Model Trees for Identifying Exceptional Players in the NHL and NBA Draft

by

Yejia Liu

B.Sc., South China University of Technology, 2016

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2018

Approval

Name:	**Yejia Liu**
Degree:	**Master of Science (Computing Science)**
Title:	**Model Trees for Identifying Exceptional Players in the NHL and NBA Draft**
Examining Committee:	**Chair:** Jiannan Wang Assistant Professor
	Oliver Schulte Senior Supervisor Professor
	Tim Swartz Supervisor Professor
	Maxwell Libbrecht External Examiner Assistant Professor
Date Defended:	**April 10, 2018**

Abstract

Drafting players is crucial for a team's success. We describe a data-driven interpretable approach for assessing prospects in the National Hockey League and National Basketball Association. Previous approaches have built a predictive model based on player features, or derived performance predictions from comparable players. Our work develops model tree learning, which incorporates strengths of both model-based and cohort-based approaches. A model tree partitions the feature space according to the values or learned thresholds of features. Each leaf node in the tree defines a group of players, with its own regression model. Compared to a single model, the model tree forms an ensemble that increases predictive power. Compared to cohort-based approaches, the groups of comparables are discovered from the data, without requiring a similarity metric. The model tree shows better predictive performance than the actual draft order from team's decision. It can also be used to highlight strongest points of players.

Keywords: player ranking; Logistic Model Trees; M5 regression trees; National Hockey League; National Basketball Association; Spearman rank correlation

Dedication

To anyone who is interested in the player drafting through data-driven methods.

Acknowledgements

First, I want to express special appreciation and thanks to my supervisor, Dr.Oliver Schulte, for his continuous support, great patience and immense knowledge during my Master study at SFU. His encouragement inspires me to overcome challenges in my research and studies. I could not have imagined having a better mentor.

Besides my supervisor, I also want to acknowledge Dr.Tim Swartz, Dr.Maxwell Libbrecht and Dr.Jiannan Wang for being my committee members and giving insightful comments and feedback.

I feel lucky to collaborate with many wonderful people who helped me to accomplish the results in this thesis. I am grateful to all my collaborators. I also want to thank all my friends for giving me sound advice in writing and encouraging me to strive towards my goal.

Last but not least, I owe my deepest thanks to my parents. They raise me up and give me unconditional love and support to what I want to pursue. They always back me up when I suffer from depressions and setbacks. Words cannot express my gratitude for them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Overview

Drafting players is one of the most important tasks in any sport in order to build a successful team. This process can take millions of dollars and thousands of man hours [1]. In this thesis, we focus on the draft of two well-known leagues, National Hockey League (NHL) and National Basketball Association (NBA). Every year teams systematically select prospects in the Entry Draft. In the Entry Draft, players who recently become eligible to play in a league are allocated to a team. Both the NHL and NBA Entry Draft use a lottery system to determine which team gets the top picks, so that every team can have a chance to sign a superstar. In this system, the best player is expected to be the first draft pick, the second best would be the second draft pick, and so forth. However, history has shown there are many misfires in draft picks. In 2008 NHL entry draft, Nikita Filatov was the 6th overall pick, taken before the well-known Erik Karlsson (No.15), but only played 53 games and scored 6 goals in NHL. In the NBA draft, the most notorious pick belongs to the Portland Trail Blazers, who chose Sam Bowie over Michael Jordan in 1984. To draft prospects, the team often relies heavily on scouts, who may only be able to watch a player a handful of times a season. In the NHL, teams have access to rankings from the Central Scouting Services (CSS), a department to rank players for the Entry Draft a few times during the hockey season, but they also rely on reviews from scouts. To find a more effective and economical way to access draftees, many sport experts and statisticians turn to data-driven methods. In this thesis, we consider predicting player future success in the NHL and NBA based on datasets from junior leagues (colleges in NBA), then ranking them with prediction results, with the purpose of supporting draft decisions.

Previous work for analyzing NHL/NBA draft datasets mainly include regression approaches or similarity-based approaches. Regression approaches build a predictive model that takes as input a set of player features, such as demographics metrics (age, height, weight etc.) and pre-draft performance metrics (goals scored, plus-minus, shoots, minutes player etc.), and output a predicted success metric (number of games played for NHL, player efficiency rating (PER) in NBA) [29; 11]. Cohort-based approaches divide players into groups of comparables and predict future success based on the player cohort. For ex-

ample, the PCS model [37] clusters ice hockey players according to their age, height, and scoring rates. One advantage of the cohort model is that predictions can be explained by reference to similar known players, which many domain experts find intuitive. Thus, many commercial sports analytic systems, such as Sony's Hawk-Eye system, have been developed to identify groups of comparables to predict a player's future performance. Yale university has built a clustering system to classify players in NBA according to their play style *(http://sports.sites.yale.edu/clustering-nba-players)*.

In this thesis, we develop new tree models ~~to the~~ pre-draft data that achieves the best of both approaches, regression-based and similarity-based [9; 19]. Each node in the tree defines a yes or no question until a leaf is reached. Based on answers to these questions, each player is allocated to a group corresponding to a leaf. In each leaf node, a regression model is built. Figure 1.1 shows an example model tree. Compared to a single regression model, the tree defines an ensemble of regression models, based on non-linear thresholds. This increases the expressive power and predictive accuracy of the model. The tree also represents complex interactions between player features and player groups. For example, if the data indicates that players from different junior leagues are sufficiently different to warrant building distinct models, the tree can introduce a split to distinguish different leagues. ~~While compared to~~ a similarity-based model, tree construction learns groups of players from the data, without requiring the analyst to specify a similarity metric. It selects splits that increase predictive accuracy. The learned distinctions between the groups are guaranteed to be relevant to future career success. Also, the tree models create a model for each group, which allows for differentiating players from the same group.

In the NHL draft, only about half of the drafted prospects finally played a game in NHL [36], which raises an excess-zeros problem [17] to predict future success as measured by the number of NHL games played. We therefore approach prospect ranking not by directly predicting the future number of games played, but by predicting whether a prospect will play any number of games at all. In terms of machine learning, we develop a classification model rather than a regression model. We learn a logistic regression model tree, and rank players by the probability that the logistic regression model tree assigns to them playing at least one game. Intuitively, if we can be confident that a player will play at least one NHL game, we can also expect the player to play many NHL games. While ~~in~~ NBA draft, linear regression model tree is built directly since there is no such excess-zeros issue: According to our collected data, over 80% of drafted prospects played at least one game in the NBA.

Following the work of Schuckers et al. (2016) and Greene (2015) [29; 11], we evaluate the model trees ranking results by comparing it to a ranking based on the player future success, measured as the number of career games they played after 7 years for NHL players, or player efficiency rating (PER) for NBA players. The results of our experiments show tree models perform better than the actual draft pick in predicting player future performance. We also show in case studies that the feature weights learned from the data can be used

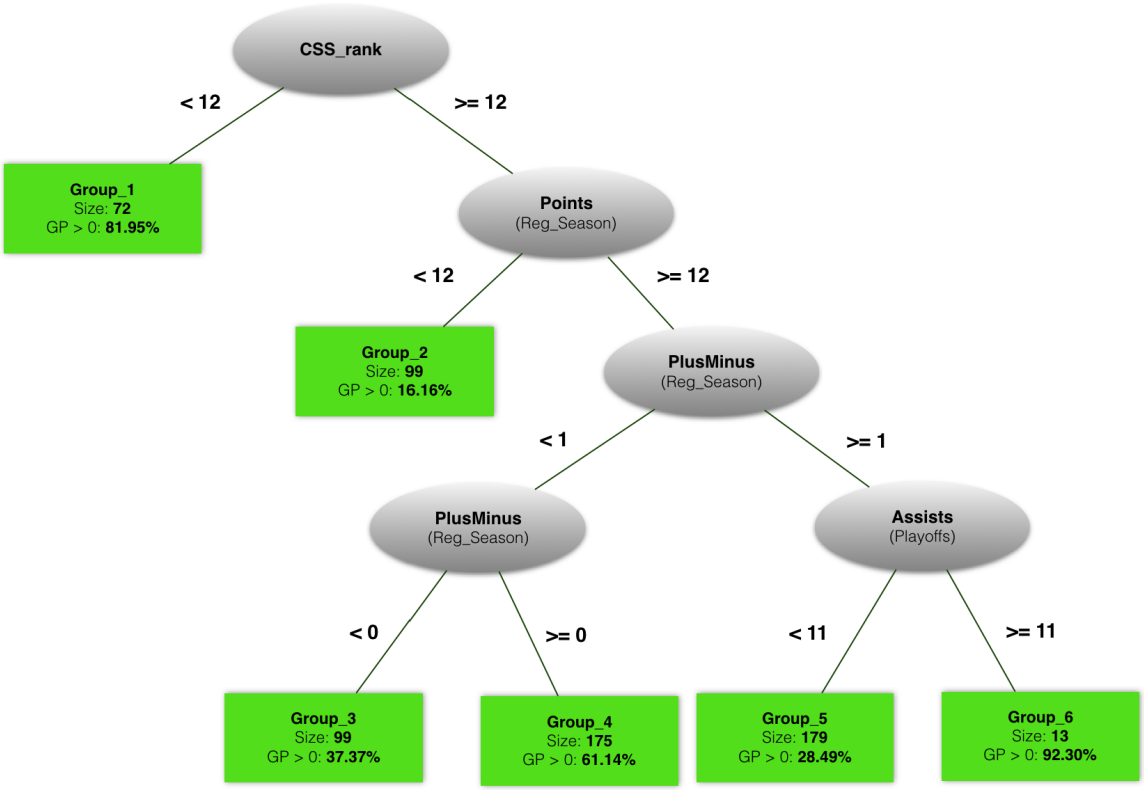

Figure 1.1: Logistic Regression Model Trees for the 2004, 2005, 2006 cohort in NHL. The tree was built using the LogitBoost algorithm implemented in the LMT package of the Weka Program [8; 12]. Each leaf defines a group of players. For each group, the figure shows the proportion of players who played at least one game in the NHL. Each leaf contains a logistic regression model for its group (not shown), which produces different predictions for players from the same group but with different features.

to explain the ranking in terms of which player features contribute the most to an above-average ranking. In this way the model tree can be used to highlight exceptional features of a player that scouts and teams can take into account in their evaluation.

Thesis Outline. We first review background about the NHL/NBA drafts and related work for model trees. Then we describe our datasets and carry out an explanatory analysis. After data exploration, the construction of model trees is presented. In the Results part, the rank correlations are reported to evaluate predictive accuracy. Case studies give examples of strong players in different groups and show how the model can be used to highlight the strongest points of exceptional players.

Chapter 2

Background, Literature Review and Problem Formulation

For different data types, there are different approaches for player ranking. With play-by-play datasets, Markov models have been widely used to analyze player performance [29; 15]. In the work of Thomas et al. (2013), the NHL player ability ratings are calculated by modelling team scoring rate as the semi-Markov process, with hazard functions depending on players on the ice [33]. Cervone et al. (2016) propose a Pointwise function to predict the number of points that an NBA player is expected to score by the end of an action [4]. For data that records the presence of players when a goal is scored, regression models have been applied to extend the classic wins-contribution metrics. For example, Macdonald (2011) develops two weighted least squares regression models to evaluate an NHL player's effect on team success in scoring and goals, independent of the opponents [21]. ~~While in~~ NBA, Sill (2010) enhances the traditional plus-minus by ridge regression to produce more accurate results [31]. In our work, we utilize player statistics that aggregate player pre-draft performance into a single set of numbers. While these datasets are less informative than play-by-play statistics, they are easier to obtain, interpret and process.

2.1 Previous Models

Regression Approaches. Wilson's (2016) predictive models (Generalized Linear Model, Artificial Neural Network, Support Vector Machine and LOESS) predict whether a drafted player can play more than 160 games after 7 career seasons. In his work, the pre-draft season statistics and the first four NHL season performance statistics are both used [38]. This encourages us to look into pre-draft season-by-season data. Since almost half of the drafted players will not play a game in the NHL based on our analysis [36], we decide to predict whether a drafted player will play at least one game or not in the NHL. The closest predecessor to our NHL work is due to Schuckers [29], who utilizes the pre-draft datasets to predict future NHL game counts by using a single generalized additive model. His results

show strong correlations between player career performance and pre-draft statistics. For the NBA, regression approaches have also been widely used to analyze player performance. Coates and Oguntimein (2008) examines the relationship between college metrics and the analogous metrics of NBA productivity through least square regression. Their results show some college statistics are significant in predicting NBA statistics, such as college scoring and rebounding, which do well in predicting NBA minutes played [5]. In this thesis, we mainly follow Greene's work [11], who builds a linear regression model to quantify the likelihood of a drafted college player having a successful NBA career. The inputs of his model are quite extensive, including the player draft picks, college statistics and physical qualities, adjusted by rookie year stats. His work presents a better predictive performance than the actual NBA draft when it comes to the top 100 prospects going to the draft.

