EVALUATING HOW NHL PLAYER SHOT SELECTION IMPACTS EVEN-STRENGTH GOAL OUTPUT OVER THE COURSE OF A FULL SEASON

By

Elliott Barinberg, B.Sc CS

A thesis submitted to the Graduate Committee of

Ramapo College of New Jersey in partial fulfillment

of the requirements for the degree of

Master of Science

in Data Science

August 2023

Committee Members:

Scott Frees, Advisor

Amanda Beecher, Reader

Debbie Yuster, Reader

COPYRIGHT

by

Elliott Barinberg

2023

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor and readers for their tremendous feedback and patience. I could not have undertaken this work without my defense committee who generously provided feedback, knowledge, and expertise not only this summer but throughout my time at Ramapo College of New Jersey both as an undergraduate Computer Science student, and now as a member of a graduate program.

I am also grateful to the NHL and the New Jersey Devils for inspiring my love of hockey. Thanks should also go to the X (Twitter) NHL analytics community for the belief and inspiration to make an impact in this field.

Lastly, I would be remiss in not mentioning my family, especially my parents and siblings, as well as my friends, who share my passion for hockey and who have inspired and supported me throughout my work.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

CHAPTER

I. INTRODUCTION

II. BACKGROUND

Basic Hockey Lesson

Why the NHL?

A Brief History of Sports Analytics and Their NHL Applications

The Definition of the Modern NHL

III. DATASET

Data Collection

Data Shape

Data Features of Note

Data Manipulation

Data Filtration

IV. MODELING THE DATA

Features

Dummy Model

Linear Regression

Logistic Regression

Random Forest

Gradient Boosted Regression

Cumulative Model Comparison

Model Validation

V. RESULTS

VI. CONCLUSION

VII. FUTURE WORK

LIST OF TABLES

TABLE … PAGE

t_1: Games Table … 26

t_2: Players Table … 26-27

t_3: Teams Table … 27

t_4: Rosters Table … 28

t_5: Events Table … 28-29

t_6: Games Table .. 29

t_7: Event Counts … 30-31

t_8: Secondary Shot Counts … 31

t_9: Random Forest Final Hyperparameters .. 56

t_10: Gradient Boosting Final Hyperparameters … 62

t_11: Top 5 Model Outputs … 77

LIST OF FIGURES

FIGURE … PAGE

f_1: The Dimensions of a Hockey Rink … 16

f_2: The Dump and Chase Strategy … 17

f_3: Lord Stanley and an Early Iteration of the Cup … 18

f_4: Jim Corsi and His Famous Mustache … 20

f_5: Terry Sawchuk, Former NHL Goalie … 23

f_6: The Event Data Location Grid … 30

f_7: Flipping Shot Locations to One Side of the Ice … 33

f_8: Shot Totals Organized by Year … 38

f_9: Shot Totals Organized by Year Without 2019 & 2020 … 39

f_10: Dummy Model Area Under the Curve … 39

f_11: Dummy Model Cumulative Expected Goals vs Reality Histogram … 40

f_12: Dummy Model Cumulative Expected Goals vs Reality Error Bars … 41

f_13: Relationship Between Dependent Variable and Independent Variables … 42

f_14: Multicollinearity Analysis on Continuous Variables … 44

f_15: Homoscedasticity Analysis Between Predictions and Residuals … 45

f_16: Normality of Residual Values … 45

f_17: Normality of Residual Values … 45

f_18: Linear Regression Area Under the Curve … 46

f_19: Linear Regression Cumulative Expected Goals vs Reality Histogram .. 47

f_20: Linear Regression Cumulative Expected Goals vs Reality Error Bars … 48

f_21: Multicollinearity Check for Logistic Regression … 50

f_22: Target Variable Binary Analysis … 51

f_23: Relationship Between Independent Variables and Log-Odds … 52

f_24: Logistic Regression Area Under the Curve … 53

f_25: Logistic Regression Cumulative Expected Goals vs Reality Histogram … 54

f_26: Logistic Regression Cumulative Expected Goals vs Reality Error Bars … 55

f_27: Sample of Multiple Trees in a Single Predictive Forest … 56

f_28: A Sample Tree from Within the Predictive Forest … 58

f_29: Random Forest Area Under the Curve … 59

f_30: Random Forest Regression Cumulative Expected Goals vs Reality Histogram … 59

f_31: Random Forest Regression Cumulative Expected Goals vs Reality Error Bars … 61

f_32: Gradient Boosted Model Decision Tree … 62

f_33: Gradient Boosted Regression Model Area Under the Curve … 63

f_34: Gradient Boosted Regression Cumulative Expected Goals vs Reality Histogram … 64

f_35: Gradient Boosted Regression Cumulative Expected Goals vs Reality Error Bars … 65

f_36: Cumulative Model Comparison Histograms … 66

f_37: Dummy Model Difference to MoneyPuck per Shot … 68

f_38: Linear Regression Model Difference to Moneypuck per Shot … 69

f_39: Logistic Regression Model Difference to Moneypuck per Shot … 69

f_40: Random Forest Regression Model Difference to Moneypuck per Shot … 70

f_41: Gradient Boosted Regression Model Difference to Moneypuck per Shot … 71

f_42: Connor McDavid Heatmap … 72

f_43: David Pastrnak Heatmap … 73

f_44: Pavel Zacha & Erik Haula Comparison Heatmaps … 74

f_45: Yegor Sharangovich & Tyler Toffoli Impact Heatmaps … 75

f_46: Cale Makar & Jonas Siegenthaler Impact Heatmaps … 76

f_47: McDavid's Net-Front Impact … 78

ABSTRACT

Within this thesis work, the applications of data collection, machine learning, and data visualization were used on National Hockey League (NHL) shot data collected between the 2014-2015 season and the 2022-2023 season. Modeling sports data to better understand player evaluation has always been a goal of sports analytics. In the modern era of sports analytics the techniques used to quantify impacts on games have multiplied. However, when it comes to ice hockey all the most difficult challenges of sports data analysis present themselves in trying to understand the player impacts of such a continuously changing game-state. The methods developed and presented in this work serve to highlight those challenges and better explain a player's impact on goal scoring for their team.

Throughout this work there are multiple kinds of modeling techniques used to try to best demonstrate a player's impact on goal scoring as a factor of all the elements the player is capable of controlling. We try to understand which players have the best offensive process and impact on goal-scoring by caring about the merit of the offensive opportunities they create. It is important to note that these models are not intended to re-create the results seen in reality, although reality and true results are used to evaluate the outputs.

This process used data scraping to collect the data from the NHL public application programming interface (API). Data cleansing techniques were applied to the collected data, yielding custom data sets which were used for the corresponding models. Data transformation techniques were used to calculate additional factors based upon the data provided, thus creating additional data within the training and testing datasets. Techniques including but not limited to linear regression, logistic regression, random forests and extreme gradient boosted regression were all used to attempt to model the true possibility of any particular even-strength event being

a goal in the NHL. Then, using formulaic approaches the individual event model was extrapolated upon to draw larger conclusions. Lastly, some unique data visualization techniques were used to best present the outputs of these models. In all, many experimental models were created which have yielded a reproducible methodology upon which to evaluate the results of any NHL player impact upon goal scoring over the course of a season.

I. INTRODUCTION

In this project the goal was two-fold. The first objective was to determine how to evaluate the events that a player is involved in. The second objective was to be able to create a methodology by which the evaluation of the first objective can be quantified and compared across players. In order to accomplish this, there was a need to take the individual events which occur throughout a season of NHL hockey and understand the specifics of each event. Then, once there is an understanding of the events, they are grouped by the player responsible for each event. This is what gives the ability to compare a player's outputs over the course of a season to another player. Eventually, the hope is to understand which players make and take the best opportunities on the ice. If the probability of scoring an individual event could be understood and aggregated over a season it could potentially help to better understand the contributions of a player. Normally when the public looks at a player's statistics the number of goals they score is on display, but that lacks critical context around the situation and significance of the goals that were scored. Was the net empty? Was it a good shot to take? Or was it the result of a lucky bounce which trickled past the goaltender? By understanding the methodology behind taking good shots we aim to create a system that answers those questions without needing to analyze each individual's volume of events manually.

Every year scouts and general managers in the NHL, as well as the fans, evaluate players. There are many statistics used to evaluate a player, including watching tape. To give context to the statistics, evaluating a player strictly on the number of goals they scored does not always provide an accurate picture of the effectiveness of said player in comparison to their peers. However, watching every NHL player's shots and making notes on their merit and results is not a feasible task for any one person. This project is looking to bridge that gap and provide the

information on which players are most effective in generating the most efficient scoring opportunities. Thus, allowing people looking to evaluate such players to have a more effective analysis and understanding of any particular player as they compare to their peers.

Within this body of work, each shot was modeled individually and with respect to the events and situation of a particular game. Then the model outputs were compiled into information which is aggregated by the player to evaluate over the course of an NHL season. Lastly, visualizations were created in order to best demonstrate the effectiveness of particular players and their ability to create high-danger chances in the NHL.

II. BACKGROUND

Basic Hockey Lesson

Ice hockey, or simply hockey, is a fast, physically demanding, and challenging sport.

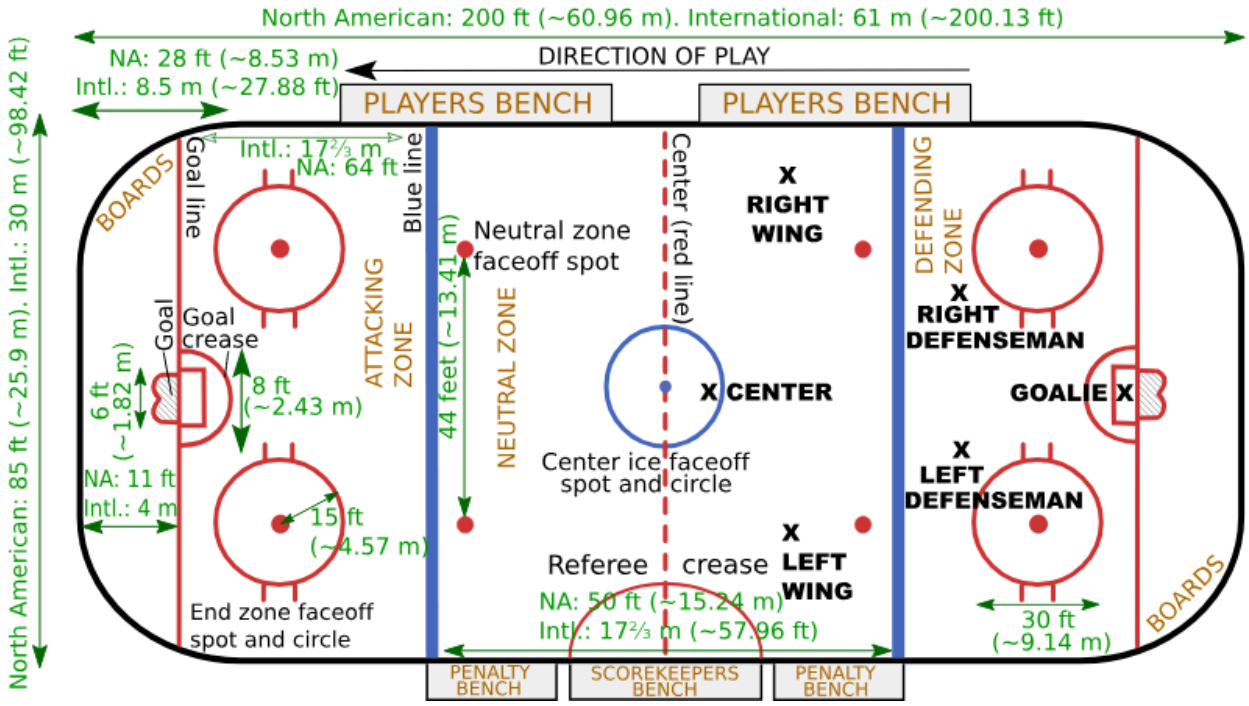

Fig f_1: The Dimensions of a Hockey Rink

As demonstrated in Figure 1, while there are some minor differences between ice hockey in North America when compared to the international style of ice hockey, this paper will focus on the ability to quantify the results as related to hockey in North America, specifically in the NHL. NHL hockey is played on a rink which is 200 ft in length and 85 ft in width [r_2].

Hockey is a continuous sport and substitutions occur during active play. Other major sports such as American football, baseball, and basketball all substitute players between plays as needed either by a player's physical condition or by the rules set within the game. In hockey however, substitutions are done during the active course of play. So if a player is tired or hurt, they need to get to their respective bench and once within 5ft of the bench their substitute can

enter the game as they make their way off the ice. It is worth noting if a player is truly incapacitated during the course of play, the game will be stopped to attempt to care for the player. There are other sports which are continuous like lacrosse or soccer which do not have the same concept of active substitutions as well. Making this on-the-fly style substitution very unique and presenting some difficult concepts to data scientists. However, hockey differs from those sports in other ways as well.

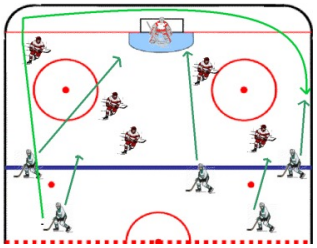

Fig f_2: The Dump and Chase Strategy

As shown in Figure 2, the NHL field of play is surrounded by walls which are active parts of the play. When playing soccer, balls go out of bounds all the time, but in the NHL the boundaries are used as part of the game and its strategies. Meaning boundary stoppages are rarer than in other sports [r_4]. However, there is one more crucial factor which separates NHL hockey from other sports. NHL hockey is played on ice[r_4] which is unlike other sports and also differs from other variations of hockey. So in addition to all the rules outlined which an NHL player must master, they also need to be expert ice skaters to be good NHL hockey players.

NHL teams carry rosters usually consisting of 12 forwards, 6 defensemen, and 2 goalies totalling rosters of 20 active players [r_4]. That is 4 offensive lines of 3 players, and 3 pairs of 2 defensemen. This is also sometimes referred to as 18 skaters and 2 goalies. Of these 20 active players, all but the backup goalie expect to find themselves on the ice during the game. Most of the game is played with 3 forwards, 2 defenseman, and 1 goalie on the field of play. Without

accounting for injuries or stoppages, substitutions occur as the skaters get tired. Usually the skaters use a rotation pattern to decide which set of players will step onto the ice next, with the coach occasionally interrupting the rotation as they see fit.

Why The NHL?

The NHL is widely recognized as the best professional hockey league in the world. It certainly holds that title in North America. It is worth mentioning that it does have international competition in the form of the KHL in Russia, the SHL in Sweden, and Liiga in Finland. Of the international leagues the KHL is regarded as the best, but just by looking at the finances of the league it is clear that the KHL organizations operate on a salary cap roughly 10% of comparable NHL organizations. This has led the best players from those countries to come to North America and to play in the NHL. Thus reinforcing the widely held belief that it is the highest quality ice hockey league in the world.

Fig f_3: Lord Stanley and an Early Iteration of the Cup

If that was not enough, the NHL has one of the most historic and amazing traditions with respect to immortalizing the winning teams and their contributions to the sport of hockey. In 1892 the Stanley Cup (named after Lord Stanley of Preston, the Governor General of Canada) was commissioned and in 1893 it was first awarded. Unlike other professional sports the Stanley Cup is not made every year to be awarded. In the past the winners would keep the cup until it

was awarded to someone else. Nowadays, access to the cup is given to the winners during the offseason and for some limited times during the next season. Part of what makes this trophy so unique is that the winning contributors of every organization have their name engraved on the layers at the bottom of the cup. Every 13 years one of these layers is removed from the cup and stored for display in the Hockey Hall of Fame [r_12]. There are plenty of other superstitions and traditions which surround this trophy. But the uniqueness of this trophy, and its lore, is what makes it so special and not just another "piece of metal" which is what the Major League Baseball commissioner, Rob Manfred, once referred to the World Series trophy as being [r_17].

A Brief History of Sports Analytics and Their NHL Applications

Sports analytics are just a fancy way to define a collection of relevant statistics that can provide information around a competitive advantage to a team or individual. Of major sports the first to really embrace sports analytics was undoubtedly baseball. In baseball and beyond, Bill James is widely regarded as a founding father of sports analytics for helping bring analytical thinking to the MLB. This approach was widely popularized by the 2011 film, *Moneyball*, in which the Oakland Athletics General Manager Billy Beane (played by Brad Pitt) relies heavily on the use of analytics to build a major league baseball roster on a minimal budget. The film was inspired by and based upon the 2002 Athletics [r_13].

Since the early days of sports analytics, many sports have taken these concepts and evolved them into new ones. They have stretched from baseball to basketball, football, soccer, golf, and hockey[r_13]. As mentioned in the hockey lesson the continuous low-event nature of hockey makes it significantly more challenging to analyze, which is part of the reason that it took so long for the NHL to begin relying on sports analytics. The NHL has kept statistics since its inception, however is a relatively new adopter of the analytics-based decision making process. In

2014 Kyle Dubas became the assistant general manager in Toronto, becoming the first member of management in the NHL with a largely analytical background, having never played a professional game. This was a very significant event, as it marked the beginning of analytics infiltrating the front-offices of NHL organizations over a decade after Billy Beane's Athletics played [r_13].

The single most influential statistic in NHL analytics has been the Corsi statistic. It has been widely adopted across teams, fans, and media to quantify output for skaters [r_3]. Originally, Corsi was created by Tim Barnes (aka Vic Ferrari) as a statistic to better measure the workload of a goaltender in a game. The fact that NHL rosters carry 2 goaltenders on an active roster means that they have the luxury to pick to start one or the other. However, in order to make this decision Tim Barnes took the sum of shots on goal, missed shots, and blocked shots to measure how much "work" a goaltender did on any given night. The higher the number, the greater the recommendation to probably rest the goaltender in the following game. Barnes, a fan of the Buffalo Sabres, had listened to an interview with former Sabres General Manager Darcy Regier where Regier spoke of shot differential and inspired Barnes.

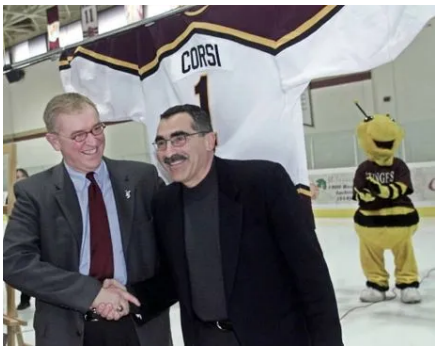

Fig f_4: Jim Corsi and His Famous Mustache

As a result Barnes almost named the statistic the Regier statistic, but after seeing a picture of Jim Corsi he chose the Corsi statistic because he liked his mustache [r_14], as demonstrated in Figure 4.

Today, the Corsi statistic is used to approximate shot differentials for teams and players. It provides some indication of the ratio of time spent in the offensive zone for teams and players relative to that in the defensive zone. Because the statistic is a ratio, it is often displayed in a manner of percentages. Effectively making it a proxy statistic for possession. On a team level from a management perspective if the team is not winning but has a positive Corsi statistic then that indicates that the team is creating more opportunities than they are giving up despite their losses. However, if the team is both losing and posting a significantly negative Corsi statistic then maybe it is time to consider rebuilding the team. The same analysis can happen on a player level by looking at the opportunities created with an individual on the ice. Most players will have Corsi statistics between 40%-60% with players above 55% being considered to be elite. The Corsi of 55% in this context means that for every 100 shots that happen while the player is on the ice 55 will be offensive zone shots for, and 45 will be defensive zone shots against [r_14].

