor that conditioned his performance: the team where he plays. Golden State Warriors had Klay Thompson and Kevin Durant in their roster in season 2018/2019, which are very good offensive players. Then, it was surely difficult for the opponents to always pays attention in perfectly guarding all three players, which means that Curry could have more space to shoot in many situations. Considering Lillard, Portland Trail Blazers were a good team, but they had two main offensive threats (not three like Golden State): Lillard and McCollum, consequently it is probable that the former took more contested and difficult shots than Curry. The situation is slightly different for Doncic: he was the best player of his team at his first year in NBA and Dallas Mavericks were not a good team, then his inexperience and the absence of another very good offensive player may have conditioned his shooting performance.

The next step of the analysis consists in comparing the shooting performance of the three players with respect to their own partition of the court induced by fitting the classification tree as describe in section 2.4.1. Before fitting the model, the shots attempted beyond the half court are removed. In preliminary analyses, I set parameter $minsplit = 300$ for Lillard, because of the high number of attempted shots, but a partition of the court with only 3 areas was produced and I want to obtain a more detailed picture of player's shooting performance. Then, I reduced the value of $minsplit$ to 150 to find a more informative court partition. Considering Doncic,

since the number of attempted shots is lower, I set $minsplit = 150$ and 3 terminal nodes are obtained. However, decreasing $minsplit$ to 100 or even 80, produces only one more small area that does not improve the information. Therefore, for both Luka Doncic and Damian Lillard a classification tree model is fitted with complexity parameter $cp = 0.005$ and minimum number of observations in a node for having a split equal to $minsplit = 150$. The obtained classification trees and the court maps with their own partitions are displayed in figure 10, figure 11, figure 12 and figure 13.

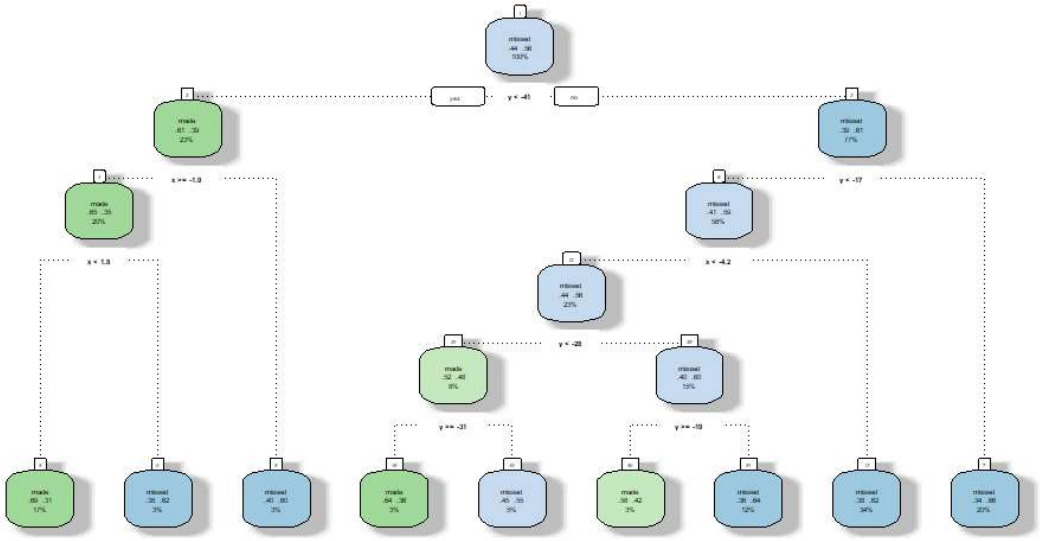

Figure 10: Classification tree - Damian Lillard, NBA regular season and playoff 2018/2019