Similarity-Based Approaches assume a similarity metric and group similar players to predict performance. A sophisticated example from baseball is the nearest neighbour analysis in the PECOTA system [32]. In the ice hockey field, the Prospect Cohort Success (PCS) model [37], defines cohorts of draftees based on age, height, and scoring rates. For basketball, many clustering approaches focus on defining appropriate roles or positions for a player. In Lutz's work [20], NBA players are clustered to several types like Combo Guards, Floor Spacers and Elite Bigs. They are grouped based on games played, minutes played per game, assists, turnover rate, rebound, steals and blocks. His results have shown different types of players can affect the team differently. The sports analytics group of Yale University has also developed an NBA clustering system to cluster players from season 2011-2012 to season 2015-2016 through hierarchical clustering methodology with their season performance statistics as inputs(`http://sports.sites.yale.edu/clustering-nba-players`). Our model tree learning provides an automatic method for identifying cohorts together with predictive validity. *We refer to grouping results as groups to avoid confusion with cohorts or clusters used in [32; 20].* Because tree learning is computationally efficient, our model tree is able to take into account a broad set of features. Also, it provides a separate predictive model for each group that assigns group-specific weights to different features. In contrast, Weissbock (2015) and Lutz (2012) make the same prediction for all players in the same cohort [37; 20]. So far, PCS has been applied to predict whether a player will play more than 200 games in their career. Tree learning can easily be modified to make predictions for any game count threshold.

2.2 Explanation of Career Success Metrics

The growing business of professional sports has resulted in an increasing demand for effective metrics to quantify player contribution to a team's success. Broadly there are three types of summary statistics to compare players: goal-based metrics, shot-based metrics and assists.

In this section, we discuss the success metrics used in the NHL and NBA to evaluate a player career performance. Among those success metrics, games played for NHL and player efficiency rating(PER) for NBA are chosen as response variables in our model trees.

2.2.1 NHL Player Evaluation Metrics

Goal scoring is an infrequent event in ice hockey compared to other high-scoring sports like basketball. In the NHL there are approximately 10 shots taken to score 1 goal. This higher number of shot events leads to measurements focusing on shots taken and shots allowed. In the work of Thomas [34], shots are assumed to be proxies for zone possession time, which itself is usually considered as a proxy for the team success. However, according to the analytic report from Found, the goal-based metrics (e.g., relative plus-minus/minute of ice time) always outperform the shot-based metrics (e.g., relative Corsi/minute of ice time), when it comes to accessing an individual player's contribution to the team winning percentage [7]. Thus, it is natural for sports analysts and statisticians to develop new measuring models which take several generally used success metrics into consideration, with the goal of evaluating players more completely.

Win Shares [13], first created in the world of baseball, are also widely used in other sports to evaluate player performance. Inspired by the Win Shares system, Hockey-Reference has built the Point Shares system, where one point is equivalent to one point share, and the player contribution is calculated by marginal goals, points and time on ice (`https://www.hockey-reference.com/about/point_shares.html`). Below are the main formulas in Point Share system for skaters:

$$\text{Skaters Point Shares} = (\text{marginal goals}) \ / \ (\text{marginal goals per point})$$

$$\text{marginal goals} = \text{goals created} - (\tfrac{7}{12}) * \text{time on ice} * (\text{goals created by forwards or defensemen} \ / \ \text{time on ice for forwards or defensemen})$$

$$\text{marginal goals per point} = (\text{league goals}) \ / \ (\text{league points})$$

The Point Shares System applies commonsense methods to calculating point shares and has been proven to have lower average absolute error in comparison with team's point total on NHL datasets from season $1917-1918$ to season $2009-2010$. However, the usage of magical numbers in this system may reduce its generality to other seasons datasets. The Total Hockey Rating(ThoR) proposed by Schuckers and Curro has gone beyond simple counting statistics, where ridge regression models have been adopted [28]. The ThoR is a comprehensive rating model which accounts for the impact of where a shift starts, and also non-shooting events including turnovers and hits that occur when a player is on the ice. The contribution of these actions is then quantified by the probability that it leads to a goal for the player's team. ThoR has been applied to all the $2010-2011$ and $2011-2012$ NHL

ice events, which produced convincing rating lists for both defensemen and forwards among these seasons. Nevertheless, ThoR and other regression models usually require play-by-play data, which are hard to get from public resource and often computationally expensive. Recently, there have also been a lot of studies about Wins Above Replacement(WAR), which mainly uses aggregate and count data. In the report from Thomas and Ventura [35], the shot rates, shot quality(likelihood of a shot becoming a goal), penalty rates and game states have been accounted for, resulting in a scalable statistic model.

In this thesis, we follow Schucker's work, using the total number of games played in a player's first seven years after being drafted. The teams also have the rights to players for at least seven years after they are drafted. Compared to other single metrics like plus-minus or complex models like WAR, games played is more intuitive and easier to interpret. Also, games played represents the usage rate of a player.

2.2.2 NBA Player Evaluation Metrics

Basketball, as one of the most popular sports in the world, has been well studied with respect to the use of success metrics to evaluate players. Two of the most intriguing and famous ones are Player Efficiency Rating (PER) (`www.basketball-reference.com/about/per.html`) and Win Shares (WS) (`www.basketball-reference.com/about/ws.html`), which are discussed in this section.

Player Efficiency Rating

The player efficiency rating (PER), created by John Hollinger, takes nearly every aspect of a player's contribution into consideration. It encompasses ~~a player~~ almost every accomplishment, such as field goals, free throws, 3-pointers, assists, rebounds, blocks and steals. Meanwhile, the negative results, such as missed shots, turnover and personal fouls are also accounted for the rating system. Compared to the traditional success metrics like wins, which highly depend on opportunities created by a player's teammates, PER aims to measure a single player's per-minute performance. In addition, PER is usually adjusted by team pace. Because Hollinger notes that a player's opportunities to accumulate statistics are dependent on the number of minutes played as well as the pace of the game.

The average league PER is always 15, which allows for comparing players across seasons. It has a rough scale which demonstrates the productivity of a player in a given year, listed in Table 2.1. This table provides a good guide to a player performance over his career. For example, Michael Jordan is widely recognized as one of the best players in NBA and his PER supports this claim. He currently has one of the highest career PER 27.91. There are only about 60 players in the history of the NBA with a career PER above 20.

The calculation of PER is as follows:

$$PER = (uPER \times \frac{lgPace}{tmPace}) \times \frac{15}{lguPER}$$

A Year for the Ages	35.0+
Runaway MVP Candidate	30.0-35.0
Strong MVP Candidate	27.5-30.0
Weak MVP Candidate	25.0-27.5
Definite All-Star	22.5-25.0
Borderline All-Star	20.0-22.5
Second offensive option	18.0-20.0
Third offensive option	16.5-18.0
Slightly above-average player	15.0-16.5
Rotation player	13.0-15.0
Non-rotation player	11.0-13.0
Fringe roster player	9.0-11.0
Player who won't stick in the league	0-9.0

Table 2.1: Player productivity and scale based on PER. *Referring from* ***https://en.wikipedia.org/wiki/Player_efficiency_rating***.

where $uPER$ is the unadjusted PER, calculated using many variables, including points, rebounds, assists, field goals, free throws, turnovers, and three pointers, as well as team and league statistics. In the above formula, lg is the prefix indicating of league rather than of player, and tm is the prefix indicating of team.

While PER is a scalable, interpretable and relatively comprehensive metric to evaluate player performance, it still suffers from criticisms. Some argue that PER is not a reliable measure of a player's defensive acumen because it largely measures offensive performance but only includes two defensive statistics, blocks and steals, in the formula.

In this thesis, we use PER as the target variable to build the M5 regression trees using drafted NBA player college datasets, following Greene's work in the NBA draft [11].

Win Shares

Similar to Win Shares in baseball and ice hockey, the win shares in basketball can be divided into two categories: offensive win shares and defensive win shares. Offensive win shares are calculated using points produced and offensive possessions, where the offensive possessions are predicted for each player. An offensive possession ends when 1) the team scores, 2) the team misses and the opponent gets the rebound, 3) the team turns over the ball, or 4) shooting free throws and either making the last shot or not securing the offensive rebound. Using these numbers from a game, the total number of possessions can be estimated for that game. In contrast, defensive win shares are calculated through defensive rating, which is concerned with opponent points and opponent possessions [11].

In our experiments, we also tried using career win shares as a response variable of linear regression model tree. However, it only produces a single regression model with lower ranking correlation results compared to PER, so we do not discuss it in the thesis.

Chapter 3

Datasets Description and Exploration

In this chapter, we first describe how we retrieve the data and how we preprocess it. Then summaries of our data are provided. Next, we discuss the distribution of some import attributes with respect to their relationship with target variables. We apply Python 2.7 to data collection, preprocessing and statistical analysis. ~~As~~ for plots, we use the *ggplot2* library in R.

3.1 Data Provenance

3.1.1 Ice Hockey Data

Our ice hockey data is obtained from a broad set of public-domain on-line sources, depending on whether a player played for the NHL or not after being drafted. For players who played at NHL after being drafted, their career statistics, demographic information as well as pre-draft statistics are recorded on the NHL official website (`www.nhl.com`), where we crawl players drafted between 1998 to 2008. For players who were drafted but not played in the NHL, we obtain their pre-draft statistics and demographic information from `www.eliteprospects.com`. Following the work of Schuckers [29], we also collect CSS_rank (rankings from the Central Scouting Services) from `www.draftanalyst.com`. In addition, we are indebted to David Wilson for sharing his NHL performance dataset.

After data collection, we clean and preprocess our datasets. We notice that some player information is not available online. This issue ~~reflects~~ most in *CSS_rank* and playoff stats (po_GP, po_G, po_A, po_P); see below. Our main processing steps for data can be summarized as follows:

- If a player was not ranked by the Central Scouting Service (CSS), we assign **(1+ the maximum rank for his draft year)** to his CSS rank value.
- Replacing missing playoff stats (po_GP, po_G, po_A, po_P) by 0.

- Pooling all European countries into a single category.
- If a player played for more than one team in his draft year (e.g., a league team and a national team), we add up ~~this~~ counts from different teams.
- Eliminating players drafted in year 2003 since a large portion of them has no CSS_rank.

3.1.2 Basketball Data

Our basketball datasets are obtained from `www.basketball-reference.com`, a rich resource of NBA player data, containing both their pre-draft and career information. We consider players who were drafted into NBA between 1985 and 2011, inclusive. This draft range ensures that a player has enough time (at least 7 years) to accumulate his career performance statistics.

In our datasets, we exclude players whose college performance statistics are not available. There are also some players whose career statistics are missing and instead, have college statistics. We replace their career statistics **(PER)** by $min(x) - std(x)$, where $min(x)$ is the minimum career PER and $std(x)$ is the standard deviation of career PER for players drafted in the same year. These players are considered to be worse than players who played in the NBA. For the players who miss both values, we discard them. In Table 3.1, we summarize the count of these players in our datasets.

College stats	NBA stats	count	Preprocessing
1	0	15	replaced by $min(x) - std(x)$
0	1	173	excluded
0	0	35	excluded
1	1	1405	kept

Table 3.1: Summary of statistics availability.*1 denotes stats are available, otherwise, it is 0.*

3.2 Data Fields Explanation

3.2.1 Ice Hockey Datasets

Our full dataset is posted on the worldwide web *`https://github.com/liuyejia/Model_Trees_Full_Dataset`*. We consider players drafted into the NHL between 1998 to 2008 (excluding goalies). Following the work of Schuckers et al. (2016) [29], we took as our dependent variable the total number of games g_i played by a player i after 7 years under an NHL contract. The first seven seasons are chosen because NHL teams have at least seven-year rights to players after they are drafted [28]. Our dataset also includes the total time on ice after 7 years. The results for time on ice were similar to number of games, so we discuss only the results for number of games. The independent variables include demographic factors *(e.g., age)*, performance metrics for the year in which a player was

drafted (e.g., goals scored), and the rank assigned to a player by the NHL Central Scouting Service (CSS). Table 3.2 lists all data columns and their descriptions. Figure 3.1 shows an excerpt from the dataset.