There are plenty of criticisms of the Corsi statistic and it is important to note that it is by no means a perfect measure of neither puck possession nor offensive output. What is mostly missed is the quality of shot opportunities. Corsi operates by the law of large numbers, assuming that as the shot count goes up the average shot will become most prevalent. But Corsi cannot be adjusted to understand that certain shots happen to score at higher rates. Corsi also cannot understand usage very well [r_3]. Although NHL hockey is a continuous rotational game, any given forward will spend most of their time on a mostly static *line* with 2 other forwards. If a star player is tasked with helping to boost 2 other players who may be below average, then naturally

their Corsi may be lower than normal. Conversely, a worse player can be uplifted by playing on a line with better players. This means that in order to understand these statistics additional context is necessary. Nonetheless, Corsi remains the backbone of NHL analytics today. However, other contextual statistics also exist now. One such statistic is known as PDO, which is not an acronym for anything [r_14]. PDO looks at a team's shooting percentage and its save percentage in order to see if a team is under or over performing either in the goaltending or scoring departments. Recently, the NHL has begun to call the statistic Shooting Plus Save Percentage, or SPSV. Zone starts is another statistic used to explain usage and add context to Corsi statistics by providing information on where a player finds themselves for face-offs relative to the offensive or defensive side of the ice. This attempts to explain a player's desired usage by a coach and can provide context to Corsi statistics that may be influenced by a player either starting on offense or defense more often [r_14].

The Definition of the Modern NHL

Ice Hockey has gone through many changes over the course of its rich history. There were decades when players did not wear helmets or other protective gear as the puck was rarely lifted off the ground. Even after NHL players began to perfect shooting, goalies continued to play without helmets even though they would consistently try to use their face to stop the puck, as shown in Figure 5.

Fig f_5: Terry Sawchuk, Former NHL Goalie

The NHL style of play was mostly standardized throughout the late 20th century and into the early 21st. The 2004-2005 lockout resulted in an initial iteration of the NHL salary cap being created in 2006. This salary cap unfortunately had some loopholes and teams and players began to figure out ways to circumvent the cap. The most notable such instance is Ilya Kovalchuk, who originally signed to a 17 year contract totalling $102,000,000 USD the first iteration of the contract was simply rejected by the league. A revised 15 year contract totalling $100,000,000 USD was eventually accepted. This contract was signed in 2010 and it included years in which Kovalchuk was paid over $10,000,000 and years in which his salary was only $1,000,000 towards the end. The organization was actually paying the player close to $10,000,000 in the years he would be playing, then adding a bunch of cheap years to the end of the contract to keep the annual average value (AAV) of his cap hit lower. This cap circumvention became a primary focus for the league in future collective bargaining agreements [r_15].

In 2012-2013 the league entered another lockout over these negotiations. There were some lasting changes made to the salary cap after the lockout. These changes included but were not limited to 8-year maximum contract terms and a maximum of 50% variance in player salaries

over the course of a contract. Contracts signed before the agreement were continued to be honored with Sydney Crosby's 12-year $104,400,000 deal being upheld as it was signed 3 months before the lockout started. With these changes began what is considered to be the modern era of NHL hockey. Around this same time the NHL began to collect data related to player and puck position on the ice and they now make that data available for download.

III. DATASET

Data Collection

The NHL has made official data available for download using a publicly available API. Data exists going further back than 2014, but the event data lacks the corresponding location information prior to 2012. However, given the lockout which occurred shortly after 2012, the decision was made to limit the data to 2014 and onwards. The methodology by which the data gets collected is pretty straightforward.

To begin with, a date range is provided to the NHL Application Programming Interface (API). This call returns a response which includes game identifier data (game id) for all teams between the dates as well as team identifier data (team ids) for the teams which took part in the games. Once those game ids are retrieved, there is a check which is executed verifying that there are existing records in the teams and players data for both the home and away team. If these do not exist there were separate smaller scripts used to collect them. Once data for the teams is populated if it did not already exist then the corresponding game id is used to retrieve the details of the game via the API. These details include, but are not limited to, events, boxscore, and varying pieces of content such as pictures and highlights of the game events. All of these details are stored in a relational database instance. This included 6 tables which were named *content*, *events*, *games*, *players*, *rosters*, and *teams*. A description of the data in each table is provided in Tables 6, 5, 1, 2, 4, 3 respectively.

After populating the player data it is aggregated along with the game information and stored in the rosters table. There is a mapping created between the player id and the game id that the player participated in, as well as the team id that they played for. This prevents any stale roster information from having an effect on the data. Without this crucial information it is

impossible to properly assign a player to a team as players switch teams during the season as well as the offseason. Once rosters are set in the database the content of the game data can finally be parsed and stored in the events table. Lastly, for some of the events that are stored there is an API that will provide links to the corresponding video content. If those links are available they are stored in the content table with details about which game and event they correspond to.

This iteration continues through each game of each day between the start of the 2014-2015 season and the end of the most recent 2022-2023 season. The result is 32 teams, 5,274 players, 12,523 games, 636,130 mappings of rosters in these games, 3,984,779 events, and 63,652 links to corresponding content. It is worth mentioning that a lot of the links provided seem to rotate and sometimes these links stop working entirely [r_1].

Table t_1: Games Table

Column	Type	Description
game_id	Integer	Unique identifier of the game
season	Integer	Season in which the game took place (i.e. 20142015)
type	String	Indicates what kind of game it is (i.e. 'R' for regular season, or 'PR' for pre-season)
home_team_id	Integer	Unique identifier of the home team
away_team_id	Integer	Unique identifier of the away team
game_date	Date	The day the game was played

Table t_2: Players Table

Column	Type	Description
player_id	Integer	A unique identifier for a player
full_name	String	A player's full name
first_name	String	A player's first name
last_name	String	A player's last name
primary_number	Integer	A player's primary jersey number
birth_date	Date	A player's Date of Birth

current_age	Integer	A player's "current age"
birth_city	String	The city the player was born in
birth_state_province	String	The state/province the player was born in
birth_country	String	The country the player was born in
nationality	String	The nationality the player belongs to
height	String	The player's height (often stale)
weight	Integer	The player's weight (often stale)
active	Boolean	Is the player still active
alternate_captain	Boolean	Is the player an alternate captain
captain	Boolean	Is the player a captain
rookie	Boolean	Is the player a rookie
shoots_catches	Character	The player handedness
roster_status	Character	Is the player on a roster
primary_position_code	String	The player’s primary position

Table t_3: Teams Table

Column	Type	Description
team_id	Integer	A unique identifier for the team
full_name	String	The organization's full name (i.e. New Jersey Devils)
team_city	String	The city the team plays in (i.e. Newark)
time_zone_id	String	The time zone the team is located in
abbreviation	String	The team's shorthand abbreviation (i.e. NJD)
team_name	String	The team's name (i.e. Devils)
location_name	String	The location the team represents (i.e. New Jersey)
first_year_of_play	Integer	The year the team first played in the NHL
division_id	Integer	A unique identifier for the division the team plays in
conference_id	Integer	A unique identifier for the conference the team plays in
short_name	String	The shorthand way to address the organization's name (i.e. New Jersey)
official_site_url	String	The link to the team's website
franchise_id	Integer	A unique identifier for the franchise (in case they move/rebrand)
is_active	Boolean	An indicator if the team is still active in the NHL

Table t_4: Rosters Table

Column	Type	Description
team_id	Integer	A reference to the team’s unique identifier
player_id	Integer	A reference to the player’s unique identifier
game_id	Integer	A reference to the game’s unique identifier
toi	String	The amount of time the player spent on the ice (if null, they didn't dress)
ev_toi	String	The amount of time the player spent on the ice at even strength (if null, they didn't dress)

Table t_5: Events Table

Column	Type	Description
event_id	Integer	A unique identifier for the event
event_idx	Integer	A secondary identifier for the event
game_id	Integer	A reference to the game identifier
event_type	String	The type of event
player_1_id	Integer	A reference to the first player involved in the event, if applicable
player_1_type	String	The first player's role in the event
player_2_id	Integer	A reference to the second player involved in the event, if applicable
player_2_type	String	The second player's role in the event
player_3_id	Integer	A reference to the third player involved in the event, if applicable
player_3_type	String	The third player's role in the event
player_4_id	Integer	A reference to the fourth player involved in the event, if applicable
player_4_type	String	The fourth player's role in the event
event_description	String	A description of the event
period	Integer	The period in which the event occurs in
period_type	String	The type of period it is (i.e. OT)
period_time	String	The time of the period at which the event occurs (format HH:MM)
period_time_remaining	String	The time remaining in the period when the event occurs (format HH:MM)
x_coordinate	Decimal	The x coordinate on the ice at which the event occurs
y_coordinate	Decimal	The y coordinate on the ice at which the event occurs
home_goals	Integer	The number of goals the home team has at the time when the event occurs

away_goals	Integer	The number of goals the away team has at the time when the event occurs
secondary_type	String	The secondary event type
penalty_severity	String	The severity of the event if its a penalty
penalty_minutes	Integer	The amount of minutes the penalty is for

Table t_6: Games Table

Column	Type	Description
game_id	Integer	A reference to the game
event_id	Integer	A reference to the event
event_url	String	A link to the video content
title	String	A title to the video
blurb	String	A short description of the video
description	String	A long description of the video

Data Features of Note

The most important thing to note about the data is the way the event is reported. As demonstrated in the table(s) above, each event comes with up to 4 related players, some information about the primary and secondary event types, data about when the event happened, and most importantly the X and Y coordinates at which the data occurs. This data is reported by foot on the ice where (0,0) is the center of the ice. X ranges from -100 to 100 representing the length of the rink and Y ranges from -47 to 47 representing the width of the rink [r_1], as shown in Figure 6.

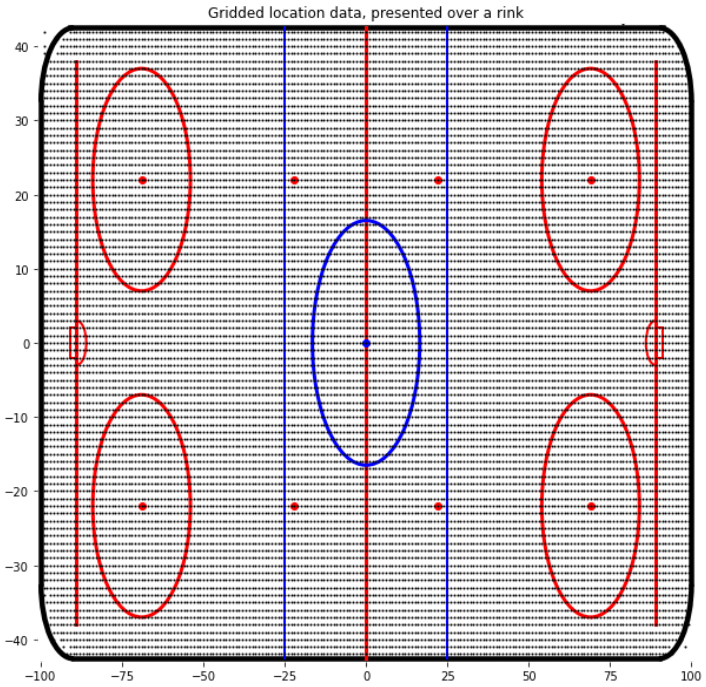

Fig f_6: The Event Data Location Grid

Table t_7: Event Counts

Event Type	Count
FACEOFF	729824
SHOT	695595
HIT	579272
STOP	568474
BLOCKED_SHOT	351130
MISSED_SHOT	292768
GIVEAWAY	226369
TAKEAWAY	173508
PENALTY	96497
GOAL	74313
PERIOD_START	41115
PERIOD_READY	41065
PERIOD_END	41058
PERIOD_OFFICIAL	41058

GAME_END	12414
GAME_SCHEDULED	12409
GAME_OFFICIAL	5188
CHALLENGE	1633
SHOOTOUT_COMPLETE	1002
FAILED_SHOT_ATTEMPT	28
EARLY_INT_END	24
EARLY_INT_START	24
EMERGENCY_GOALTENDER	11

Of these event types the ones of importance for modeling the chances of any shot being a goal are *shot*, *missed shot,* and *goal*. Table 7 details the kinds of events that were reported and how many of each exists in the dataset.

It is worth noting that although there are plenty of *blocked shots* reported, the location which is received is not the location at which the shot is taken, but the location at which the block itself occurs therefore it is actually a defensive location, and not an offensive event.

Table t_8: Secondary Shot Counts

Secondary Type	Count
Wrist Shot	417103
Slap Shot	117466
Snap Shot	112113
Backhand	62010
Tip-In	39791
Deflected	12706
Wrap-around	7599
Poke	454
Unknown	386
Batted	221
Between Legs	56
Cradle	3

As far as secondary types related to the events there are 12 secondary shot types which matter to the eventual models including the ones reported without a known shot type. The kinds of secondary shot types provided and the record count for each are shown in Table 8.

Clearly the sum of the secondary event type counts does not equal the sum of the primary types and the reason for this is that *missed shot* events are included in the eventual dataset, but those events are not reported along with a secondary type. In fact those shots highlight a lot of the data which is lacking. Namely, there is no information about the speed of the shot and no information about the angle of the shot. Meaning the eventual models cannot tell a hard shot from a soft shot in the dataset and also cannot tell where exactly a player was aiming nor how much they may have missed by and the assumption has to be made that all shots were effectively shooting for the midpoint of the net.

The x and y coordinates represent the place the puck is at the point where each event occurs and it acts as a sort of "eventually consistent" location. Meaning if you look at any non-standard event like a *period start* or *period end* they may mark the puck at (0,0) or center ice. Which is eventually true as that is where the next faceoff will always occur. If you look at other events like *hit* or *penalty* then the location reported is actually the location of the puck [r_4].

Data Manipulation

Once all the data is stored there are a few critical pieces of information still needed in order to accomplish creating a model of the data. Using jupyter notebooks all the data from the MySQL instance is stored in pandas dataframes which are manipulated to standardize some critical information. Firstly, the period time, which is provided in "mm:ss" format, is split into 2

separate fields, minutes and seconds. This allows a statistical understanding of the time at which an event occurs in a numeric format.

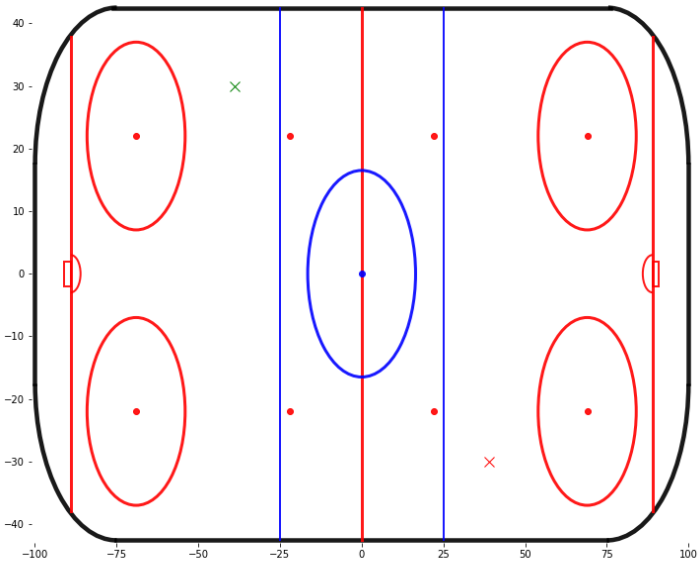

Fig f_7: Flipping Shot Locations to One Side of the Ice

Next, the goal is to standardize the point at which an event occurs. To accomplish this, 2 fields known as the adjusted x and y coordinates are created which take any shots located left of the y-axis in the reporting and rotate them 180 degrees onto the positive end of the dataset as demonstrated by the green and red X in Figure 7. This effectively doubles the number of shots in the dataset on the one side of the ice without changing any details about the shots themselves.

Once any negative coordinates have been transposed to the positive side of the ice a relatively simple math formula is used to retrieve the distance and the angle that the event is from the net. The distance formula calculation is used to find the distance in feet from the shot to the center of the net. Arctangent is used to calculate the angle on the ice, in degrees, that the event occurs from the center of the net.

Next, the goal was to be able to try to adjust for rebounds and some sustained pressure. To do so, the raw x and y coordinates are used again. Each event in each game is sorted by the time at which it occurs and a 1 row lookback is used to find the x and y coordinates and time of the most recent event in the game prior. These new fields titled *previous_x*, *previous_y*, and *previous_time_in_seconds* are then used to calculate the distance and time from the last event.

This project is based upon goals which occur when teams have the same number of skaters, also known as even-strength. To understand which goals happen at even strength there needs to be an understanding of all events that happen at even strength. So a new dataframe is created which focuses solely on the penalty data. This dataframe stores the penalty event as well as the time at which the penalty is set to expire. Then iterating through each row and returning the number of active penalties on the home and away team provides context to the situation during which the event may have occurred. When these penalty numbers are equal, then the event occurred at even-strength. The data could be reverse-engineered to understand if the event occurs at 5 on 5, or 4 on 4, or 3 on 3 but there is no demand to know that in this project, just that the event occurred at even-strength.

Data Filtration

For the shot data the work builds upon the same shot data Corsi uses, except *blocked shots* are excluded. This is commonly known as Fenwick shots, a popular alternative to Corsi [r_14]. The event data begins as 3,960,724 unique events.

- Filters Applied:
    1. All events are categorized as *shot, goal,* or *missed shot*
        a. 1,056,553 remaining events
        b. These are the event types we are looking for

2. All events occur during the regular season
    a. 929,139 remaining events
    b. Excluded the preseason and postseason as the player subset is different
3. All events have a goalie attached
    a. 925,913 remaining events
    b. We are excluding empty net goals which would skew data
4. All events happen in the offensive half
    a. 908,302 remaining events
    b. We are excluding shots which happen to go on net from the defensive half of the ice
5. Events are not penalty shots or shootouts
    a. 901,809 remaining events
    b. This 1 on 1 skater vs goalie event is not a part of the normal course of play
6. Shooter dressed for 10 or more games
    a. 893,903 remaining events
    b. This is an effort to make sure the shooter was given an opportunity to establish a comparable sample of shots in the given season
7. Events occur at even-strength
    a. 729,196 remaining events
    b. Using the aforementioned penalty logic events that occur when there is an imbalance of skaters on the ice

Once all the shots which do not occur in even strength scenarios are excluded there is a final dataset consisting of 729,196 even-strength, non-empty-net, offensive-zone shot-events which occur during the regular season between the 2014-2015 season and the 2022-2023 season inclusively, in which any given shot corresponds to a player who played at least 10 games in the season during which the event occurs.

IV. MODELING THE DATA

The next step to accomplishing the goals set out in the introduction is to use supervised learning for the analysis. The idea behind supervised learning is to create an algorithm with labeled data, to be able to predict the particular change of any given piece of data resulting in a goal being scored. In this case a regression model is applicable as the output of the model is the chance of the particular shot being a goal and not whether or not it was a goal. So, the data will show if an event results in a goal or not, but this model predicts the likelihood of a given shot being a goal.

The most recent season (2022-2023) which is 88,853 shots are used as the testing data. The previous data between 2014-2015 and the 2021-2022 season has 640,343 training data points to build our model. This ends up being a training to test split of about 88% training to predict 12%.

Features

The final feature set which the data used to train the model is the distance to the net in feet, the angle of the shot, the type of the shot, the time at which the event occurs in seconds, the time since the last event occurred in seconds, the distance from the most recent event in feet, the deficit the shooter takes their shot at (a range from -3 to 3), and the shooter handedness.