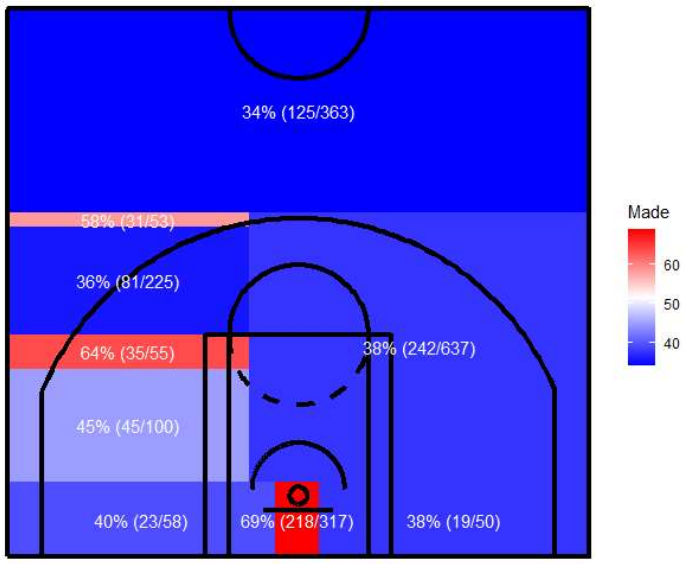

Figure 11: Partition of the court induced by the classification tree, with rectangles coloured according to shooting percentage (bottom) - Damian Lillard, NBA regular season and playoff 2018/2019

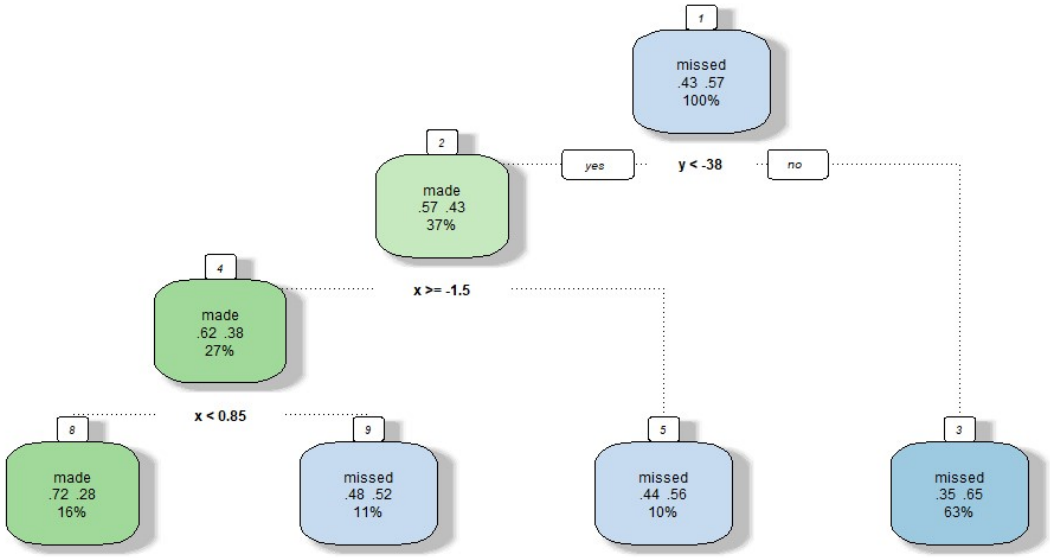

Figure 12: Classification tree – Luka Doncic, NBA regular season 2018/2019

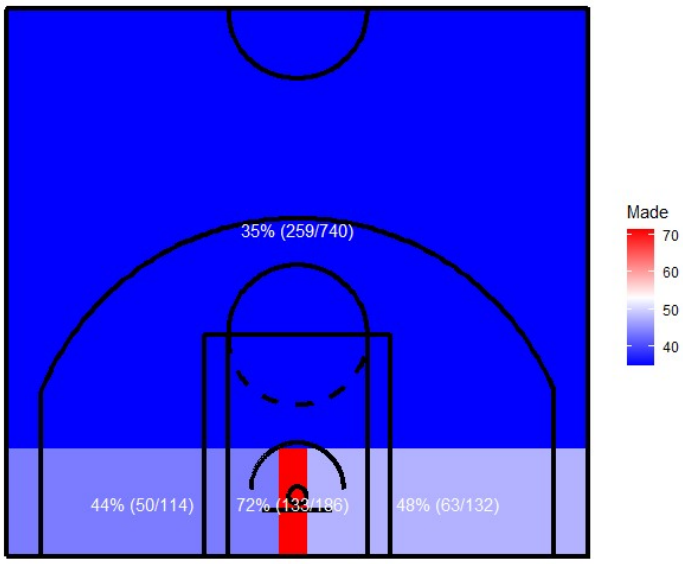

Figure 13: Partition of the court induced by the classification tree, with rectangles coloured according to shooting percentage (bottom) – Luka Doncic, NBA regular season 2018/2019