Variable Names	Descriptions
id	nhl.com id for NHL players, otherwise Eliteprospects.com id
DraftAge	Age in Draft Year
Country	Nationality. Canada -> 'CAN', USA -> 'USA', countries in Europe -> 'EURO'
Position	Position in Draft Year. Left Wing -> 'L', Right Wing -> 'R', Center -> 'C', Defencemen -> 'D'
Overall	Overall pick in NHL Entry Draft
CSS_rank	Central scouting service ranking in Draft Year
rs_GP	Games played in regular seasons in Draft Year
rs_G	Goals in regular seasons in Draft Year
rs_A	Assists in regular seasons in Draft Year
rs_P	Points in regular seasons in Draft Year
rs_PIM	Penalty Minutes in regular seasons in Draft Year
rs_PlusMinus	Goal Differential in regular seasons in Draft Year
po_GP	Games played in playoffs in Draft Year
po_G	Goals in playoffs in Draft Year
po_A	Assists in playoffs in Draft Year
po_P	Points in playoffs in Draft Year
po_PIM	Penalty Minutes in playoffs in Draft Year
po_PlusMinus	Goal differential in playoffs in Draft Year
sum_7yr_GP	Total NHL games played in player's first 7 years of NHL career
sum_7yr_TOI	Total NHL Time on Ice in player's first 7 years of NHL career
GP_7yr_greater_than_0	Played a game or not in player's first 7 years of NHL career

Table 3.2: Player Attributes listed in dataset *(excluding weight and height)*.

id	Player Name	Draft Age	Country	Height (in)	Weight (lbs)	Position	Overall	CSS_rank	rs_GP	rs_G	rs_A	rs_P	rs_PIM	rs_Plus Minus	sum_7yr_GP	sum_7yr_TOI	GP_7yr > 0
847-4141	Patrick Kane	19	USA	71	177	R	1	2	65	67	87	154	94	44	515	9927	yes
847-3419	Brad Marchand	18	CAN	69	181	L	71	80	68	29	37	66	83	40	218	3418	yes
27	Yared Hagos	18	EURO	73	218	C	70	24	43	11	26	37	24	1	0	0	no

Figure 3.1: Sample Player Data for their draft year. rs = regular season. We use the same statistics for the playoffs *(not shown)*.

3.2.2 Basketball Datasets

Following Greene's work [11], we choose career PER as our response variable. Our datasets also include career win shares and ws_48. However, M5 regression tree generated when using them as target variables is only a single node (a single linear regression model) with weaker predictive power *(correlation)* compared to career PER. Thus we don't present them in the thesis.

In our experiment, the datasets are divided into training *(1985-2005 drafts)* and testing datasets *(2006-2011 drafts)* according to the ratio 6/4. Table 3.3 lists all the data columns and their descriptions.

Variables	Descriptions
age	Player age in his draft year
height	Player height in his draft year
weight	Player weight in his draft year
position	Player position in his draft year
shoots	Player shoots style, left-handed or right-handed
ah	If a player gained amateur honour (such as *McDonald's All American*) in college before being drafted, then the value is 1, otherwise, 0
g	Games played by the player in his draft year
mp	Minutes played in the player draft year *(total and per game statistics in the player draft year are both collected)*
fg	Field goals gained by the player in his draft year *(total and per game statistics in the draft year are both collected)*
fga	Field goals attempts made by the player in his draft year
3p	3-point field goals obtained by the player in his draft year
3pa	3-point field goal attempts made by the player in his draft year
ft	Free throws made by the player in his draft year *(total and per game statistics in the player draft year are both collected)*
fta	Free throw attempts made by the player in his draft year
orb	Offensive rebounds made by the player in his draft year
trb	Total rebounds made by the player in his draft year *(total and per game statistics are both collected)*
ast	Assists made by the player in his draft year *(total and per game statistics are both collected)*
stl	Steals made by the player in his draft year
blk	Blocks made by the player in his draft year
tov	Turnovers of the player in his draft year
pf	Player personal fouls in his draft year
pts	Points gained by the player in his draft year *(total and per game statistics are both collected)*

Table 3.3: Player Attributes listed in datasets.

3.3 Data Exploration

In this section, we focus on exploring the distribution of some important predictors which includes CSS_rank and Position, and their relationship with the response variable in our obtained ice hockey and basketball datasets, respectively.

3.3.1 Features Analysis for Ice Hockey Datasets

CSS_rank. Each year the CSS rank is given by the full-time professional scouts in NHL Central Scouting Bureau. The Bureau ranks players ~~based on how well they would be~~ by ~~simulating~~ their performance in junior leagues to the performance in NHL. The CSS rank is stratified by player position (Skaters versus Goalies) and player location (North America versus Europe). In the work of Schuckers et al. (2016) [29], the CSS rank plays an important role in predicting player career performance. It was converted to Cescin (multiply 1.35 for North American players while 6.27 for European players) for each player. In our experiment, we use the original CSS_rank directly from raw data since the position and country are also considered in our model trees. Figure 3.2 shows the relationship between CSS_rank and sum_7yr_GP.

Country_group and major junior league. The distribution of player sum_7yr_GP grouped by country_group and major junior league OHL, QMJHL, WHL are shown in Table 3.4 and Table 3.5. Table 3.4 illustrates that drafted players from Canada have higher sum_7yr_GP compared to American and European players, along with a bigger number. For major junior league, the drafted players from OHL perform better than from other leagues based on their statistics displayed in Table 3.5.

country_ group	size	mean	std	min	25%	50%	75%	max	CSS_rank (mean)
CAN	903.0	66.75	116.21	0.0	0.0	0.0	86.50	524.0	86
EURO	856.0	50.85	102.74	0.0	0.0	0.0	34.25	475.0	100
USA	460.0	57.79	105.73	0.0	0.0	0.0	68.0	515.0	99

Table 3.4: Statistic overview of country_group vs.sum_7yr_GP.

League	size	mean	std	min	25%	50%	75%	max	CSS_rank (mean)
OHL	352.0	84.12	129.80	0.0	0.0	3.5	132.75	524.0	87
QMJHL	218.0	55.12	112.24	0.0	0.0	0.0	39.50	471.0	96
WHL	344.0	70.28	115.27	0.0	0.0	0.0	103.00	494.0	90
others	1305.0	49.5	99.29	0.0	0.0	0.0	36.00	504.0	97

Table 3.5: Statistic overview of major league vs.sum_7yr_GP.

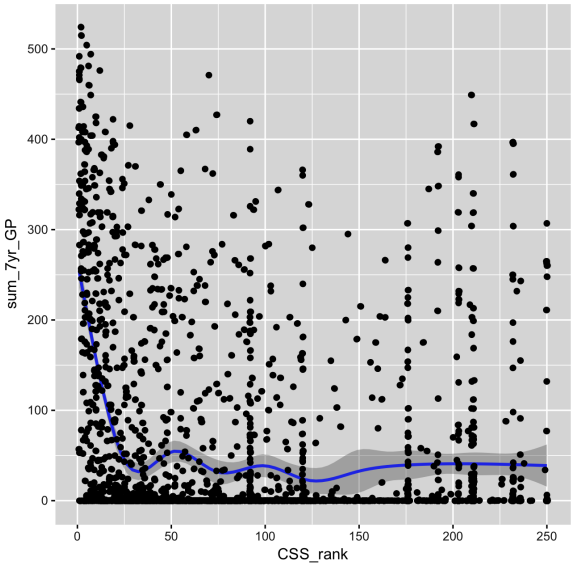

Figure 3.2: Scatter plot of CSS_rank vs.sum_7yr_GP. *Smoothed by generalized additive model.*

3.3.2 Features Analysis for Basketball Datasets

Position. Every basketball player is assigned a position to describe how they play in the court, which is usually determined by player physical size and skills. For example, if a player is physically fit and strong, then he is likely to be in the position of *center* or *power forward*. If he is a guard and scores well, he is potentially assigned as the *shooting guard*. Different player positions contribute differently to winning the game. According to a report from Mazique [23], the bigs *(power forwards and centers)* carry the responsibility of scoring and defending for the team, and contributes most to the team success. The wings positions *(small forwards and shooting guards)* require a truly excellent player to elevate the team. Based on the importance of player position in previous studies [20; 23], we analyze the relationship between the player draft position and career PER in our datasets, shown in Table 3.6. As shown in Table 3.6 in bold, *center and power forward* has the highest career PER on average, in accordance with most studies of position in the NBA.

position	size	mean	std	min	25%	50%	75%	max
Center and Power Forward	162	**14.93**	3.5	7	11.85	13.75	16.33	24.6
Shooting Guard and Point Guard	115	13	3.28	4.4	10.7	12.65	14.75	24.2
Power Forward and Small Forward	108	12.94	3.35	-1.5	11.1	13.35	15.9	20.8
Small Forward and Shooting Guard	68	12.81	3.03	7	10.68	12.65	14.45	25.2
Point Guard	181	12.61	6.7	-6.8	9.7	12.2	15.1	76.1
Power Forward	134	11.95	6.94	-30.2	9.8	11.85	14.78	58.3
Small Forward	142	11.05	4.77	-5.6	8.7	11.05	13.9	31.3
Shooting Guard	142	10.66	4.9	-11.4	8.58	11.45	13.4	22.2
Center	227	9.96	9.78	-48.6	8.6	11.3	13.8	66.8
Guard	10	-14.2	18.97	-57.62	-24.25	-7.87	-1.07	1.61
Forward	12	-14.87	20.73	-57.62	-15.13	-5.48	-1.44	-0.88
Forward-Center	7	-14.24	13.58	-27.62	-27.62	-15.13	-1.63	1.61

Table 3.6: Overview of position and career PER statistical analysis, sorted by the mean career PER value of each position.

Chapter 4

Model Tree Construction

In this chapter, we discuss the construction of model trees used in our experiments, Logistic Model Trees (LMT) and M5 regression trees (M5P). For LMT construction, we introduce the main algorithm to build logistic models, tree splitting and pruning methods. Then, we display the initialization, splitting and pruning of M5P.

4.1 Logistic Model Trees

Logistic Model Tree (LMT) has been created ~~aiming at combining~~ advantages of tree induction methods and linear logistic models. In each leaf node, explicit class probability estimates can be calculated rather than just a classification label. In the logistic variant, the LogitBoost algorithm is adapted to produce a logistic regression model in every node [18], and then the tree is split based on the C4.5 splitting criterion [26] (see section 4.1.2). LMT has been tested on 36 datasets from UCI repositories [2]. The results show its classification accuracy outperforms C4.5, CART, Naïve Bayes trees and Lotus [19] on some datasets. In our thesis, the LMT is used to produce ice hockey prospect groups with a predictive model in each group. In the following subsection, we overview the relevant concepts and algorithms to LMT.

4.1.1 LogitBoost Algorithm

In LMT, every node in the tree contains a logistic model. To estimate the parameters of ~~logistic model,~~ the LogitBoost algorithm has been applied, which looks for parameter values that maximize the likelihood of observed data points. The pseudocode for this algorithm is shown in Table B.1.

LogitBoost fits a regression model in each iteration: it first computes a response variable which encodes the error of the currently fitting model on the training data points (in terms of probability estimates), and then tries to improve the model by adding a function to the ultimate classifier; the function is chosen to fit the response variables, using a least-squared

error criterion. As shown in [9], this process is similar to performing a quasi-Newton step in every iteration.

LMT adopts the SMOTI algorithm to induce model trees [22]. The idea is to construct the final multiple regression function at a leaf from simple regression functions that are fit at different levels in the tree, from the root down to that particular leaf. This has a smoothing effect, in that the models fit at child nodes are close to the parent model, and therefore closer to each other.

4.1.2 Splitting Strategies

According to Landwher's description in [18], LMT uses almost the same splitting strategies applied in C4.5, a landmark decision tree algorithm for classification. It utilizes information entropy to make splitting decisions. The attribute with the highest normalized information gain is chosen to make the splitting decision each time. The pseudocode of C4.5 is summarized in the Appendix C [16].

When applying C4.5 algorithm, LMT makes two adjustments to grow more reliable model trees. First, if a node contains less than 15 examples, no splitting would happen. Since the leaves of LMT are complex models, more examples are required for model fitting. Second, boosting happens only at nodes which contain at least 5 examples, which is the minimum number of examples required by cross-validation in LogitBoost to determine the best number of iterations.

4.1.3 Tree Pruning

Compared to ordinary classification trees, LMT are usually much smaller since a single node with its complex regression model can lead to the best generalization performance. But the same principle still applies here: every split and local refinement at child node still increase the model complexity. LMT employs the pruning method from the CART algorithm to avoid overfitting. CART is a suite of algorithms to build classification and regression trees [3]. The pruning method of CART is a post-pruning strategy using a combination of training error and penalty term for model complexity to make pruning decisions, with the purpose to optimize the cost-complexity function, which is shown as follows:

$$R_\alpha(T) = R(T) + \alpha \times |f(T)|$$

where $R(T)$ is the training/learning error, $f(T)$ is a function that returns the set of leaves of tree T, and α is a regularization parameter.

4.2 M5 Regression Trees

M5 model trees (M5P) are designed for tasks to predict a numeric value associated with a case rather than just the class which the case belongs to [25]. In the leaves, a multivariate

linear regression model is built instead of just a numeric value. Compared to standard Classification and Regression Tree (CART), M5P is generally smaller in tree size and more accurate, meanwhile, they can also deal with high dimensionality attributes. In our thesis, the M5P is used to predict the player efficiency rating (PER) for drafted players in NBA based on their college statistics. In the subsections, the construction of M5P is reviewed.