It is worth reiterating here that the goal is to attempt to evaluate a player on the merit of the shots they take for the purpose of comparison between each other. With that in mind, information on the shooter or the goaltender is removed from the model of any event. In reality, the models we created indicated that the most important factor in understanding any given shot's chance of scoring is the goaltender. Since the goal is to predict and compare goal scoring output as a result of the methodology or merit of the shots a player takes and not a result of who the

goaltender is on any given shot those data points are excluded from the model. Following the same methodology the shooter is excluded as a datapoint as well as the shooter would become a categorical variable which could detract from the true merit of the shot and indicate reliance on pure shooting skill.

Dummy Model

In order to properly evaluate the more complex models which follow, the process begins with the absolute default dummy model. In the training dataset there are 38,195 goals resulting from 640,343 shots. Using those numbers, an average conversion rate (shooting %) of about 6.0% is derived.

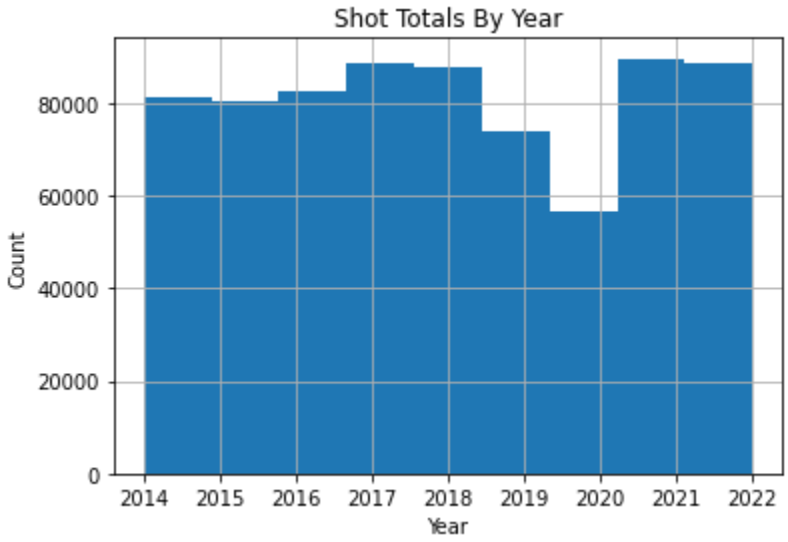

Fig f_8: Shot Totals Organized by Year

As demonstrated in Figure 8, the 2019-2020 and 2020-2021 season observed a significant reduction in shot totals. This is a function of the covid pandemic and it meant that in those seasons the total number of observations was lower, thus players scored less goals per season. So to adjust for this the simplest solution would be to exclude those 2 years from the shooting totals

for the purpose of collecting this data as the games were played at completely different schedules from the other seasons in the dataset, as is represented in Figure 9 below.

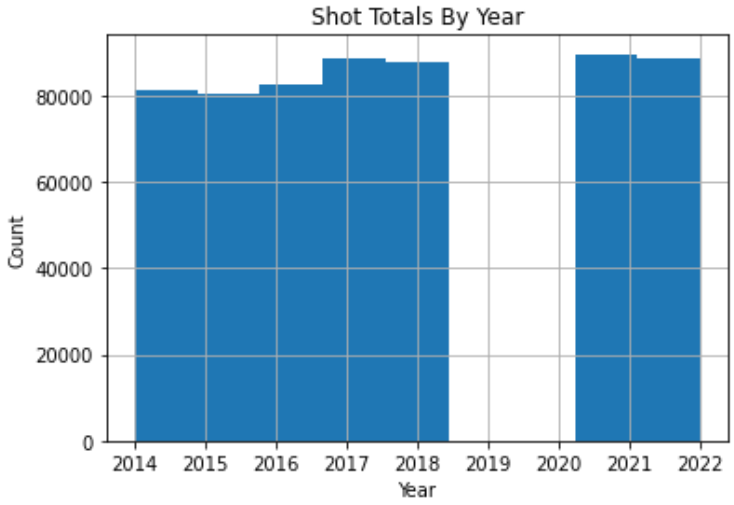

Fig f_9: Shot Totals Organized by Year Without 2019 & 2020

After removing the covid years from the dataset the scoring rate doesn't change from the original 6.0%. This 6.0% figure is assigned to every event in the dataset as the expected goals (xG) for each event. How much more or less accurate would a supervised learning algorithm be than this dummy model?

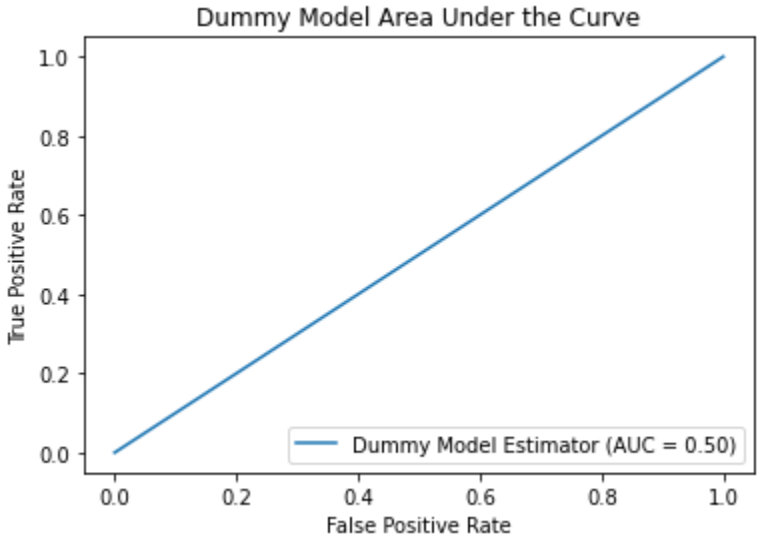

Fig f_10: Dummy Model Area Under the Curve

As Figure 10 shows, the dummy value splits the true-positive and false-positive rates evenly. Resulting in an area under the curve of exactly 0.50. Once we have the output of the other models the methodology to evaluate a player's season is relatively straightforward. The data is grouped by player and the total of each player's expected goal output is taken, known as the *Cumulative Expected Goals By Player* and it is compared to the *Cumulative Goals Scored in Reality By Player* to evaluate the aggregate model outputs by player.

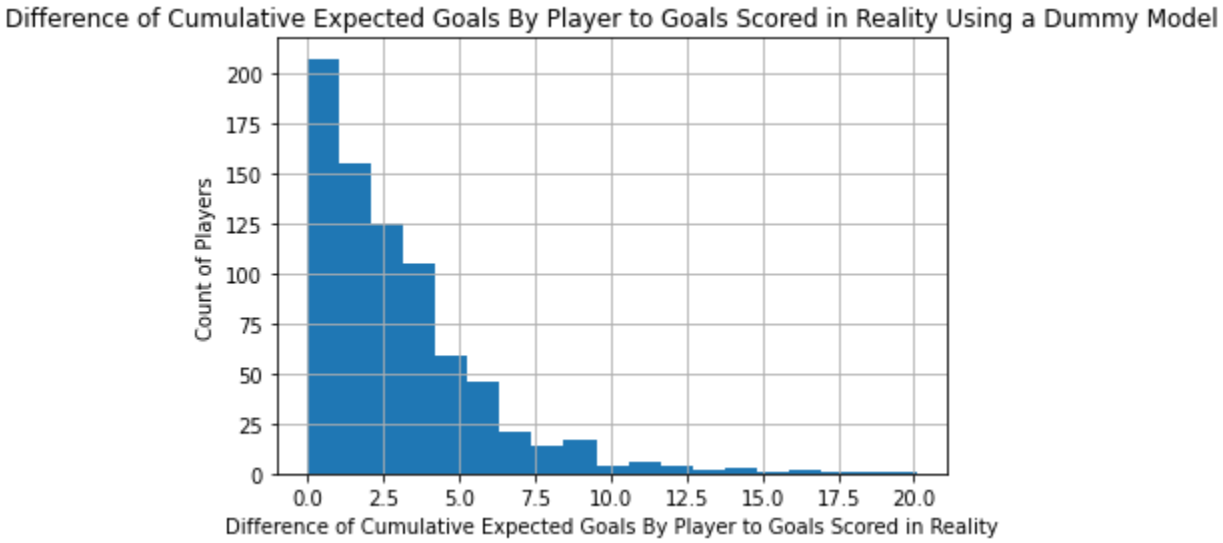

Fig f_11: Dummy Model Cumulative Expected Goals vs Reality Histogram

As Figure 11 demonstrates, a dummy value equivalent to the avg shooting % as an expected goal value does yield some decent results in terms of player predictions. There are about 200 players for which the model is able to make a prediction within 1 goal of reality. The amount by which the model misses decreases in a relatively linear fashion until about the 10 goals mark. The largest miss in the model was 20 goals on a given player. However, there is not much more to really be gleaned from the figure until we are able to compare it with the outputs of other supervised learning models.

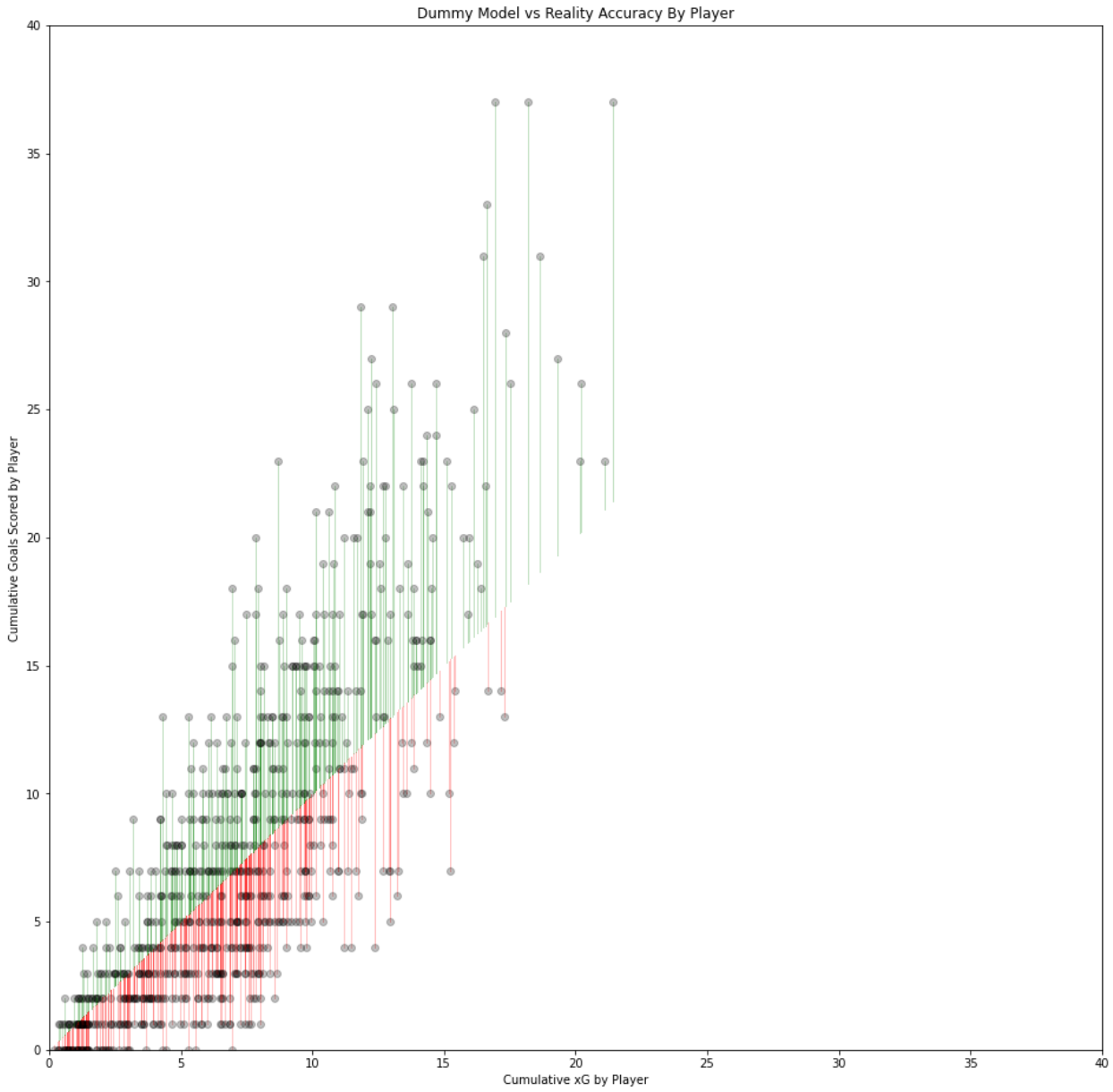

Fig f_12: Dummy Model Cumulative Expected Goals vs Reality Error Bars

Looking at this data with error bars, shown in Figure 12, gives a better idea of the bias which exists in the dummy model. The X-Axis in this example is the sum of the model outputs by the player. The Y-Axis is the number of goals the player has scored. An error bar is drawn from the model output sum to reality giving a visual representation of how much the model missed by for that player. The dummy model is doing a relatively good job of predicting the

lower quadrant of NHL player performance. However, there is no player for whom the model predicts more than 25 goals in the season, where in reality there was a decent chunk of players who scored more than 25 goals. This is a result of using a dummy model, in order to score 25 goals with a 6.0% shooting percentage a player needs to take 417 shots. So this indicates that no player was able to cross that threshold within the dataset. After taking the average absolute value of all the error bars and averaging those, the model misses by about 3.010 goals per player over the course of the year.

Linear Regression

Simple linear regression enables the prediction of a variable based upon the information about another variable. Linear regression attempts to establish a relationship between two variables along a straight line. Multiple regression is a type of regression where the dependent variable shows a linear relationship with two or more independent variables. It can also be nonlinear where the independent and dependent variables do not follow a straight line [r_5].

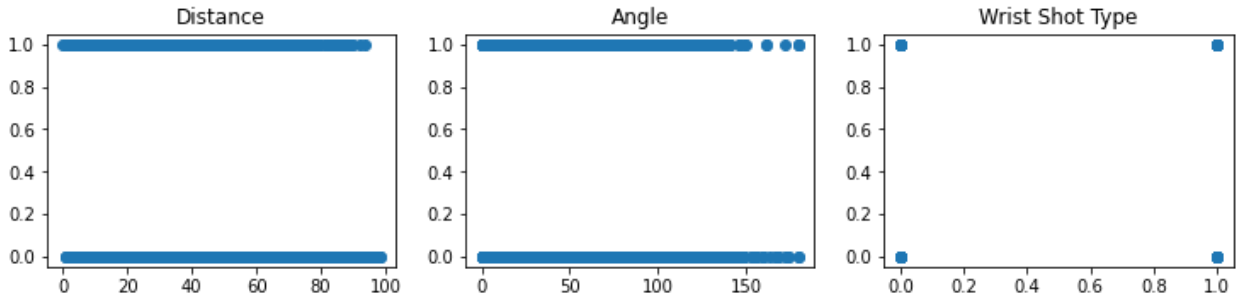

Fig f_13: Relationship Between Dependent Variable and Independent Variables

Multiple linear regression is based on assumptions. The first assumption is that there is a linear relationship between the dependent variable and independent variables [r_5]. Looking at Figure 13 it is evident from the scatterplots that the independent variables do not all share linear relationships with the result. This is the first indication that linear regression may not be the ultimate way to answer the questions the model is expected to answer as there is a binary

outcome and so a linear relationship cannot be established to any of the continuous or categorical variables in the dataset.

The next assumption made is that the predictive variables are not highly correlated with each other. If independent variables show multicollinearity, then that could lead to problems with the algorithm's ability to attribute which variable is the one which is actually contributing to the dependent variable [r_6]. As demonstrated in Figure 14, of the feature set the only 2 variables which seem to share some relation are distance and angle, in the sense that at certain angles it is impossible to shoot from further than at others.

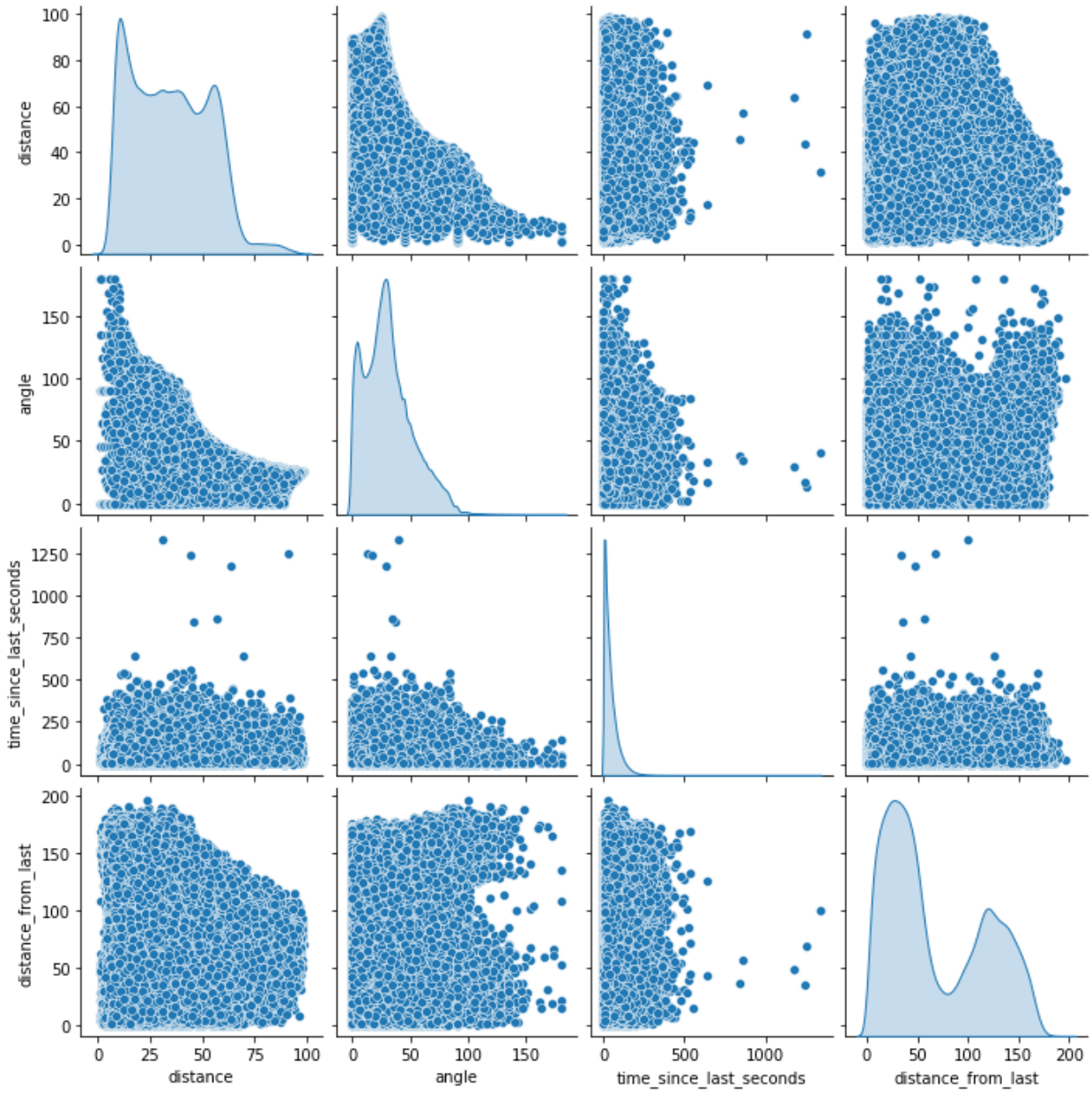

Fig f_14: Multicollinearity Analysis on Continuous Variables

Another assumption made by multiple linear regression is the constant variance of residuals. Known as homoscedasticity, multiple linear regression assumes that the standardized residuals will have even distribution against predicted values.