Considering Damian Lillard, his own partition of the court underlines the evidence that he performs better when he shoots from his right. In particular he has good shooting percentages in the red rectangle that ranges from the elbow to the sideline, both for 3-point and middle distance shots, and in the light red rectangle beyond the 3-point line in middle-right zone. Another interesting aspect regards his shooting performance in the restricted area, where he is accurate from very close to the basket, but his efficacy is definitely lower near the arc. This latter detail did not come out analysing predetermined areas on the court. For what concerns Luka Doncic, the evidence that his shooting performance is quite homogeneous throughout the court is confirmed: his own partition is composed by only four areas. The red rectangle close to the basket highlights his ability in shooting in the restricted area and again, his bigger size with respect to direct defenders should help him in performing that good. Comparing the court's partitions of Curry, Doncic and Lillard, it is straightforward to notice that Curry's shooting performance is the best one. His partition presents more and larger red rectangles, which show that he shoots with confidence in more spots than Lillard and Doncic. Consequently, it should be more difficult for opponents to make him shoot from zones where he performs worse.

Finally, the last phase of the analysis consists in fitting random forests and gradient boosting for estimating scoring probabilities in every point of the court and displaying the result in a court map. Before fitting the models, the sample is randomly divided into training set (75% of the sample) and test set (25% of the sample), in order to assess the performance of the models. Then, analysing Lillard the training set is composed by 1393 shots and the test set by 465 shots, whereas for Doncic there are 879 shots in the training set and 293 shots in the test set.

First of all, consider random forests. For both players I fitted the model on the training set with these tuning parameters: number of trees equals to $ntree = 500$ and number of variables candidates for splitting equals to $mtry = 1$. For what concerns gradient boosting, I optimized the tuning parameters by 10-fold cross-validation with the function `train` of R package `caret` and the obtained values are the same the two players. The model fitted on Lillard's and Doncic's training sets have the following optimal parameters: number of iterations equals to $nround = 50$, size of constituent trees equals to $maxdepth = 1$ and shrinkage parameter equals to $eta = 0.3$. The confusion matrices and the measures of accuracy are reported in table 3, table 4, table 5 and table 6 below.

Training set	Made	Missed
Made	589	22
Missed	22	760
Accuracy	96.8%	

Test set	Made	Missed
Made	93	99
Missed	115	158
Accuracy	54.0%	

Out-of-bag sample	Made	Missed
Made	292	319
Missed	287	495
Accuracy	56.5%	

Table 3: Confusion matrices and accuracy for random forests – Damian Lillard, NBA regular season and playoff 2018/2019

Training set	Made	Missed
Made	183	102
Missed	428	680
Accuracy	62.0%	

Test set	Made	Missed
Made	69	46
Missed	139	211
Accuracy	60.2%	

Table 4: Confusion matrices and accuracy for gradient boosting – Damian Lillard, NBA regular season and playoff 2018/2019

Training set	Made	Missed
Made	370	10
Missed	4	495
Accuracy	98.4%	

Test set	Made	Missed
Made	60	36
Missed	71	126
Accuracy	63.5%	

Out-of-bag sample	Made	Missed
Made	163	211
Missed	160	345
Accuracy	57.8%	

Table 5: Confusion matrices and accuracy for random forests – Luka Doncic, NBA regular season 2018/2019

Training set	Made	Missed
Made	140	100
Missed	234	405
Accuracy	62.0%	

Test set	Made	Missed
Made	55	28
Missed	76	134
Accuracy	64.5%	

Table 6: Confusion matrices and accuracy for gradient boosting – Luka Doncic, NBA regular season 2018/2019

The model performances are similar to the ones obtained analysing Curry. Random forests clearly overfit the data: nearly perfect accuracy in making prediction in the training set, but the model performance definitely deteriorates for predictions in test set and on out-of-bag sample. Gradient boosting's performance is more stable for both training and test sets. The main thing to notice is a bit of underfitting in the model fitted for Doncic's data, since the prediction accuracy is slightly higher in test set (64.5%) than in training set (62.0%). However, the prediction accuracy is higher in the models for Lillard and Doncic, than in models for Curry and accuracy values over 60% show a good model performance. This underlines how determinant is the position of the court in influencing the outcome of shots, but a further consideration has to be done: observing the confusion matrices, it is clear that both gradient boosting and random forests have more difficulties in predicting made shots than missed shots. Since for both Lillard and Doncic the overall number of missed shots is higher than the number of made shots, it is reasonable that models' prediction performance improves.

