Increasing Offensive Efficiency in the NCAA Women’s Ice Hockey

Spencer Fascetta

Endicott College

**Dedication**

-To my parents, Kevin and Tammy, for always supporting me, and providing me the opportunities to become the person I am today.

-To my sister, Sydney, who always pushes me to be the best version of myself, holds me accountable, and whose continued on-ice dominance inspires me to keep pushing the boundaries of women’s ice hockey. Watching her play is surreal, and the amount of pride I have each time she steps on the ice cannot be properly expressed in words. Go get ‘em, Squid.

-To my grandmother, Nancy Hager, for her unwavering love and support, and allowing me to jabber on about a topic that she knew absolutely nothing about. Love you Fance.

-To Bill Bowes, for embodying the idea that attitude overcomes all, and kicking cancer’s butt.

AND

-To all those who fear their future. Never doubt yourself, or your abilities. You need only the adoration of one person. Yourself.

***“Things aren’t always what they seem. Our fears can play tricks on us, making us afraid to change course, afraid to move on, but usually hidden behind our fears are second chances waiting to be seized. Second chances at life… at glory… at family… at love. And these opportunities don’t come around every day. So, when they do, we have to be brave, take a chance, and grab them while we can.”***

**-*Barry Allen***

**Acknowledgements**

Thanks to Dr. Katherine Kilty and Professor Michael Diehl, both of Endicott College, for their help navigating a thesis paper for a double major for which there was no blueprint. To Endicott Sports Information Director Shawn Medeiros for his constant support in developing strategies to track live stats at games, and for providing me an outlet to ramble on about my topic to who actually understood what I was talking about. To Matthew Barlowe for his assistance diagnosing and error checking the Python code in the programs written for this study. To Ryan Stimson of *The Athletic* and the Rochester Institute of Technology for his insight on prior analytics research. To Shawn Ferris of *Stanley Cup of Chowder*, Ian Tulloch of *The Athletic* and *The Leafs Geeks Podcast*, and Dmitri Filipovic of *ESPN*, *Yahoo Sports*, and *The Hockey PDOCast* for being excellent sounding boards for the ideas presented in this paper. To my friends, Anna Giles and Erika Graves, who let me blather on about this idea of mine with little-to-no clue what in the heck I was talking about. To my friend Mike Cratty for being a close ally to bounce ideas off of. To the great people at *BlackNGold Hockey*, especially Mark Allred, Court Lalonde, and Rob Tomlin for the constant support in finding my voice in this medium. To Head Coach Hilary Witt, Associate Coach Stephanie Jones, Assistant Coach Bill Bowes, Director of Women’s Hockey Operations Stephanie Walsh, Team Manager Dana Valetti, and each and every one of the players on the 2018-19 University of New Hampshire Women’s Ice Hockey Team for providing me the positive environment and means to conduct my research, give real-time feedback, and for welcoming me into their family with open arms. It was truly an honor.

Abstract

This paper aims to determine whether or not there is a more logical and efficient means of tracking offensive generation in NCAA Women’s Ice Hockey. It utilizes shot attempt data collected over the 2018-19 season to compare the relationships between shot outcomes and the type, quality, and danger of each attempt, and number of shot assists preceding each attempt. Multiple programs were written in Python to separate and analyze the data, and a logistical model was produced to assess the accuracy and repeatability of the data collected. While the dataset was heavily influenced by the nature in which the data was collected, it was still determined that the number of shot assists preceding a shot had an unexpected influence on the probability of scoring a goal, and redirections and rebounds were the most efficient mean through which to generate a goal.

*Keywords:* shot attempt, shot assist, shot danger, shot quality, shot type, ice hockey

**Table of Contents**

**Acknowledgments – 3**

**Abstract – 4**

**Table of Figures – 6**

**Table of Tables – 7**

**Chapter 1 – Introduction – 8**

**Chapter 2 – Review of Literature – 12**

**Chapter 3 – Procedures and Methodology - 25**

**Chapter 4 – Results - 29**

**Chapter 5 – Discussion – 39**

**Chapter 6 – Summary and Conclusions – 43**

**Bibliography – 46**

**Appendices – 50**

**Appendix A: Pertinent Terminology - 50**

**Appendix B: Python Code – 58**

**Appendix C: Decisions Made – 93**

**Appendix D: Filtered Data - 95**

**Table of Figures**

|  |  |  |
| --- | --- | --- |
| Figure Number | Title | Location(s) |
| 1 | Shot Outcomes by Shot Danger | 31 |
| 2 | Shot Outcomes by Shot Type | 32 |
| 3 | Shot Outcomes by Number of Shot Assists | 33 |
| 4 | Shot Outcomes by Shot Quality | 34 |
| 5 | Shot Outcomes by Shot Danger and Shot Quality | 35 |
| 6 | Shot Outcomes by Shot Danger and Shot Type | 36 |
| 7 | Shot Outcomes by Shot Quality and Shot Type | 35 |
| 8 | Receiver Operating Characteristic | 92 |

Table of Tables

|  |  |  |
| --- | --- | --- |
| Table Number | Title | Location(s) |
| 1 | Individual Variables for Data Collection | 26 |
| 2.1 | Shot Danger vs Shot Outcome Data | 95 |
| 2.2 | Shot Type vs Shot Outcome Data | 96 |
| 2.3 | Shot Quality vs Shot Outcome Data | 97 |
| 2.4 | Number of Shot Assists vs Shot Outcome Data | 98-99 |
| 2.5 | Shot Danger and Shot Type vs Shot Outcome Data | 100-102 |
| 2.6 | Shot Danger and Shot Quality vs Shot Outcome Data | 102-104 |
| 2.7 | Shot Quality and Shot Type vs Shot Outcome Data | 104-107 |
| 3.1 | Shot Danger vs Shot Outcome Binary Data | 107 |
| 3.2 | Shot Type vs Shot Outcome Binary Data | 108 |
| 3.3 | Shot Quality vs Shot Outcome Binary Data | 108 |
| 3.4 | Number of Shot Assists vs Shot Outcome Binary Data | 109 |
| 3.5 | Shot Danger and Shot Type vs Shot Outcome Binary Data | 110 |
| 3.6 | Shot Danger and Shot Quality vs Shot Outcome Binary Data | 111 |
| 3.7 | Shot Quality and Shot Type vs Shot Outcome Binary Data | 112-113 |

**Chapter 1**

**INTRODUCTION**

The introduction of analytics into sport has come rapidly within the last decade (Fried & Mumcu, 2017; Fry & Ohlmann, 2012). Through the introduction and implementation of increasingly advanced and complex machine learning techniques, it has become plausible to predict player production and value over the span of their career, rather than basing personnel decisions solely on the eye test (Passfield & Hopker, 2017; Riley, 2017). While relatively rudimentary research has been completed and published for the sport of hockey into offense generation (Tulsky et. al, 2013) and individual player impact (Weissbock & Inkpen, 2014-A; Weissbock & Inkpen, 2014-B), there is still much that is unknown about the subtle intricacies of the sport of ice hockey. As such, there are likely many market inefficiencies that can and should be exploited by the teams and organizations with fewer resources than the juggernauts of professional sport. This paper investigates the feasibility of increasing offensive efficiency, or the rate at which offensive scoring chances are successfully generated and converted into goal, and how the style or system of play for a team can be adapted to take advantage of these proposed theories. Furthermore, it attempts to determine if this notion of offensive efficiency is not only an accurate predictor of future success, but a means of finding undervalued or underutilized players that will succeed in a different role or on a different team. This would, theoretically, identify an untapped market inefficiency in player acquisition and analysis, which would provide a competitive advantage for underfunded or small-market teams in the sport of ice hockey.

**Statement of the Problem**

Hockey, as a sport, has always been slow to modernize. For example, they didn’t mandate the wearing of protective helmets for players until the early 1980s; players weren’t required to wear a protective face shield until the early 2010s; they still insist that there is no evidence of the sport leading to Chronic Traumatic Encephalopathy, or CTE, despite the litany of scientific proof that has been researched and presented. Naturally, the sport has been slow to adopt a modern mathematical approach. Publicly accessible data did not exist until the late 2000s. from databases such as NaturalStatTrick and Corsica, the public finally began to scratch the surface of the potential benefits statistical analysis could have for the sport. While this has undoubtedly led to important developments and alterations to how the sport is played, such as a larger emphasis placed on shot generation and location and zone entries and exits, there is much more that has still yet to be explored. Through the use of machine learning and predictive modelling techniques, it should be possible to create an equation that defines offensive efficiency within the context of the offensive zone in ice hockey, as there is no current research within this specific field.

**Definition of Terms**

Please see Appendix A for any terms used or referenced in this paper.

**Delimitations**

There are a few pertinent delimitations to this experiment. First and foremost, I collected the data by hand, with no other assistance. While the model was generated using machine learning techniques, there is almost certainly a level of human error in the data itself. Likewise, I chose the specific parameters that the data was measured on, meaning they are subject to my own personal biases in how to interpret each definition. Finally, my lack of experience in writing the code for a predictive model was also a hindrance. Much of this was built upon a single semester course, titled Data Mining and Visualization, which is hardly the level of experience one would need to build many of the intricacies of a model into a program. Regardless, I did the best I could with the knowledge I possess.

**Limitations**

In addition to the noted delimitations, the study also presented with several limitations. The shot attempts are raw data, and therefore do not adjust for score or venue, both of which have been shown to have a tangible impact on shot attempt data. This is due to a lack of necessary data in order to make these adjustments. Ideally, a study such as this would incorporate significantly more data, most likely the results from every game and every game throughout more than a single season. This data has the consistent presence of a single team, potentially skewing results based on the strengths and weaknesses of that team.

**Significance of the Problem**

While there has been a lot of research done into the generation of offensive opportunities, such as the research paper from Eric Tulsky, Geoffrey Detweiler, Robert Spencer, and Corey Sznajder (2013) which discusses the importance of successful controlled zone entries and their impact on offensive generation, there is currently very little publicly available on how to maximize the efficiency of these opportunities. By determining this, the hope is to uncover a hidden means of player and system evaluation, particularly how said player fits within a given coaching system. This effects player salary and worth and finding undervalued and therefore underpaid players who would be beneficial to teams with fewer financial resources than others. It could apply to professional teams, as franchises such as the Florida Panthers, or Ottawa Senators, routinely pay lower salaries to their players, and are forced to allow others to leave the organization because they cannot afford them. It could, however, help collegiate or junior programs as well, by identifying under-scouted or underrecruited players for programs that are at a geographical disadvantage, such as the University of Alabama-Huntsville, or Nebraska-Omaha, neither of which lay in what one would consider a hotspot for the development of hockey players.

**Research Question**

What is the most efficient means of generating consistent offensive opportunities in the sport of ice hockey, and how can teams utilize this information to gain a competitive edge against their opposition?

**Chapter 2**

**REVIEW OF LITERATURE**

The following review of literature attempts to understand the growing use of analytics in the sport of ice hockey, and how they can be utilized and/or manipulated so as to determine the most efficient means of generating offense, thereby uncovering a potential market inefficiency in player evaluation. It looks at the integration of analytics into sport in general, and the various ways in which the business world and sports world mirror each other in their use of the medium. It then attempts to discover the ways machine learning and predictive modeling can be utilized in order to better analyze the complex data that a sport such as ice hockey inherently produces. Finally, it looks to existing research in analytics in ice hockey, how that information is currently being applied to the sport, and how that existing research can be adapted for the use of this study.

**Integration of Analytics into Sport**

The advent of analytics has become increasingly prevalent in the sports world over the past decade. Fry and Ohlmann (2012) stated “There has been an increase in the number of front office personnel with quantitative training and (or) the appreciation for the power of analytics to help improve the performance of their teams both on the field of play and from a business perspective” (p. 105). While the idea of using analytics is typically referred to in the context of player and team improvement and performance, it has also begun penetrating the business side of sports. As Fried and Mumcu (2017) point out, “Traditionally, sport teams have applied analytics to player and team strategies on the field or court focusing on player and team statistics; however, the use of analytics has made its way also to the front offices of sport organizations, as shown by the increase in the use of analytics to make evidence-based decisions in each of the industry’s segments” (p. 421). Mondello and Kamke (2014) also point out that, “While data can serve as an incredibly valuable resource, the utility of data is largely dependent on how well it is analyzed and more importantly communicated to a broader audience” (p. 2).

**Use of Analytics-Driven Departments in Non-Sports Organizations**

As sports teams and leagues have begun embracing the use of statistical data in their day-to-day operations, so too have non-sport business entities. Mondello and Kamke (2014) again provide an excellent example of how an insurance company, Assurant Solutions, has implemented the use of analytics into their business model successfully, According to the two, Assurant “sells credit insurance, debt protection, and competes for market share in the highly competitive credit insurance business where customer retention remains a significant industry problem” (pp.3-4). Despite the 16% retention rate for Assurant Solutions being consistent with industry-wide standards, it had five out of every six customers drop their coverage, and cease interacting with the company. Mondello and Kamke (2014) note that “Although they were analyzing the key to keeping customers loyal, they did so with the wrong approach. Consequently, Assurant’s leadership decided to implement a new analytical strategy” (pp. 3-4).

Mondello and Kamke (2014) then compare this example to one of the Orlando Magic implementing the use of analytics into their business operations in addition to their on-court performance. **“**Once the redeemed tickets were returned, the Magic organization followed up with consumers and subsequently collected valuable consumer information to help quantify the elusive ROI. Among their findings: 26% of the consumers Sports Analytics 6 had never visited a Tijuana Flats restaurant, 66% would not have visited without the promotion, and 85% indicated they would visit again. The Magic could now tangibly provide the sponsor with several ROI metrics linked to the promotion and identify strategies to increase awareness and ultimately revenues” (p. 5-6). While the use of analytics has become more widespread, it has become apparent that there is much to learn in the field. Franks, D’Amour, Cervone, and Bornn (2016) point out. “The core idea of our work is that quantifying sources of variability – and how these sources are related across metrics, players, and time – is essential for understanding how sports metrics can be used” (p. 151).