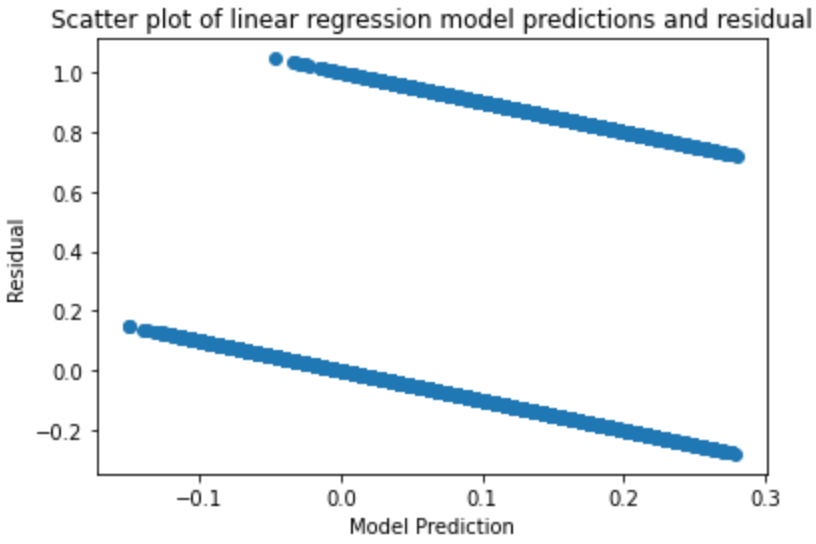

Fig f_15: Homoscedasticity Analysis Between Predictions and Residuals

However as seen in Figure 15 when the model predictions and the residuals are scattered, there are very clear trends in this data. This is the opposite of the assumptions needed for the linear regression model to be effective, as the desired result would be a random distribution.

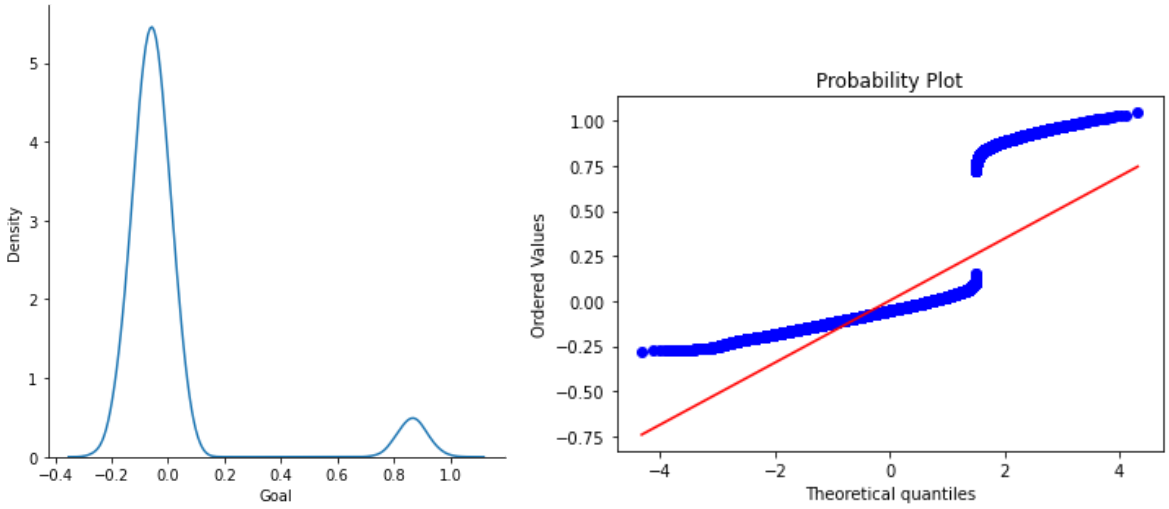

Fig f_16 & f_17: Normality of Residual Values

The next assumption which is checked is the normality of the residual values [r_7]. The expectation is that residual values should follow a normalized distribution and the values in a probability plot should straddle the centerline. However, as demonstrated in the distribution and

probability plots in Figures 16 and 17 the binary outcome continues to not satisfy assumptions made by the linear regression algorithm.

The fifth assumption made by linear regression is the independence of observations [r_5]. The data confidently satisfies this assumption, as by nature each shot is an independent event fed into the model. Even though the data does use the previous shot location and time as a factor, it is not dependent upon that previous shot. This subset of data provides context to the game state at which the shot occurs, giving better context to an independent event.

At this point there is an abundance of evidence that the data fails to satisfy the assumptions needed for linear regression algorithms. Nonetheless, the results are included, although conclusions should be carefully considered since the data does not fit the assumptions well.

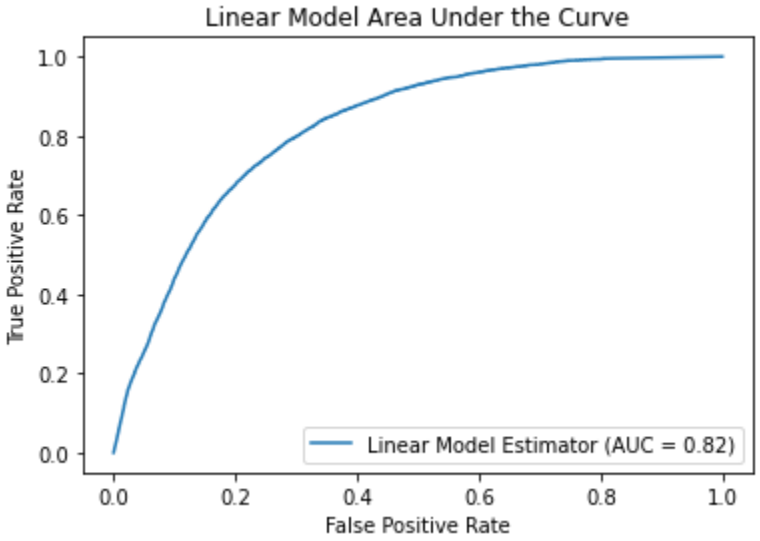

Fig f_18: Linear Regression Area Under the Curve

Figure 18 shows a Receiver Operating Characteristic (ROC) curve evaluating the Area Under the Curve (AUC) metric. This shows the multiple linear regression model, despite its shortcomings, does a decent job on a shot by shot basis of predicting the chance of a given shot

to be a goal. Showing an increase in the AUC from 0.5 in the dummy model to 0.82 in the linear regression model.

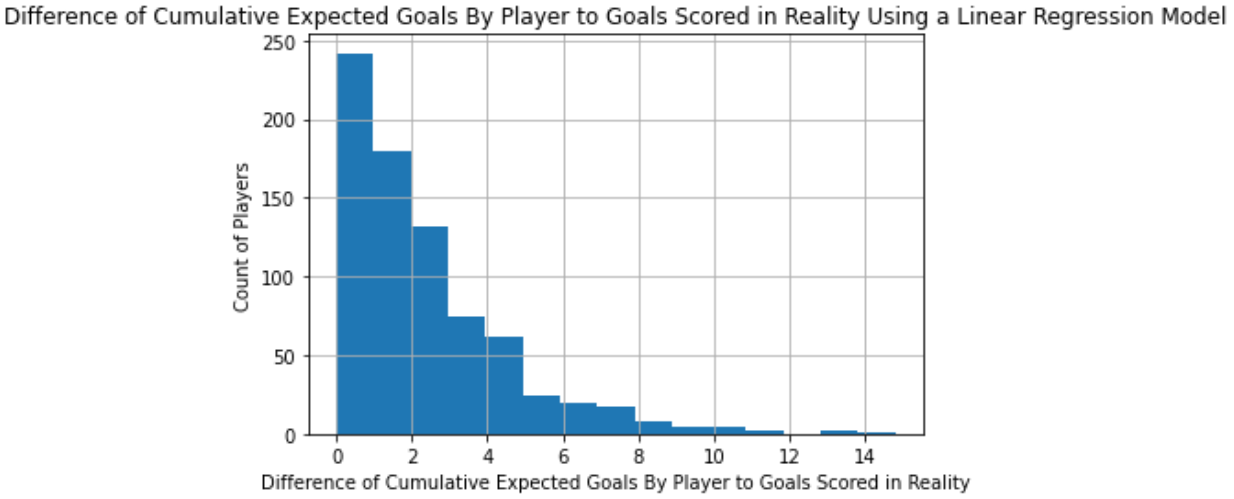

Fig f_19: Linear Regression Cumulative Expected Goals vs Reality Histogram

Figure 19 provides additional context about where the linear model out-performs the dummy model. There are now close to 250 players for which the model predictions are within 1 goal. The maximum the model misses by is decreased from 20 goals to about 15 goals. The relationship between the misses also changes. There are more players clustered towards the left side of the X-Axis. This means the slope looks more like an exponential curve, however between 0 and 5 the relationship is still very linear.

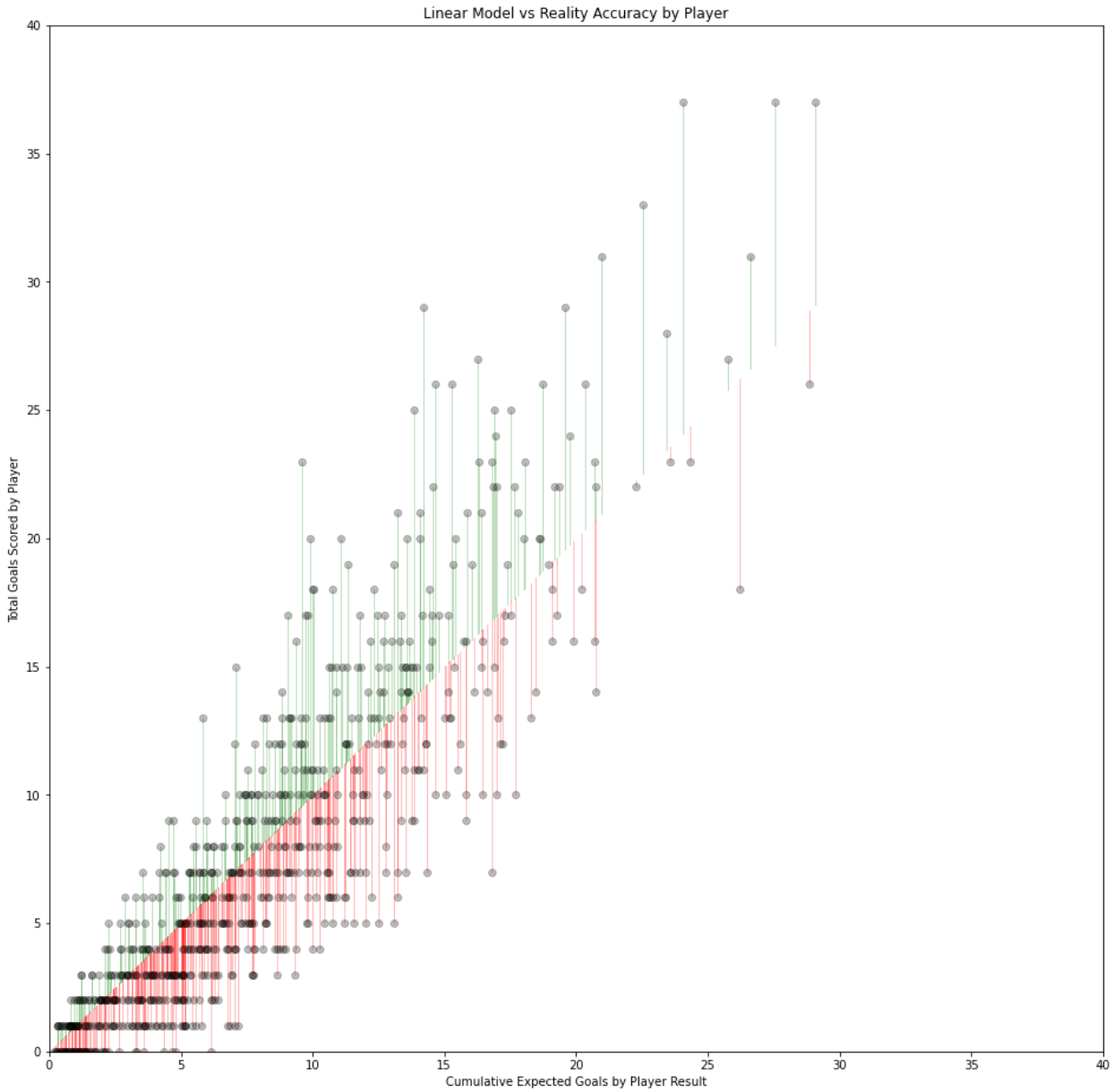

Fig f_20: Linear Regression Cumulative Expected Goals vs Reality Error Bars

Figure 20 uses the same style of error bar graph and begins to shape some ideas of the model shortcomings. Although the model does a decent job with predictions overall the amount by which it underpredicts a large portion of the players is on display. This is demonstrated when we see no player's predicted to score over 30 goals in the model. Overall, the model is a bit more

accurate than the dummy model, moving the average error per player down from 3.010 goals per player to 2.350 goals per player.

These results are certainly better than the dummy model. However, the shape of the data demonstrates that linear regression is probably not the best way to evaluate the chances of a particular shot becoming a goal. So moving forward other models will be evaluated.

Logistic Regression

Multiple binary logistic regression is a form of supervised learning used to predict a binary dependent variable using one or more other variables. Given that the variable the model is predicting is a binary goal or no goal, binary logistic regression can be used as the algorithm to predict the classification of a shot. However, this will lead to issues as just classifying the shot as goal or no goal in the binary outcome will result in the overwhelming vast majority of events being classified as not goals. The reason for this is the low shooting % which was observed in the previous examples. Unless the logistic algorithm can predict something as being more than 50% chance as a goal it will not classify the event as being a goal. In reality a shot opportunity with a 20% chance of being a goal is already a high-danger opportunity. Meaning if a player shoots 5 shots each with a 20% chance of being a goal, despite the logistic model potentially classifying each shot as not converting to a goal, the expectation based upon the sum of the expected goal values would be that the player has scored one. However, using logistic regression the probability values of the outcome can be outputted which the supervised learning algorithm comes up with instead of the final classification results, thus giving a similar output to the linear regression model in the form of the chance of the shot becoming a goal [r_16].

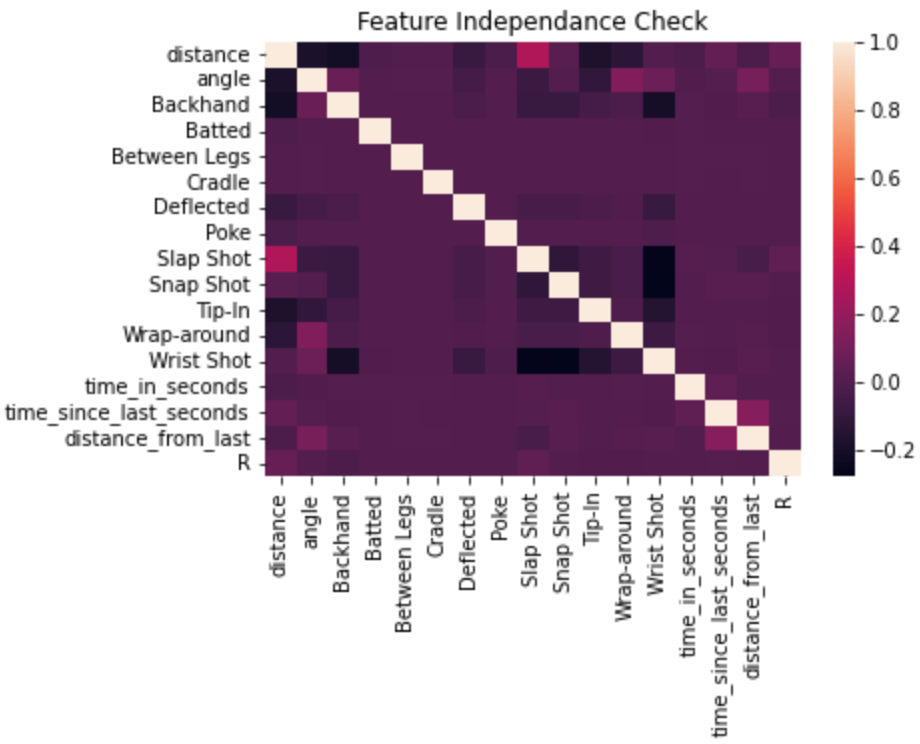

Fig f_21: Multicollinearity Check for Logistic Regression

Multiple logistic regression also has assumptions about the data in order to work. Some of these assumptions are shared with the linear regression analysis above and some are unique to logistic regression. The 2 assumptions logistic regression shares with the above linear algorithm are the multicollinearity of the data and the independence of each observation within the dataset. Which means that there should be a minimal relationship between independent variables of the dataset [r_16]. Figure 21 demonstrates that the variables are not related. The closest relationship exists between the distance and the categorical variable indicating the Slap Shot type of shot. This makes sense, as most players probably use similar areas on the ice to execute Slap Shots, but even then the relationship is not statistically significant. Also, it was already confirmed in the previous analysis but worth re-stating that each NHL shot is an independent event even if the models use information on the previous event to understand the context better.

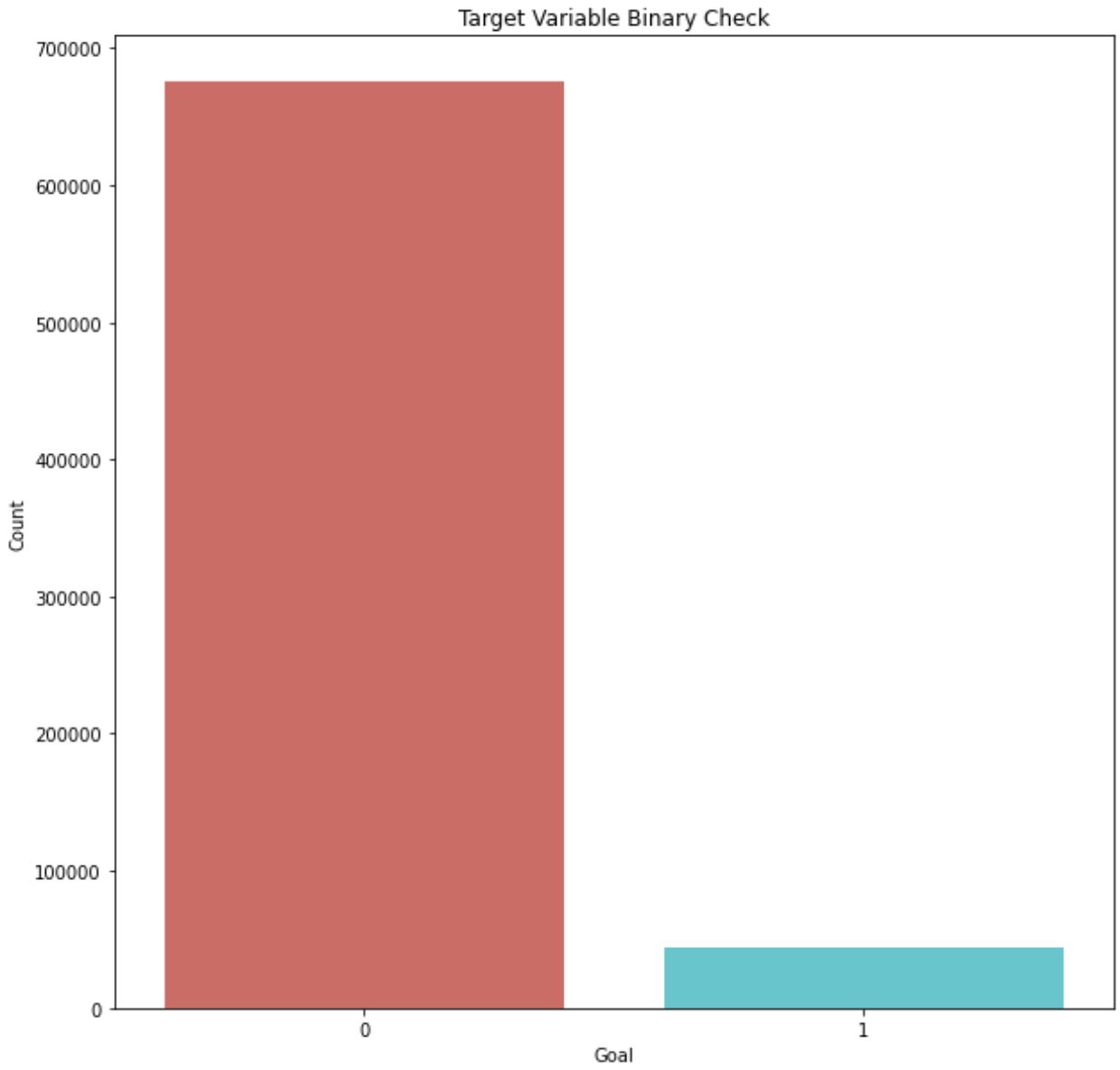

Fig f_22: Target Variable Binary Analysis

There are other assumptions the logistic regression makes about the data that the linear regression does not necessarily. The first is a binary target variable, is the shot a goal or no [r_16]. Meaning there is a yes or no target variable, which satisfies this assumption as demonstrated by Figure 22.