Successively, the dataset with new observations, which represent shots from every possible position on the court is created and the previous described models are used to predict the outcome of those shots. The probabilities of each observation of belonging to class "made" are computed and can be interpreted as estimated scoring probabilities. These scoring probabilities are divided in four groups as described in section 2.4.2, in order to produce a partition of the

court in areas that are not constrained to be rectangles. The SPGs obtained in this way for Lillard and Doncic are displayed respectively in figure 14 and figure 15.

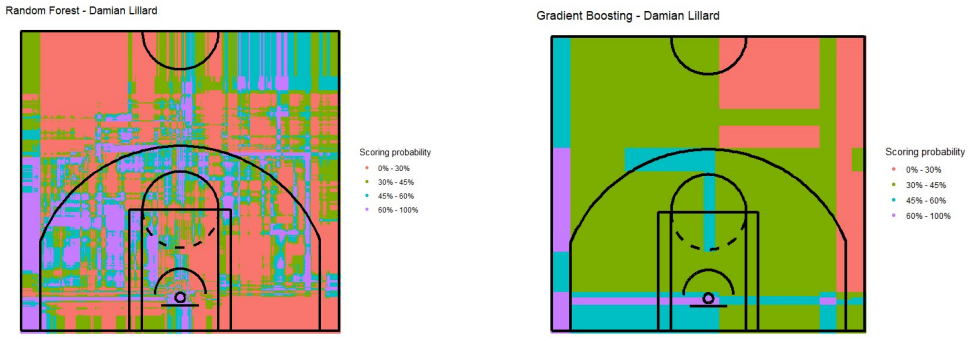

Figure 14: Partition of the court obtained by estimating scoring probabilities with random forests (left) and gradient boosting (right) – Damian Lillard, NBA regular season and playoff 2018/2019

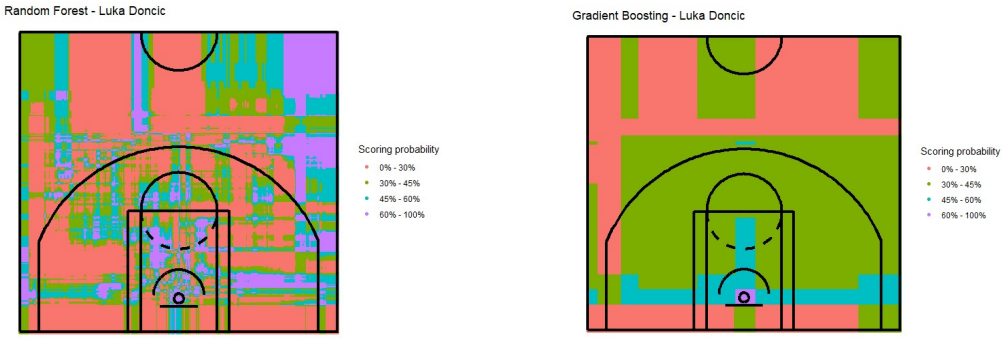

Figure 15: Partition of the court obtained by estimating scoring probabilities with random forests (left) and gradient boosting (right) – Luka Doncic, NBA regular season 2018/2019

The effects of overfitting for random forests are straightforward to be noticed in the court maps of both players, because of the presence of areas coloured in violet (estimated scoring probability over 60%) and in blue (estimated scoring probability between 45% and 60%) close to half court line. Even for gradient boosting, the zone near half court is not totally coloured in

red, but the estimates of scoring probabilities look more realistic. These SPGs confirm several evidences from the previous analyses: the peculiarity of Lillard of shooting better from his right, the ability of Doncic in shooting from inside the painted area thanks to his big size and also his homogeneous shooting performance throughout the court, except for a slightly better performance from the area along the baseline (same result of the analysis with classification trees). Moreover, these court maps highlight the longer shooting range of Curry and Lillard with respect to Doncic, since their red area is nearer to the half court line. In general, the portion of the court coloured in blue and violet is larger in Curry's court map and this is another proof that his shooting performance is definitely better compared to the ones of Doncic and Lillard.

In conclusion, these analyses prove the efficacy of the proposed tools in evaluating and comparing players' shooting performances. In this sense, data visualizations are fundamental, because they allow fast and easy interpretations of the results, even understandable by a non-technical audience. In real life applications, this is a necessary condition for an effective communication between analysts and coaching staff.