**The Use of Physical Data and Technology**

Teams and organizations have begun embracing the development of technological aides for the purposes of informing their decision-making process. Mondello and Kamke (2014) point out that “While the utility of analyzing data to increase new business development has been successfully integrated within the professional sports industry, franchises continue to explore different analytical data techniques to drive decision making. Consequently, sport organizations have recognized the added value new technologies present. Leaders in sport business and sales are becoming increasingly savvy with analytics. Data-driven analysis has become a competitive advantage in driving business strategies and challenging the industry overall to invest resources or risk the possibility of falling behind the competition” (p. 4). They also note that “Technological growth within various digital and mobile platforms assist in making data collection easier and more seamless than previous periods. Concurrently, sport consumers have higher expectations for smart, relevant marketing. With increased competition within teams, rival leagues, and other entertainment options, attracting and retaining consumers is especially important for the long-term viability of professional sports teams. Subsequently, understanding who the consumer is and furthermore establishing a deeper relationship with their individual preferences is important” (p. 7).

In terms of practical applications of the technology, Mondello and Kamke (2014) reference the use of RFID chips in replica jerseys by the Tampa By Lightning. The organization developed a customer loyalty program through which free customized team jerseys were offered to season ticket members. The two note the duality of the impact of this type of program, saying, “For the organization, fans wearing jerseys to the games helps foster a home-team atmosphere within the arena and on television. Other guests attending games or watching on television see these jerseys and may subsequently inquire about becoming season ticket holders” (p.8). The team also incentivized the season ticket holders to wear these jerseys to games, offering substantial discounts on concessions and merchandise if they did so through the use of the previously noted RFID chips that were sewn into the jersey sleeve, which were programmed to collect all purchase information made by that particular season ticket holder. Each individual who receives one of these jerseys is informed that such data collection is taking place, and the data is used by the organization to improve the overall fan experience at home games. (Mondello and Kamke, 2014 p. 8).

As technology advances more and more rapidly, one would expect the level of integration to increase exponentially. Passfield and Hopker (2017) note the additional practical applications these advances in tangible technology could have on sports moving forward, describing that, “In recent years there has been an explosion in the use of information technology in the sport and exercise fields”. The two also note the impact that this development has had thus far on the field at large, as well as the potential it has for increased impact moving forward. The use of websites to accumulate and aggregate increasingly larger repositories of both primary and secondary data has allowed sport and exercise scientists to access datasets that would have been too large to collate by hand. In particular, the two describe the development and use of wearable technology to be one of the more important developments in recent years. “The invention of wearable technology enables extensive measurements to be gathered during exercise, training, and competition. Increasingly, athletes and coaches recognize that such detailed, high-quality data can be used to inform objective decision making on aspects of training and performance. In this paper we discuss how rigorous analysis of large data sets may hold the potential to change sport, as well as the nature of its related sciences” (Passfield and Hopker, p. 851).

As stated previously, data can be tremendously helpful, but only paints half the picture. In order to understand whether a particular statistic or equation is useful, it needs to demonstrate that it has a reasonable level of accuracy when predicting future results. Through the use of predictive modeling, these statistics are subjected to hundreds of thousands of simulations to determine how well they predict existing data. As hockey is an inherently random sport (statistically speaking), these models can often become quite complex.

**Predictive Modeling and Machine Learning**

**Structural Equation Modeling**

The use of Structural Equation Modeling allows for the use of multivariate analysis. Riley (2017) discusses the intricacies of a structural equation model, or SEM, describing that, “At its core, a SEM consists of two categories of variables (measured and latent) and a path diagram that specifies the relationships between these variables” (p. 5). The idea of a SEM is developed from the notion that some constructs are impossible to be captured by a single variable. Riley also provides a hockey-specific example. “For example, the construct of offense in hockey cannot be fully captured by points alone (a player with 20 goals and 80 assists is very different than a player with 80 goals and 20 assists), but rather exists as some combination of multiple measured variables (e.g. points, goals, assists, and-so-on)” (p. 5). The measured variables in SEM must be variables that have been observed and directly collected data on, while latent variables are unobserved variables that are inferred from measured variables. Again, Riley (2017) relates this to hockey, as offense impact is typically inferred from goals, assists, points, and other more easily collected measurements. (p. 5).

The relationship between measured and latent variables is determined by using a confirmatory factor analysis (also known as a CFA). Each latent variable becomes a linear combination of its measured variables, whose relationships can then be used to calculate factor scores for each latent variable. This measurement creates a broader baseline of analysis for each latent variable in the model. Riley (2017) clearly outlines the goal of a SEM, stating that it is, “to specify a model whose estimated means and covariances (referred to as parameter estimates) fit the observed data. If a model produces parameter estimates that closely match the data, that model is said to be accepted; if the parameter estimates do not match the data, then the model is said to be rejected” (p. 6).

Harring, McNeish, and Hancock (2017) present a more nuanced evaluation of a SEM as an effective predictive tool for analysis, stating that, “If one views a model in the broad sense of a comprehensive mechanism for understanding the behavior of data in a population, then any extent to which that mechanism is incorrect constitutes a misspecification” (p.616). A model could potential posit multivariate normality, or even independence of observations as a part of a larger picture. A reliance on either of these instances is therefore flawed in nature. This can be prevented by relying on a complex, multilevel equation which accounts for potential fallacies within the context of the problem at hand.

**Machine Learning**

While incredibly helpful, an SEM is difficult to calculate by hand. This is where the idea of machine learning is key to the development and implementation of statistical analysis. Many programmers prefer to utilize languages such as R or Python to create a predictive model and teach the AI to adapt to the equations being fed into the program.Thomas, Ventura, Jensen, and Ma (2013) indicate the increased efficiency of using R to analyze large datasets, stating **“**Execution time varies with the total number of covariates, with the simplest cases (200,000 outcomes and 60 covariates) taking 30 processor minutes, to the more complicated runs (200,000 outcomes and 2600 covariates) requiring roughly 60 processor-hours” (p. 1506). They also “used multiple parallel chains with sufficient burn-in periods to collect a sufficient number of uncorrelated samples” (p.1506).

Weissbeck and Inkpen (2014-A) provide a definition for the idea of machine learning, stating that it is, “a branch of artificial intelligence that applies statistical methods with algorithmic modeling in order to teach a computer (machine) tasks such as ‘recognition, diagnosis, planning, robot control, prediction etc.” (p. 3). Using the techniques presented allows for the analysis of large amounts of data for patterns, which then allow for the discovery of relationships between actions and intended outcomes. The two also relate machine learning directly to its applications in ice hockey and the current lack of knowledge in this space, noting that “if it is not easily possible, we want to know why this is” (p. 3). Jawed, Ziad, Khan, and Asrar (2018) also provide an impetus for the use of machine learning in common, everyday activities, noting that, “The convenience provided by this intelligence has caused humans to rely more and more on machines for all sorts of activities from simple communications to completely entrusting a business into the hands of a machine” (p. 1698).

**Regression**

As the individual events throughout a hockey game are inherently random, the use of multivariate equations offers a more accurate approach to analysis of the sport. The best form to analyze multivariate equations tends to be in some sort of regression-based computation and model. As Macdonald (2011) states, “The main benefit of the weighted linear regression model is that the resulting adjusted plus-minus statistics for each player should in theory be independent of that player’s teammates and opponents. The traditional plus-minus statistic in hockey is highly dependent on a player’s teammates and opponents, and the use of the regression removes this dependence” (p. 1). Thomas et al. (2013) chose to utilize a Cox proportional hazards model for their particular research, so as to force the hazard function to have separate components for time dependence and predictors “as h(X, t) = h0(t)h1(X), where X can represent various factors such as the players and/or team on the ice” (p. 1502). They chose to begin with h0(t) = 1, so that the time until the next event is exponentially distributed with rate h1(X); the baseline rate is established with a corresponding intercept term in X” (p. 1502). The group then modeled the scoring rate of each team as a logarithm-linear Poisson process, with the intercept terms representing the baseline scoring rates for the home and away teams, as “the overall scoring rate for the home team is greater than for the away team; in this way, we explicitly detect a home-ice advantage” (p. 1502).

Thomas et al. (2013) describe the advantages of this method, noting that “rather than trying to model a single outcome, such as goal differential, we can simultaneously calculate both the offensive and the defensive player ability parameters for each player, which are known to be distinct” (p. 1503). In their study, they determine that such parameters must have a meaningful impact on game-outcomes, as it has a correlative effect on scoring rates. Therefore, one can assess individual player impact by comparing the expected number of goals scored and allowed by a player’s team given their ratings against the same data with ratings set to a zero sum.

Anderson-Bergman notes describes the use of interval-censoring in such a problem, stating, “Interval-censoring occurs when observations are not known exactly, but rather up to an interval. For example, suppose a component of a machine is inspected at time *c1* and *c2*. The component is observed to be operational at *c1* but broken at *c2*. In such a case, while the exact failure time is not known, it is known that the event occurred inside the interval (*c1, c2*]. In some cases, these intervals are small, and the interval-censored aspect of the data can be ignored with only minor biases” (p. 487). According to Anderson-Bergman, in order to completely define a parametric survival regression model, one needs to specify the both the baseline distribution of the model, and the effect of the covariates on the baseline distribution. He references icenReg, the program which his article utilizes, and denotes that Weibull, gamma, exponential, logarithmic-normal and logarithmic-logistic survival baseline distributions are included in the program. It also supports proportional hazards, AFT, and proportional odds. (488).

**Existing Work in Hockey**

**Existing Metrics**

Hockey is far behind the other three major sports in North America when it pertains to the use and integration of statistical analysis. Weissbeck & Inkpen (2014) note that while hockey is complex in both its watchability and play, it is still in its infancy as it pertains to the use of analytics, particularly when comparing its standing in the development and implementation of analytics to that of other sports, such as basketball or baseball. While this is important to establish the context of current research into analytics in hockey, there have still been a number of important developments within the last decade.

PDO, or the “luck” stat as it is colloquially known, has become one of the more widely accepted and utilized metrics available in the public AND private sectors. Dermody (2013) notes that while PDO has plenty of descriptive qualities, it has relatively poor repeatability, and therefore is somewhat limited in its applications. However, it is still recognized that, over the course of a standard game of hockey, a higher all-strength PDO tends to indicate which team shall win said game.

Another important innovation has been the research and development of aging curves, showing the prime years of a player’s productivity in the NHL. This allows teams to better allocate financial resources. As Desjardins (2009) points out, although the correlation between a player’s playing time and point production peaks between the ages of 23 and 26, the point production per game reaches the 90% mark of its peak value significantly earlier than that timeframe. It also maintains its value beyond the age of 30. He uses this to propose that the quality of a given league can be estimated utilizing the ratio of the player’s points per game production in a given league and year.

**Team and Individual Distinctions**

Aging curves are an excellent insight into individual production and productivity. As previously mentioned, Desjardins (2009) indicated that a points-per-game measurement of a player is the most consistently reliable measurements of player value. He states that, “for the PPG [points-per-game] production of players born after 1948 reaches its 90% level at age 21 or 22 and does not fall below that level until age 30 or 31, provided the player is still active” (p. 1). The key to understanding individual impact is the separation of the individual from the team, typically utilizing a form of a regression model to distinguish performance. As Macdonald (2011) points out, “the main benefit of the weighted linear regression model is that the resulting adjusted plus-minus statistics for each player should in theory be independent of that player’s teammates and opponents. The traditional plus-minus statistic in hockey is highly dependent on a player’s teammates and opponents, and the use of the regression removes this dependence” (p.1).

The important factor in deciding how to apply a statistic in ice hockey is whether it is accurately predictive of a player’s future performance. In their 2016 study of the viability and applicability of various sports metrics, Franks et. al. note that the classification of signal and noise is particularly different. They argue that repeatability is of no importance when analyzing individual performance, but it is key when analyzing player acquisition. Hence, they conclude that, “chance and team context are still relevant signals when making an attribution decision but are sources of noise for an acquisition decision” (p. 152).

**Predictivity**

There has been quite a bit of research into predictive models, as they offer teams the most tangible evidence of the positive impact of statistics in hockey. Thomas et al (2013) indicate a Deviance Information Criterion, which they calculate utilizing individual and average overall samples can be applied to the likelihood that the original data, or fitted data, in-sample fit, and the withheld data for out-of-sample confirmation. While these can be surface level predictions for many in the public sphere, they still demonstrate the viability of many of the available metrics long term. As Weissbeck and Inkpen (2014) point out, the use of PDO has been popular in the public sphere for several years. Using PDO and puck possession data has developed into a common means of making long term projections for team success. It is assumed that teams with strong possession skills, but have a low PDO value, are “unlucky” and therefore should see an increase in winning percentage. Therefore, the opposite is assumed to be true. The two point out two specific case studies, identifying the 2011/2012 Minnesota Wild and 2013/2014 Toronto Maple Leafs as two teams who had very low possession metrics, extremely high PDO values, and found themselves in first place in the league two months into the season, and finished low in the standings, missing the playoffs.

A difficulty with any statistical analysis in the NHL is the league’s insistence on not utilizing a simplistic win/loss system. Birnbaum (2013) points this out, stating, “Following the usual method for breaking down performance luck, we start by figuring out the expected standard deviation due to randomness alone.  That's a little harder than usual, because the NHL doesn't just have wins and losses; it also has pity points for overtime or shootout losses” (p. N/A).