Another assumption made by the logistic regression algorithm is that there is a large enough sample size to be able to determine outcomes efficiently and there are no major outliers

in the data as logistic regression is very sensitive to outliers. The general rule with sample size is that for logistic regression there need to be at least 10 events with the least frequent outcome for each independent variable [r_16]. In the feature set there are 16 independent variables, and one of them was unable to fit that assumption. As demonstrated in Table 8, there were not enough shots classified as *Cradle* by this rule.

To check for other outliers, an anomaly detection technique is used. To make sure there were no outliers or anomalies in the dataset all categorical variables were standardized.

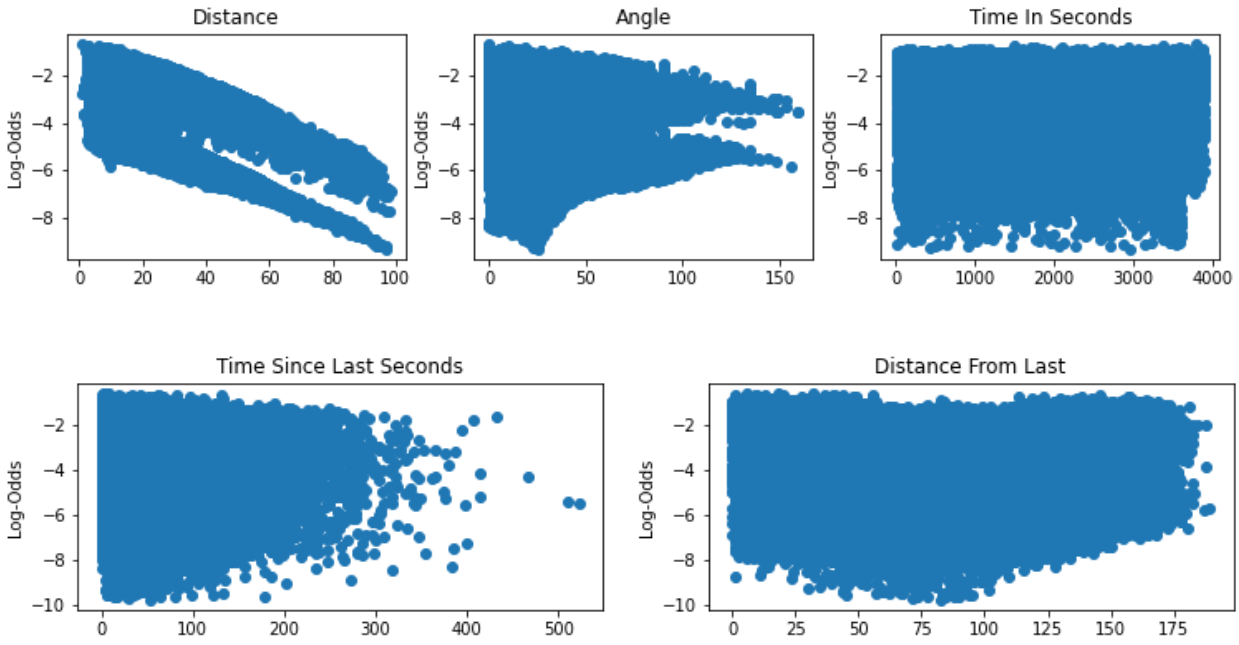

Fig f_23: Relationship Between Independent Variables and Log-Odds

The last assumption the multiple logistic regression makes about the data is that there is a linear relationship between the independent variables and the logit of the target variable [r_16]. Logit, also known as a log-odds function, is tested by plotting the continuous predictors against the log-odds. Figure 23 provides the first indications that logistic regression may not be accurate. Although it may fit most of the assumptions made, the only independent variable which shares a linear relationship with the log-odds is the distance from which the shot is taken.

Having seen how the data fits with the assumptions made by a logistic regression algorithm, the following observations were made with respect to the comparison between the dummy and logistic models.

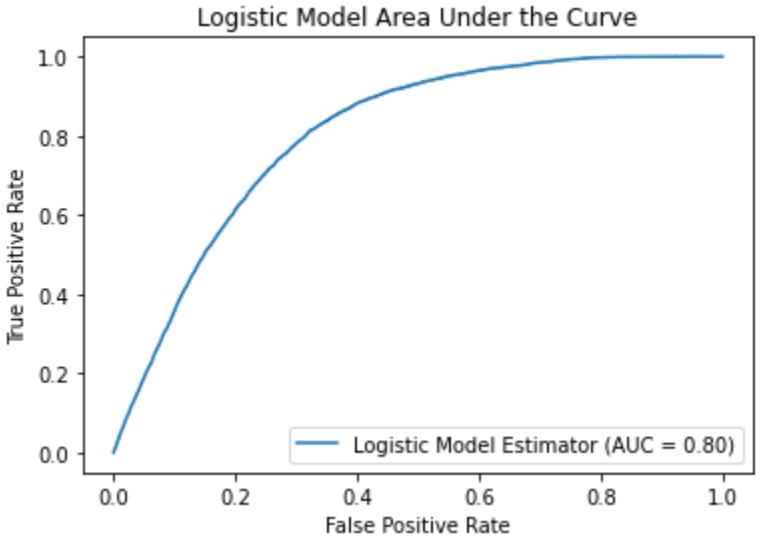

Fig f_24: Logistic Regression Area Under the Curve

As demonstrated by the ROC curve in Figure 24, the multivariate binary logistic model shows significant statistical improvement to the area under the curve over the dummy model. Moving the area under the curve from 0.5 to 0.8. The next step is to evaluate the results of the model over the course of a season.

Fig f_25: Logistic Regression Cumulative Expected Goals vs Reality Histogram

Figure 25 shows the results of the model predictions over a season. Immediately, the shape of the curve is more parabolic than both the dummy and linear models. This follows from about 100 additional observations which fall between 0-1 difference than in the dummy model for a total of 300 players falling within that range.

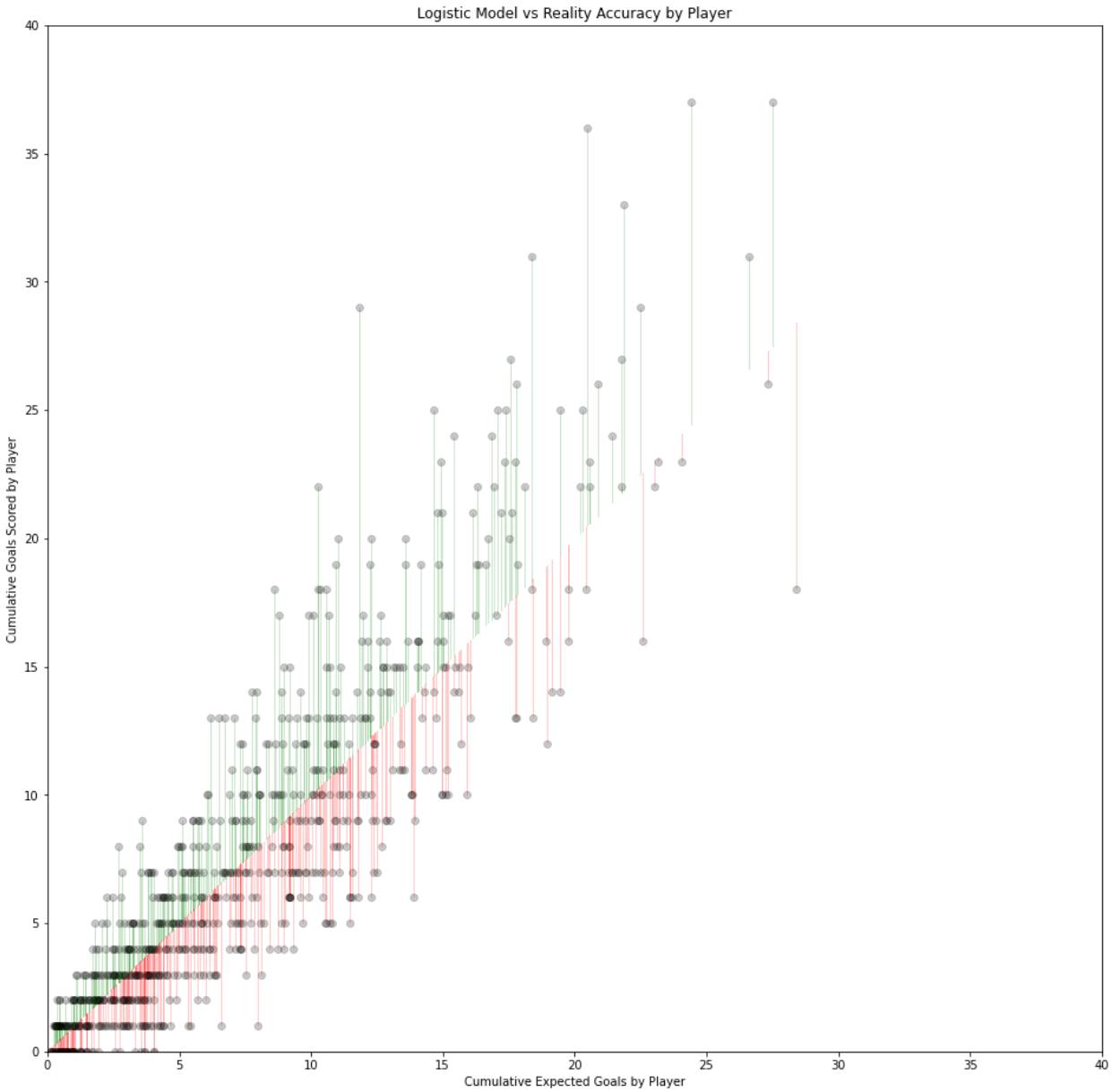

Fig f_26: Logistic Regression Cumulative Expected Goals vs Reality Error Bars

Figure 26 shows the error bars again demonstrating what this model does well relative to the dummy model. It is also doing a good job of predicting the lower quadrant of NHL player performance like the dummy model did. However, there are now a handful of players for which the model predicts more than 25 goals. There are still no players predicted to score over 30, even though multiple players did in fact score more than 30 goals over the course of the season. After

taking the absolute value of all the error bars and averaging those, the logistic model misses by about 2.164 goals per player over the course of the year.

Random Forest

Random forest regression is a supervised learning algorithm using an ensemble learning method. Random forest regression is a bagging technique built upon decision trees. These trees in random forests run in parallel, meaning there is no interaction between these trees while building the trees. Once all the trees are created, the random forest algorithm merges the decision trees to combine to a final prediction, as depicted in Figure 27 [r_10].

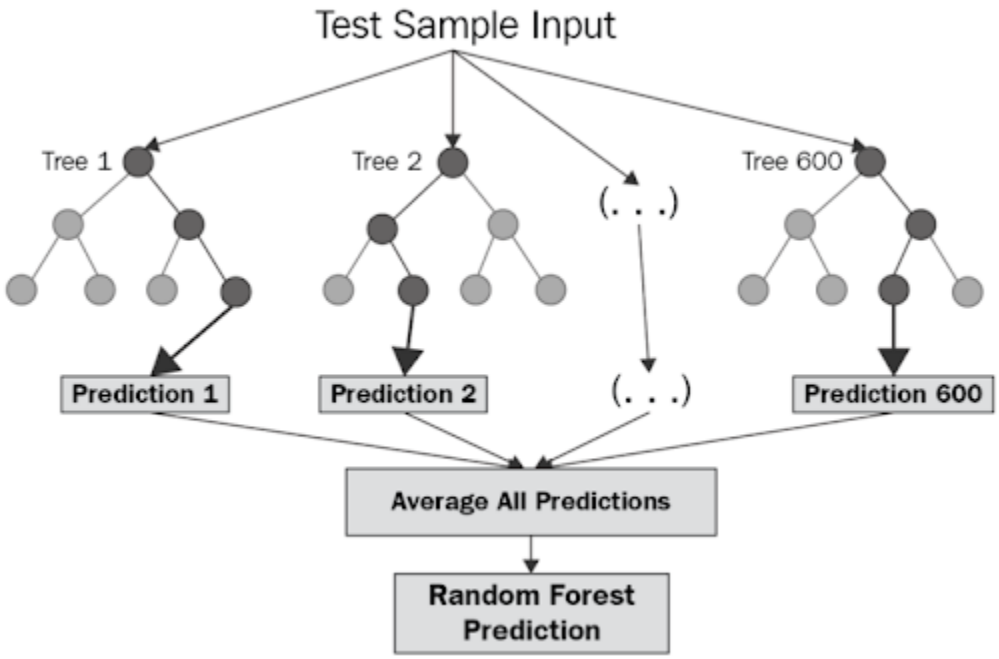

Fig f_27: Sample of Multiple Trees in a Single Predictive Forest

Random forest regression usually provides higher accuracy than other algorithms, especially for complex datasets with feature sets that may or may not contain relationships between them. However, this comes at the cost of understanding the algorithm. While random forests often achieve higher accuracy than a single decision tree, being able to interpret the data

across multiple trees is challenging as the forest will average the results across decision trees to create the prediction [r_10].

This means the algorithm cares significantly less about the relationship between the features of the data and significantly more about the tuning details which the model is given.

Table t_9: Random Forest Final Hyperparameters

Num Estimators (Tree Count)	Max Depth	Max Features	Max Leaf Nodes	Min Samples Leaf	Min Samples Split	Bootstrap
100	None	Auto	25	5	12	TRUE

Table 9 details the final hyperparameter set which the model was tuned to. The final parameter set used was 100 trees in the forest, no max depth, automatic feature selection, a 25 maximum on leaf nodes, 12 minimum samples to split a tree, and 5 minimum samples for each leaf node. Figure 28 depicts the first tree (of 100) in the forest to give a visual representation of how these trees look within the model.

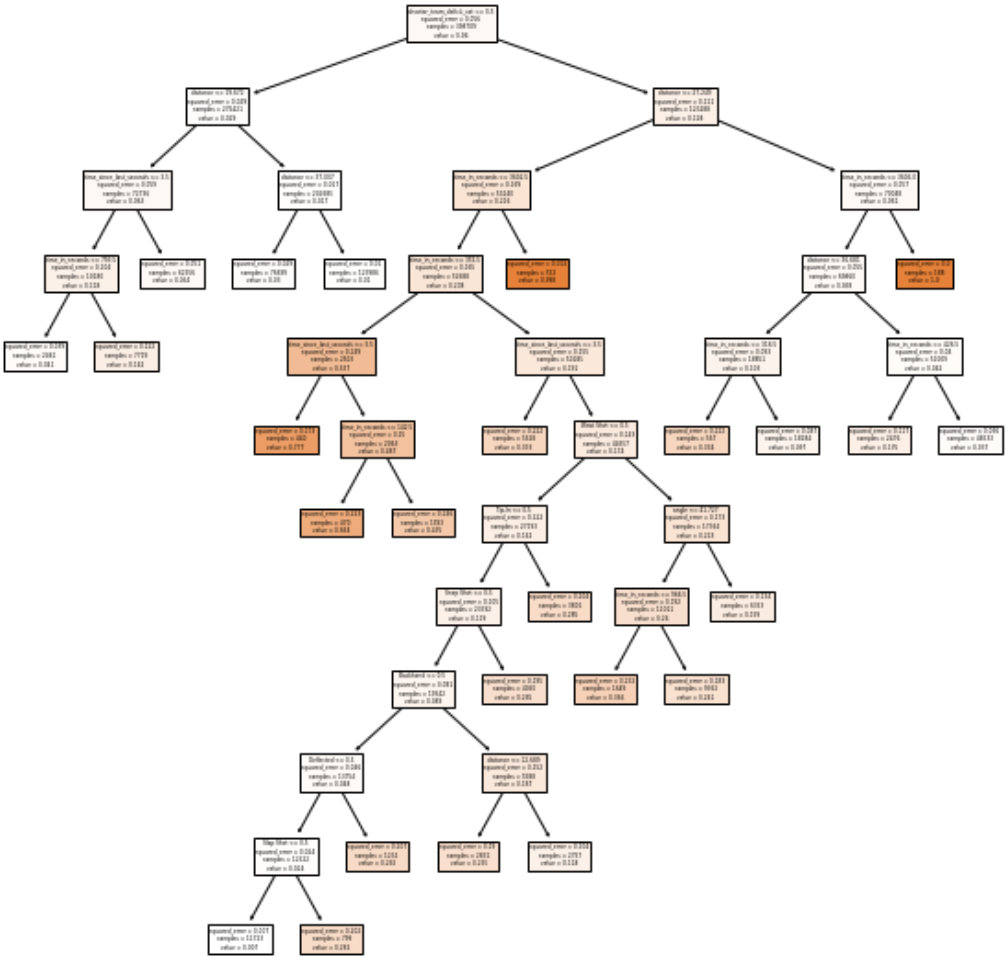

Fig f_28: A Sample Tree from Within the Predictive Forest

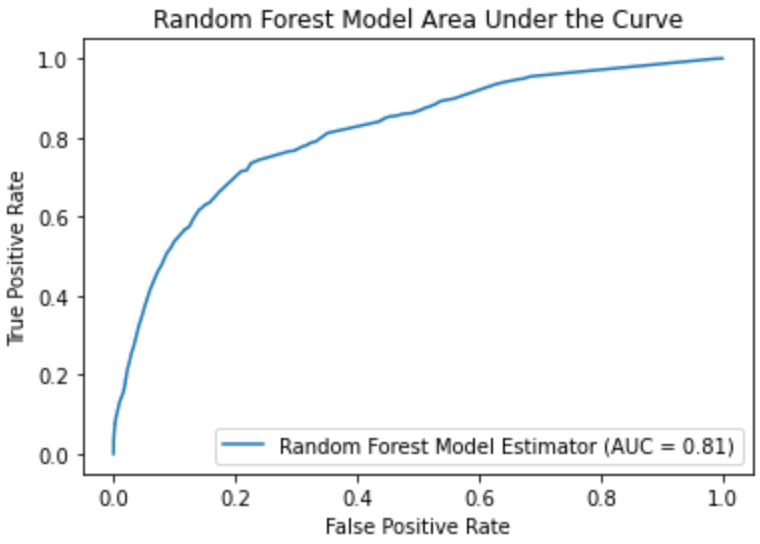

Fig f_29: Random Forest Area Under the Curve

Figure 29 shows the ROC curve for the random forest model. Overall, the random forest model shows a marginal improvement over the logistic regression model and still a significant increase over the dummy model. Given this is another relatively accurate model on a shot-by-shot basis, then once again the next step is to extrapolate how the model performs over a season.