4. CONCLUSIONS

Evaluating the shooting performance is a relevant topic in basketball analytics. In this thesis, I examined the use of different machine learning methods to assess the shooting performance of players or teams in the basketball court by means of the production of graphical representations. I presented various analyses, starting from the simpler one and proceeding with others, which are deeper and more complex under the statistical point of view. First of all, I descripted the basic graphical tools that are commonly used by analysts. One of such graphical tools permits to study players' or teams' shooting performances in predetermined area of the court. Consequently, the key idea is to overcome the previous analysis by evaluating the shooting performance in areas that are not determined in advance, but that are obtained basing on data. In this way, every player or team has his own partition of the court. Then, I replicated an analysis proposed by Paola Zuccolotto, Marica Manisera and Marco Sandri in the paper *Basketball spatial performance indicators and graph*, where the use of classification trees is proposed to obtain an optimal partition of the court in rectangles, such that the shooting performance is maximally homogeneous within them. Finally, I extended the latter analysis proposing the application of random forests and gradient boosting for predicting the scoring probability in every point of the court and exploiting the obtained results to produce a court map with areas coloured according to the predicted scoring probability. These areas are not constrained to be rectangles as in the analysis with classification trees, but they are free-shaped. Considering the last analysis, it resulted that gradient boosting performs better than random forests for this specific task, because the latter method clearly overfits the data.

All the proposed tools can be extremely useful for the coaching staff for the definition of game strategies: identifying the best shooting spots for each player and developing offensive schemes that aim to conclude with well-built shots from those spots, but also observing the opponents and developing a defensive strategy with the objective to deny easy shots from their favourite areas. Moreover, the development of specific training programmes for each player and the comparison between players in scouting are other possible applications. In particular, I presented an example of the latter, where the shooting performances of three NBA players (Stephen Curry, Damian Lillard and Luka Doncic) are compared.

The proposed case studies are based on play-by-play data, which are a particular type of data that describes the main events happened during a basketball match. A lot of knowledge can be gathered from play-by-play datasets, however there is not information about players' movement with and without the ball in their hands, which can be a determinant factor for the evaluation of

their shooting performance and for predicting shot outcomes. Several machine learning, computer vision and statistical methods can be proposed to solve many problems in basketball analytics basing on the so-called tracking data, which are detected by technological devices (GPS sensors, cameras, wearable technologies etc.) and contain information about players' movement. Therefore, for future research, it would be interesting to perform analyses for the shooting performance evaluation in the basketball court basing on the information provided by tracking data.

ACKNOWLEDGEMENTS

This thesis represents the last step of a long and hard, but beautiful, journey in the world of Statistics started in September 2014. No citation is more appropriate than the famous “no pain, no gain”. Now, the satisfaction is extreme and I am definitely conscious that I could have never done a better choice in my life than specializing in this fascinating and amazing subject.

However, nothing of that would have been possible without the support of all the following people. First of all, thanks to professor Stefania Mignani for having been my supervisor for the thesis, both for the Bachelor and for the Master, and for having been my guide at university. Thanks to professors Paola Zuccolotto and Marica Manisera for having giving me the opportunity to combine my passions, Statistics and basketball, during the beautiful internship experience in Brescia and in writing this thesis. Thanks to all the friends of mine for having always been close to me. Thanks to my parents, Roberta and Massimiliano, to my grandparents, Roberto, Iris, Silvana and Mario, and to all my family for having been patient with me and for having always believed in me, even in the most difficult moments. Finally, thanks to Aneliya for supporting me every day and for having changed my life. I am sure that you are the main reason of my graduation. I love you.

Gianluca Piromallo

REFERENCES

Alagappan, M., 2012. From 5 to 13: Redefining the positions in basketball, in: 2012 MIT Sloan Sports Analytics Conference. http://www.sloansportsconference.com.

Annis, D. H. 2006. Optimal End-Game Strategy in Basketball. Journal of Quantitative Analysis in Sports 2, 1.

Ante, P., Slavko, T., Igor, J., 2014. Interdependencies between defence and offence in basketball. Sport Science 7, 62–66.

Avugos, S., Köppen, J., Czienskowski, U., Raab, M., Bar-Eli, M., 2013. The "hot hand" reconsidered: A meta-analytic approach. Psychology of Sport and Exercise 14, 21–27.

Bianchi, F., Facchinetti, T., Zuccolotto, P., 2017. Role revolution: towards a new meaning of positions in basketball. Electronic Journal of Applied Statistical Analysis 10, 712–734.