4.2.1 Initial Tree Construction

The growing and splitting of M5P is based on the standard deviation of the target variables in training cases. Supposing we have a set of training examples T, and T_i represents the ith subset of T. A calculation which determines the subset of cases related to each outcome is carried out in every possible T_i. After examining all the possible test cases, the one with maximum error reduction is chosen. The expected error reduction is defined as:

$$\Delta error = sd(T) - \sum_i \frac{|T_i|}{|T|} \times sd(T_i)$$

where $sd(T)$ is the standard deviation of target values of all training cases and $sd(T_i)$ is the standard deviation of target values of cases in T_i.

4.2.2 Linear Models Development

A multivariate linear model is built at every tree node with standard regression methods. In the construction process, M5P compares the accuracy of using linear models with using subtree to make decisions. After the linear model is obtained, M5P uses a greedy search to remove variables that contribute little to the model in order to simplify linear models.

4.2.3 Tree Pruning

The pruning process utilizes an estimate of expected error which will be applied to each node for data. First, the absolute difference between the actual value and predicted value is averaged for each training case that reaches that node and then multiplied by $(n+v)/(n-v)$, where n is the number of training examples and v is the number of parameters in the model.

M5P generates a regression model for each interior node in the unpruned tree. This model is calculated using only the attributes that are tested in the subtree below this node. Then, the regression model is simplified by dropping terms to minimize the expected estimated error mentioned above. Terms are dropped one by one, greedily, as long as the expected estimated error decreases. Finally, the tree is pruned back from the leaves once a model is in place for each interior node.

4.2.4 Smoothing

Smoothing is used in M5P to improve the prediction accuracy [24]. The predicted value of a case given by the model at the leaf is adjusted to reflect average of the predicted values

at nodes along the path from the root to the leaf. The formula of adjustment is defined as follows:

$$PV(S) = \frac{n_i \times PV(S_i) + k \times M(S)}{n_i + k}$$

where S_i is the branch of subtree S, n_i is the number of training cases at S_i, $PV(S_i)$ function denotes the predicted value at S_i, and $M(S)$ is the value given by the model at S and k is a smoothing constant *(default* 15*)*.

Chapter 5

NHL Results and Case Studies

We present our ~~experiment~~ results for ice hockey data in this chapter. We first go through how we construct LMT for our datasets, then display the modelling results and learned groups. Next, we propose a method to analyze the strongest points of exceptional players, which is informative to scouts and teams. Python 2.7, Weka and R are the main tools used in our experiments.

5.1 Predictive Models and Evaluation

In this section, we first show the construction of our logistic model tree, and then display predicted correlation results in comparison with the actual draft order from team's decisions. Last but not least, we analyze the learned groups with respect to dependent and independent variables, as well as their interactions.

5.1.1 Model Trees Construction

Model trees are a flexible formalism that can be built for any regression model. An obvious candidate for a regression model would be linear regression; alternatives include a generalized additive model [29], and a Poisson regression model specially built for predicting counts [27]. We introduce a different approach: a logistic regression model to predict whether a player will play any games at all in the NHL ($g_i > 0$). The motivation is that many players in the draft never play any NHL games at all (up to 50% depending on the draft year) [36]. This poses an extreme excess-zeros problem for any regression model that aims to predict directly the number of games played. In contrast, for the classification problem of predicting whether a player will play any NHL games, the excess-zeros issue means that our data is balanced between the classes. This classification problem is interesting in itself; for instance, a player agent would be keen to know what chances their client has to participate in the NHL. The logistic regression probabilities $p_i = P(g_i > 0)$ can be used not only to predict whether a player will play any NHL games, but also to rank players such

that the ranking correlates well with the actual number of games played. Our method is therefore summarized as follows.

1. Build a tree whose leaves contain a logistic regression model.
2. The tree assigns each player i to a unique leaf node l_i, with a logistic regression model $m(l_i)$.
3. Use $m(l_i)$ to compute a probability $p_i = P(g_i > 0)$.

In our experiments, we follow Schucker's work [29], dividing our data into the same two cohorts. The first cohort contains players drafted in the years 1998 through 2002. The second cohort is for players drafted from 2004 to 2008. Figure 1.1 shows the logistic regression model tree learned for our second cohort by the LogiBoost algorithm. It places CSS rank at the root as the first splitting attribute which has the highest information gain. Players ranked better than 12 form an elite group, of whom almost 82% play at least one NHL games. For players at rank 12 or below, the tree considers next their regular season points total. Players with rank and total points below 12 form an unpromising group: only 16% of them play an NHL game. Players with rank below 12 but whose points total is 12 or higher, are divided by the tree into three groups according to whether their regular season plus-minus score is positive, negative, or 0. (A three-way split is represented by two binary splits). If the plus-minus score is negative, the prospects of playing an NHL game are fairly low at about 37%. For a neutral plus-minus score, this increases to 61%. For players with a positive plus-minus score, the tree uses the number of playoff assists as the next most important attribute. Players with a positive plus-minus score and more than 10 playoff assists form a small but strong group that is 92% likely to play at least one NHL game.

5.1.2 Modelling Results

Following [29], we consider three rankings:

1. the performance ranking based on the actual number of NHL games that a player played.
2. the ranking of players based on the probability p_i of playing at least one game (Tree Model SRC).
3. the ranking of players based on the order in which they were drafted by team (Draft Order SRC).

The draft order can be viewed as the ranking that reflects the judgment of NHL teams. Like [29], we evaluate the predictive accuracy of the Draft Order and the LMT model using the Spearman Rank Correlation (SRC) between (i) Draft order and actual number of games,

Training Data NHL Draft Years	Out of Sample Draft Years	Draft Order SRC	LMT Classification Accuracy	LMT SRC
1998, 1999, 2000	2001	0.43	82.27%	**0.83**
1998, 1999, 2000	2002	0.30	85.79%	**0.85**
2004, 2005, 2006	2007	0.46	81.23%	**0.84**
2004, 2005, 2006	2008	0.51	63.56%	**0.71**

Table 5.1: Predictive Performance (Draft Order, our Logistic Model Trees) using Spearman Rank Correlation (SRC). Bold indicates the best values.

and (ii) p_i order and actual number of games, as shown in Table 5.1. We provide the formula for the Spearman correlation in the Appendix A.

Other Approaches. We also tried designs based on a linear regression model tree, using the M5P algorithm implemented in the Weka program. The result is a decision stump that splits on CSS rank only, which had substantially worse predictive performance, shown in Table 5.2 (i.e., Spearman correlation of only 0.4 for the $2004 - 2006$ cohort). For the generalized additive model (gam), the reported correlations were 2001: 0.53, 2002: 0.54, 2007: 0.69, 2008: 0.71 [29]. Our correlation is not directly comparable to the gam model because of differences in data preparation: the gam model was applied only to drafted players who played at least one NHL game, and the CSS rank was replaced by the Cescin conversion factors: for North American players, multiply CSS rank by 1.35, and for European players, by 6.27 [10]. The Cescin conversion factors represent an interaction between the player's country and the player's CSS rank. A model tree offers another approach to representing such interactions: by splitting on the player location node, the tree can build a different model for each location. Whether the data warrant building different models for different locations is a data-driven decision made by the tree building algorithm. The same point applies to other sources of variability, for example the draft year or the junior league. Including the junior league as a feature has the potential to lead to insights about the differences between leagues, but would make the tree more difficult to interpret; we leave this topic for future work. In the next section we examine the interaction effects captured by the model tree in the different models learned in each leaf node.

Training Data NHL Draft Years	Out of Sample Draft Years	Draft Order SRC	M5P SRC
1998, 1999, 2000	2001	0.43	**0.53**
1998, 1999, 2000	2002	0.30	**0.47**
2004, 2005, 2006	2007	**0.46**	0.36
2004, 2005, 2006	2008	**0.51**	0.44

Table 5.2: Predictive Performance (M5 regression trees, over all draft ranking) using Spearman Rank Correlation. Bold indicates the best values.

5.1.3 Groups and Variables Interaction

In this section, we examine the learned group regression models, first in terms of the dependent success variable, then in terms of the player features.

Learned Groups and Dependent Variable. Figure 5.1 shows boxplots for the distribution of our dependent variable g_i. The strongest groups are, in order, 1, 6, and 4. The other groups show weaker performance on the whole, although in each group some players reach high numbers of games. Most players in Group 2&3&4&5 have GP equals to zero while Group 1&6 represent the strongest cohort in our prediction, where over 80% players played at least 1 game in NHL. The tree identifies that among the players who do not have a very high CSS rank (worse than 12), the combination of regular season $Points >= 12$, $rs\_PlusMinus > 0$, and $po\_Assists > 10$ is a strong indicator of playing a substantive number of NHL games (median $g_i = 128$).

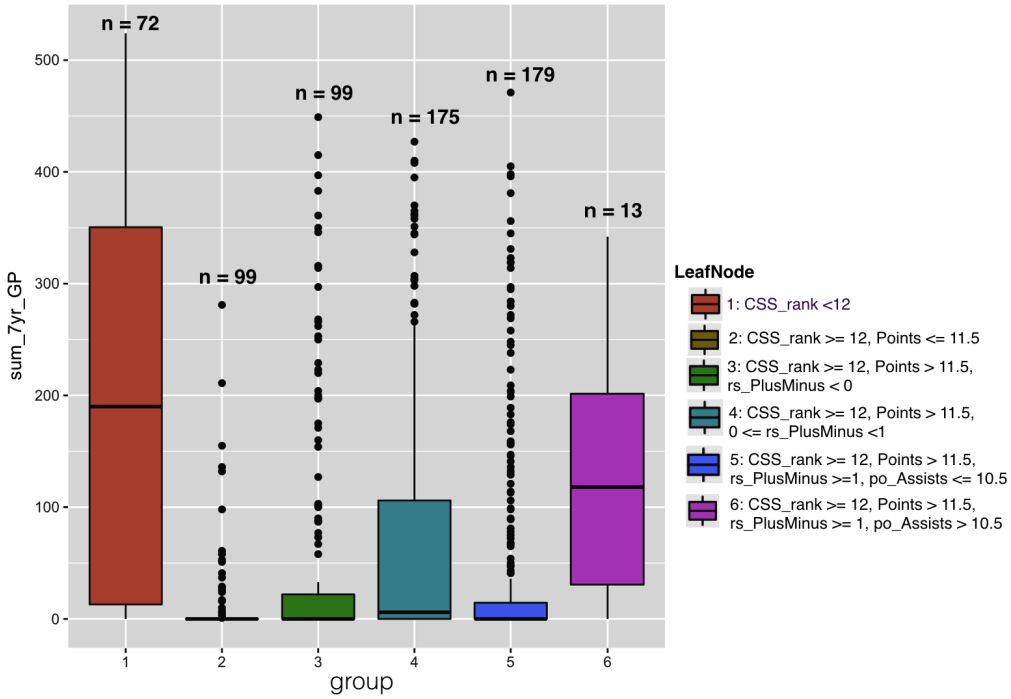

Figure 5.1: Boxplots for the dependent variable g_i , the total number of NHL games played after 7 years under an NHL contract. Each boxplot shows the distribution for one of the groups learned by the logistic regression model tree. The group size is denoted n.

Learned Groups and Independent Variables. Figure 5.2 shows the average statistics by group and for all players. The CSS rank for Group 1 is by far the highest. The data validate the high ranking in that 82% players in this group went on to play an NHL game. Group 6 in fact attains an even higher proportion of 92%. The average statistics of this group are even more impressive than those of group 1 (e.g., 67 regular season points

in group 6 vs. 47 for group 1). But the average CSS rank is the lowest of all groups. So this group may represent a small group of players ($n = 13$) overlooked by the scouts but identified by the tree. Other than Group 6, the group with the lowest CSS rank on average is Group 2. The data validate the low ranking in that only 16% of players in this group went on to play an NHL game. The group averages are also low (e.g., 6 regular season points is much lower than other groups).

Group	Mean Points											
	rs_P	rs_A	CSS_ rank	po_A	Weight	Height	rs_ Plus Minus	rs_GP	rs_ PIM	Draft Age	po_ GP	po_P
1	47	27	7	4	204	74	6	55	63	18	8	7
2	6	4	94	1	206	74	-2	38	56	18	8	1
3	40	23	76	2	201	73	-9	63	78	18	7	4
4	44	25	86	4	198	73	0	47	59	19	10	8
5	43	26	71	3	199	73	12	62	73	19	9	5
6	67	44	107	14	193	71	23	65	83	19	19	19
all	*39*	*23*	*101*	*2*	*201*	*73*	*3*	*55*	*67*	*18*	*6*	*4*

Figure 5.2: Statistics for the average players in each group and all players.

Learned Group Weights and Variable Interactions. Figure 5.3 illustrates logistic regression weights by group. A positive weight implies that an increase in the covariate value predicts a large increase in the probability of playing more than one game, compared to the probability of playing zero games. Conversely, a negative weight implies that an increase in the covariate value decreases the predicted probability of playing more than one game. Bold numbers show the groups for which an attribute is most relevant. The table exhibits many interesting interactions among the independent variables; we discuss only a few. Notice that if the tree splits on an attribute, the attribute is assigned a high-magnitude regression weight by the logistic regression model for the relevant group. Therefore our discussion focuses on the tree attributes.