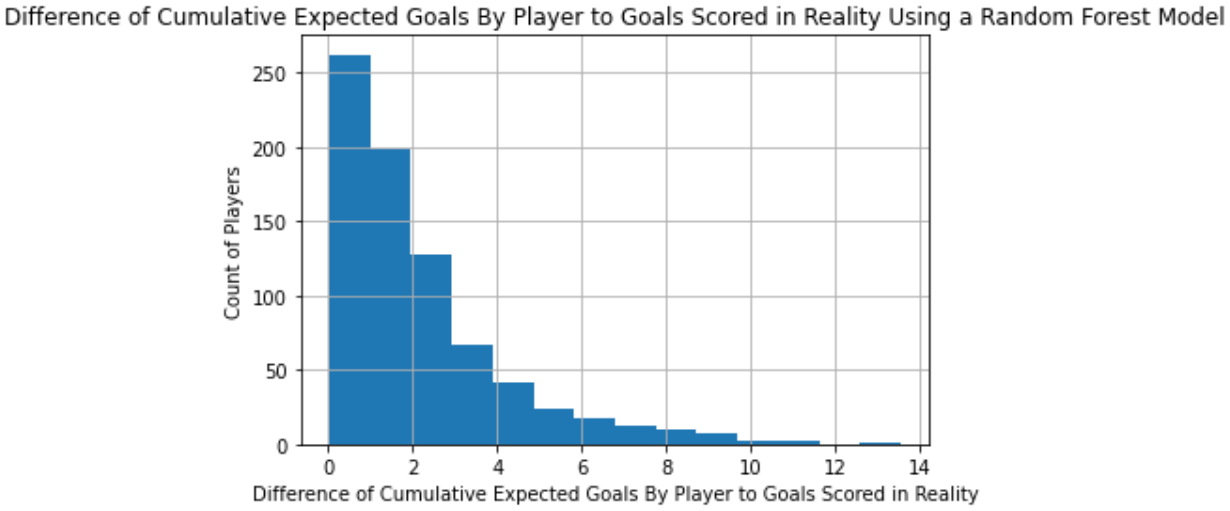

Fig f_30: Random Forest Regression Cumulative Expected Goals vs Reality Histogram

Figure 30 shows that the random forest model gets more accurate at the higher-end of the spectrum, but loses accuracy to the logistic algorithm in the low-end. There is no player whose predicted goals are more than 13 away than their actual number of goals in the test data, whereas the logistic regression function had up to 17 away. However, the random forest was a little less exact in predicting reality as there were more players in the logistic regression section for which the model was within 1 goal.

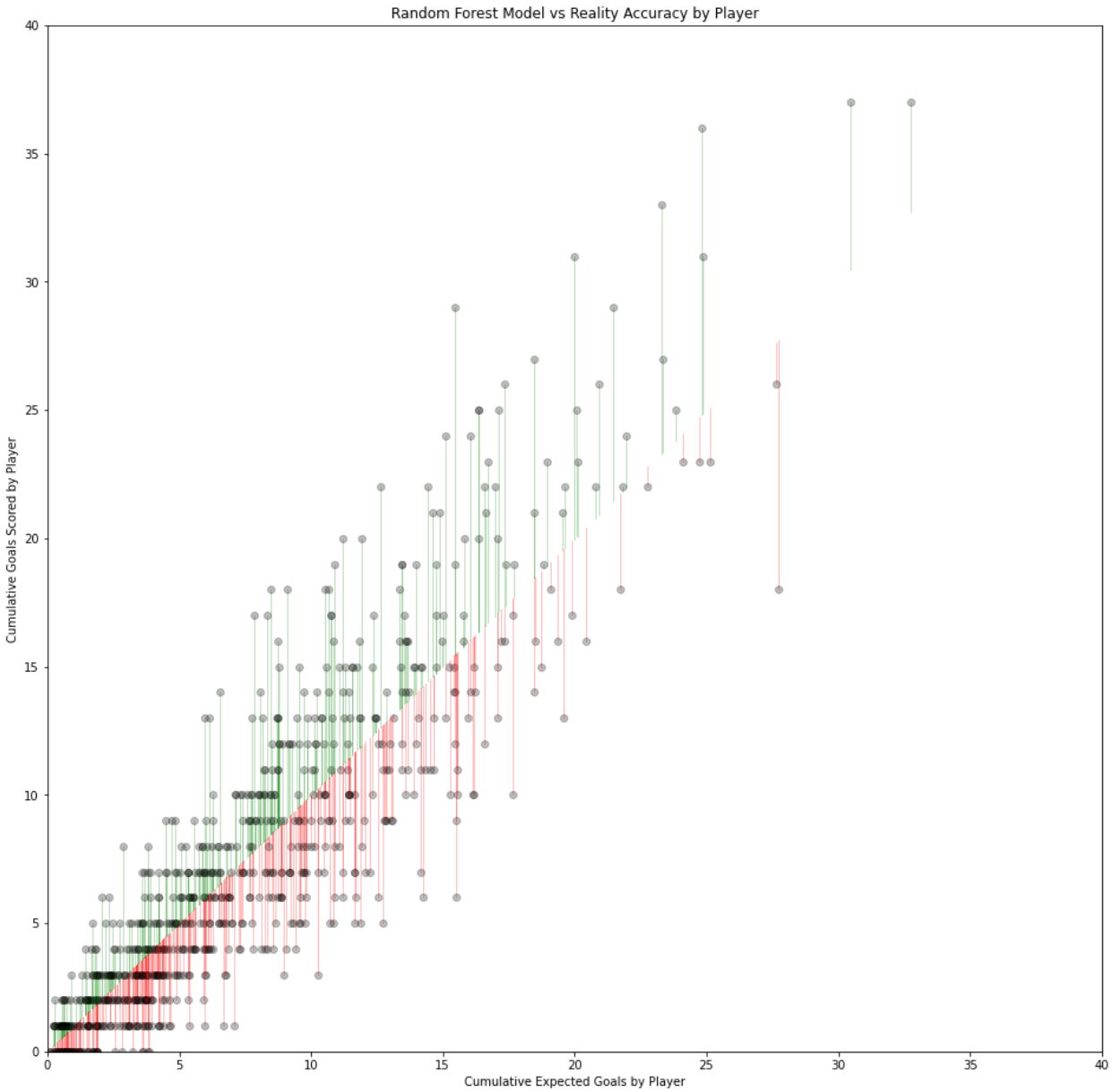

Fig f_31: Random Forest Regression Cumulative Expected Goals vs Reality Error Bars

Figure 31 shows each individual player's cumulative expected goals visualized with an error bar to the actual result they achieved. It becomes clear that the random forest model is doing a much better job of predicting the top-end scorers than the logistic regression model was able to. There are now a couple of players for which the model predicts more than 30 goals which is closer to the reality. The model continues to do a pretty good job at predicting the

lower-end scorers as well. After taking the absolute value of each of these error bars, the model misses on a per-player basis by about 2.150 goals per season. A moderate improvement over the 2.169 which the logistic regression had been able to accomplish and a significant improvement over the dummy model.

Gradient Boosted Regression

Gradient Boosting is another supervised learning technique based around decision trees, but as the name suggests it is a boosting technique and not a bagging technique like random forests. Meaning that as the algorithm creates a tree it uses previously created trees to determine if one particular tree is better per the parameters given to the algorithm. This means that the algorithm is able to be more accurate than random forest. Because the decision tree is trained to correct other tree errors, gradient boosted outputs are capable of capturing more complex patterns in the data [r_11]. However, this also makes them susceptible to overtraining and overfitting. For this reason it is important to be careful and not let the model overtrain.

Fig f_32: Gradient Boosted Model Decision Tree

Figure 32 demonstrates the final decision tree after hyperparameter tuning. The gradient boosting algorithm is capable of handling a multitude of data types and excels at relationships between categorical and continuous data relative to other supervised learning techniques. It is able to natively handle sparsity and it is capable of handling categorical variables within the algorithm [r_11]. In the case of this model another cross validation technique is used to help find the best algorithm for the data by hyperparameter tuning.

Table t_10: Gradient Boosting Final Hyperparameters

Tree Method	Objective	Evaluation Metric	Minimum Child Weight	Max Delta Step	Gamma	Eta	ColSample_ bytree	SubSa mple	Max Depth
GPU_HIST	BINARY:LOGISTI C	AUC	12	3	0.75	0.17	0.74	0.81	None

Following the documentation the model hyperparameters which were most effective are presented in Table 10. The most notable ones are the objective being a binary logistic boosted model and the evaluation is done using the area under the curve.

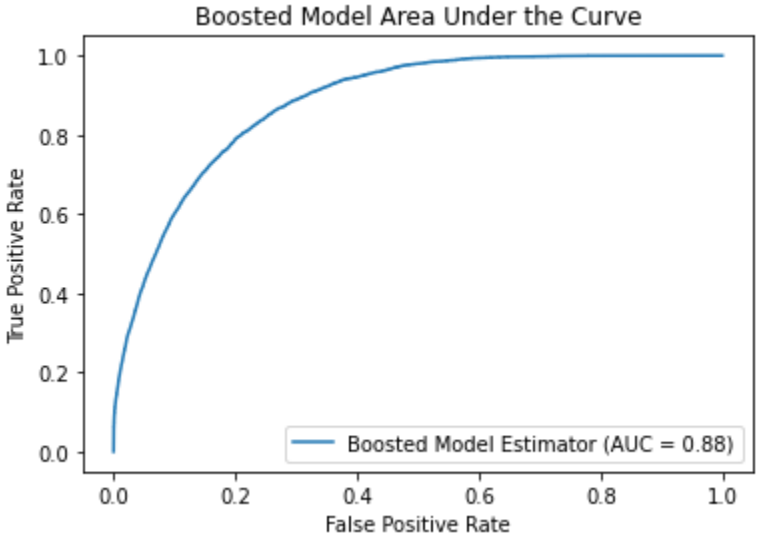

Fig f_33: Gradient Boosted Regression Model Area Under the Curve

Figure 33 demonstrates the relationship between true and false positives within the boosted model. This model outperformed the dummy model and even showed significant improvement on a per-shot basis in comparison to the logistic and random forest models. Once again, the output of the model was explored on a season-long basis.

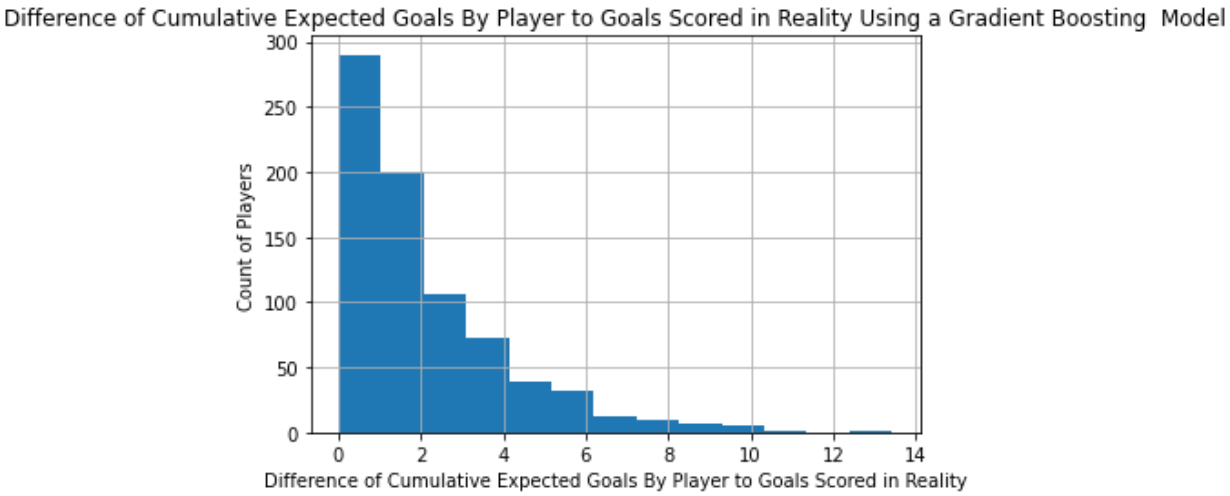

Fig f_34: Gradient Boosted Regression Cumulative Expected Goals vs Reality Histogram

As seen in Figure 34, the boosted model demonstrated even more raw accuracy on a per player basis than the random forest model. Over 300 players were predicted to within one goal of their real life results, and the biggest misses in the model are smaller and even more rare. This model demonstrated its effectiveness on a per-season basis.

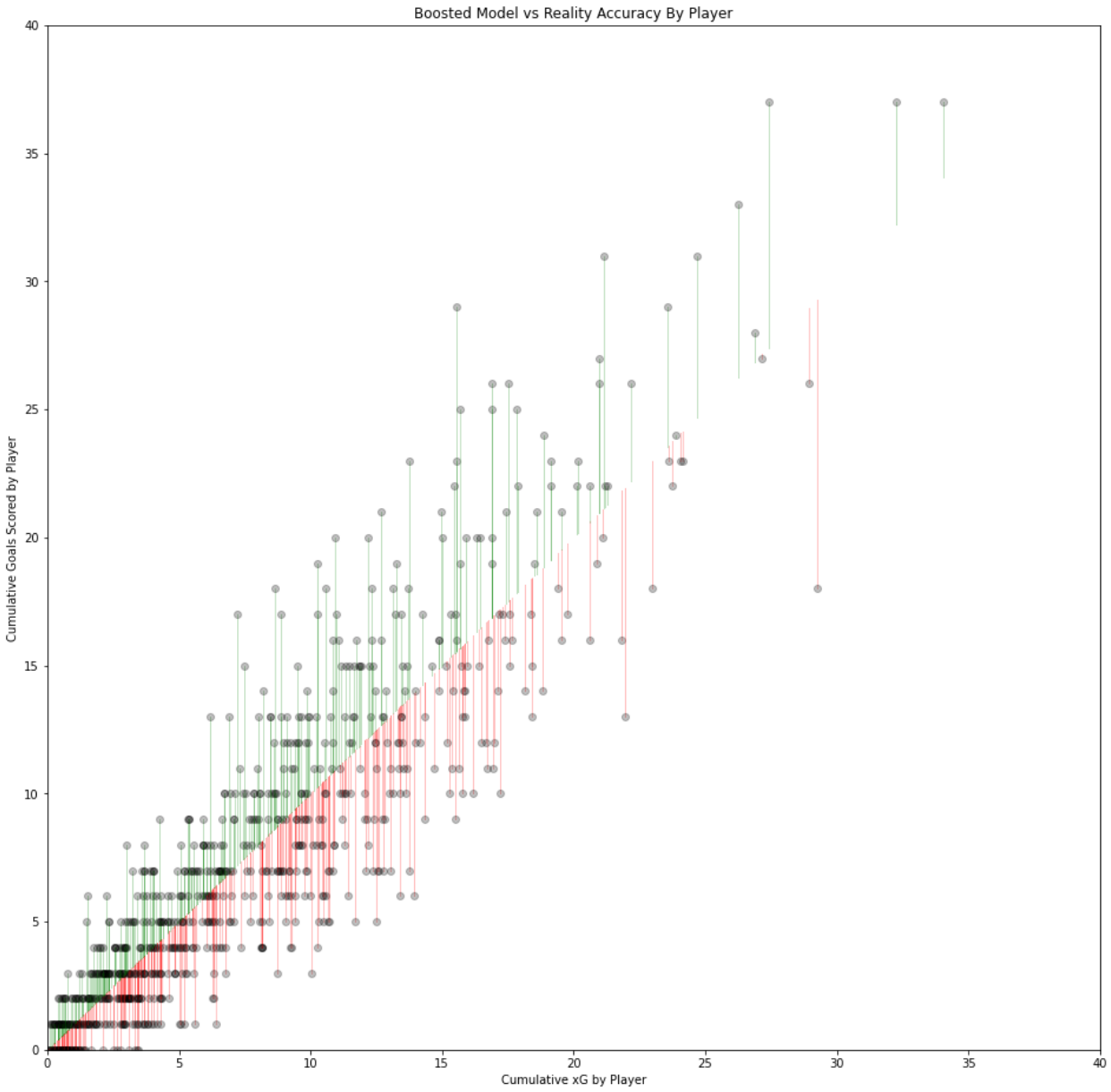

Fig f_35: Gradient Boosted Regression Cumulative Expected Goals vs Reality Error Bars

As depicted in Figure 35 the observations seem a little better than the random forest model. On the low-end of the graph, the error bars more closely straddle a slope of one. On the high-end of the graph the model still does struggle to adjust for people who store more than 30 goals, but it is closer overall than any other. The boosted model misses the cumulative total of

the average player's output by 2.081 goals. Making it overall the most accurate model when compared to reality.

Cumulative Model Comparison

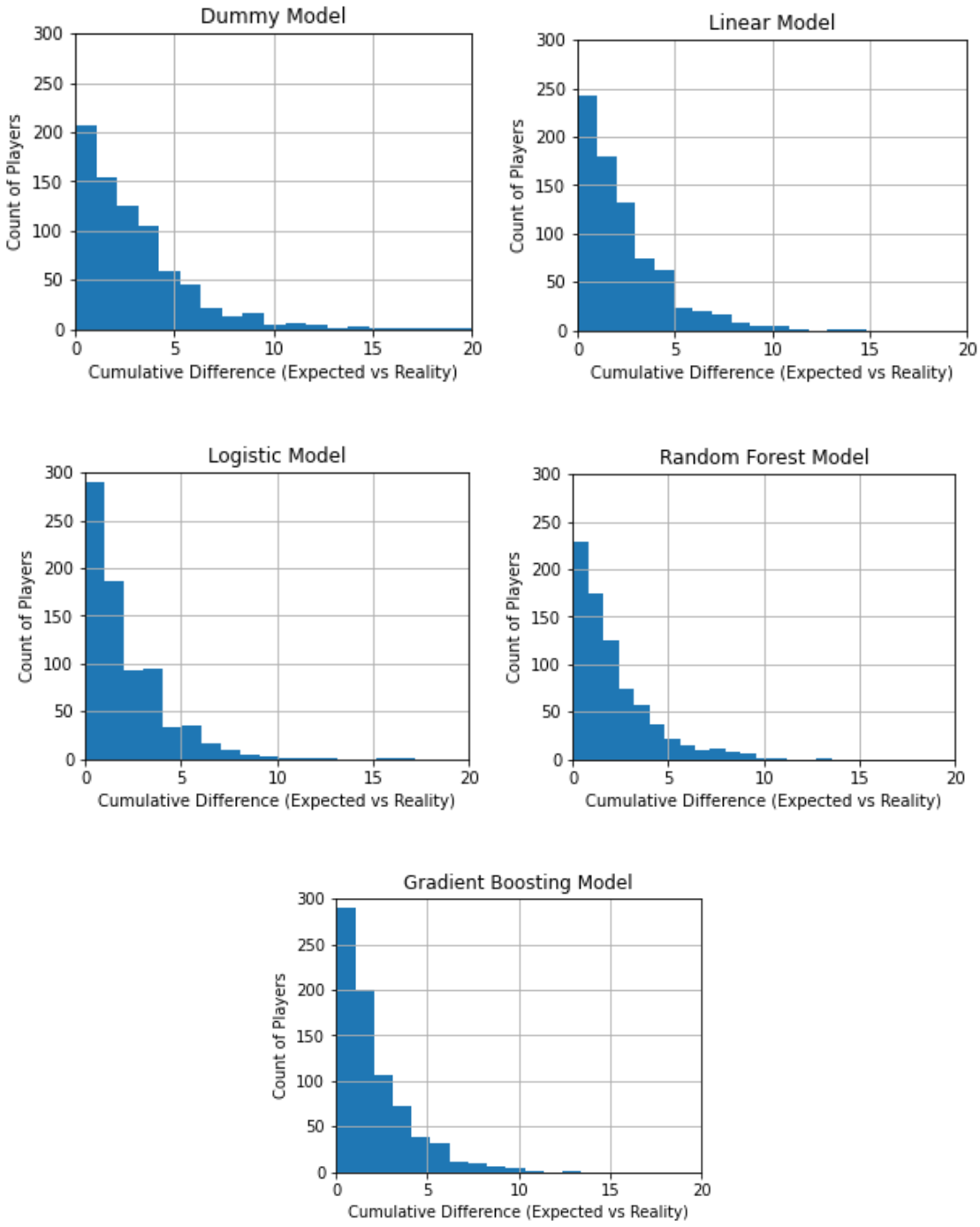

Fig f_36: Cumulative Model Comparison Histograms

Using Figure 36 to compare the cumulative totals of the model outputs it becomes evident which models do well over a large sample. There are a few interesting observations of note. The first, is just how good the logistic model was overall. Putting almost 300 unique players in the dataset to within 1 goal of what they produced in reality is quite good. Comparing that with the random forest model, it becomes clear that although the random forests were better about not missing by more than 15 goals on any player which the logistic model did. However, the random forest model was significantly worse than the logistic model in terms of being able to evaluate a player to within 1 goal of reality. The gradient boosting model really seems to take the best of both the logistic and random forest model. It was able to predict about the same number of players to within 1 goal of their totals in reality like the logistic model. Additionally, it did not miss by more than 15 goals on any player like the random forest model.