Bornn, L., Cervone, D., Franks, A., Miller, A., 2017. Studying basketball through the lens of player tracking data, in: Handbook of Statistical Methods and Analyses in Sports. Chapman and Hall/CRC, pp. 245–269.

Breiman, L., Friedman, J.H., Stone, C.J., Olshen, R.A., 1984. Classification and regression trees. CRC press.

Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32.

Brown, M. and J. Sokol. 2010. An Improved LRMC Method for NCAA Basketball Prediction. Journal of Quantitative Analysis in Sports 6, 1–23.

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, 585–599.

Clemente, F.M., Martins, F.M.L., Kalamaras, D., Mendes, R.S., 2015. Network analysis in basketball: inspecting the prominent players using centrality metrics. Journal of Physical Education and Sport 15, 212.

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, 51–72.

Engelmann, J., 2017. Possession-based player performance analysis in basketball (adjusted +/– and related concepts), in: Handbook of Statistical Methods and Analyses in Sports. Chapman and Hall/CRC, pp. 215–227.

Erčulj, F., Štrumbelj, E., 2015. Basketball shot types and shot success in different levels of competitive basketball. PloS one 10, e0128885.

Fearnhead, P., Taylor, B.M., 2011. On estimating the ability of NBA players. Journal of Quantitative Analysis in Sports 7.

Fewell, J.H., Armbruster, D., Ingraham, J., Petersen, A., Waters, J.S., 2012. Basketball teams as strategic networks. PloS one 7, e47445.

Franks, A.M., D'Amour, A., Cervone, D., Bornn, L., 2016. Meta-analytics: tools for understanding the statistical properties of sports metrics. Journal of Quantitative Analysis in Sports 12, 151–165.

Gabel, A., Redner, S., 2012. Random walk picture of basketball scoring. Journal of Quantitative Analysis in Sports 8, 1416.

García, J., Ibáñez, S.J., De Santos, R.M., Leite, N., Sampaio, J., 2013. Identifying basketball performance indicators in regular season and playoff games. Journal of Human Kinetics 36, 161–168.

Gudmundsson, J., Horton, M., 2016. Spatio-temporal analysis of team sports. a survey. arXiv preprint arXiv:1602.06994 .

Gupta, A.A., 2015. A new approach to bracket prediction in the NCAA men's basketball tournament based on a dual-proportion likelihood. Journal of Quantitative Analysis in Sports 11, 53–67.

Hastie, T., Tibshirani, R., Friedman, J., H., 2001, The Elements of Statistical Learning. Springer.

Han, J., Kamber, M., Pei, J., 2000. Data Mining. Concepts and Techniques. Morgan Kaufmann.

Koh, K.T., Wang, C.K.J., Mallett, C., 2011. Discriminating factors between successful and unsuccessful teams: A case study in elite youth Olympic basketball games. Journal of Quantitative Analysis in Sports 7.

Koh, K.T., Wang, C.K.J., Mallett, C., 2012. Discriminating factors between successful and unsuccessful elite youth Olympic female basketball teams. International Journal of Performance Analysis in Sport 12, 119–131.

Kubatko, J., Oliver, D., Pelton, K., Rosenbaum, D.T., 2007. A starting point for analyzing basketball statistics. Journal of Quantitative Analysis in Sports 3, 1–22.

Lamas, L., De Rose Jr., D., Santana, F.L., Rostaiser, E., Negretti, L., Ugrinowitsch, C., 2011. Space creation dynamics in basketball offence: validation and evaluation of elite teams. International Journal of Performance Analysis in Sport 11, 71–84.

Li, R., Belford, G., 2002. Instability of decision tree classification algorithms. KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 2002, Pages 570–575

Loeffelholz, B., E. Bednar, and K. W. Bauer. 2009. Predicting NBA Games using Neural Networks. Journal of Quantitative Analysis in Sports 5, 1–15.

Lopez, M.J., Matthews, G.J., 2015. Building an NCAA men's basketball predictive model and quantifying its success. Journal of Quantitative Analysis in Sports 11, 5–12.

Manisera, M., Sandri, M., Zuccolotto, P., 2019. BasketballAnalyzeR: the R package for basketball analytics., in: Conference Smart Statistics for Smart Applications, Pearson. pp. 395–402. 19th-21st June 2019.

Manner, H., 2016. Modeling and forecasting the outcomes of NBA basketball games. Journal of Quantitative Analysis in Sports 12, 31–41.