At the tree root, *CSS rank* receives a large negative weight of -17.9 for identifying the most successful players in Group 1, where all CSS ranks are better than 12. Figure 5.4a shows that the proportion of above-zero to zero-game players decreases quickly in Group 1 with worse CSS rank. However, the decrease is not monotonic. Figure 5.4b is a scatterplot of the original data for Group 1. We see a strong linear correlation ($p = -0.39$), and also a large variance within each rank. The proportion aggregates the individual data points at a given rank, thereby eliminating the variance. This makes the proportion a smoother dependent variable than the individual counts for a regression model.

Group 5 has the smallest logistic regression coefficient of -0.65. Group 5 consists of players whose CSS ranks are worse than 12, regular season points above 12, and plus-minus above 1. Figure 5.5a plots CSS rank vs. above-zero proportion for Group 5. As the proportion plot shows, the low weight is due to the fact that the proportion trends downward

Metrics / Group	CSS_ rank	Draft Age	Height	Weight	Country _group	Position	Games _played	Goals	Assists	Points	Penalty_ in_ Minutes	Plus Minus
1	**-17.9**	-3.91	-2.69	2.35	E: -0.77 C: 1.23 U:-0.49	D: -0.54 L: 2.09 R: -0.68	rs: -2.43 po: 4.15	rs: -0.03 po: **-9.8**	rs: 1.97 po: 8.89	rs: 1.73 po: 0.3	rs: **7.98** po: **-7.6**	rs: -0.45
2	-1.12	-1.1	-4.8	**6.7**	E: -0.13 C: 0.28 U:-0.22	D: -1.1 L: -0.45 R: 1.89	rs: 5.9 po: -14.1	rs: -2.17 po: -2.9	rs: **11.8** po: **21.6**	rs: **14.2** po:**11.1**	rs: -2.72 po: 5.2	rs: 1.57
3	-2.6	**6.95**	-7.4	**6.7**	E: -2.4 C: 1.04 U: 2.34	D: 0.39 L: 0.68 R: -0.4	rs: 3.21 po: -1.05	rs: -0.52 po: -0.6	rs: 1.36 po: -0.39	rs: 0.54 po:-2.6	rs: -1.88 po: 2.7	rs: 13.16
4	-2.4	5.2	-4.2	-0.52	E: 1.08 C: -0.40 U: -0.6	D: -0.03 L: -0.24 R: 0.14	rs: 3.58 po: -4.5	rs: -2.16 po: 1.58	rs: -0.12 po: 1.71	rs: -1.4 po: 1.6	rs: -2.72 po: 3.45	rs: 0
5	-0.65	-3.89	**0.01**	4.68	E: -1.26 C: 0.74 U: 0.47	D: 0.91 L: -0.64 R: 0.05	rs: 2.24 po: -0.25	rs: 3.59 po: -1.7	rs: -0.23 po: 0.33	rs: 2.19 po:-0.8	rs: -4.05 po: 6.86	rs: -0.73
6	-8.89	6.64	-14.91	0.34	E: -28.1 **C: 5.9** U: 7.2	D: 3.32 L: 0.74 R: -28.12	**rs: 16.7** po: 2.74	rs: **21.6** po: -9.7	rs: -0.34 po: -0.43	rs: -0.5 po:-0.4	rs: 1.3 po: -1.6	rs: **21.9**

Figure 5.3: Group 200$(4 + 5 + 6 + 7 + 8)$ Weights Illustration. E = Europe, C = Canada, U = USA, rs = Regular Season, po = Playoff. Largest-magnitude weights are in bold. Underlined weights are discussed in the text.

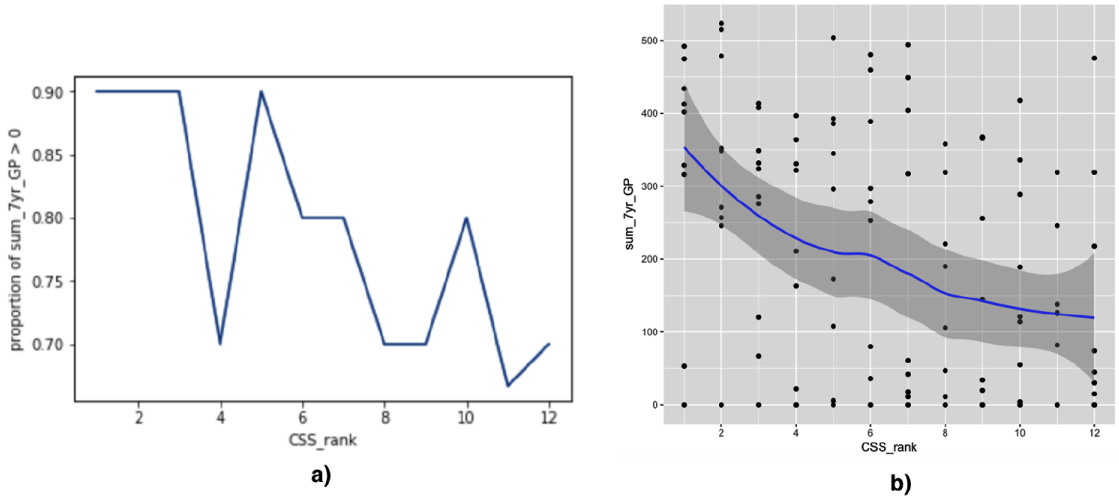

Figure 5.4: Proportion and scatter plots for CSS_rank vs. sum_7yr_GP in Group 1.

only at ranks worse than 200. The scatterplot in Figure 5.5b shows a similarly weak linear correlation of -0.12.

Regular season points are the most important predictor for Group 2, which comprises players with CSS rank worse than 12, and regular season points below 12. In the proportion plot Figure 5.6, we see a strong relationship between points and the chance of playing more than 0 games (logistic regression weight 14.2). In contrast, in Group 4 (overall weight -1.4),

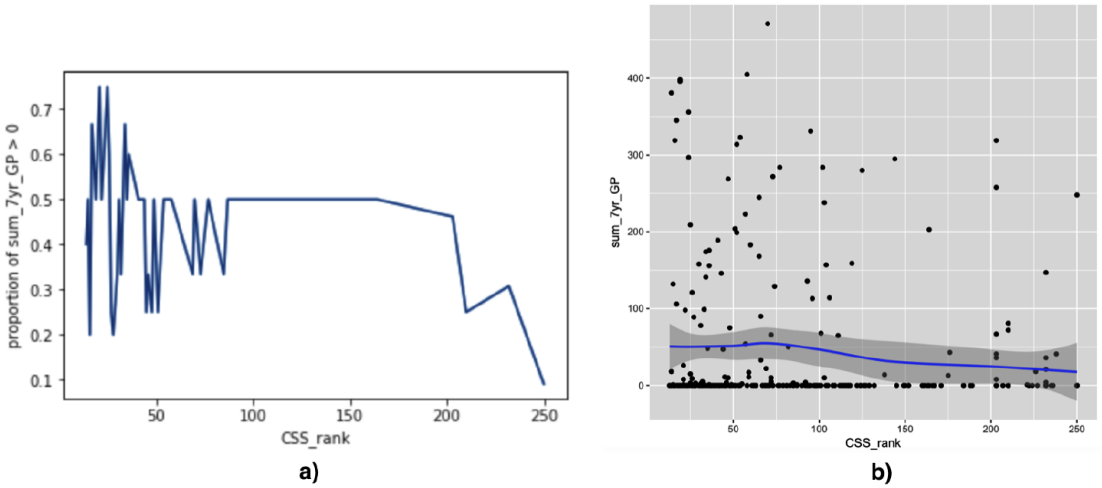

Figure 5.5: Proportion and scatter plots for CSS_rank vs.sum_7yr_GP in Group 5.

there is essentially no relationship up to 65 points; for players with points between 65 and 85 in fact the chance of playing more than zero games slightly decreases with increasing points.

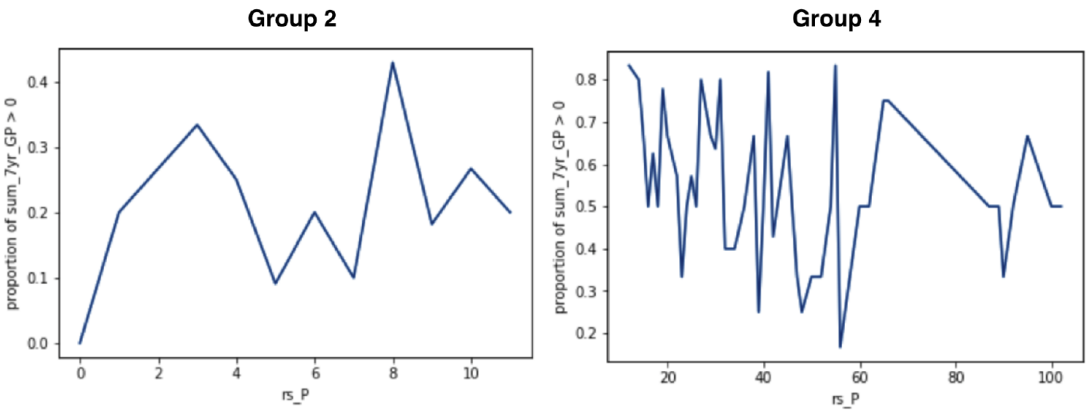

Figure 5.6: Proportion_of_Sum_7yr_GP_greater_than_0 vs. rs_P in Group 2&4.

In Group 3, players are ranked at level 12 or worse, have collected at least 12 regular season points, and show a negative plus-minus score. The most important feature for Group 3 is the *regular season plus-minus* score (logistic regression weight 13.16), which is negative for all players in this group. In this group, the chances of playing an NHL game increase with plus-minus, but not monotonically, as Figure 5.7 shows. As Figure 5.3 shows, Group 3 is the only large group where plus-minus receives a weight above 10.

For *regular season goals*, Group 5 assigns a high logistic regression weight of 3.59. However, Group 2 assigns a surprisingly negative weight of −2.17. Group 5 comprises players at

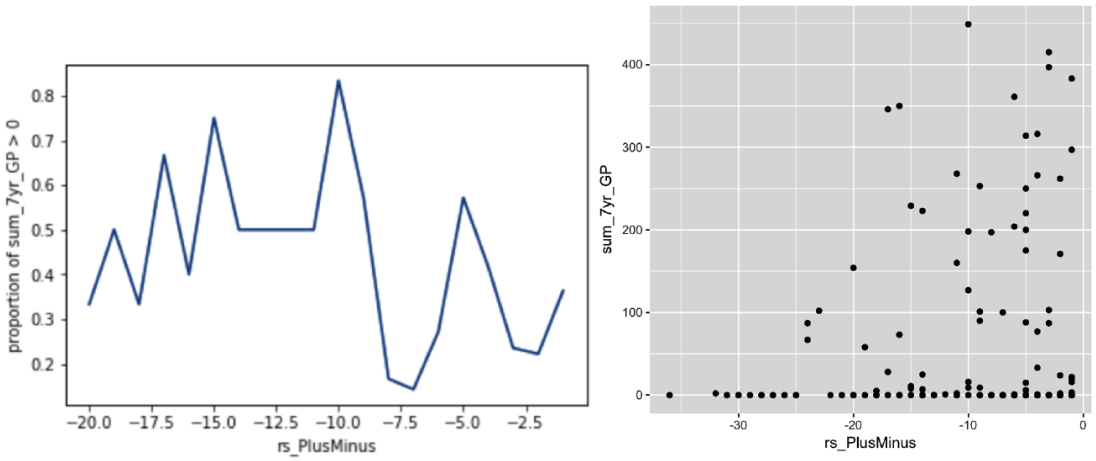

Figure 5.7: Proportion and scatter plots for rs_PlusMinus vs.sum_7yr_GP in group 3.

CSS rank worse than 12, regular season points 12 or higher, and positive plus-minus greater than 1. About 64.8% in this group are offensive players (see Figure 5.8). The positive weight therefore indicates that successful forwards score many goals, as we would expect.

Group 2 contains mainly defensemen (61.6%; see Figure 5.8). The typical strong defenseman scores 0 or 1 goals in this group. Players with more goals tend to be forwards, who are weaker in this group. In sum, the tree assigns weights to goals that are appropriate for different positions, using statistics that correlate with position (e.g., plus-minus), rather than the position directly.

5.2 Case Studies: Exceptional Players and Strongest Points

Teams make drafting decisions not based on player statistics alone, but drawing on all relevant source of information, and with extensive input from scouts and other experts. As Cameron Lawrence from the Florida Panthers put it, 'the numbers are often just the start of the discussion' [14]. In this section we discuss how the model tree can be applied to support the discussion of individual players by highlighting their special strengths. The idea is that the learned weights can be used to identify which features of a highly-ranked player differentiate him the most from others in his group.