Model Validation

A large inspiration source for this project was Peter Tanner, and his creation MoneyPuck which is a publicly accessible set of hockey analytics available for free on the internet. The goal at MoneyPuck is to try to predict reality and compare it to the public markets in the form of public betting lines. This is slightly different from trying to determine the methodology of which players take the best shots at the best times. However, the fact they make their shot specific model available for anyone means that it can be used to validate the predictions which come from the models. In order to combine the datasets between the outputs of the newly created models and the data available on MoneyPuck the data points are combined by game id, time of event, and the location on the ice. The reason the location on the ice needs to be added is because there are sometimes multiple events which take place at the same time, such as a penalty being recorded at the same time a save is made. By including the location of the shots in the model

outputs, those additional data points that MoneyPuck includes in their model are excluded from the comparison allowing a better understanding of the difference between the model outputs and better evaluate the outputs.

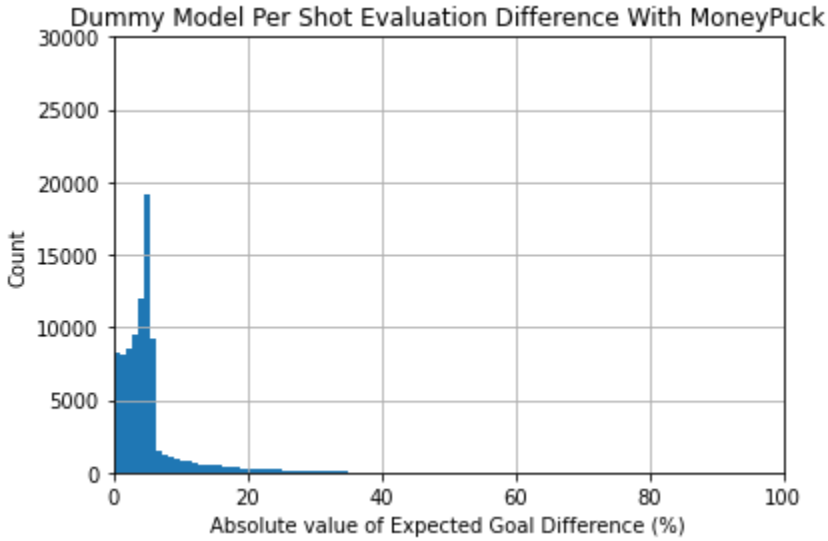

Fig f_37: Dummy Model Difference to MoneyPuck per Shot

Figure 37 demonstrates that the dummy model's constant nature does make a decently accurate model however it does struggle significantly with high-danger shots and generally misses by about 5-10% per shot.

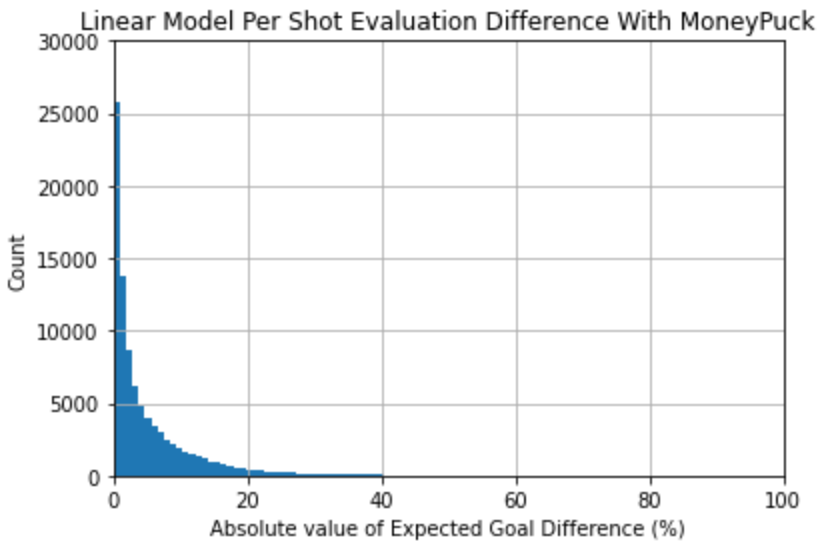

Fig f_38: Linear Regression Model Difference to Moneypuck per Shot

Figure 38 demonstrates once again, that despite the data shortcomings with respect to linear regression, it still does a thoroughly better job than the dummy model in terms of comparison to MoneyPuck and reality.

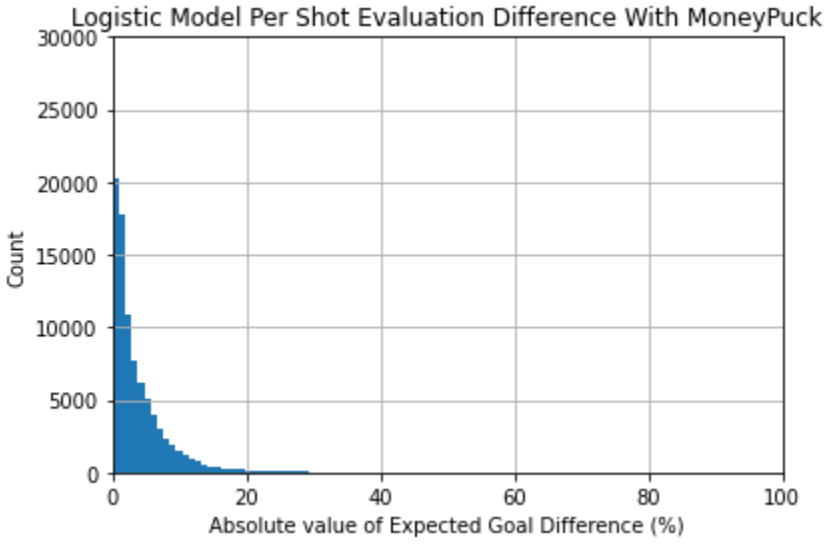

Fig f_39: Logistic Regression Model Difference to Moneypuck per Shot

Figure 39 demonstrates a stark comparison of the shot difference to MoneyPuck from the logistic regression model. On a per-shot basis it lines up much closer to the MoneyPuck model,

and some error is always to be expected between the 2 models, however even in these models it continues to miss on the high-danger shots.

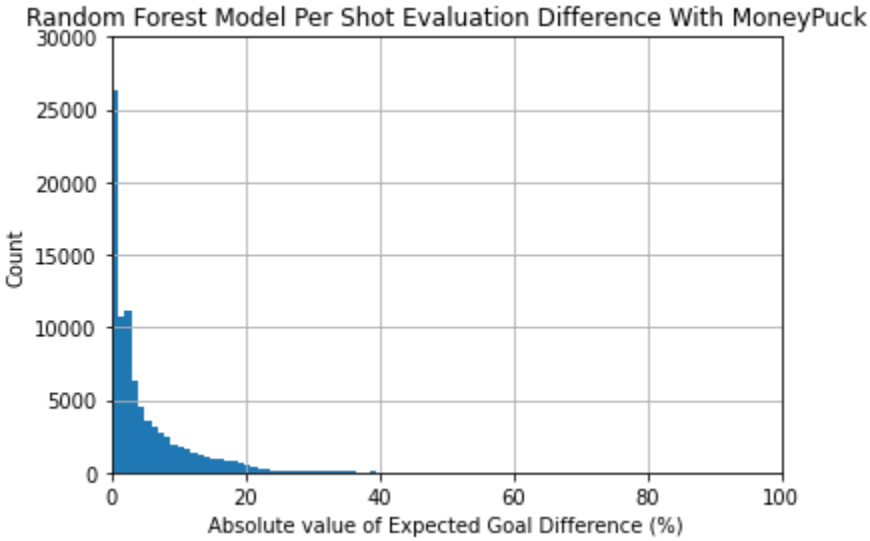

Fig f_40: Random Forest Regression Model Difference to Moneypuck per Shot

Figure 40 demonstrates the additional accuracy the random forest model was able to achieve. Increasing the number of events within 1% of the MoneyPuck model from just over 20,000 to over 25,000 shots in the test data

.

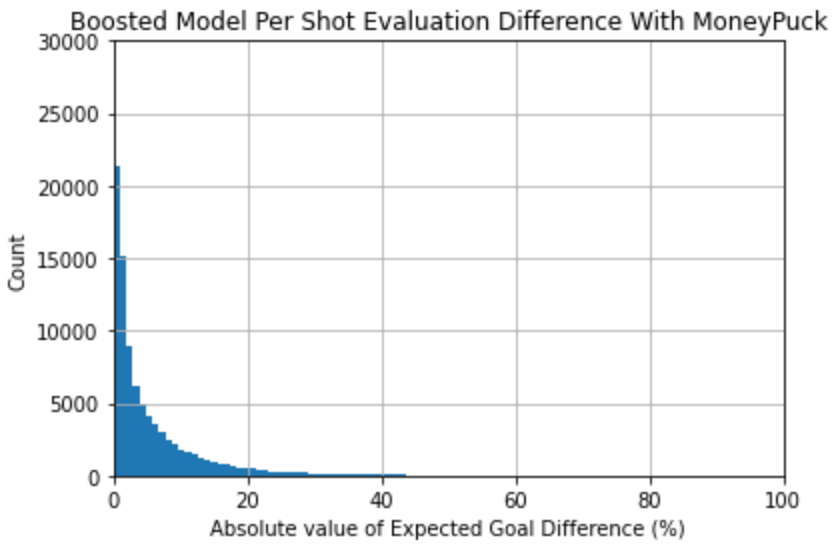

Fig f_41: Gradient Boosted Regression Model Difference to Moneypuck per Shot

Interestingly, Figure 41 shows that the boosted model deviates from moneypuck a little more relative to the random forest model despite it being the most accurate out of the models created for this project. Nonetheless, it does demonstrate that the model still very closely ties out to other publicly available models. This gives confidence moving forward to take the analysis on the events on a player level.

V. RESULTS

No matter how the data is cut it seems every popular model in the world considers Connor McDavid the best player in the world, and these models are no different. To visualize the data, players expected goal outputs are compared to other players at the same position. The difference between expected goal outputs from various locations on the ice are compiled and compared to the average NHL and then transposed over the offensive end of the rink.

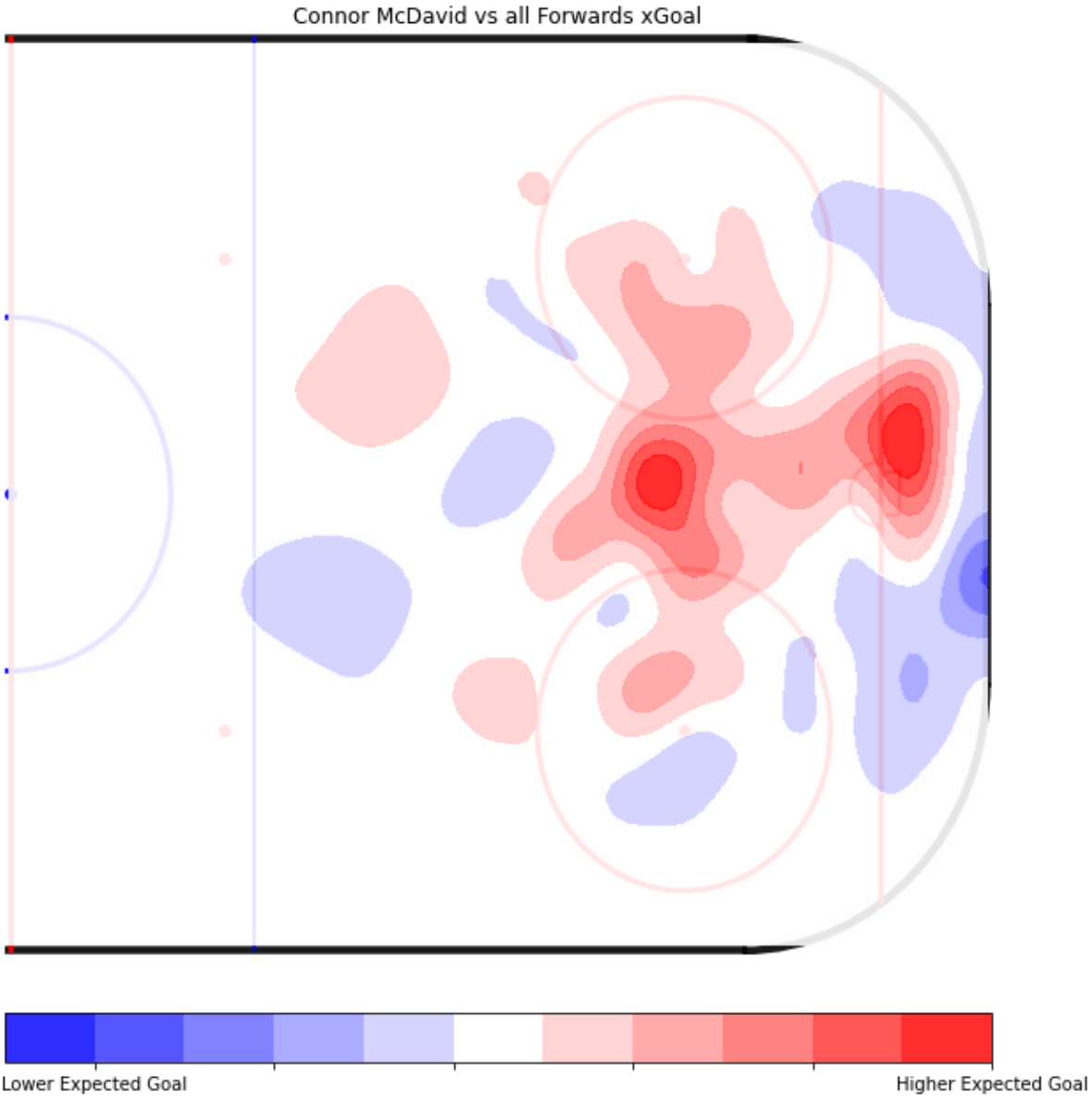

Fig f_42: Connor McDavid Heatmap

Figure 42 is generated for Connor McDavid. It is clear that his abilities are on par with average throughout the ice, and in front of the net his abilities are statistically better than other centers in the NHL. The graph also shows McDavid's preference to attack from the left side more than the right, this is mostly a result of his handedness.

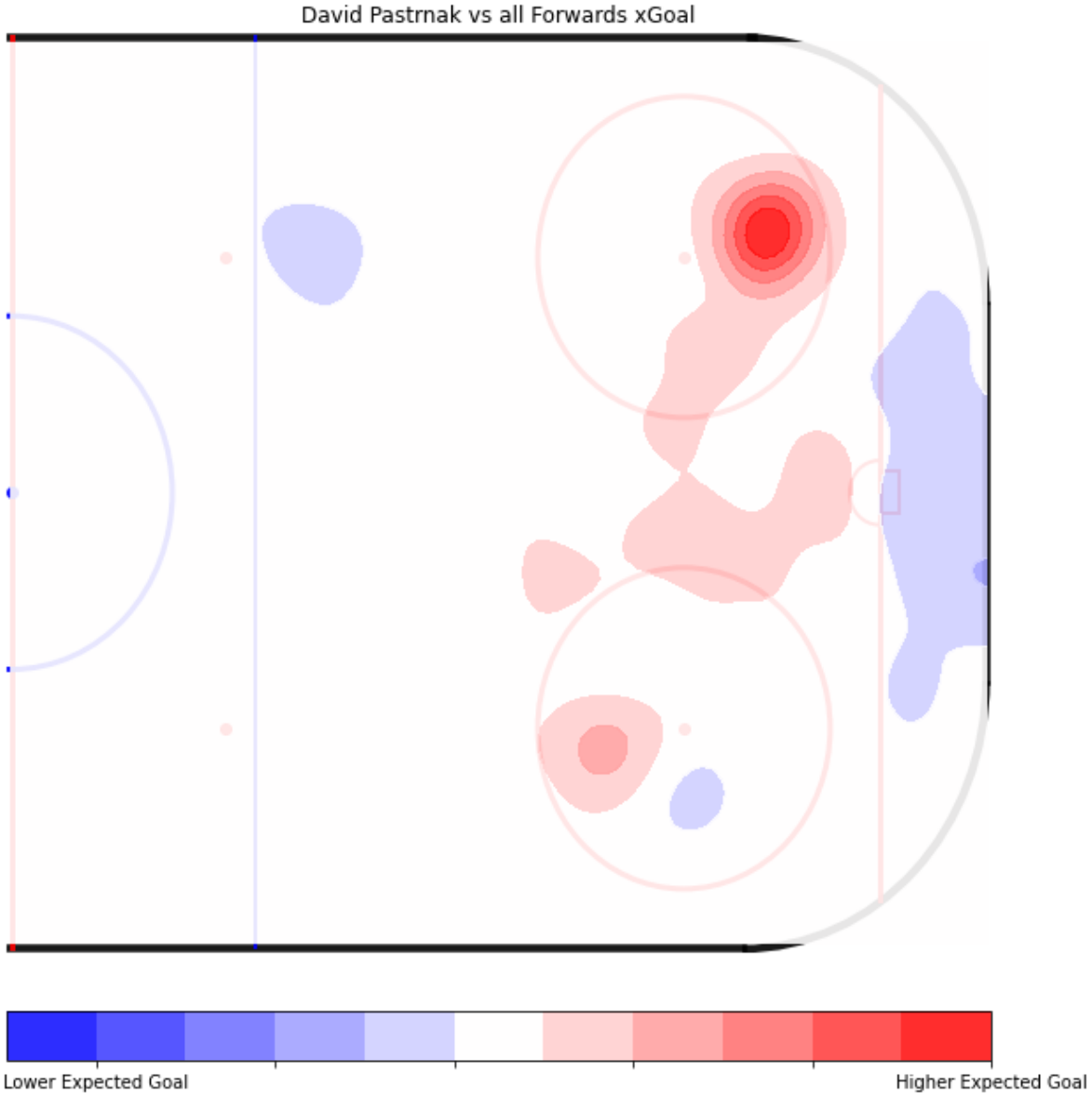

Fig f_43: David Pastrnak Heatmap

The next best player in the model is David Pastrnak, visualized in Figure 43, a right winger who plays for Boston. It is important to note when reading these heatmaps, that although

they provide visualizations from center-ice, the offensive game is played inside the blue line and most offenses are looking to create scoring chances closer to the goal than to the blue line. For this reason, a lot of the shooting data points from the blue line area are usually attempts to get the puck closer to the net before a final shot is taken. Often players will shoot towards the net, hoping for the puck to find a deflection on its flight. For these reasons, when observing data around the offensive blue line it is important to keep in mind the context of what these shots were likely intended to do.