**Summary**

Hockey has traditionally been slow to adapt to changes in the sporting and societal landscapes. While they have begun to delve into the use of analytics relatively recently, there are still significant unknowns within the field as a result of the lack of resources currently afforded to it. One way to further integrate the use of advanced metrics into sport is through the use of machine learning and predictive modeling. Game-by-game information can be useful, but for the numbers to impact year-to-year projections and salary distributions, they need to be sufficiently predictive of future outcomes. Due to the complex nature of the sport of hockey (and sports in general), multivariate equations appear to offer the most accurate portrayal of the numerous aspects of the game that may or may not influence a player’s performance. These are inherently large equations to implement, and the use of computer programming to analyze larger sets of data, and to run near-infinite simulations of different models significantly widens the potential impact of analytics on the sport. Some important discoveries in the field to this point included, but are not limited to, the introduction of puck possession metrics, the focus on shot generation as a measurement of offensive impact rather than traditional counting stats, and the introduction of relative statistics as a means of contextualizing a player’s production within that of the team. The purpose of the following study is to determine the most efficient means of generating consistent offensive opportunities in the sport of ice hockey, and how teams can then utilize this information to gain a competitive edge against their opposition.

**Chapter 3**

**PROCEDURES**

**Statement of the Problem**

Hockey, as a sport, has always been slow to modernize. For example, they didn’t mandate the wearing of protective helmets for players until the early 1980s; players weren’t required to wear a protective face shield until the early 2010s; they still insist that there is no evidence of the sport leading to Chronic Traumatic Encephalopathy, or CTE, despite the litany of scientific proof that has been researched and presented. Naturally, the sport has been slow to adopt a modern mathematical approach. Publicly accessible data did not exist until the late 2000s. from databases such as NaturalStatTrick and Corsica, the public finally began to scratch the surface of the potential benefits statistical analysis could have for the sport. While this has undoubtedly led to important developments and alterations to how the sport is played, such as a larger emphasis placed on shot generation and location and zone entries and exits, there is much more that has still yet to be explored. Through the use of machine learning and predictive modelling techniques, it should be possible to create an equation that defines offensive efficiency within the context of the offensive zone in ice hockey, as there is no current research within this specific field.

**Source of the Data**

The data has been collected over the course of the 2018-19 season for the University of New Hampshire Women’s Ice Hockey Team. At the request of the coaching staff, all analysis has been conducted after anonymizing player-specific data.

**Instrumentation**

This is a descriptive study and is therefore quantitative in nature.

**Procedures**

The collection has been conducted over the course of a 6-month timeframe, both during games and utilizing game film. All shot-specific data was collected utilizing the video-analysis software *XOS Thunder*. Shot attempts were entered manually into the system, and each period of each game provided a comma-separated values file (.csv) of the data. These files were then compiled into a master sheet, allowing for ease of analysis. All shot attempts were categorized in the following:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Strength | | | | Number of Shot Assists | | | | Shot Quality | Shot Type | Shot Danger | Shot Outcome |
| 5v5 | 5v4 | | 5v3 | 0 | 1 | | 2 | Clean | Backhand | High | Goal |
| 4v4 | 4v5 | | 4v6 | 3 | 4 | | 5 | Rebound | One-Timer | Mid | Save |
| 3v4 | 3v5 | | 3v6 | 6 | 7 | | 8 | Rush | Slap | Low | Blocked |
| 6v5 | 6v4 | | 6v3 | 9 | 10 | | 11 | Redirection | Snap |  | Miss |
| 4v3 | | 3v3 | | 13 | | 14 | | Transition | Wrap |  | Post |
|  | | | |  | | | |  | Wrist |  | Deflection |

Once collected, the data was read into a program written in Python, using a Jupyter Notebook, which allowed the data to be filtered into multiple groups. Using the shot outcome as the common variable, I separated the data into 7 subgroups: grouped by Shot Quality, then Shot Outcome (1); grouped by Shot Type, then Shot Outcome (2); grouped by the Number of Shot Assists, then Shot Outcome (3); grouped by Shot Danger, then Shot Outcome (4); grouped by Shot Danger, then Shot Quality, then Shot Outcome (5); grouped by Shot Danger, then Shot Type, then Shot Outcome (6); and grouped by Shot Quality, then Shot Type, then Shot Outcome (7). Once this process was completed, each subgroup was then plotted into bar graphs where the number of shot attempts was the x-axis, and the categories being compared were the y-axes. Analysis on this information was then conducted. An additional column was then added to each subgroup’s DataFrame to calculate the percentage of the total subgroup each outcome was. This information was then output to a secondary Excel file for storage.

Once this process was complete, a new Jupyter Notebook was created in order to build a logistic model. The same process of reading in, sorting, and grouping the data was repeated, although the Shot Outcomes were pared down into a binary variable, either a Goal, or No Goal. This provided a response variable for the model to function properly. Once this process was completed, the data was once again output into a separate Excel file for storage. This completed the initialization of the data.

In order to build the model, dummy variables were created from the initial DataFrame. Dummy variables only have two values, either 0 or 1, which allow them to function properly within a logistical model. The data was split into randomized groupings for testing and training the model. Then, the training data was balanced using SMOTE, or Synthetic Minority Oversampling Technique). This creates an even, more reliable dataset for the model to be trained accurately. By only balancing the training data, the integrity of the test data was maintained. Recursive Feature Elimination was then used to determine the viability of each feature of the model (the previously defined variables). This lists how trustworthy each variable is. Upon doing this, any shot attempt which came after 5 or more shot assists was eliminated from the model, as each of those instances had too few occurrences to be reliable in a stable predictive model. Finally, a logistical model was then built using the Shot Outcome as the response variable. The function produced a value of 0.499892 and iterated 35 times. It was then retrained as a logistic regression, and the accuracy of the logistic regression classifier on the test set of data was calculated. A confusion matrix was calculated, as well as the precision, recall, F-measure, and support values for the model. Finally, a Receiving Operator Characteristic (ROC) Curve was produced for the model. I received help in the mathematical functionality of the model from Professor Michael Diehl of Endicott College, assistance with the code writing and error checking from Matthew Barlowe of the NHL’s statistical analysis department, and assistance from Ryan Stimson of RIT and *The Athletic* with the hockey applications.

**Chapter 4**

**RESULTS**

Hockey, as a sport, has always been slow to modernize. For example, they didn’t mandate the wearing of protective helmets for players until the early 1980s; players weren’t required to wear a protective face shield until the early 2010s; they still insist that there is no evidence of the sport leading to Chronic Traumatic Encephalopathy, or CTE, despite the litany of scientific proof that has been researched and presented. Naturally, the sport has been slow to adopt a modern mathematical approach. Publicly accessible data did not exist until the late 2000s. from databases such as NaturalStatTrick and Corsica, the public finally began to scratch the surface of the potential benefits statistical analysis could have for the sport. While this has undoubtedly led to important developments and alterations to how the sport is played, such as a larger emphasis placed on shot generation and location and zone entries and exits, there is much more that has still yet to be explored. Through the use of machine learning and predictive modelling techniques, it should be possible to create an equation that defines offensive efficiency within the context of the offensive zone in ice hockey, as there is no current research within this specific field. Through the processes outlined in the previous chapter, the following information was gleaned from this study in an attempt to find a viable solution to the hypothesis proposed.

Both the data, and the Python code written to group the data, build these graphs, and construct the logistical model can all be found in the appendices of this paper.

**Shot Outcomes by Shot Danger**

A screenshot of a cell phone

Description automatically generated The data was first grouped by the danger of the shot attempt taken, then by the outcome of that attempt, producing *Figure 1*.

Figure 1

When grouping the data by shot danger, using shot outcome as the response variable, some very clear trends appear. In this sample, high danger shot attempts result in a goal 9.88% of the time, mid-danger shot attempts result in a goal 2.22% of the time, and low danger shot attempts result in a goal 1.58% of the time. The most likely outcome when taking a low danger shot attempt is for the shot to be blocked, occurring 39.15% of the time. In contrast, the vast majority of both mid and high danger shot attempts result in a save, occurring at 52.76% and 65.88% rates respectively. Of the 3996 total shot attempts in this dataset, 1650, or 41.29% of them were low danger attempts, 1304, or 32.63% of them were mid danger attempts, and the remaining 1042, or 26.08% of them were high danger attempts.

**Shot Outcomes by Shot Type**

A screenshot of a cell phone

Description automatically generatedThe data was then grouped by the type of shot attempt taken, then by the outcome of that attempt, producing the *Figure 2.*

Figure 2

When grouping the data by shot type, and using shot outcome as the response variable, goals were found to be most likely to be scored off of a one-timer, occurring at a rate of 8.84%. Backhand shots resulted in the second highest conversion rate, at 6.25%. The least likely to result in a goal that actually had a goal scored in the dataset was a slap shot, converting at a 2.58% rate. Wrap-arounds did not result in a single goal in this dataset. The most frequent shot type taken in this dataset was a wrist shot, with 1796 total occurrences, or 44.94% of the entire dataset.

**Shot Outcomes by the Number of Shot Assists Preceding an Attempt**

A screenshot of a cell phone

Description automatically generatedThe data was then grouped by the number of shot assists preceding each shot attempt, then by the outcome of that attempt, producing *Figure 3.*

Figure 3

Grouping by the number of shot assists prior to a shot attempt and using the Shot Outcome as the response variable should give an approximation of the effectiveness and importance of pre-shot movement in this dataset. There were significantly more shot attempts that had 2 or fewer shot assists prior to the attempt, and a point of diminishing returns was reached after approximately 4 shot assists. In the usable subsection of this grouping (any attempt on which more than 4 shot assists were recorded was not used in the logistical model), a shot attempt resulted in a goal most often following 2 shot assists, being converted at a rate of 4.51%. In comparison, 1 shot assist resulted in a goal 3.80% of the time in this dataset, 3 shot assists resulted in a goal 3.54% of the time in this dataset, and an attempt without any shot assists resulted in a goal 3.79% of the time in this dataset. More than half of the total shot attempts recorded either had 0 or 1 total shot assists, with more than 1000 more attempts featuring 0 shot assists than 1 shot assist.

**Shot Outcomes by Shot Quality**

A screenshot of a cell phone

Description automatically generated The data was then grouped by shot quality, then shot outcome, producing *Figure 4.*

Figure 4

When grouping the data by the quality of the shot, and using the outcome as the response variable, redirections resulted in goals 16.88% of the time in this dataset. Clean shots resulted in a goal only 1.86% of the time, while shots off the rush resulted in a goal 3.32% of the time. Transition shots resulted in goals 5.34% of the time, and rebounds resulted in goals 10.85% of the time. Clean shots were the most frequently attempted in this dataset, with 1669, or 41.77% total attempts, while the least frequently occurring attempt were redirections, with only 77 total attempts, or 1.93% of the entire dataset.

**Shot Outcomes by Shot Danger and Shot Quality**

A screenshot of a cell phone

Description automatically generatedThe data was then grouped by shot danger, then again by shot quality, then by shot outcome, producing *Figure 5.*

Figure 5

When grouping by shot danger, then shot quality, and using shot outcome as the response variable, it was found that goals were most likely to be scored on low danger redirections, at a rate of 50.00%. However, only two shot attempts were recorded as low danger redirections in the entire dataset. The next most likely attempt to result in a goal is a high danger redirection, at a rate of 17.86%, followed by high danger rebounds, at a rate of 15.00%, and mid danger redirections, at a rate of 10.53%. The attempts found to be the least likely to result in a goal were low danger rush shots, at a rate of 0.99%, mid danger clean shots, at a rate of 1.62%, low danger clean shots, at a rate of 1.67%, and low danger transition shots, at a rate of 1.78%. The most frequently occurring style of shot attempt in this subgroup of the data was a low danger clean shot, with more than 1000 individual occurrences.

**Shot Outcomes by Shot Danger and Shot Type**

The data was then grouped by shot danger, then shot type, and shot outcome was used as the response variable, producing *Figure 6.*

A screenshot of a cell phone

Description automatically generatedThree attempt types resulted in no recorded goals in the dataset: high danger slap shots, high danger wrap arounds, and low danger backhand shots. The most likely type of attempt to result in a goal was a high danger one-timer, converting at a rate of 13.85%. The next most likely was a high danger wrist shot, converting at a rate of 11.41%, followed by high danger snap shots, at a rate of 10.67%, and mid danger one-timers, at a rate of 7.59%. The attempts which had goals recorded found to result in a goal the least were low danger snap shots, at a rate of 0.45%, mid danger wrist shots, at a rate of 1.49%, mid danger snap shots, at a rate of 1.69%, and low danger slap shots, at a rate of 1.92%. The most frequent type of shot attempt in this subgroup of data was a low danger wrist shot.

Figure 6

**Shot Outcomes by Shot Quality and Shot Type**

A screenshot of a cell phone

Description automatically generatedIn this subgroup of data, the dataset was grouped by the quality of the shot, then the type of shot, and the outcome was used as the response variable, producing *Figure 7.*

Figure 7

Of this subgroup, 7 of the 29 existing parameter combinations did not result in a goal being scored. These were transition wrap-arounds, rush wrap-arounds, redirection slap-shots, rebound one-timers, rebound wrap-arounds, clean wrap-arounds, and clean one-timers. In total, this accounted for 68 total attempts, or 1.70% of the total dataset. Redirections, as previously mentioned, also accounted for a relatively small fraction of the total dataset. Therefore, while wrist shot redirections (35.71% conversion rate), snap shot redirections (15.38% conversion rate), and one-timer redirections (13.51% conversion rate) were the three parameter combinations most likely to result in a goal, the dataset does not provide a large enough sample size to declare these groupings to be indicative of a repeatable statistic. Likewise, backhand redirections (8.33% conversion rate) only occurred a total of 12 times throughout the dataset. When eliminating these parameter groupings, along with the 7 which did not result in a single goal, there are then 18 possible combinations of parameters to consider. Of that new grouping, goals were found to be scored on a rebound snap shot most frequently, at 12.86%. The next highest conversion rates were for rebound wrist shots, at 12.40%, and rebound backhand shots, at 12.31%. The least likely combination to result in a goal of this subgroup was found to be clean snap shots, converted at a rate of 0.63%.