5.2.1 Explaining the Rankings: identify strong points

Our method is as follows. For each group, we find the average feature vector of the players in the group, which we denote by $\overline{x_{g1}}, \overline{x_{g2}}, ..., \overline{x_{gm}}$ (see Figure 5.2). We denote the features of player i as $x_{i1}, x_{i2}, ..., x_{im}$. Then given a weight vector $(w_1, ..., w_m)$ for the logistic regression model of group g, the log-odds difference between player i and a random player in the group is given by

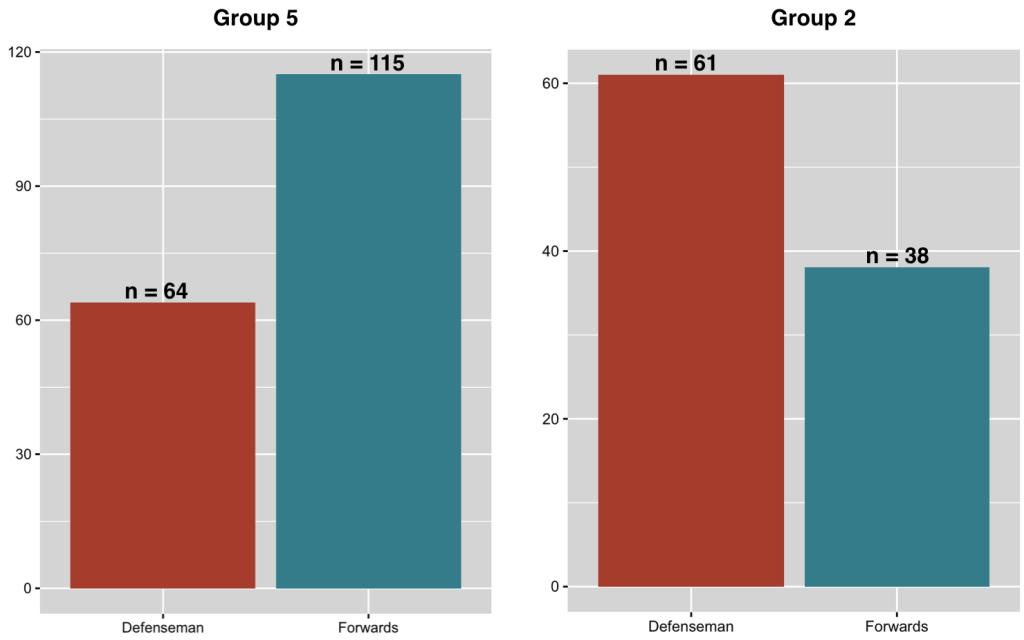

Figure 5.8: Distribution of Defenseman vs. Forwards in Group 5&2. The size is denoted as n.

$$\sum_{j=1}^{m} w_j(x_{ij} - \overline{x_{gi}})$$

We can interpret this sum as a measure of how high the model ranks player i compared to other players in his group. This suggests defining as the player's strongest features the x_{ij} that maximize $w_j(x_{ij} - \overline{x_{gi}})$, and as his weakest features those that minimize $w_j(x_{ij} - \overline{x_{gi}})$. This approach highlights features that are i) relevant to predicting future success, as measured by the magnitude of w_j, and ii) different from the average value in the player's group of comparables, as measured by the magnitude of $x_{ij} - \overline{x_{gi}}$.

5.2.2 Case Studies

Table 5.3 shows, for each group, the three strongest points for the most highly ranked players in the group. We see that the ranking for individual players is based on different features, even within the same group. The table also illustrates how the model allows us to identify a group of comparables for a given player. We discuss a few selected players and their strong points. The most interesting cases are often those where the ranking differs from CSS rank from scouts. We therefore discuss the groups with lower rank first.

Among the players who were not ranked by CSS at all, our model ranks *Kyle Cumiskey* at the top. Cumiskey was drafted in place 222, played 132 NHL games in his first 7 years, represented Canada in the World Championship, and won a Stanley Cup in 2015 with the Blackhawks. His strongest points were being Canadian, and the number of games played (e.g., 27 playoff games vs. 19 group average).

In the lowest CSS-rank group 6 (average 107), our top-ranked player *Brad Marchand* received CSS rank 80, even below his Boston Bruin teammate's Lucic's. Given his Stanley Cup win and success representing Canada, arguably our model was correct to identify him as a strong NHL prospect. The model highlights his superior play-off performance, both in terms of games played and points scored. Group 2 (CSS average 94) is a much weaker group. *Matt Pelech* is ranked at the top by our model because of his unusual weight, which in this group is unusually predictive of NHL participation. In group 4 (CSS average 86), *Sami Lepisto* was top-ranked, in part because he did not suffer many penalties although he played a high number of games. In group 3 (CSS average 76), *Brandon McMillan* is ranked relatively high by our model compared to the CSS. This is because in this group, left-wingers and shorter players are more likely to play in the NHL. In our ranking, *Milan Lucic* tops Group 5 (CSS average 71). At 58, his CSS rank is above average in this group, but much below the highest CSS rank player (Legein at 13). The main factors for the tree model are his high weight and number of play-off games played. Given his future success (Stanley Cup, NHL Young Stars Game), arguably our model correctly identified him as a star in an otherwise weaker group. The top players in Group 1 like *Sidney Crosby* and *Patrick Kane* are obvious stars, who have outstanding statistics even relative to other players in this strong group.

The overall message is that teams should look carefully at players who have features that make them stand out from the comparables in their group, even if they are not highly ranked by the CSS. This is especially true for players in group 6, i.e. with a high number of points and assists and a positive plus-minus.

Group	**Top Players**	**Strongest Points** ($\overline{x}$ = group mean)		
1	Sidney Crosby	rs_P 188 ($\overline{x}$ = 47)	rs_A 110 ($\overline{x}$ = 27)	CSS_rank 1 ($\overline{x}$ = 7)
	Patrick Kane	rs_P 154 ($\overline{x}$ = 47)	rs_A 87 ($\overline{x}$ = 27)	CSS_rank 2 ($\overline{x}$ = 7)
	Sam Gagner	rs_P 118 ($\overline{x}$ = 47)	po_A 22 ($\overline{x}$ = 4)	rs_A 83 ($\overline{x}$ = 27)
2	Matt Pelech	Weight 230 ($\overline{x}$ = 206)	CSS_rank 41 ($\overline{x}$ = 94)	rs_A 4 ($\overline{x}$ = 4)
	Adam Pineault	CSS_rank 25 ($\overline{x}$ = 94)	rs_P 8 ($\overline{x}$ = 6)	Height 73 ($\overline{x}$ = 74)
	Roman Wick	rs_P 10 ($\overline{x}$ = 6)	CSS_rank 36 ($\overline{x}$ = 94)	rs_PlusMinus 73 ($\overline{x}$ = 74)
3	A.J.Jenks	CSS_rank 20 ($\overline{x}$ = 76)	Weight 205 ($\overline{x}$ = 201)	Country USA
	Bill Sweatt	CSS_rank 27 ($\overline{x}$ = 76)	Position L	rs_PlusMinus -1 ($\overline{x}$ = -2)
	Brandon McMilan	CSS_rank 44 ($\overline{x}$ = 76)	Position L	Height 71 ($\overline{x}$ = 73)
4	Sami Lepisto	CSS_rank 25 ($\overline{x}$ = 86)	rs_GP 61 ($\overline{x}$ = 47)	rs_PIM 30 ($\overline{x}$ = 59)
	Linus Omark	CSS_rank 55 ($\overline{x}$ = 86)	Height 70 ($\overline{x}$ = 73)	DraftAge 20 ($\overline{x}$ = 19)
	Oscar Moller	CSS_rank 20 ($\overline{x}$ = 86)	Height 70 ($\overline{x}$ = 73)	rs_GP 68 ($\overline{x}$ = 47)
5	Milan Lucic	Weight 236 ($\overline{x}$ = 199)	po_GP 23 ($\overline{x}$ = 9)	CSS_rank 58 ($\overline{x}$ = 71)
	Michael Del Zotto	Position D	Country CAN	po_GP 15 ($\overline{x}$ = 9)
	Steven Delisle	Weight 234 ($\overline{x}$ = 199)	Country CAN	po_GP 19 ($\overline{x}$ = 9)
6	Brad Marchand	Country CAN	po_GP 25 ($\overline{x}$ = 19)	po_P 23 ($\overline{x}$ = 19)
	Mathieu Carle	Country CAN	CSS_rank 53 ($\overline{x}$ = 107)	rs_GP 67 ($\overline{x}$ = 65)
	Kyle Cumiskey	Country CAN	po_GP 27 ($\overline{x}$ = 19)	rs_GP 72 ($\overline{x}$ = 65)

Table 5.3: Strongest Statistics for the top players in each group. Underlined players are discussed in the text.

Chapter 6

NBA Results and Case Studies

Our results for basketball data are shown in this chapter. We first go through how we develop the M5P for our datasets, then display the learned groups and models. Last, we analyze the strongest points of exceptional players identified by our model tree.

6.1 Predictive Models and Evaluation

In this section, we first show the construction of M5P and predicted correlation results in comparison with actual draft order and a baseline method *(ordinary linear regression)*. Last, we analyze the learned groups in terms of the relationship between weights of group models and career PER.

6.1.1 Model Trees Construction

Different from NHL, most drafted basketball players would play at least one game in NBA (over 80% in our datasets depending on the draft year). Since there is no excess-zeros issue, we use a regression approach, which links predictors to the continuous response variable directly.

Our method is summarized as follows:

1. Build a tree whose leaves contain a linear regression model.
2. The tree assigns each player i to a unique leaf node I_i, with a linear regression model $m(I_i)$.
3. Use $m(I_i)$ to compute predicted career PER.

Figure 6.1 shows the M5P for all our datasets. The attribute *position* is placed at the root as the most import attribute, corresponding to a previous study which clusters players by their positions [20]. Players who are from Position_Union_1 have an average PER about 13, forming a better group compared to other players. Position_Union_1 is a list of positions from Table 3.6, automatically grouped by the M5P algorithm. The values are

shown in Appendix B. For players who are not from Position_Union_1, the tree takes *age* as the next splitting attribute. Players who are older than 24 years old and are not from Position_Union_1, belong to a less promising group with PER around 10. Then, the tree chooses *position* as another splitting point again, reflecting its significance. For players who not belong to Position_Union_1 but belong to Position_Union_2, with age smaller than or equal to 24, they form an average level group. The average PER value of players in this group is around 7. Position_Union_2 is a list of positions from Table 3.6, automatically grouped by the M5P algorithm. The values are shown in Appendix B. Lastly, the tree chooses *blk (blocks)* as the splitting feature. However, the size of Group 1 and Group 2 are relatively small (8 and 18).

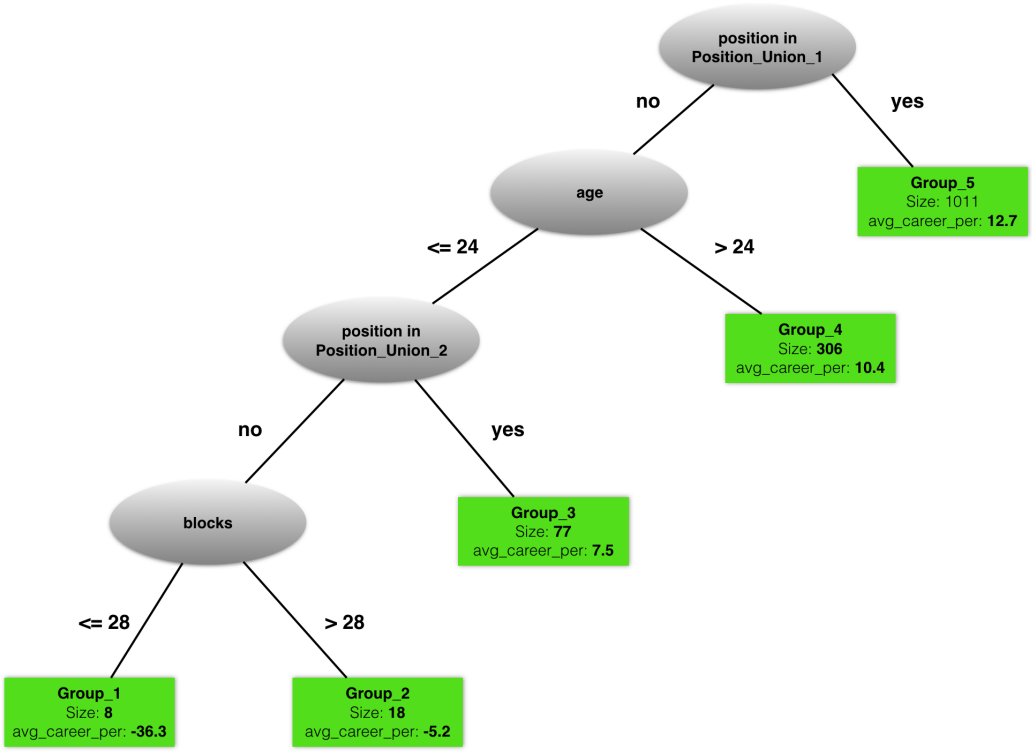

Figure 6.1: M5 regression trees for all the drafted players in 1985-2011 drafts. Each leaf defines a group of players. For each group, the figure shows the average career PER. Each leaf contains a linear regression model for its group (not shown). The group model assigns weights to *all* player features, and produces different predictions for players from the same group but with different features. The values of Position_Union_1 and Position_Union_2 are listed in Appendix D.