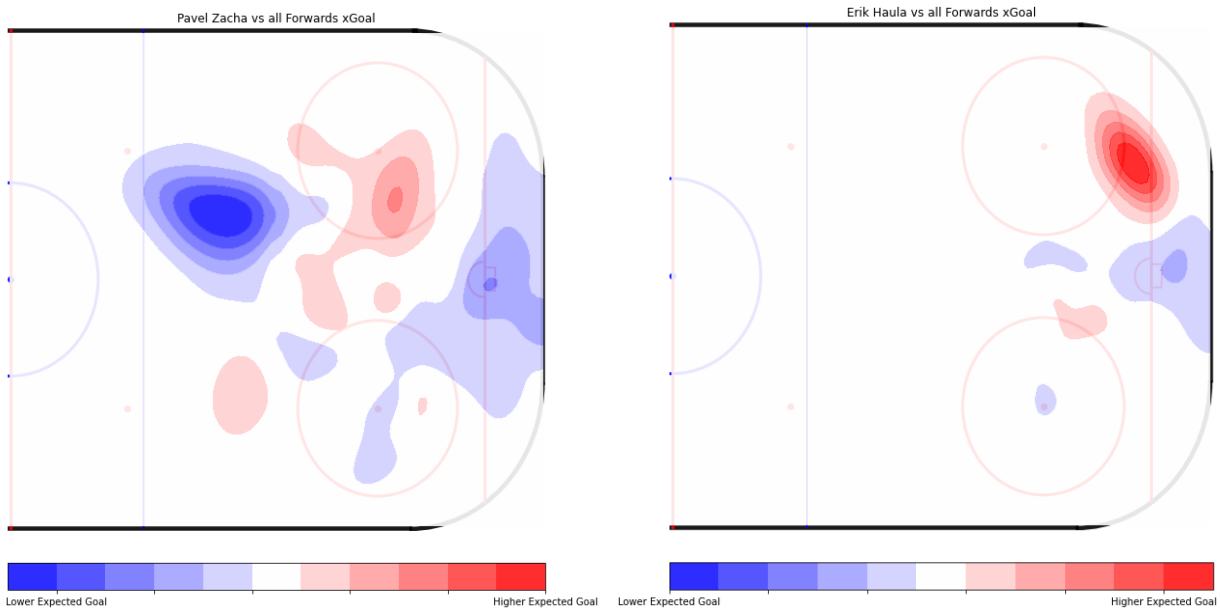

Fig f_44: Pavel Zacha & Erik Haula Comparison Heatmaps

Figure 44 is an example of how to compare players in evaluations. The 2 players in this example were traded for each other between New Jersey and Boston. Looking at their charts it seems that Pavel Zacha and Erik Haula are about equal overall. This is in contrast to their results in reality where Pavel Zacha was able to score 17 goals in 82 games played, whereas Erik Haula managed only 10 goals in 80 games played. This is where the interpretation of the player's results needs to be explained. The model does not show much difference between the two players, and yet their results in reality were pretty different. There are a couple of factors at play.

Firstly, the model predicted both players as having cumulative expected goals between 13 and 14 based on their shot selection and play. However, Pavel Zacha found himself on a Boston team which set a new record for total points (in NHL history) in the 2022-2023 NHL season. It was a legendary regular season for Boston where every measurable statistic and metric indicated that Boston was outperforming historically dominant NHL teams in an unprecedented manner since the beginning of the modern NHL era. It is likely that playing on such a dominant team boosted Pavel Zacha's output. As mentioned previously when discussing Corsi, the teammates on a player's line make all the difference. Conversely, Erik Haula found himself on quite a good team with NJ in 2022-2023, but despite the team making the playoffs his final output was not on par with that of Pavel Zacha, even though the model estimated their cumulative shot selection to be relatively even.

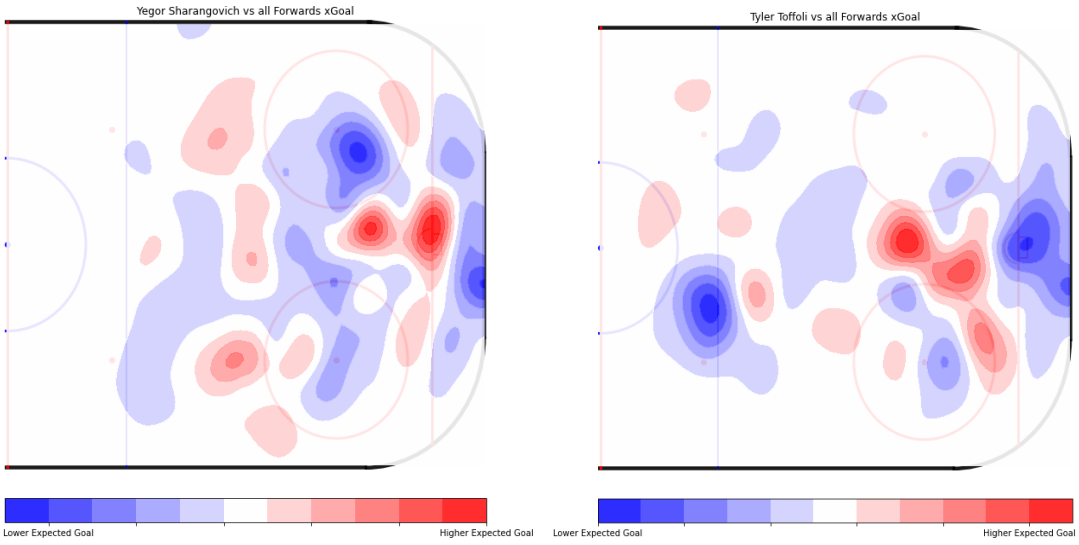

Fig f_45: Yegor Sharangovich & Tyler Toffoli Impact Heatmaps

Figure 45 illustrates another two players who were traded for each other. In this case the player on the left, Yegor Sharangovich, was packaged with a draft-pick for the player on the right, Tyler Toffoli. Interestingly, they have relatively similar effects on the ice. However, it is

clear that although both players are effective in front of the net, Yegor Sharangovich struggles a bit more than Tyler Toffoli as the shot gets further from the net towards the circles. The final results support these conclusions as Sharangovich was predicted to score a little above 11 goals in the 75 games he played. Whereas, Toffoli was predicted to score just over 21 goals in the 82 games he played. In reality, Sharangovich managed to score 9 goals to Toffoli's 20 within the dataset. This is where the analysis becomes interesting, because the natural question which arises is whether the draft-pick will make up for the roughly 10 goal difference between the player outputs.

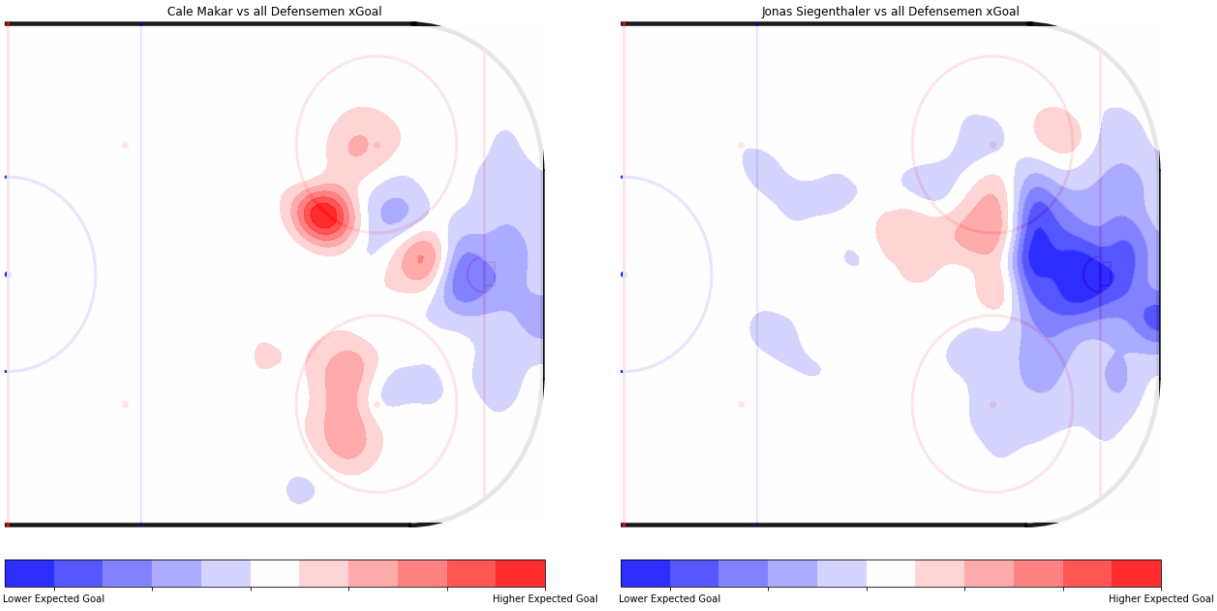

Fig f_46: Cale Makar & Jonas Siegenthaler Impact Heatmaps

The last subset worth demonstrating is the defensemen as the model is also able to evaluate the merits of the shots they take. On the left side of Figure 46 is Cale Makar, widely considered one of the most gifted offensively-capable defensemen in the NHL. On the right side of Figure 46 is Jonas Siegenthaler, who has quietly become a talented stay-at-home style defensive-defenseman. Given that the defensive players' responsibilities are mostly to keep the other team from scoring, their offensive outputs are considered to be bonuses to whichever team

they play on. However, observing the model outputs it becomes clear just how good Cale Makar is on offense when compared to someone like Siegenthaler. This also certainly agrees with the observations of reality. Makar, despite only playing only 60 games, was able to score 11 goals when the model predicted just over 9 in that span. Siegenthaler, on the other hand, played in 80 games throughout which the model predicted a total of just under 2 goals. In reality, Siegenthaler scored 3 goals in the dataset. Nonetheless, the model was able to accurately capture the difference in offensive output between these two defensemen, regardless of the number of games they played.

In addition to the comparison to Cale Makar, the Jonas Siegenthaler heatmap also provides a good context for what a less offensive player looks like. By the front of the net there are multiple steps of blue indicating the inability to find dangerous shot selection from dangerous areas on the ice. Contrasting a heatmap like that to Figure 43 for McDavid, the observation which stands out is the ability to find or create high probability chances from closer to the net.

Being able to compare the merits of the shots which NHL players take gives the ability to compare their predicted goal scoring. Overall, some trends present themselves in these charts as well. Firstly, the better players seem to do close to the net, the better their overall performance is. Also, somewhere around the blue-line as well as past the goal-line the impact results become less significant.

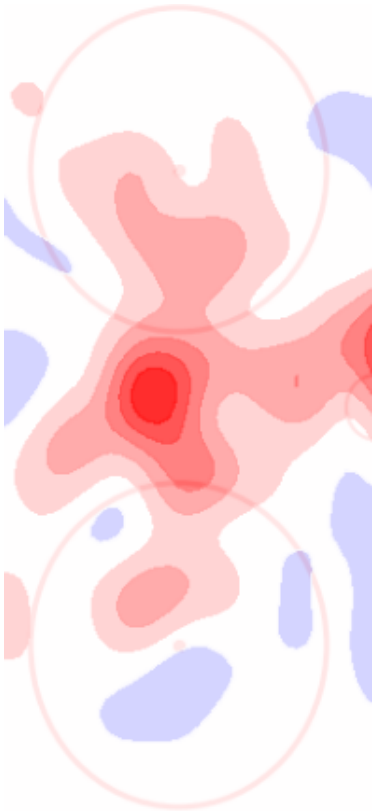

Fig f_47: McDavid's Net-Front Impact

The inverse of this indicates where to look by zooming in on the area from the front of the net to the edges of the circles. This area displayed in Figure 47 correlates best to the cumulative model outputs. Going back to the original goals of this project, the intention was to be able to compare players as fans or general managers in order to be able to contrast their goal scoring. These visualizations do allow for the effective comparison between players.

Table t_11: Top 5 Model Outputs

Player ID	Player Name	Cumulative xG	Even Strength Goal Total
8478402	McDavid	34.05	37
8477956	Pastrnak	32.24	37
8475786	Hyman	29.28	18
8479318	Matthews	28.97	26
8478420	Rantanen	27.41	37

Lastly, table 11 provides the top five cumulative expected goal outputs across the entire NHL according to the model. All players here are some sort of forward, and it is worth noting the biggest miss there is Hyman. This is ironically easily explained by knowing that he is teammates with the top player on the list. Hyman playing with McDavid has led to him getting some of the best shot selection opportunities in the NHL. The linemates with which these players play make a significant impact on their projections over a season, and playing with someone who almost won the MVP award unanimously lifted the analytics behind Hyman's shots even if the final numbers in reality did not demonstrate that effect necessarily.

VI. CONCLUSION

In this thesis, multiple supervised learning techniques were used to explore and create effective goal scoring models in the NHL. The objective of this work was to explore the methodology of shooting in the NHL in hopes of understanding the players who create the best scoring chances throughout the NHL on the merit of the shots they take. The work done in this project was done almost exclusively using Python in Jupyter Notebooks, storing data in SQL and using the Sci-Kit Learn and Matplotlib libraries for machine learning and visualization, respectively.

NHL shot data going back 9 years was collected and organized. A dummy model was created by using the NHL average shooting percentage, and from there 4 different kinds of supervised learning models were used in order to attempt to model the chances of any particular shooting opportunity resulting in a goal scored.

Multiple thousand different models were created, and ran, across different supervised learning methods. The results were extrapolated and compared on individual events and on the sum of the season as a whole.

Through all the experiments and analysis it was determined that an extreme gradient boosted regression model did best with accurately predicting the chance of any particular shot becoming a goal. This model not only was able to achieve the best performance on a per-shot basis, but it also most accurately reflected the cumulative reality. Using this model, the cumulative results over the course of a season were able to provide a good understanding of player performance in comparison to peers.

VII. FUTURE WORK

With respect to continuing the research in this thesis, the ultimate goal is to develop a more comprehensive understanding of events on the ice and how that may affect the model. There are multiple data points which the model could have benefited from knowing.

Information such as the shot speed on any given shot would help in providing the model more context on the shot which was taken. Also, an understanding into how long players have been on the ice could indicate if the shooter or the defensemen are more fresh or more tired which may have an effect on individual shot events. It would also be nice to know the goaltender position at the time of a shot, giving the model an indication for if the goaltender is facing the shot squarely or if the shot is coming from a strange angle. Lastly, if there was the ability to tell the true angle that the puck took off of a player's shot attempt that would allow the model to get a little better in its predictions. If the angle and goalie position information were both included in the model then the real situation would be even more accurately reflected in the model, and hopefully result in more accurate predictions.

Given time to collect and aggregate this information the model may adjust for shooting talent and more specific situations. Hockey's continuous low-event nature means that the analytics will always be tricky to model accurately. Having these additional variables can only make the models more accurate in assessing a player's offensive methodology.

Additionally, with respect to the comparison methodology, it is impossible to truly quantify a player's impact on a per-minute basis as the usage of the coach does interfere. However, it may be possible in the future to group together players by their ice time. For example it is unfair to compare a player who averaged less than 10 minutes per game to a player who averaged over 20 minutes. However, perhaps those players who played less than 10 minutes

can be compared to each other as well as the other buckets which players may fall into. Having this additional metric on which to compare could potentially provide more accurate comparisons in similar players.

REFERENCES

[r_1] "National Hockey League." *NHL Public API*, statsapi.web.nhl.com/api/v1/. Accessed 7 Aug. 2023.

[r_2] "Ice Hockey Rink." *Wikipedia*, 13 June 2023, en.wikipedia.org/wiki/Ice_hockey_rink.

[r_3] Macdonald, Brian. *Adjusted Plus-Minus for NHL Players Using Ridge Regression ... - Arxiv.Org*, 3 Oct. 2012, arxiv.org/pdf/1201.0317.pdf.

[r_4] *NHL 2022 Official Rules*, 2021, cms.nhl.bamgrid.com/images/assets/binary/326142322/binary-file/file.pdf.

[r_5] Taylor, Sebastian. "Multiple Linear Regression." *Corporate Finance Institute*, 12 May 2023, corporatefinanceinstitute.com/resources/data-science/multiple-linear-regression/.

[r_6] Vaidya, Dheeraj. *Multicollinearity - Definition, Types, Regression, Examples*, www.wallstreetmojo.com/multicollinearity/. Accessed 7 Aug. 2023.

[r_7] Sargent, Thomas J, and John Stachurski. "Multivariate Normal Distribution." *Intermediate Quantitative Economics with Python*, python.quantecon.org/multivariate_normal.html. Accessed 7 Aug. 2023.

[r_8] "Scipy Stats Multivariate Normal." *Scipy.Stats.Multivariate_normal - SciPy v1.11.1 Manual*, docs.scipy.org/doc/scipy/reference/generated/scipy.stats.multivariate_normal.html. Accessed 7 Aug. 2023.

[r_9] Chakure, Afroz, and Brennan Whitfield. *Random Forest Regression in Python Explained*, 27 Apr. 2023, builtin.com/data-science/random-forest-python.

[r_10] Ho, Tin Kam. *Random Decision Forests*, 17 Apr. 2016, https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf.

[r_11] Simic, Milos. "Gradient Boosting Trees vs. Random Forests." *Baeldung on Computer Science*, 15 May 2023, www.baeldung.com/cs/gradient-boosting-trees-vs-random-forests#:~:text=Gradient%20boosting%20trees%20can%20be,and%20start%20modeling%20the%20noise.

[r_12] "Stanley Cup History." *HHOF*, www.hhof.com/thecollection/stanleycup_history.html. Accessed 7 Aug. 2023.

[r_13] "Sports Analytics." *Wikipedia*, 31 July 2023, en.wikipedia.org/wiki/Sports_analytics.

[r_14] "Corsi (Statistic)." *Wikipedia*, 25 July 2022, en.wikipedia.org/wiki/Corsi_(statistic).

[r_15] Fitzsimmons, Scott. "Long Contracts Are Making a Mockery of the NHL Salary Cap." *Bleacher Report*, 2 Oct. 2010, bleacherreport.com/articles/422577-long-contracts-making-a-mockery-of-the-nhl-salary-cap.

[r_16] Pierson, Lillian. "Logistic Regression Example in Python (Source Code Included)." *Data Mania*, 16 Mar. 2023, www.data-mania.com/blog/logistic-regression-example-in-python/.

[r_17] Nesbitt, Andy. “MLB Commissioner Rob Manfred Spit in the Face of All Baseball Fans on Sunday.” *USA Today*, Gannett Satellite Information Network, 17 Feb. 2020, ftw.usatoday.com/2020/02/mlb-rob-manfred-spits-in-face-of-all-fans.