**Chapter 5**

**DISCUSSION**

The key to the proper analysis and application of this data is to ensure that it is both trustworthy and repeatable. In short, the information gleaned from this research should allow one to reasonably predict future outcomes using the data provided. To ensure that the data was accurate, a logistical model was built from a binary version of the dataset, which simplified the possible shot outcomes to either goal or not a goal. The data was then split into a training group and a testing group. The training group was balanced using Synthetic Minority Oversampling Technique, so as to make the overall dataset more reliable when constructing a model. Recursive Feature Elimination tested each individual feature to determine whether or not it was viable. This process eliminated any shot attempt for which the number of shot assists recorded was greater than 4. The model was then run, producing a logistic regression classifier accuracy of 0.87. This indicates that the model, and therefore the data, is accurate enough to apply with a broader approach, while keeping in mind the previously noted inherent limitations of the dataset in question.

The information concerning shot danger is relatively straightforward. It is clear that the probability of scoring a goal increases the closer to the net the shot is taken. This lines up with common sense and previous research (Tulsky et. al, 2013), although the difference in conversion rate between high danger attempts and mid danger attempts being so much larger than the difference in conversion rate between mid-danger attempts and low danger attempts was an interesting development. This squares with the vaunted eye test, as that makes sense when you think about hockey in general terms. However, much of the data actually demonstrated some ideologies that are relatively antithetical to common hockey assumptions. It is apparent that pre-shot movement and obfuscation are the preeminent means of generating consistent offense. In any subgroup that considered shot quality, redirections and rebounds were consistently more successful at converting shot attempts into goals than clean looks and rush attempts. In turn, rush attempts being marginally more successful than clean looks was relatively unsurprising. The subgroup which considered the number of shot assists preceding a shot attempt also found at least two shot assists preceding an attempt increases the probability of scoring a goal by more than 18%, while the difference between one shot assist and zero shot assists is only a paltry 0.13% increase. However, the diminishing returns shown by the steadily increasing number of shot assists indicates that there needs to be a purpose to pre-shot movement. This is an excellent topic of study for future research.

As previously noted, there is a high probability that the data in this study is skewed based on the on-ice talent and game strategy of the University of New Hampshire’s Women’s Ice Hockey team, as north of 50% of the entire dataset is comprised solely of shot attempts made by that team. This should not necessarily preclude any knowledge from being applicable to the widespread women’s ice hockey community, but it prevents any broad applications to be generalized. Despite this, there is a dearth of research in the women’s game, particularly in NCAA hockey, simply due to the lack of available resources for the league. Therefore, while much of the research is reflective of similar to existing research, such as Buttrey, Washburn, & Price (2011), and Weissbock & Inkpen (2014-A; 2014-B), it is nevertheless encouraging to see the level of similarity that exists given the fundamental differences between the men’s and women’s games.

As redirections and rebounds presented significantly higher conversion rates than their other counterparts, it can be postulated that, despite actually converting for goals at a demonstrably low rate, low danger shots are important for generating offense. It is the way that they are used, however, that is the key. A shot to score from the point is very different than a shot for a rebound, or a shot for a deflection. Therefore, it is unwise to rely on consistently scoring from the top of the offensive zone. Additionally, the conversion rates of transition shots were strangely low relative to their rebound and redirection counterparts. This runs counterintuitive to the idea that forcing a goaltender to move laterally in the net is the best way to draw them out of position. Perhaps a better strategy would be to utilize smaller area passes coupled with quick releases. This preserves the shooter’s ability to prevent the goaltender from getting set in position prior to shooting the puck, but also increases the probability of a shot assist being successful by reducing the amount of space the puck has to travel.

Overall, this dataset is indicative of a team that lacks an abundance of elite-level shooters. As approximately half of the shot attempts in the dataset come from the same team, that has the potential to significantly skew the results. Nonetheless, it would be recommended that teams rely more heavily on pre-shot movement and confusing the opponent to generate offense rather than attempting to beat a defender one-on-one or beat the goaltender with a superior shot. Based on the information provided, it appears that hand-eye coordination (i.e. redirection ability) is a heavily underrated and underutilized talent. Likewise, teams who shoot with a purpose more often (shot-passes, intentionally shooting wide of the net to create a ricochet off the end boards, shooting intentionally off of a goaltender’s pads to generate a loose puck in a high danger area) necessitates players with high Hockey-IQ, or an elevated ability to process the game. Generating offense in this manner also does not require that a team have an elite level shooter on their roster to be successful. While one should never turn down the opportunity to add an elite shooter to their roster, it is not a necessity for them to be successful in the long run. Therefore, it can be argued that a player’s stick skills, skating, and processing ability are more valuable to a team than the player’s ability to shoot the puck. In a sport that loves to embrace so-called “snipers”, teams and coaching staffs can often become too reliant on the inherently inconsistent scoring ability of a few players. Balancing the remainder of their roster with players who think the game better than their skillset, can skate efficiently and agilely, and have excellent hand-eye coordination is the most efficient pathway to long-term success and sustainability.

**Chapter 6**

**SUMMARY AND CONCLUSIONS**

The introduction of analytics into sport has come rapidly within the last decade (Fried & Mumcu, 2017; Fry & Ohlmann, 2012). Through the introduction and implementation of increasingly advanced and complex machine learning techniques, it has become plausible to predict player production and value over the span of their career, rather than basing personnel decisions solely on the eye test (Passfield & Hopker, 2017; Riley, 2017). While relatively rudimentary research has been completed and published for the sport of hockey into offense generation (Tulsky et. al, 2013) and individual player impact (Weissbock & Inkpen, 2014-A; Weissbock & Inkpen, 2014-B), there is still much that is unknown about the subtle intricacies of the sport of ice hockey. As such, there are likely many market inefficiencies that can and should be exploited by the teams and organizations with fewer resources than the juggernauts of professional sport.

The goal of this paper was to determine whether or not there was a feasible means of increasing the efficiency of offensive generation, how the style or system of play for a team can be adapted to fit these theories, and how teams can better exploit any newly discovered market inefficiencies. In order to do so, shot attempt data was collected throughout the 2018-19 season for the University of New Hampshire’s Women’s Ice Hockey Team. This data was then read into a program written in Python to filter and group it into smaller data subsets. These subsets were then built into a logistical model, in order to determine the validity of the data, and how accurately it could predict future outcomes. This provided a categorical relevance of 0.87, making the data relatively accurate within the context of its limitations and delimitations.

Ultimately, there were no groundbreaking new theories or ideas gleaned from the data. This is in large part due to the limitations and delimitations previously noted about the dataset. The overabundance of one team in the dataset creates an imbalanced view of the information and prevents a more widespread application. Therefore, any conclusions drawn must be done so tentatively and contextually. With that in mind, it was clear from the data that there is a point of diminishing returns with every incremental increase of the number of shot assists preceding a shot attempt, tapering off somewhere between one and two passes. Additionally, shot attempts coming off of rebounds or redirections appeared to provide a significantly higher conversion rate as compared to a clean shot attempt, opposite to relative conventional wisdom.

The analysis of the impact of shot danger on offensive generation mirrored previous research in the area (Tulsky et. al, 2013), despite that research being solely into NHL play. There are significant differences in the men’s and women’s game, primarily the use of body contact, which could conceivably cause the NHL data to not fully be able to be applied to the women’s game. This research suggests that there is a larger parallel between the two than previously thought. Unfortunately, as mentioned, much of the information found through this course of study necessitate additional context and further investigation to determine concrete conclusions. It would be prudent to look into how zone entry and exit data correlates with each shot attempt, as this may provide better context for rush attempts. Isolating even strength data would also be a prudent decision to make, as would isolating the difference between shot attempts for and shot attempts against. The most important aspect that could be further analyzed would necessitate the construction of an expected goals model to apply to the data. The difficulty with this is it would require accurate player- and puck-tracking software and information, so as to convert the ice surface into a coordinate plane, and place each shot attempt within said plane. At this moment in time, this is not a reasonable expectation for collegiate women’s hockey, as the NHL has yet to fully integrate this style of software into their own league, and the funding for the NHL far outweighs that of a Division I Women’s Ice Hockey program.

Summarily, while the data provided a considerable amount of new starting points for additional research, it did not provide an answer to the proposed research question, and it is not plausible to identify a new market inefficiency in player analysis using solely this research. However, slight alterations to shot selection strategies and offensive zone puck movement could provide a tangible positive difference in offensive generation. This would theoretically be achieved by minimizing the risk of offensive zone passing, while being significantly more purposeful with shot attempts coming from low danger areas; that is, not necessarily shooting to score from further out, but consistently shooting to generate a higher danger rebound opportunity, or shooting directly at a teammate’s stick to create a redirection. This could be accomplished with existing rosters and player talents but would need a committed coaching staff to implement these different offensive zone strategies.

**BIBLIOGRAPHY**

Akbilgic, O. (2015). Classification trees aided mixed regression model. *Journal of Applied Statistics,42*(8), 1173-1781. doi:10.1080/02664763.2015.1006394

Algamal, Z. Y., & Lee, M. H. (2015). Penalized Poisson Regression Model using adaptive modified Elastic Net Penalty. *Electronic Journal of Applied Statistical Analysis,8*(2), 236-245. doi:10.1285/i20705948v8n2p236

Anderson-Bergman, C. (2017). Bayesian Regression Models for Interval-censored Data in R. *The R Journal,9*(2), 487-498. Retrieved October 24, 2018.

Birnbaum, P. (2013, January 21). Luck vs Talent in the NHL Standings. Retrieved from http://blog.philbirnbaum.com/2013/01/luck-vs-talent-in-nhl-standings.html

Brock, T. R., PhD, CPT, CRP, ID (S&L ). (3028). Performance Analytics: The Missing Big Data Link Between Learning Analytics and Business Analytics. *Performance Improvement,56*(7), 6-16. doi:10.1002/pﬁ.21701

Buttrey, S. E., Washburn, A. R., & Price, W. L. (2011). Estimating NHL Scoring Rates. *Journal of Quantitative Analysis in Sports,7*(3). doi:10.2202/1559-0410.1334

Chiu, W., Rodriguez, F. M., & Won, D. (2016). *Psychological Reports,119*(2), 2016th ser., 435-449. doi:10.1177/0033294116662880

Desjardins, G. (2009). *League Equivalencies*(pp. 1-4). HockeyAnalytics.com.

Franks, A. M., D’Amour, A., Cervone, D., & Bornn, L. (2016). Meta-analytics: Tools for understanding the statistical properties of sports metrics. *Journal of Quantitative Analysis in Sports,12*(4), 151-165. doi:10.1515/jqas-2016-0098

Fried, G., & Mumcu, C. (Eds.). (2017). Sport Analytics: A Data-Driven Approach to Sport Business and Management. *International Journal of Sport Communication,*(10), 2017th ser., 421-422. doi:10.1123/jsc.2017-0069

Fry, M. J., & Ohlmann, J. W. (2012). Introduction to the Special Issue on Analytics in Sports, Part I: General Sports Applications. *Interfaces,42*(2), 105-108. doi:10.1287/inte.1120.0633

Harring, J. R., McNeish, D. M., & Hancock, G. R. (2017). Using Phantom Variables in Structural Equation Modeling to Assess Model Sensitivity to External Misspecification. *Psychological Methods,22*(4), 2017th ser., 616-631. Retrieved from 10.1037/met0000103.

Jawed, H., Ziad, Z., Khan, M., & Asrar, M. (2018). Anomaly detection through keystroke and tap dynamics implemented via machine learning algorithms. *Turkish Journal of Electrical Engineering & Computer Sciences,*(26), 2018th ser., 1698-1709. doi:10.3906/elk-1711-410

Kaplan, E. H., Mongeon, K., & Ryan, J. T. (2014). A Markov Model for Hockey: Manpower Differential and Win Probability Added. *INFOR: Information Systems and Operational Research,52*(2), 39-50. doi:10.3138/infor.52.2.39

Krzywicki, K. (2010, June). *NHL Shot Quality 2009-10: A look at shot angles and rebounds*[Scholarly project].

Lamas, L., Drezner, R., Otranto, G., & Barrera, J. (2018). Analytic method for evaluating players’ decisions in team sports: Applications to the soccer goalkeeper. *Plos One,13*(2), 1-18. doi:10.1371/journal.pone.0191431

Ma, Z., Wang, P., Gao, Z., Wang, R., & Khalighi, K. (2018). Ensemble of machine learning algorithms using the stacked generalization approach to estimate the warfarin dose. *Plos One,13*(10), 1-12. doi:10.1371/journal.pone.0205872

Macdonald, B. (2011). A Regression-Based Adjusted Plus-Minus Statistic for NHL Players. *Journal of Quantitative Analysis in Sports,7*(3). doi:10.2202/1559-0410.1284

Mondello, M., & Kamke, C. (n.d.). The Introduction and Application of Sports Analytics in Professional Sport Organizations. *Journal of Applied Sport Management,6*(2), 2014th ser., 1-12. Retrieved from EBSCO.