Figure 6.2 visualizes the distribution of our response variable, career PER, among each leaf node. Although the size of Group 5 is the largest, the variance between players' career PER in this group is smaller than other groups. In order, the strongest groups are 5, 4 and 3.

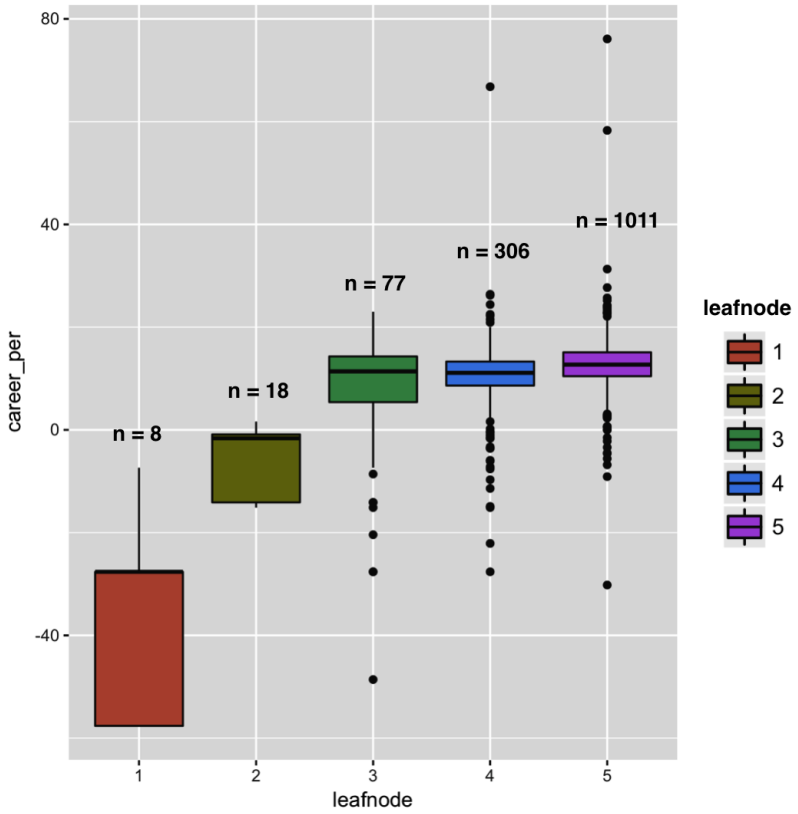

Figure 6.2: Box plots for career PER vs. leaf node. The group size is denoted as n.

6.1.2 Predictive Performance

To evaluate the predictive results, we use both Pearson Correlation and Spearman Rank Correlation to compare the predictive power of our tree models to the actual draft order and a baseline *(ordinary linear regression)*. The results are displayed in Table 6.1, which shows our model tree performs better than the actual draft order and ordinary linear regression.

6.1.3 Group Models

Figure 6.3 illustrates the weights of each learned group. The most relevant attributes which have the largest magnitude are bold and underlined. A positive weight means an increase in the covariate value predicts an increase in the predicted career PER, otherwise, it ~~brings decrease~~ in the predicted career PER. It is noteworthy that if tree splits on an attribute, the attribute is assigned a high-magnitude regression weight by the M5P for the relevant group, similar to the LMT.

For Group 1&2, *blk (blocks)* receives the largest positive weight, in contrast to the one in Group 3. This verifies the splitting node on *blk (blocks)* for Group 1 and Group 2 in the tree. In Group 3, the *trb_per (total rebounds per game)* has the largest positive weight, in contrast

Evaluation / Method	Pearson Correlation	Spearman Rank Correlation	RMSE
Draft Order	0.42	0.39	NaN
Linear Regression	0.45	0.40	7.14
Our Model Trees	**0.55**	**0.43**	**6.16**

Table 6.1: Comparison of predictive performance between draft order, linear regression and our tree models. *Bold indicates the best values*

with the negative weight in Group 5 (the strongest group). *Pts_per (points per game)* plays an important role in Group 4, while it impacts little in Group 1&2. In Group 5 (the strongest group), *ft (total number of free throws)* is the most import attribute. This result is in accord with the empirical experience in the NBA: 'free throws can normally be shot at a high percentage by good players' (`https://en.wikipedia.org/wiki/Free_throw`). *Age* receives the largest negative weight in Group 4, in comparison to Group 1&2 and Group 3. It corresponds to the splitting on *age* in model trees. The weight of *fta (free throw attempts)* is negative among all groups, especially in Group 4 and Group 5.

Metrics / Group	age	position	g	mp	ft	fta	trb	ast	blk	pts	ah
1	-0.04	10.95 0.05 0.07	22.78	all: 0 per: 0	all: 2.25 per: -0.14	-1.89	all: 0.21 per: 25.92	all: 0 per: 0.11	**73.31**	all: 0 per: 0.31	0.04
2	-0.04	10.95 0.05 0.07	22.78	all: 0 per: 0	all: 2.25 per: -0.14	-1.89	all: 0.21 per: 25.92	all: 0 per: 0.11	**51.55**	all: 0 per: 0.31	0.04
3	-0.04	6.95 0.05 0.07	10.22	all: 0 per: 0	all: 2.25 per: -0.14	-1.89	all: 0.21 per: **11.55**	all: 0 per: 0.11	1.53	all: 0 per: 0.31	0.04
4	-34.35	1.92 0.05 0.07	12.89	all: 0 per: 5.08	all: 2.25 per: -0.14	-18.63	all: 0.21 per:0	all: 0 per: 0.11	9.51	all: -11.24 per: **17.91**	0.04
5	-2.39	0.36 0.83 0.53 1.34	-6.54	all: 4.27 per: -4.91	all: **10.57** per: -5.33	-10.69	all: 10.05 per: -4.95	all: 5.66 per: 0.04	2.41	all: 2.92 per: 4.01	1.03

Figure 6.3: Weights Illustration. The largest weights are in bold. The smallest weights are underlined.

6.2 Case Studies: Exceptional Players and Strongest Points

Similar to discovering NHL exceptional players and their strengths, we apply the same method to NBA datasets to find the strongest points of exceptional players in each group. We find two interesting cases: *Matt Geiger* and *Dejuan Blair*, discussed below.

Table 6.2 shows the top players in each group together with their strongest points. In Group 1, the weakest group, players whose strongest points are in *trb_per (total rebounds*

per game) and *blk (blocks)* are ranked higher compared to the rest of players. Group 2 has almost the same distribution of strongest points for exceptional players as Group 1. Group 3 is a relatively average group. In this group, *Shawn Bradley* is ranked as the greatest player. He is one of the most controversial players in the NBA draft history, well-known for his advantageous height. However, according to the results of our method, his strongest points is in his blocks ability, which is probably related to his height. This finding is in accord with his career performance in NBA. *Benoit Benjamin* in Group 4 has the $3rd$ overall pick in his draft year. According to Table 6.2, he is excellent in scoring points and free throws. Group 5 is the strongest group by our model. The most prestigious player *Chris Webber* identified by our model is a superstar in the NBA. He is a five-time NBA All-Star, a five-time All-NBA Team member, and NBA Rookie of the Year *(1994)*. His strongest points in his pre-draft years are identified as *trb (total rebounds)*, *mp (minutes played)* and *ast (assists)*.

Our model also discovers players who should have received a better draft order than their actual draft order. *Matt Geiger* in Group 4, was picked at $42th$ in 1992, after *Todd Day (*$8th$*), Bryant Stith (*$13th$*) Anthony Peeler (*$15th$*)*. However, his career PER is 15.2, above those players drafted before him. A more recent case is *Dejuan Blair*, who has the $37th$ overall draft pick in 2009, taken after *Jordan Hill (*$8th$*), Ricky Rubio (*$5th$*)*, but he obtained almost the same career PER as them. In addition, Blair joined two-time 'The Basketball Tournament' defending champion Overseas Elite in summer 2017 and his team, Overseas Elite won its third straight 'The Basketball Tournament' championship with a $86-83$ victory over Team Challenge ALS on ESPN (`https://en.wikipedia.org/wiki/DeJuan_Blair`). The statistics of these two underestimated players are shown in Table 6.3.

The overall message is that teams should look carefully at players who have features that make them stand out from the comparables in their group, even if they have bad draft picks in their draft year.

Group	**Top Players**	**Strongest Points** ($\overline{x}$ = group mean)		
1	Paccelis Morlende	trb_per 18.3 ($\overline{x}$ = 6.12)	blk 27 ($\overline{x}$ = 25)	ft 138 ($\overline{x}$ = 121.5)
	Sani Becirovic	trb_per 18.3 ($\overline{x}$ = 16.2)	blk 27 ($\overline{x}$ = 25)	ft 138 ($\overline{x}$ = 121.5)
	Cenk Akyol	trb_per 16.4 ($\overline{x}$ = 6.12)	blk 26 ($\overline{x}$ = 25)	g 31 ($\overline{x}$ = 32)
2	Latavious Williams	blk 46 ($\overline{x}$ = 34)	trb_per 7.19 ($\overline{x}$ = 6.41)	fg_per 0.51 ($\overline{x}$ = 0.50)
	Ryan Richards	blk 46 ($\overline{x}$ = 34)	trb_per 7.19 ($\overline{x}$ = 6.41)	fg_per 0.51 ($\overline{x}$ = 0.50)
	Petteri Koponen	blk 37 ($\overline{x}$ = 34)	fg_per 0.51 ($\overline{x}$ = 0.50)	height 193 ($\overline{x}$ = 203)
3	Shawn Bradley	blk 177 ($\overline{x}$ = 32)	trb_per 7.7 ($\overline{x}$ = 6.6)	position Center
	Kosta Koufos	position Center	g 37 ($\overline{x}$ = 32)	trb_per 6.7 ($\overline{x}$ = 6.6)
	Paulao Prestes	position Center	trb_per 7.19 ($\overline{x}$ = 6.6)	g 33 ($\overline{x}$ = 32)
4	Benoit Benjamin	ft 172 ($\overline{x}$ = 116)	pts_per 21.5 ($\overline{x}$ = 16.86)	age 20 ($\overline{x}$ = 21.38)
	Hersey Hawkins	ft 284 ($\overline{x}$ = 116)	pts_per 36.3 ($\overline{x}$ = 16.86)	age 21 ($\overline{x}$ = 21.38)
	Chris Kaman	ft 206 ($\overline{x}$ = 116)	pts_per 22.4 ($\overline{x}$ = 16.86)	age 20 ($\overline{x}$ = 21.38)
5	Larry Johnson	ft 162 ($\overline{x}$ = 122)	trb 380 ($\overline{x}$ = 214)	ast 104 ($\overline{x}$ = 84)
	Anfernee Hardaway	trb 273 ($\overline{x}$ = 214)	ast 204 ($\overline{x}$ = 84)	mp 1196 ($\overline{x}$ = 929)
	Chris Webber	trb 362 ($\overline{x}$ = 84)	mp 1143 ($\overline{x}$ = 929)	ast 90 ($\overline{x}$ = 84)

Table 6.2: NBA exceptional players in each group and their strongest points [9].

name	draft_year	draft pick	career PER	predicted PER	comparables (career_per, pick)
Matt Geiger	1992	42	15.2	11.7	Anthony Peeler (12.9, 15th)
Dejuan Blair	2009	37	16.5	17.2	Jordan Hill (16.3, 8th)

Table 6.3: Underestimated players.

Chapter 7

Conclusion

We have proposed building regression model trees for ranking draftees in the NHL & NBA, or other sports, based on a list of player features and performance statistics. The model tree groups players according to the values of discrete features, or learned thresholds for continuous performance statistics. Each leaf node defines a group of players that is assigned its own regression model. Tree models combine the strength of both regression and cohort-based approaches, where player performance is predicted with reference to comparable players. An obvious approach is to use a linear regression tree for predicting dependent variable, ~~like what~~ we did with NBA datasets. Also, this regression tree method can also be applied to the NHL datasets. However, we found that a linear regression tree performs poorly in the NHL due to the excess-zeros problem (many draft picks never play any NHL game). Instead, we introduced the idea of using a logistic regression tree to predict whether a player plays any NHL game within 7 years. Players are ranked according to the model tree probability that they play at least 1 game.

Key findings include the following. 1) The model tree ranking correlates well with the actual success ranking according to the actual number of games played, better than draft order. 2) The model tree can highlight the exceptionally strongest points of draftees that make them stand out compared to the other players in their group.