Metulini, R., Manisera, M., Zuccolotto, P., 2017a. Sensor analytics in basketball, in: Proceedings of the 6th International Conference on Mathematics in Sport.

Metulini, R., Manisera, M., Zuccolotto, P., 2017b. Space-time analysis of movements in basketball using sensor data, in: Statistics and Data Science: new challenges, new generations - SIS2017 proceeding.

Metulini, R., Manisera, M., Zuccolotto, P., 2018. Modelling the dynamic pattern of surface area in basketball and its effects on team performance. Journal of Quantitative Analysis in Sports 14, 117–130.

Miller, A.C., Bornn, L., 2017. Possession sketches: Mapping NBA strategies. MIT Sloan Sports Analytics Conference 2017.

Oliver, D., 2004. Basketball on paper: rules and tools for performance analysis. Potomac Books, Inc.

Özmen, U.M., 2012. Foreign player quota, experience and efficiency of basketball players. Journal of Quantitative Analysis in Sports 8, 1–18.

Page, G.L., Barney, B.J., McGuire, A.T., 2013. Effect of position, usage rate, and per game minutes played on NBA player production curves. Journal of Quantitative Analysis in Sports 9, 337–345.

Passos, P., Araújo, D., Volossovitch, A., 2016. Performance analysis in team sports. Taylor & Francis.

Passos, P., Davids, K., Araújo, D., Paz, N., Minguéns, J., Mendes, J., 2011. Networks as a novel tool for studying team ball sports as complex social systems. Journal of Science and Medicine in Sport 14, 170–176.

Piette, J., Pham, L., Anand, S., 2011. Evaluating basketball player performance via statistical network modeling, in: MIT Sloan Sports Anal. Conf.

Ruiz, F.J., Perez-Cruz, F., 2015. A generative model for predicting outcomes in college basketball. Journal of Quantitative Analysis in Sports 11, 39–52.

Sandri, M., forthcoming. The R package BasketballAnalyzeR. CRC Press. chapter 6. In: Zuccolotto P. and Manisera M., Basketball Data Science. With Applications in R.

Sandri, M., Zuccolotto, P., Manisera, M., 2018. BasketballAnalyzeR: an R package for the analysis of basketball data. URL: https://github.com/sndmrc/BasketballAnalyzeR.

Schumaker, R.P., Solieman, O.K., Chen, H., 2010. Sports Data Mining. Springer.

Schwarz, W., 2012. Predicting the maximum lead from final scores in basketball: A diffusion model. Journal of Quantitative Analysis in Sports 8.

Shortridge, A., Goldsberry, K., Adams, M., 2014. Creating space to shoot: quantifying spatial relative field goal efficiency in basketball. Journal of Quantitative Analysis in Sports 10, 303–313.

Skinner, B., Goldman, M., 2017. Optimal strategy in basketball, in: Handbook of Statistical Methods and Analyses in Sports. Chapman and Hall/CRC, pp. 229–244.

Travassos, B., Araújo, D., Davids, K., Esteves, P.T., Fernandes, O., 2012. Improving passing actions in team sports by developing interpersonal interactions between players. International Journal of Sports Science & Coaching 7, 677–688.

Vračar, P., Štrumbelj, E., Kononenko, I., 2016. Modeling basketball play-by-play data. Expert Systems with Applications 44, 58–66.

Wu, S., Bornn, L., 2018. Modeling offensive player movement in professional basketball. The American Statistician 72, 72–79.

Yuan, L.H., Liu, A., Yeh, A., Kaufman, A., Reece, A., Bull, P., Franks, A., Wang, S., Illushin, D., Bornn, L., 2015. A mixture-of-modelers approach to forecasting NCAA tournament outcomes. Journal of Quantitative Analysis in Sports 11, 13–27.

Zhang, T., Hu, G., Liao, Q., 2013. Analysis of offense tactics of basketball games using link prediction, in: Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on, IEEE. pp. 207–212.

Zuccolotto, P., Manisera, M., Sandri, M., 2018. Big data analytics for modeling scoring probability in basketball: The effect of shooting under high-pressure conditions. International Journal of Sports Science & Coaching 13, 569–589.

Zuccolotto, P., Manisera, M., Sandri, M., 2019. Basketball spatial performance indicators and graphs.

Zuccolotto, P., Manisera, M., 2020. Basketball Data Science. With Applications in R. CRC Press.

http://www.bigdataball.com, Data Source.