Nastase, P., & Stoica, D. (2010). A New Business Dimension - Business Analytics. *Accounting and Management Information Systems,9*(4), 603-618.

Nikota, J. (2013, September 16). Leafs' Attack Time at The Halfway Mark. Retrieved from https://www.pensionplanpuppets.com/2013/9/16/4727746/leafs-attack-time-at-the-halfway-mark

Passfield, L., & Hopker, J. G. (2017). A Mine of Information: Can Sports Analytics Provide Wisdom From Your Data? *International Journal of Sports Physiology and Performance,12*(7), 851-855. doi:10.1123/ijspp.2016-0644

Riley, S. N. (2017). Investigating the multivariate nature of NHL player performance with structural equation modeling. *Plos One,12*(9). doi:10.1371/journal.pone.0184346

Shimizu, Y. (2016). Moment convergence of regularized least-squares estimator for linear regression model. *The Institute of Statistical Mathematics, Tokyo,2017*(69), 1141-1154. doi:10.1007/s10463-016-0577-6

STUDYING LUCK & OTHER FACTORS IN PDO. (2013, January 10). Retrieved from http://nhlnumbers.com/ 2013/1/10/studying-luck-other-factors-in-pdo

Thomas, A. C., Ventura, S. L., Jensen, S. T., & Ma, S. (2013). Competing process hazard function models for player ratings in ice hockey. *The Annals of Applied Statistics,7*(3), 1497-1524. doi:10.1214/13-aoas646

Tulsky, E., Detweiler, G., Spencer, R., & Sznajder, C. (2013). Using Zone Entry Data To Separate Offensive, Neutral, And Defensive Zone Performance. *MIT Sloan Sports Analytics Conference,7*.

Vululleh, P. (2018). Determinants of students’ e-learning acceptance in developing countries: An approach based on Structural Equation Modeling (SEM). *International Journal of Education and Development Using Information and Communication Technology,14*(1), 141-151. Retrieved from EBSCO.

Weissbock, J., & Inkpen, D. (2014-A). Combining Textual Pre-game Reports and Statistical Data for Predicting Success in the National Hockey League. *Advances in Artificial Intelligence Lecture Notes in Computer Science,*251-262. doi:10.1007/978-3-319-06483-3\_22

Weissbock, J., & Inkpen, D. (2014-B) *Forecasting Success in the National Hockey League Using In-Game Statistics and Textual Data*(Unpublished doctoral dissertation). University of Ottawa.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques with Java implementations*. Amsterdam: Elsevier.

**APPENDICES**

**Appendix A: Pertinent Terminology**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Corsi | | | | | | | | | 5-on-5 shot attempts | | |
|  | | | | Corsi Relative | | | | | Shot attempt differential relative for a player in comparison to their rest of the team | | |
|  | | | | Corsi For | | | | | Shot attempts taken by a team | | |
|  | | | | Corsi Against | | | | | Shot attempts against a team | | |
|  | | | | Corsi On | | | | | Shot attempts taken by a team while a particular player is on the ice | | |
|  | | | | Corsi Off | | | | | Shot attempts taken by a team while a particular player is not on the ice | | |
| Fenwick | | | | | | | | | Unblocked 5-on-5 shot attempts | | |
|  | | | | Fenwick Relative | | | | | Unblocked shot attempt differential for a player in comparison to the rest of their team | | |
|  | | | | Fenwick For | | | | | Unblocked shot attempts taken by a team | | |
|  | | | | Fenwick Against | | | | | Unblocked shot attempts against a team | | |
|  | | | | Fenwick On | | | | | Unblocked shot attempts taken by a team while a particular player is on the ice | | |
|  | | | | Fenwick Off | | | | | Unblocked shot attempts taken by a team while a particular player is not on the ice | | |
| Shot Attempt | | | | | | | | | Any time a puck is shot with the intention of scoring a goal | | |
| Shot Danger | | | | | | | | | The location from which a shot attempt is taken based upon the likelihood that it turns into a goal scored | | |
|  | | | | High Danger | | | | | A shot attempt taken from an area that approximately encompasses the space between the hashmarks in the offensive zone, from the goal line up to halfway between the top of the hashmarks and the top of the circle | | |
|  | | | | Mid Danger | | | | | A shot attempt taken from an area that encompasses the space below the tops of the circles in the offensive zone, but not including the area defined as a High Danger Shot Attempt | | |
|  | | | | Low Danger | | | | | A shot attempt taken from an area that encompasses the space above the top of the circles in the offensive zone. | | |
| Shot Quality | | | | | | | | | The way in which a shot attempt is made | | |
|  | | | | | Clean Shot | | | | Any shot attempt that comes from a player with an unimpeded sight-path to the net or from a stationary position | | |
|  | | | | | Redirection | | | | Any shot attempt that comes by deflecting a secondary shot, typically as the puck moves from high in the offensive zone to low in the offensive zone | | |
|  | | | | | Rebound | | | | Any shot attempt that comes directly after the opposing goaltender has made a save on a previous shot attempt without either team gaining possession of the puck. | | |
|  | | | | | Transition | | | | Any shot attempt that comes after a pass that either moves across the face of the net, or from low in the offensive zone to high in the offensive zone | | |
|  | | | | | Rush | | | | Any shot attempt that comes up to four seconds after a zone entry that does not qualify as a transition shot | | |
| Shot Outcome | | | | | | | | | The outcome of any shot attempt | | |
|  | | | | | Blocked | | | | Any shot attempt that is impeded by an opposing player | | |
|  | | | | | Deflection | | | | Any shot attempt that makes contact with a shooting player’s teammate prior to reaching the goal line | | |
|  | | | | | Goal | | | | Any shot attempt that results in a legal hockey goal | | |
|  | | | | | Miss | | | | Any unblocked shot attempt that is not a save, goal, or post hit | | |
|  | | | | | Post | | | | Any shot attempt that directly makes contact with the crossbar or either goalpost without either entering the net in a legal fashion immediately after making contact with the post or crossbar or makes contact with the goaltender or an opposing player immediately prior to making contact with the post or crossbar. | | |
|  | | | | | Save | | | | Any unblocked shot attempt that the goaltender successfully prevents from entering the net in a legal fashion | | |
| Shot Type | | | | | | | | | The type of shot taken by a player. This is either a backhand shot, one-timer, slapshot, snapshot, wrist shot, or wrap-around attempt | | |
| Zone Entry | | | | | | | | | The act of the puck crossing the opposing blue line from the neutral zone in the direction of the net of the opposing team. A successful zone entry is defined as one which the team attempting to enter the zone maintains possession of the puck for at least 3 seconds after the puck has crossed the blue line into the offensive zone | | |
|  | | | Stretch Pass Entry | | | | | | Any zone entry which is initiated by a pass that either spans at least half the width of the rink or is longer than the distance between the center ice line and a blueline, and that is received between the center ice line and the opposing blue line | | |
|  | | | Small Area Pass Entry | | | | | | Any zone entry which is initiated by a pass that is any distance shorter than that of a stretch pass entry that is received between the center ice line and the opposing blue line | | |
|  | | | Rush Entry | | | | | | Any zone entry which is initiated by a singular player carrying the puck from the neutral zone into the offensive zone, without any passes received between the center ice line and the opposing blueline. | | |
|  | | | Controlled Entry | | | | | | Any zone entry which the team attempting to enter the zone maintains full possession of the puck across the opposing blueline. This is either a rush, stretch pass, or small area pass entry | | |
|  | | | Dump-In | | | | | | Any zone entry in which the puck is shot into the zone behind the opposing defensemen in an attempt to either initiated a line change, or create a loose puck battle below the bottom of the circles in the offensive zone | | |
|  | | | Chip In | | | | | | Any zone entry in which the puck is directed around an opposing defender at or immediately prior to the opposing blue line with the goal of the player or a teammate winning the race to the loose puck immediately after crossing the opposing blue line | | |
| Zone Exit | | | | | | | | | The act of the puck crossing the defensive blue line from the defensive zone in the direction of the net of the opposing team and into the neutral zone. A successful zone exit is defined as one which the team attempting to exit the zone maintains possession of the puck for at least 3 seconds after the puck has entered the neutral zone | | |
|  | | | | Carry Exit | | | | | Any zone exit which the attempting player skates with the puck from anywhere below the top of the circles in their defensive zone across the defensive blueline | | |
|  | | | | Pass Exit | | | | | Any zone exit which the attempting player passes the puck directly to a teammate who is either immediately behind, at, or in front of the defensive blueline. | | |
|  | | | | Controlled Exit | | | | | Any zone exit which the attempting player or team maintains full control of the puck throughout the act of exiting the zone. This is either a pass exit or a carry exit | | |
|  | | | | Dump-Out | | | | | Any zone exit where the attempting player tries to create a loose-puck battle in the neutral zone by sending the puck to open space. | | |
|  | | | | Pressure Exit | | | | | Any zone exit attempted while the attempting player is being attacked by an opposing forechecker throughout the entire process of the exit | | |
|  | | | | Clear | | | | | Any zone exit (typically while killing a penalty) where the intent of the exit is for the puck to end up behind the extended goal line in the offensive zone | | |
| Plus/Minus (+/-) | | | | | | | Calculated by subtracting the total number of goals allowed by a player's team while the player is on the ice (at even strength or on the power play) from the total number of goals scored by the player's team while the player is on the ice (at even strength or short-handed). | | | | |
| Point Shares | | | | | | | | | An estimate of the total number of standings points contributed by a player to a team’s overall record | | |
|  | Offensive Point Share | | | | | | | | An estimate of the number of standings points contributed by a player due to his offense. | | |
|  | Defensive Point Share | | | | | | | | An estimate of the number of standings points contributed by a player due to his defense. | | |
|  | Goalie Point Share | | | | | | | | An estimate of the number of standings points contributed by a player due to his play in goal | | |
| Goals Created | | | | | | | | Calculated by adding goals scored to 0.5 times assists, then multiplying by team goals divided by team goals plus 0.5 times team assists. | | | |
| Through Percentage (Thru%) | | | | | | | | | The percentage of total shot attempts which result in either a save or goal for the shooting team | | |
| Shooting Percentage (SH%) | | | | | | | | | The percentage of shots on goal taken by a team or player that result in a goal | | |
| True Shooting Percentage (TSH%) | | | | | | | | | The percentage of shot attempts taken by a team or player that result in a goal | | |
| Save Percentage (SV%) | | | | | | | | | The percentage of shots against a team which are saved by a goaltender | | |
| Goals Saved Above Average (GSAA) | | | | | | | | | | The number of goals a goaltender saves on average as compared to an average goaltender in their league | |
| Expected Goals (xG) | | | | | | | | | The number of goals expected to result from a team or player’s shot attempts | | |
| Expected Points (xP) | | | | | | | | | The number of standings points that are to be expected from the way a team has played | | |
| PDO | | | | | | | | | The sum of shooting percentage and save percentage. Also referred to as “the luck stat” | | |
| Goals For Percentage (GF%) | | | | | | | | | The percentage of all goals that are scored by a team | | |
| Types of Classifications | | | | | | | | |  | | |
|  | | | | | | Per 60 (/60) | | | Any stat divided by the total number of Time on Ice a player or team has accumulated, expressed as a ratio of that stat per hour of ice time played | | |
|  | | Quality of Competition (QoC) | | | | | | | | |  |
|  | | Quality of Teammates (QoT) | | | | | | | | |  |
|  | | | | | | Per Game (/GP) | | |  | | |
|  | | | | | | Expected (x-) | | |  | | |

**Appendix B: Python Code**

In [1]:

#Spencer Fascetta

#Thesis Code - Initial Analysis

#April 4, 2019

#Endicott College

#Import all necessary mathematical libraries

import pandas as pd

import numpy as np

import nltk

import json

import math

import seaborn as sea

import math

In [2]:

import matplotlib.pyplot as plt

%matplotlib inline

In [3]:

#Reads in Master Tracking data

excel\_file="UNH Master Tracking - Thesis.xlsx"

In [4]:

#Creates dataframe from Master Tracking Excel file

data=pd.read\_excel(excel\_file, 'Shot Attempts')

In [5]:

#Defines categorical variables as arrays

shot\_qualities=['Clean', 'Rebound', 'Rush', 'Redirection', 'Transition']

shot\_types=['Backhand','One-Timer','Slap','Snap','Wrap', 'Wrist']

shot\_outcomes=['Goal', 'Miss', 'Deflection', 'Blocked', 'Save', 'Post']

shot\_dangers=['High', 'Mid', 'Low']

In [6]:

#Removes Index column from dataframe

data.drop(['Shot Attempt'], axis=1, inplace=True)

In [7]:

#Removes On-Ice skater data from dataframe

data1=data.drop(['On Ice 1', 'On Ice 2', 'On Ice 3', 'On Ice 4',

'On Ice 5', 'On Ice 6'], axis=1)

row=data1.shape[0]

#Groups data into necessary subgroups

danger\_outcome=data1.groupby(['Shot Danger', 'Shot Outcome']).size()

type\_outcome=data1.groupby(['Shot Type', 'Shot Outcome']).size()

quality\_outcome=data1.groupby(['Shot Quality', 'Shot Outcome']).size()

shot\_assists\_outcome=data1.groupby(['Number of Shot Assists', 'Shot Outcome']).size()

danger\_quality\_outcome=data1.groupby(['Shot Danger', 'Shot Quality', 'Shot Outcome']).size()

danger\_type\_outcome=data1.groupby(['Shot Danger', 'Shot Type', 'Shot Outcome']).size()

quality\_type\_outcome=data1.groupby(['Shot Quality', 'Shot Type',

'Shot Outcome']).size()