Tree models are flexible and can be applied to other prediction problems to discover groups of comparable players as well as predictive models. For example, we can predict future NHL success from past NHL success, similar to Wilson [38] who used machine learning models to predict whether a player will play more than 160 games in the NHL after 7 years. Another direction is to apply the model to other sports, for example drafting for the National Football League.

Bibliography

[1] J. Albert, M. E. Glickman, T. B. Swartz, and R. H. Koning. *Handbook of Statistical Methods and Analyses in Sports*. CRC Press, 2017.

[2] C. Blake and C. Merze. UCI repository of machine learning databases. 1998.

[3] L. Breiman, H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. 1984.

[4] D. Cervone, A. D'Amour, L. Bornn, and K. Goldsberry. Pointwise: Prediting points and valuing decisons in real time with nba optical tracking data. In *MIT Sloan Sports Analytics Conference*, 2016.

[5] D. Coates and B. Oguntimein. The length and success of nba careers: Does college production predict professional outcomes? *International Journal of Sport Finance*, 5(1), 2008.

[6] E. C. Fieller, H. O. Hartley, and E. S. Pearson. Tests for rank correlation coefficients. i. *Biometrika*, 44:470–481, 1957.

[7] R. Found. Goal-based metrics better than shot-based metrics at predicting hockey success. Technical report, 2016.

[8] E. Frank, M. Hall, and I. Witten. *The Weka Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*. Fourth edition, 2016.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[10] I. Fyffe. Evaluating central scouting. Technical report, 2011.

[11] A. C. Greene. The success of nba draft picks: Can college career predict nba winners. Master's thesis, St. Cloud State University, 2015.

[12] M. Hall, E. Frank, G. Homes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: An update. sigkdd explorations. 11:10–18, 2009.

[13] B. James and J. Henzler. *Win Shares*. STATS, Inc., 2002.

[14] E. Joyce and C. Lawrence. Blending old and new: How the florida panthers try to predict future performance at the nhl entry draft, 2017.

[15] E. H. Kaplan, K. Masri, and J. T. Ryan. A markov game model for hockey: Manpower differential and win probability added. *INFOR: Information Systems and Operational Research*, 52(2):39–50, 2014.

[16] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31:249–268, 2007.

[17] P. A. Lachenbruch. Analysis of data with excess zeros. *Statistical Methods in Medical Research*, 11, 2002.

[18] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Kluwer Academic Publishers*, 2006.

[19] W.-Y. Loh. *GUIDE User Manual*. University of Wisconsin-Madison, 2017.

[20] D. Lutz. A cluster analysis of nba players. In *MIT sloan Sports Analytics Conference*, 2012.

[21] B. Macdonald. An improved adjusted plus-minus statistic for nhl players. In *MIT sloan Sports Analytics Conference*, 2011.

[22] D. Malerba, A. Appice, M. Ceci, and M. Monopoli. Trading-off local versus global effects of regression nodes in model trees. In *In: M.-S. Hacid, Z. W. Ras, D. A. Zighed, and Y. Kodratoff (eds.): ISMIS 2002.*, page 393–402. Springer-Verlag, 2002.

[23] B. Mazique. Dissecting which position is most important towards winning an nba championship. Technical report, 2012.

[24] D. Pregibon. *private communication*. 1989, 1992.

[25] J. R. Quinlan. Learning with continuous classes. *World Scientific*, pages 343–348, 1992.

[26] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[27] A. Ryder. Poisson toolbox, a review of the application of the poisson probability distribution in hockey. Technical report, Hockey Analytics, 2004.

[28] M. Schuckers and J. Curro. Total hockey rating (thor): A comprehensive statistical rating of national hockey league forwards and defensemen based upon all on-ice events. In *MIT sloan Sports Analytics Conference*, 2013.

[29] M. E. Schuckers and LLC Statistical Sports Consulting. Draft by numbers: Using data and analytics to improve national hockey league(NHL) player selection. In *MIT sloan Sports Analytics Conference*, 2016.

[30] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(379-423), 1948.

[31] J. Sill. Improved nba adjusted +/− using regularization and out-of-sample testing. In *MIT sloan Sports Analytics Conference*, 2010.

[32] N. Silver. *PECOTA 2004: A Look Back and a Look Ahead*, pages 5–10. New York: Workman Publishers, 2004.

[33] A. Thomas, S. Ventural, S. Jensen, and S. Ma. Competing process hazard function model for player ratings in ice hockey. *The Annals of Applied Science*, 7(2):1497–1524, 2013.

[34] A. C. Thomas. The impact of puck possession and location on ice hockey strategy. *Journal of Quantitative Analysis in Sports*, 2(1), 2006.

[35] A. C. Thomas and S. Ventura. The highway to war: Defining and calculating the components for wins above replacement. 2015. Available at `https://aphockey.files.wordpress.com/2015/04/sam-war-1.pdf`.

[36] P. Tingling, K. Masri, and M. Martell. Does order matter? an empirical analysis of nhl draft decisions. *Sport, Business and Management: an International Journal*, (2):155–171, 2011.

[37] J. Weissbock. Draft analytics: Unveiling the prospect cohort success model. Technical report, 2015.

[38] D. R. Wilson. Mining nhl draft data and a new value pick chart. Master's thesis, University of Ottawa, 2016.

Appendix A

Spearman Rank Correlation

Spearman's correlation measures the relevance and direction of monotonic association between two variables [6]. The standard formula for calculating is based on the squared rank differences:

(1) $p = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$, formula for no tied ranks. n = number of ranks, d_i = difference in paired ranks. This is the formula we applied in Table 5.1.

In the case of tied ranks, an alternative is to use the Pearson correlation between ranks, computed by the formula:

(2) $p = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y}^2)}}$, where x_i = rank of player i according to ranking x, ditto for y_i.

Players who have played zero NHL games are tied when ranked by the number of NHL games; this is the only case of ties. Table A.1 repeats the calculation of Table 5.1 using the Pearson correlation among ranks (2) rather than the squared rank differences (1). With this measure also, the model ranking correlates more highly with actual number of games played than the team draft order.

Training Data NHL Draft Years	Out of Sample Draft Years	Draft Order Pearson Correlation	Tree Model Pearson Correlation
1998, 1999, 2000	2001	0.43	0.69
1998, 1999, 2000	2002	0.45	0.72
2004, 2005, 2006	2007	0.48	0.60
2004, 2005, 2006	2008	0.51	0.58

Table A.1: Pearson Correlation of NHL ranks.

Appendix B

LogitBoost Algorithm Pseudocode

LogitBoost (J classes)

1. Start with weights $w_{ij} = 1/n,\ i = 1, \ldots, n,\ j = 1, \ldots, J,\ F_j(x) = 0$ and $p_j(x) = 1/J\ \ \forall j$

2. Repeat for $m = 1, \ldots, M$:

   (a) Repeat for $j = 1, \ldots, J$:

      i. Compute working responses and weights in the jth class

$$z_{ij} = \frac{y^*_{ij} - p_j(x_i)}{p_j(x_i)(1 - p_j(x_i))}$$

$$w_{ij} = p_j(x_i)(1 - p_j(x_i))$$

      ii. Fit the function $f_{mj}(x)$ by a weighted least-squares regression of z_{ij} to x_i with weights w_{ij}

   (b) Set $f_{mj}(x) \leftarrow \frac{J-1}{J}(f_{mj}(x) - \frac{1}{J}\sum_{k=1}^{J} f_{mk}(x)),\ \ F_j(x) \leftarrow F_j(x) + f_{mj}(x)$

   (c) Update $p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$

3. Output the classifier $\underset{j}{\text{argmax}}\ \ F_j(x)$

Figure B.1: LogitBoost Algorithm [9].

The variable y^*_{ij} represents the observed class probabilities for instance x_i. The $p_j(x_i)$ are the estimates of the class probabilities for an instance x given by the model so far. M in the pseudocode is the set of data points and J denotes the set of class variables. The $f_{mj}(x)$ can be considered as a weak classifier (not necessarily linear) to improve the ultimate classifier $F_j(x)$.

Appendix C

C4.5 Splitting Strategy

The splitting strategy of C4.5 is summarized as follows.

1. Check for the following cases:
   (a) When all the samples in the list belong to the same class, a leaf node for that class is created;
   (b) When none of the attributes provide any information gain, a decision node higher up the tree is created using the expected value of that class;
   (c) When encountering a previously-unseen class, a decision node higher up the tree is created using the expected value of that class.
2. For each attribute a, find the normalized information gain ratio from splitting on a.
3. Let a_best be the attribute with the highest normalized information gain.
4. Create a decision node that splits on a_best.
5. Recur on the subsets obtained by splitting on a_best, and added these node as children of the node.

In the above algorithm, the information gain ratio (IGR) is calculated by information gain (IG) dividing intrinsic value (IV). The IG and IV are defined as follows:

$$IG(Ex, a) = H(Ex) - \sum_{v \in values(a)} \left(\frac{|x \in Ex | value(x,a) = v|}{|Ex|} \cdot H(x \in Ex | value(x, a) = v) \right)$$

$$IV(Ex, a) = - \sum_{v \in values(a)} \left(\frac{|x \in Ex | value(x,a) = v|}{|Ex|} \cdot \log_2 \left(\frac{|x \in Ex | value(x,a) = v|}{|Ex|} \right) \right)$$

where Ex is the set of all training examples, $value(x, a)$ denotes the value of a specific example x for an attribute a and $value(a)$ function defines the set of all possible values of the attribute a. In the above formulas, H represents the entropy, a measure of unpredictability of data values [30].

Appendix D

Values of Position_Union in the NBA Tree

The values of *Position_Union_1* and *Position_Union_2* in the M5P for the NBA data (Figure 6.1) are as follows:

Position_Union_1 = (Small Forward, Point Guard and Shooting Guard and Small Forward, Power Forward and Shooting Guard and Small Forward, Power Forward, Small Forward and Point Guard and Shooting Guard, Small Forward and Power Forward, Point Guard, Shooting Guard and Small Forward and Point Guard, Point Guard and Shooting Guard, Small Forward and Shooting Guard, Small Forward and Shooting Guard and Power Forward, Small Forward and Power Forward and Center, Shooting Guard and Power Forward,Power Forward and Small Forward, Shooting Guard and Point Guard,Shooting Guard and Small Forward, Shooting Guard and Small Forward and Power Forward, Center and Power Forward, Power Forward and Center, Point Guard and Small Forward and Shooting Guard,Small Forward and Power Forward and Shooting Guard, Small Forward and Shooting Guard and Point Guard, Center and Small Forward and Power Forward, Power Forward and Center and Small Forward, Small Forward and Center and Power Forward, Center and Power Forward and Small Forward, Shooting Guard and Power Forward and Small Forward)

Position_Union_2 = (Center-Forward, Center and Small Forward, Small Forward and Center, Center, Shooting Guard and Point Guard and Small Forward, Power Forward and Small Forward and Shooting Guard, Shooting Guard, Small Forward, Point Guard and Shooting Guard and Small Forward, Power Forward and Shooting Guard and Small Forward, Power Forward, Small Forward and Point Guard and Shooting Guard, Small Forward and Power Forward, Point Guard, Shooting Guard and Small Forward and Point Guard, Point Guard and Shooting Guard, Small Forward and Shooting Guard, Small Forward and Shooting Guard and Power Forward, Small Forward and Power Forward and Center, Shooting Guard and Power Forward, Power Forward and Small Forward, Shooting Guard and Point Guard, Shooting Guard and Small Forward, Shooting Guard and Small Forward and Power

Forward, Center and Power Forward, Power Forward and Center, Point Guard and Small Forward and Shooting Guard, Small Forward and Power Forward and Shooting Guard, Small Forward and Shooting Guard and Point Guard, Center and Small Forward and Power Forward, Power Forward and Center and Small Forward, Small Forward and Center and Power Forward, Center and Power Forward and Small Forward, Shooting Guard and Power Forward and Small Forward)

Appendix E

Datasets and Code

Our data and main code can be retrieved from our GitHub repositories. Their stored places are summarized in Table E.1.

Code for crawling the pre-draft data of draftees in the NHL	https://github.com/sfu-cl-lab/Yeti-Thesis-Project/blob/master/Decision_Trees/LMT/python_code/crawl_predraft_NHL_stats.py
Code for crawling the career performance data of players in the NHL	https://github.com/sfu-cl-lab/Yeti-Thesis-Project/blob/master/Decision_Trees/LMT/python_code/crawl_NHL_career_stats.py
Code for crawling the NBA drafted players' college and career data	https://github.com/sfu-cl-lab/Yeti-Thesis-Project/blob/master/NBA_work/crawl_basketball_stats.py
Code for calculating NHL draftees' strongest points	https://github.com/sfu-cl-lab/Yeti-Thesis-Project/blob/master/Decision_Trees/LMT/python_code/NHL_strongest_points.py
Ice hockey datasets	https://github.com/liuyejia/Model_Trees_Full_Dataset
Basketball datasets	https://github.com/sfu-cl-lab/Yeti-Thesis-Project/blob/master/NBA_work/joined_drafted_all_players_original.csv

Table E.1: Our datasets and code.