In [8]:

#Creates new dataframes for each individual subgroup previously defined

danger\_outcome\_df=pd.DataFrame(danger\_outcome)

type\_outcome\_df=pd.DataFrame(type\_outcome)

quality\_outcome\_df=pd.DataFrame(quality\_outcome)

shot\_assists\_outcome\_df=pd.DataFrame(shot\_assists\_outcome)

danger\_quality\_outcome\_df=pd.DataFrame(danger\_quality\_outcome)

danger\_type\_outcome\_df=pd.DataFrame(danger\_type\_outcome)

quality\_type\_outcome\_df=pd.DataFrame(quality\_type\_outcome)

In [ ]:

#Plots distribution of Shot Danger and Shot Outcome dataframe

plot1=danger\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250), fontsize=450)

plot1.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot1.get\_figure()

fig.suptitle('Shot Outcomes by Shot Danger',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Shot Dangers', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot1.png")

In [ ]:

#Plots distribution of Shot Type and Shot Outcome dataframe

plot2=type\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250), fontsize=450)

plot2.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot2.get\_figure()

fig.suptitle('Shot Outcomes by Shot Type',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Shot Types', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot2.png")

In [ ]:

#Plots distribution of Number of Shot Assists and Shot Outcome dataframe

plot3=shot\_assists\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250),

fontsize=450)

plot3.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot3.get\_figure()

fig.suptitle('Shot Outcomes by Number of Shot Assists',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Number of Shot Assists', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot3.png")

In [ ]:

#Plots distribution of Shot Quality and Shot Outcome dataframe

plot4=quality\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250), fontsize=450)

plot4.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot4.get\_figure()

fig.suptitle('Shot Outcomes by Shot Quality',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Shot Qualities', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot4.png")

In [9]:

#Plots distribution of Shot Danger, Shot Quality, and Shot Outcome dataframe

plot5=danger\_quality\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250),

fontsize=450)

plot5.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot5.get\_figure()

fig.suptitle('Shot Outcomes by Shot Danger and Shot Quality',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Shot Dangers, Shot Qualities', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot5.png")

In [ ]:

#Plots distribution of Shot Danger, Shot Type, and Shot Outcome dataframe

plot6=danger\_type\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250),

fontsize=450)

plot6.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot6.get\_figure()

fig.suptitle('Shot Outcomes by Shot Danger and Shot Type',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Shot Dangers, Shot Types', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot6.png")

In [ ]:

#Plots distribution of Shot Quality, Shot Type, and Shot Outcome dataframe

plot7=quality\_type\_outcome\_df.unstack().plot(kind='barh', figsize=(500,250),

title='Shot Outcomes by Shot Quality and Shot Type',

fontsize=450)

plot7.legend(["Blocked", "Deflection", "Goal", "Miss", "Post", "Save"], fontsize=450)

fig=plot7.get\_figure()

fig.suptitle('Shot Outcomes by Shot Quality and Shot Type',fontsize=500)

plt.xlabel('Shot Attempts', fontsize=450)

plt.ylabel('Shot Qualities, Shot Types', fontsize=450)

#Saves plot to .png picture file

fig.savefig(r"C:\Users\Spencer\Desktop\School\Internship\Game Clipping Files\Master\plot7.png")

In [ ]:

#Pulls the individual totals for each master variable - these are stored in lists

quality=data['Shot Quality'].value\_counts()

types=data['Shot Type'].value\_counts()

danger=data['Shot Danger'].value\_counts()

assists=data['Number of Shot Assists'].value\_counts()

outcome=data['Shot Outcome'].value\_counts()

#Raw total for each danger level

high\_total=danger['High']

mid\_total=danger['Mid']

low\_total=danger['Low']

#Recasts dataframe as type String

danger\_outcome\_df.columns=danger\_outcome\_df.columns.astype(str)

#Adds new column using the raw total of each corresponding subvariable

danger\_outcome\_df['Raw Totals']=[high\_total, high\_total, high\_total, high\_total,

high\_total, low\_total, low\_total, low\_total,

low\_total, low\_total, low\_total, mid\_total,

mid\_total, mid\_total, mid\_total, mid\_total, mid\_total]

danger\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

#Creates new column that calculates the percentage of the subtotal for each subvariable

danger\_outcome\_df['Percentages']=danger\_outcome\_df['Totals']/danger\_outcome\_df['Raw

Totals']

#Drops previously created column of raw totals for ease of analysis

danger\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

#THIS PROCESS IS REPEATED FOR EACH SUBGROUP OF DATA

In [ ]:

backhand\_total=types['Backhand']

oneT\_total=types['One-Timer']

slap\_total=types['Slap']

snap\_total=types['Snap']

wrap\_total=types['Wrap']

wrist\_total=types['Wrist']

type\_outcome\_df

type\_outcome\_df.columns=type\_outcome\_df.columns.astype(str)

type\_outcome\_df['Raw Totals']=[backhand\_total, backhand\_total, backhand\_total,

backhand\_total, backhand\_total, oneT\_total, oneT\_total,

oneT\_total, oneT\_total, oneT\_total, oneT\_total,

slap\_total, slap\_total, slap\_total, slap\_total,

slap\_total, slap\_total, snap\_total, snap\_total,

snap\_total, snap\_total, snap\_total, snap\_total,

wrap\_total, wrap\_total, wrap\_total, wrist\_total,

wrist\_total, wrist\_total, wrist\_total, wrist\_total,

wrist\_total]

type\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

type\_outcome\_df['Percentages']=type\_outcome\_df['Totals']/type\_outcome\_df['Raw Totals']

#type\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [ ]:

clean\_total=quality['Clean']

redirection\_total=quality['Redirection']

rebound\_total=quality['Rebound']

transition\_total=quality['Transition']

rush\_total=quality['Rush']

quality\_outcome\_df

quality\_outcome\_df.columns=quality\_outcome\_df.columns.astype(str)

quality\_outcome\_df['Raw Totals']=[clean\_total, clean\_total, clean\_total, clean\_total,

clean\_total, clean\_total, rebound\_total,

rebound\_total, rebound\_total, rebound\_total,

rebound\_total, rebound\_total, redirection\_total,

redirection\_total, redirection\_total,

redirection\_total, rush\_total, rush\_total,

rush\_total, rush\_total, rush\_total, rush\_total,

transition\_total, transition\_total,

transition\_total, transition\_total,

transition\_total, transition\_total]

quality\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

quality\_outcome\_df['Percentages']=quality\_outcome\_df['Totals']/quality\_outcome\_df['Raw

Totals']

#quality\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [ ]:

zero\_total=assists[0]

one\_total=assists[1]

two\_total=assists[2]

three\_total=assists[3]

four\_total=assists[4]

five\_total=assists[5]

six\_total=assists[6]

seven\_total=assists[7]

eight\_total=assists[8]

nine\_total=assists[9]

ten\_total=assists[10]

eleven\_total=assists[11]

thirteen\_total=assists[13]

fourteen\_total=assists[14]

shot\_assists\_outcome\_df

shot\_assists\_outcome\_df.columns=shot\_assists\_outcome\_df.columns.astype(str)

shot\_assists\_outcome\_df['Raw Totals']=[zero\_total, zero\_total, zero\_total, zero\_total,

zero\_total, zero\_total, one\_total, one\_total,

one\_total, one\_total, one\_total, one\_total,

two\_total, two\_total, two\_total, two\_total,

two\_total, two\_total, three\_total, three\_total,

three\_total, three\_total, three\_total,

three\_total, four\_total, four\_total,

four\_total, four\_total, four\_total, five\_total,

five\_total, five\_total, five\_total, five\_total,

six\_total, six\_total, six\_total, six\_total,

seven\_total, seven\_total, seven\_total,

seven\_total, eight\_total, eight\_total,

eight\_total, eight\_total, nine\_total,

nine\_total, ten\_total, ten\_total, eleven\_total,

thirteen\_total, fourteen\_total]

shot\_assists\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

shot\_assists\_outcome\_df['Percentages']=shot\_assists\_outcome\_df['Totals']/

shot\_assists\_outcome\_df['Raw Totals']

#shot\_assists\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [ ]:

save\_total=outcome['Save']

blocked\_total=outcome['Blocked']

miss\_total=outcome['Miss']

goal\_total=outcome['Goal']

deflection\_total=outcome['Deflection']

post\_total=outcome['Post']

danger\_quality\_outcome\_df

In [ ]:

danger\_quality\_counts=data1.groupby(['Shot Danger', 'Shot Quality']).size()

danger\_type\_counts=data1.groupby(['Shot Danger', 'Shot Type']).size()

quality\_type\_counts=data1.groupby(['Shot Quality', 'Shot Type']).size()

high\_clean\_total=danger\_quality\_counts['High']['Clean']

high\_rebound\_total=danger\_quality\_counts['High']['Rebound']

high\_redirection\_total=danger\_quality\_counts['High']['Redirection']

high\_rush\_total=danger\_quality\_counts['High']['Rush']

high\_transition\_total=danger\_quality\_counts['High']['Transition']

mid\_clean\_total=danger\_quality\_counts['Mid']['Clean']

mid\_rebound\_total=danger\_quality\_counts['Mid']['Rebound']

mid\_redirection\_total=danger\_quality\_counts['Mid']['Redirection']

mid\_rush\_total=danger\_quality\_counts['Mid']['Rush']

mid\_transition\_total=danger\_quality\_counts['Mid']['Transition']

low\_clean\_total=danger\_quality\_counts['Low']['Clean']

low\_rebound\_total=danger\_quality\_counts['Low']['Rebound']

low\_redirection\_total=danger\_quality\_counts['Low']['Redirection']

low\_rush\_total=danger\_quality\_counts['Low']['Rush']

low\_transition\_total=danger\_quality\_counts['Low']['Transition']

danger\_quality\_outcome\_df

danger\_quality\_outcome\_df.columns=danger\_quality\_outcome\_df.columns.astype(str)

danger\_quality\_outcome\_df['Raw Totals']=[high\_clean\_total, high\_clean\_total,

high\_rebound\_total, high\_rebound\_total,

high\_redirection\_total,

high\_redirection\_total,

high\_rush\_total, high\_rush\_total,

high\_transition\_total, high\_transition\_total,

low\_clean\_total, low\_clean\_total,

low\_rebound\_total, low\_rebound\_total,

low\_redirection\_total, low\_redirection\_total,

low\_rush\_total, low\_rush\_total,

low\_transition\_total, low\_transition\_total,

mid\_clean\_total, mid\_clean\_total,

mid\_rebound\_total, mid\_rebound\_total,

mid\_redirection\_total, mid\_redirection\_total,

mid\_rush\_total, mid\_rush\_total,

mid\_transition\_total, mid\_transition\_total,

]

danger\_quality\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

danger\_quality\_outcome\_df['Percentages']=danger\_quality\_outcome\_df['Totals']/

danger\_quality\_outcome\_df['Raw Totals']

danger\_quality\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

#danger\_type\_outcome\_df

In [ ]:

high\_backhand\_total=danger\_type\_counts['High']['Backhand']

high\_oneT\_total=danger\_type\_counts['High']['One-Timer']

high\_slap\_total=danger\_type\_counts['High']['Slap']

high\_snap\_total=danger\_type\_counts['High']['Snap']

high\_wrap\_total=danger\_type\_counts['High']['Wrap']

high\_wrist\_total=danger\_type\_counts['High']['Wrist']

mid\_backhand\_total=danger\_type\_counts['Mid']['Backhand']

mid\_oneT\_total=danger\_type\_counts['Mid']['One-Timer']

mid\_slap\_total=danger\_type\_counts['Mid']['Slap']

mid\_snap\_total=danger\_type\_counts['Mid']['Snap']

mid\_wrist\_total=danger\_type\_counts['Mid']['Wrist']

low\_backhand\_total=danger\_type\_counts['Low']['Backhand']

low\_oneT\_total=danger\_type\_counts['Low']['One-Timer']

low\_slap\_total=danger\_type\_counts['Low']['Slap']

low\_snap\_total=danger\_type\_counts['Low']['Snap']

low\_wrist\_total=danger\_type\_counts['Low']['Wrist']

danger\_type\_outcome\_df

danger\_type\_outcome\_df.columns=danger\_type\_outcome\_df.columns.astype(str)

danger\_type\_outcome\_df['Raw Totals']=[high\_backhand\_total, high\_backhand\_total,

high\_oneT\_total, high\_oneT\_total,

high\_slap\_total, high\_snap\_total,

high\_snap\_total, high\_wrap\_total,

high\_wrist\_total, high\_wrist\_total,

low\_backhand\_total, low\_oneT\_total,

low\_oneT\_total, low\_slap\_total, low\_slap\_total,

low\_snap\_total, low\_snap\_total, low\_wrist\_total,

low\_wrist\_total, mid\_backhand\_total,

mid\_backhand\_total, mid\_oneT\_total,

mid\_oneT\_total, mid\_slap\_total,

mid\_slap\_total, mid\_snap\_total,

mid\_snap\_total, mid\_wrist\_total,

mid\_wrist\_total

]

danger\_type\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

danger\_type\_outcome\_df['Percentages']=danger\_type\_outcome\_df['Totals']/

danger\_type\_outcome\_df['Raw Totals']

danger\_type\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

#quality\_type\_outcome\_df

In [ ]:

clean\_backhand\_total=quality\_type\_counts['Clean']['Backhand']

clean\_oneT\_total=quality\_type\_counts['Clean']['One-Timer']

clean\_slap\_total=quality\_type\_counts['Clean']['Slap']

clean\_snap\_total=quality\_type\_counts['Clean']['Snap']

clean\_wrap\_total=quality\_type\_counts['Clean']['Wrap']

clean\_wrist\_total=quality\_type\_counts['Clean']['Wrist']

rebound\_backhand\_total=quality\_type\_counts['Rebound']['Backhand']

rebound\_oneT\_total=quality\_type\_counts['Rebound']['One-Timer']

rebound\_slap\_total=quality\_type\_counts['Rebound']['Slap']

rebound\_snap\_total=quality\_type\_counts['Rebound']['Snap']

rebound\_wrap\_total=quality\_type\_counts['Rebound']['Wrap']

rebound\_wrist\_total=quality\_type\_counts['Rebound']['Wrist']

redirection\_backhand\_total=quality\_type\_counts['Redirection']['Backhand']

redirection\_oneT\_total=quality\_type\_counts['Redirection']['One-Timer']

redirection\_slap\_total=quality\_type\_counts['Redirection']['Slap']

redirection\_snap\_total=quality\_type\_counts['Redirection']['Snap']

redirection\_wrist\_total=quality\_type\_counts['Redirection']['Wrist']

rush\_backhand\_total=quality\_type\_counts['Rush']['Backhand']

rush\_oneT\_total=quality\_type\_counts['Rush']['One-Timer']

rush\_slap\_total=quality\_type\_counts['Rush']['Slap']

rush\_snap\_total=quality\_type\_counts['Rush']['Snap']

rush\_wrap\_total=quality\_type\_counts['Rush']['Wrap']

rush\_wrist\_total=quality\_type\_counts['Rush']['Wrist']

transition\_backhand\_total=quality\_type\_counts['Transition']['Backhand']

transition\_oneT\_total=quality\_type\_counts['Transition']['One-Timer']

transition\_slap\_total=quality\_type\_counts['Transition']['Slap']

transition\_snap\_total=quality\_type\_counts['Transition']['Snap']

transition\_wrap\_total=quality\_type\_counts['Transition']['Wrap']

transition\_wrist\_total=quality\_type\_counts['Transition']['Wrist']

quality\_type\_outcome\_df.columns=quality\_type\_outcome\_df.columns.astype(str)

quality\_type\_outcome\_df['Raw Totals']=[clean\_backhand\_total, clean\_backhand\_total,

clean\_oneT\_total, clean\_oneT\_total

clean\_slap\_total, clean\_slap\_total,

clean\_snap\_total, clean\_snap\_total,

clean\_wrap\_total, clean\_wrist\_total,

clean\_wrist\_total, rebound\_backhand\_total,

rebound\_backhand\_total, rebound\_oneT\_total,

rebound\_slap\_total, rebound\_slap\_total,

rebound\_snap\_total, rebound\_snap\_total,

rebound\_wrap\_total, rebound\_wrist\_total,

rebound\_wrist\_total, redirection\_backhand\_total,

redirection\_backhand\_total,

redirection\_oneT\_total, redirection\_oneT\_total,

redirection\_slap\_total, redirection\_snap\_total,

redirection\_snap\_total, redirection\_wrist\_total,

redirection\_wrist\_total, rush\_backhand\_total,

rush\_backhand\_total, rush\_oneT\_total,

rush\_oneT\_total, rush\_slap\_total,

rush\_slap\_total, rush\_snap\_total,

rush\_snap\_total, rush\_wrap\_total,

rush\_wrist\_total, rush\_wrist\_total,

transition\_backhand\_total,

transition\_backhand\_total,

transition\_oneT\_total, transition\_oneT\_total,

transition\_slap\_total, transition\_slap\_total,

transition\_snap\_total, transition\_snap\_total,

transition\_wrap\_total, transition\_wrist\_total,

transition\_wrist\_total

]

quality\_type\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

quality\_type\_outcome\_df['Percentages']=quality\_type\_outcome\_df['Totals']/

quality\_type\_outcome\_df['Raw Totals']

quality\_type\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [ ]:

#Imports ability to read OUT to an excel file

from openpyxl import Workbook

from openpyxl import load\_workbook

from openpyxl.utils.dataframe import dataframe\_to\_rows

from openpyxl.cell.cell import WriteOnlyCell

In [ ]:

#Sets path for data output

path = 'Thesis Output.xlsx'

book = load\_workbook(path)

writer = pd.ExcelWriter(path, engine = 'openpyxl')

writer.book = book

writer.sheets=dict((ws.title, ws) for ws in book.worksheets)

In [ ]:

#Output each dataframe to separate sheet in new Excel file

quality\_outcome\_df.to\_excel(writer, 'QualityOutcome')

writer.save()

danger\_outcome\_df.to\_excel(writer, 'DangerOutcome')

writer.save()

shot\_assists\_outcome\_df.to\_excel(writer, 'ShotAssistsOutcome')

writer.save()

type\_outcome\_df.to\_excel(writer, 'TypeOutcome')

writer.save()

danger\_quality\_outcome\_df.to\_excel(writer, 'DangerQualityOutcome')

writer.save()

danger\_type\_outcome\_df.to\_excel(writer, 'DangerTypeOutcome')

writer.save()

quality\_type\_outcome\_df.to\_excel(writer, 'QualityTypeOutcome')

writer.save()

writer.close()

In [1]:

#Spencer Fascetta

#Thesis Code - Binary Operations

#April 4, 2019

#Endicott College

#Import all necessary mathematical libraries

import pandas as pd

import numpy as np

import nltk

import json

import math

import seaborn as sea

import math

In [2]:

import matplotlib.pyplot as plt

%matplotlib inline

In [158]:

#Read in altered Master Tracking Excel file - replaced all

#instances of Shot Outcomes with Goal or Non Goal

excel\_file="UNH Master Tracking - Thesis Binary.xlsx"

In [159]:

#Creates dataframe using the data stored in previously noted Excel file

data=pd.read\_excel(excel\_file, 'Shot Attempts')

In [160]:

#Defines categorical variables as arrays

shot\_qualities=['Clean', 'Rebound', 'Rush',

'Redirection', 'Transition']

shot\_types=['Backhand','One-Timer','Slap','Snap',

'Wrap', 'Wrist']

shot\_outcomes=['Goal', 'No Goal']

shot\_dangers=['High', 'Mid', 'Low']

In [161]:

#Removes Index column, strength, and event initiator columns

data.drop(['Shot Attempt'], axis=1, inplace=True)

data.drop(['Event Initiator'], axis=1, inplace=True)

data.drop(['Strength'], axis=1, inplace=True)

In [188]:

#Drops on-ice player information from dataframe

data1=data.drop(['On Ice 1', 'On Ice 2', 'On Ice 3',

'On Ice 4', 'On Ice 5', 'On Ice 6'], axis=1)

row=data1.shape[0]

#Groups data into necessary subgroups

danger\_outcome=data1.groupby(['Shot Danger', 'Shot Outcome']).size()

type\_outcome=data1.groupby(['Shot Type', 'Shot Outcome']).size()

quality\_outcome=data1.groupby(['Shot Quality', 'Shot Outcome']).size()

shot\_assists\_outcome=data1.groupby(['Number of Shot Assists',

'Shot Outcome']).size()

danger\_quality\_outcome=data1.groupby(['Shot Danger', 'Shot Quality',

'Shot Outcome']).size()

danger\_type\_outcome=data1.groupby(['Shot Danger', 'Shot Type',

'Shot Outcome']).size()

quality\_type\_outcome=data1.groupby(['Shot Quality', 'Shot Type',

'Shot Outcome']).size()

In [189]:

#Creates dataframe for each subgroup

danger\_outcome\_df=pd.DataFrame(danger\_outcome)

type\_outcome\_df=pd.DataFrame(type\_outcome)

quality\_outcome\_df=pd.DataFrame(quality\_outcome)

shot\_assists\_outcome\_df=pd.DataFrame(shot\_assists\_outcome)

danger\_quality\_outcome\_df=pd.DataFrame(danger\_quality\_outcome)

danger\_type\_outcome\_df=pd.DataFrame(danger\_type\_outcome)

quality\_type\_outcome\_df=pd.DataFrame(quality\_type\_outcome)

In [190]:

#Pulls the individual totals for each master

#variable - these are stored in lists

quality=data['Shot Quality'].value\_counts()

types=data['Shot Type'].value\_counts()

danger=data['Shot Danger'].value\_counts()

assists=data['Number of Shot Assists'].value\_counts()

outcome=data['Shot Outcome'].value\_counts()

#Raw total for each danger level

high\_total=danger['High']

mid\_total=danger['Mid']

low\_total=danger['Low']

#Recasts dataframe as type String

danger\_outcome\_df.columns=danger\_outcome\_df.columns.astype(str)

danger\_outcome\_df

#Adds new column using the raw total of each corresponding subvariable

danger\_outcome\_df['Raw Totals']=[high\_total, high\_total,

low\_total, low\_total,

mid\_total, mid\_total]

danger\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

#Creates new column that calculates the percentage

#of the subtotal for each subvariable

danger\_outcome\_df['Percentages']=danger\_outcome\_df['Totals']/

danger\_outcome\_df['Raw Totals']

#Drops previously created column of raw totals for ease of analysis

danger\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

#danger\_outcome\_df

#THIS PROCESS IS REPEATED FOR EACH SUBGROUP OF DATA

In [191]:

backhand\_total=types['Backhand']

oneT\_total=types['One-Timer']

slap\_total=types['Slap']

snap\_total=types['Snap']

wrap\_total=types['Wrap']

wrist\_total=types['Wrist']

type\_outcome\_df

type\_outcome\_df.columns=type\_outcome\_df.columns.astype(str)

type\_outcome\_df['Raw Totals']=[backhand\_total, backhand\_total,

oneT\_total, oneT\_total,

slap\_total, slap\_total,

snap\_total, snap\_total,

wrap\_total,

wrist\_total, wrist\_total]

type\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

type\_outcome\_df['Percentages']=type\_outcome\_df['Totals']/type\_outcome\_df['Raw Totals']

type\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [192]:

clean\_total=quality['Clean']

redirection\_total=quality['Redirection']

rebound\_total=quality['Rebound']

transition\_total=quality['Transition']

rush\_total=quality['Rush']

quality\_outcome\_df

quality\_outcome\_df.columns=quality\_outcome\_df.columns.astype(str)

quality\_outcome\_df['Raw Totals']=[clean\_total, clean\_total,

rebound\_total, rebound\_total,

redirection\_total, redirection\_total,

rush\_total, rush\_total,

transition\_total, transition\_total]

quality\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

quality\_outcome\_df['Percentages']=quality\_outcome\_df['Totals']/

quality\_outcome\_df['Raw Totals']

quality\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [193]:

zero\_total=assists[0]

one\_total=assists[1]

two\_total=assists[2]

three\_total=assists[3]

four\_total=assists[4]

five\_total=assists[5]

six\_total=assists[6]

seven\_total=assists[7]

eight\_total=assists[8]

nine\_total=assists[9]

ten\_total=assists[10]

eleven\_total=assists[11]

thirteen\_total=assists[13]

fourteen\_total=assists[14]

shot\_assists\_outcome\_df

shot\_assists\_outcome\_df.columns=shot\_assists\_outcome\_df.columns.astype(str)

shot\_assists\_outcome\_df['Raw Totals']=[zero\_total, zero\_total,

one\_total, one\_total,

two\_total, two\_total,

three\_total, three\_total,

four\_total, four\_total,

five\_total, five\_total,

six\_total, seven\_total,

seven\_total, eight\_total,

eight\_total, nine\_total,

ten\_total, ten\_total,

eleven\_total, thirteen\_total,

fourteen\_total]

shot\_assists\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

shot\_assists\_outcome\_df['Percentages']=shot\_assists\_outcome\_df['Totals']/

shot\_assists\_outcome\_df['Raw Totals']

shot\_assists\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [194]:

goal\_total=outcome['Goal']

no\_goal\_total=outcome['No Goal']

#danger\_quality\_outcome\_df

In [195]:

danger\_quality\_counts=data1.groupby(['Shot Danger', 'Shot Quality']).size()

danger\_type\_counts=data1.groupby(['Shot Danger', 'Shot Type']).size()

quality\_type\_counts=data1.groupby(['Shot Quality', 'Shot Type']).size()

high\_clean\_total=danger\_quality\_counts['High']['Clean']

high\_rebound\_total=danger\_quality\_counts['High']['Rebound']

high\_redirection\_total=danger\_quality\_counts['High']['Redirection']

high\_rush\_total=danger\_quality\_counts['High']['Rush']

high\_transition\_total=danger\_quality\_counts['High']['Transition']

mid\_clean\_total=danger\_quality\_counts['Mid']['Clean']

mid\_rebound\_total=danger\_quality\_counts['Mid']['Rebound']

mid\_redirection\_total=danger\_quality\_counts['Mid']['Redirection']

mid\_rush\_total=danger\_quality\_counts['Mid']['Rush']

mid\_transition\_total=danger\_quality\_counts['Mid']['Transition']

low\_clean\_total=danger\_quality\_counts['Low']['Clean']

low\_rebound\_total=danger\_quality\_counts['Low']['Rebound']

low\_redirection\_total=danger\_quality\_counts['Low']['Redirection']

low\_rush\_total=danger\_quality\_counts['Low']['Rush']

low\_transition\_total=danger\_quality\_counts['Low']['Transition']

danger\_quality\_outcome\_df

danger\_quality\_outcome\_df.columns=danger\_quality\_outcome\_df.columns.astype(str)

danger\_quality\_outcome\_df['Raw Totals']=[high\_clean\_total, high\_clean\_total,

high\_rebound\_total, high\_rebound\_total,

high\_redirection\_total,

high\_redirection\_total,

high\_rush\_total, high\_rush\_total,

high\_transition\_total, high\_transition\_total,

low\_clean\_total, low\_clean\_total,

low\_rebound\_total, low\_rebound\_total,

low\_redirection\_total, low\_redirection\_total,

low\_rush\_total, low\_rush\_total,

low\_transition\_total, low\_transition\_total,

mid\_clean\_total, mid\_clean\_total,

mid\_rebound\_total, mid\_rebound\_total,

mid\_redirection\_total, mid\_redirection\_total,

mid\_rush\_total, mid\_rush\_total,

mid\_transition\_total, mid\_transition\_total,

]

danger\_quality\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

danger\_quality\_outcome\_df['Percentages']=danger\_quality\_outcome\_df['Totals']/

danger\_quality\_outcome\_df['Raw Totals']

danger\_quality\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

#danger\_type\_outcome\_df

In [196]:

high\_backhand\_total=danger\_type\_counts['High']['Backhand']

high\_oneT\_total=danger\_type\_counts['High']['One-Timer']

high\_slap\_total=danger\_type\_counts['High']['Slap']

high\_snap\_total=danger\_type\_counts['High']['Snap']

high\_wrap\_total=danger\_type\_counts['High']['Wrap']

high\_wrist\_total=danger\_type\_counts['High']['Wrist']

mid\_backhand\_total=danger\_type\_counts['Mid']['Backhand']

mid\_oneT\_total=danger\_type\_counts['Mid']['One-Timer']

mid\_slap\_total=danger\_type\_counts['Mid']['Slap']

mid\_snap\_total=danger\_type\_counts['Mid']['Snap']

mid\_wrist\_total=danger\_type\_counts['Mid']['Wrist']

low\_backhand\_total=danger\_type\_counts['Low']['Backhand']

low\_oneT\_total=danger\_type\_counts['Low']['One-Timer']

low\_slap\_total=danger\_type\_counts['Low']['Slap']

low\_snap\_total=danger\_type\_counts['Low']['Snap']

low\_wrist\_total=danger\_type\_counts['Low']['Wrist']

danger\_type\_outcome\_df

danger\_type\_outcome\_df.columns=danger\_type\_outcome\_df.columns.astype(str)

danger\_type\_outcome\_df['Raw Totals']=[high\_backhand\_total, high\_backhand\_total,

high\_oneT\_total, high\_oneT\_total,

high\_slap\_total, high\_snap\_total,

high\_snap\_total, high\_wrap\_total,

high\_wrist\_total, high\_wrist\_total,

low\_backhand\_total, low\_oneT\_total,

low\_oneT\_total, low\_slap\_total, low\_slap\_total,

low\_snap\_total, low\_snap\_total,

low\_wrist\_total, low\_wrist\_total,

mid\_backhand\_total, mid\_backhand\_total,

mid\_oneT\_total, mid\_oneT\_total,

mid\_slap\_total, mid\_slap\_total,

mid\_snap\_total, mid\_snap\_total,

mid\_wrist\_total, mid\_wrist\_total

]

danger\_type\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

danger\_type\_outcome\_df['Percentages']=danger\_type\_outcome\_df['Totals']/

danger\_type\_outcome\_df['Raw Totals']

danger\_type\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

#quality\_type\_outcome\_df

In [197]:

clean\_backhand\_total=quality\_type\_counts['Clean']['Backhand']

clean\_oneT\_total=quality\_type\_counts['Clean']['One-Timer']

clean\_slap\_total=quality\_type\_counts['Clean']['Slap']

clean\_snap\_total=quality\_type\_counts['Clean']['Snap']

clean\_wrap\_total=quality\_type\_counts['Clean']['Wrap']

clean\_wrist\_total=quality\_type\_counts['Clean']['Wrist']

rebound\_backhand\_total=quality\_type\_counts['Rebound']['Backhand']

rebound\_oneT\_total=quality\_type\_counts['Rebound']['One-Timer']

rebound\_slap\_total=quality\_type\_counts['Rebound']['Slap']

rebound\_snap\_total=quality\_type\_counts['Rebound']['Snap']

rebound\_wrap\_total=quality\_type\_counts['Rebound']['Wrap']

rebound\_wrist\_total=quality\_type\_counts['Rebound']['Wrist']

redirection\_backhand\_total=quality\_type\_counts['Redirection']['Backhand']

redirection\_oneT\_total=quality\_type\_counts['Redirection']['One-Timer']

redirection\_slap\_total=quality\_type\_counts['Redirection']['Slap']

redirection\_snap\_total=quality\_type\_counts['Redirection']['Snap']

redirection\_wrist\_total=quality\_type\_counts['Redirection']['Wrist']

rush\_backhand\_total=quality\_type\_counts['Rush']['Backhand']

rush\_oneT\_total=quality\_type\_counts['Rush']['One-Timer']

rush\_slap\_total=quality\_type\_counts['Rush']['Slap']

rush\_snap\_total=quality\_type\_counts['Rush']['Snap']

rush\_wrap\_total=quality\_type\_counts['Rush']['Wrap']

rush\_wrist\_total=quality\_type\_counts['Rush']['Wrist']

transition\_backhand\_total=quality\_type\_counts['Transition']['Backhand']

transition\_oneT\_total=quality\_type\_counts['Transition']['One-Timer']

transition\_slap\_total=quality\_type\_counts['Transition']['Slap']

transition\_snap\_total=quality\_type\_counts['Transition']['Snap']

transition\_wrap\_total=quality\_type\_counts['Transition']['Wrap']

transition\_wrist\_total=quality\_type\_counts['Transition']['Wrist']

quality\_type\_outcome\_df.columns=quality\_type\_outcome\_df.columns.astype(str)

quality\_type\_outcome\_df['Raw Totals']=[clean\_backhand\_total, clean\_backhand\_total,

clean\_oneT\_total, clean\_slap\_total,

clean\_slap\_total, clean\_snap\_total,

clean\_snap\_total, clean\_wrap\_total,

clean\_wrist\_total, clean\_wrist\_total,

rebound\_backhand\_total, rebound\_backhand\_total,

rebound\_oneT\_total, rebound\_slap\_total,

rebound\_slap\_total, rebound\_snap\_total,

rebound\_snap\_total, rebound\_wrap\_total,

rebound\_wrist\_total, rebound\_wrist\_total,

redirection\_backhand\_total,

redirection\_backhand\_total,

redirection\_oneT\_total, redirection\_oneT\_total,

redirection\_slap\_total, redirection\_snap\_total,

redirection\_snap\_total, redirection\_wrist\_total,

redirection\_wrist\_total, rush\_backhand\_total,

rush\_backhand\_total, rush\_oneT\_total,

rush\_oneT\_total, rush\_slap\_total,

rush\_slap\_total, rush\_snap\_total,

rush\_snap\_total, rush\_wrap\_total,

rush\_wrist\_total, rush\_wrist\_total,

transition\_backhand\_total,

transition\_backhand\_total,

transition\_oneT\_total, transition\_oneT\_total,

transition\_slap\_total, transition\_slap\_total,

transition\_snap\_total, transition\_snap\_total,

transition\_wrap\_total, transition\_wrist\_total,

transition\_wrist\_total

]

quality\_type\_outcome\_df.rename(index=str, columns={'0':'Totals'}, inplace=True)

quality\_type\_outcome\_df['Percentages']=quality\_type\_outcome\_df['Totals']/

quality\_type\_outcome\_df['Raw Totals']

quality\_type\_outcome\_df.drop(columns=['Raw Totals'], inplace=True)

In [198]:

#Creates a new dataframe of each single-variable subgroup combined into one

single\_variable=danger\_outcome\_df.append(type\_outcome\_df.append(

shot\_assists\_outcome\_df.append(quality\_outcome\_df)))

single\_variable.rename(index=str, columns={'Shot Danger':'Variable'}, inplace=True)

In [199]:

#Imports ability to read OUT to an excel file

from openpyxl import Workbook

from openpyxl import load\_workbook

from openpyxl.utils.dataframe import dataframe\_to\_rows

from openpyxl.cell.cell import WriteOnlyCell

In [200]:

#Sets path for data output

path = 'Thesis Output.xlsx'

book = load\_workbook(path)

writer = pd.ExcelWriter(path, engine = 'openpyxl')

writer.book = book

writer.sheets=dict((ws.title, ws) for ws in book.worksheets)

In [201]:

#Output each dataframe to separate sheet in new Excel file

quality\_outcome\_df.to\_excel(writer, 'QualityOutcome Binary')

writer.save()

danger\_outcome\_df.to\_excel(writer, 'DangerOutcome Binary')

writer.save()

shot\_assists\_outcome\_df.to\_excel(writer, 'ShotAssistsOutcome Binary')

writer.save()

type\_outcome\_df.to\_excel(writer, 'TypeOutcome Binary')

writer.save()

danger\_quality\_outcome\_df.to\_excel(writer, 'DangerQualityOutcome Binary')

writer.save()

danger\_type\_outcome\_df.to\_excel(writer, 'DangerTypeOutcome Binary')

writer.save()

quality\_type\_outcome\_df.to\_excel(writer, 'QualityTypeOutcome Binary')

writer.save()

writer.close()

In [203]:

#imports modeling libraries

from sklearn import preprocessing

plt.rc("font", size=14)

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

import seaborn as sns

sns.set(style="white")

sns.set(style="whitegrid", color\_codes=True)

In [204]:

#Creation of dummy variables. These variables only have two values - either 0 or 1

cat\_vars=['Number of Shot Assists', 'Shot Danger', 'Shot Quality',

'Shot Type', 'Shot Outcome']

for var in cat\_vars:

cat\_list='var'+'\_'+var

cat\_list = pd.get\_dummies(data1[var], prefix=var)

data2=data1.join(cat\_list)

data1=data2

cat\_vars=['Number of Shot Assists', 'Shot Danger', 'Shot Quality', 'Shot Type']

data\_vars=data1.columns.values.tolist()

to\_keep=[i for i in data\_vars if i not in cat\_vars]

In [205]:

#Sets final data columns

data\_final=data1[to\_keep]

In [240]:

data\_final.columns.values

data\_final.head()

Out[240]:

array(['Shot Outcome', 'Number of Shot Assists\_0.0',

'Number of Shot Assists\_1.0', 'Number of Shot Assists\_2.0',

'Number of Shot Assists\_3.0', 'Number of Shot Assists\_4.0',

'Number of Shot Assists\_5.0', 'Number of Shot Assists\_6.0',

'Number of Shot Assists\_7.0', 'Number of Shot Assists\_8.0',

'Number of Shot Assists\_9.0', 'Number of Shot Assists\_10.0',

'Number of Shot Assists\_11.0', 'Number of Shot Assists\_13.0',

'Number of Shot Assists\_14.0', 'Shot Danger\_High',

'Shot Danger\_Low', 'Shot Danger\_Mid', 'Shot Quality\_Clean',

'Shot Quality\_Rebound', 'Shot Quality\_Redirection',

'Shot Quality\_Rush', 'Shot Quality\_Transition',

'Shot Type\_Backhand', 'Shot Type\_One-Timer', 'Shot Type\_Slap',

'Shot Type\_Snap', 'Shot Type\_Wrap', 'Shot Type\_Wrist',

'Shot Outcome\_Goal', 'Shot Outcome\_No Goal'], dtype=object)

In [318]:

#Over-sampling, utilizing SMOTE (Synthetic Minority Oversampling Technique) technique

#Point of this is to balance the data so there is a more even dataset to train a model

#ONLY BALANCES TRAINING DATA. This maintains the integrity of the test data

X = data\_final.loc[:, data\_final.columns != 'y']

y = data\_final.loc[:, data\_final.columns == 'Shot Outcome']

#Replaces Shot Outcome column with binary results for Goal and No Goal,

#then casts the columns as type INTEGER

binary={'Goal':1, 'No Goal':0}

y['Shot Outcome'].replace(binary, inplace=True)

y.astype(int)

X['Shot Outcome'].replace(binary, inplace=True)

X.astype(int)

from imblearn.over\_sampling import SMOTE

os = SMOTE(random\_state=0)

#Splits data into test and train subsets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)

columns = X\_train.columns

os\_data\_X,os\_data\_y=os.fit\_sample(X\_train, y\_train)

os\_data\_X = pd.DataFrame(data=os\_data\_X,columns=columns )

os\_data\_y= pd.DataFrame(data=os\_data\_y,columns=['y'])

os\_data\_X.drop(['Shot Outcome\_Goal'], axis=1, inplace=True)

os\_data\_X.drop(['Shot Outcome\_No Goal'], axis=1, inplace=True)

os\_data\_X.drop(['Number of Shot Assists\_5.0', 'Number of Shot Assists\_6.0',

'Number of Shot Assists\_7.0', 'Number of Shot Assists\_8.0',

'Number of Shot Assists\_9.0', 'Number of Shot Assists\_10.0',

'Number of Shot Assists\_11.0', 'Number of Shot Assists\_13.0',

'Number of Shot Assists\_14.0'], axis=1, inplace=True)

C:\Users\Spencer\Anaconda3\lib\site-packages\pandas\core\generic.py:5886: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy

self.\_update\_inplace(new\_data)

C:\Users\Spencer\Anaconda3\lib\site-packages\sklearn\utils\validation.py:761: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

y = column\_or\_1d(y, warn=True)

In [319]:

#Recursive Feature Elimination

#Tests each feature of the model to see how well it performs

#independently, then lists the viability of each feature

#This was tested with entirety of data, and pared down into

#ONLY features that returned TRUE - maintains accuracy of model

data\_final\_vars=data\_final.columns.values.tolist()

y=['y']

X=[i for i in data\_final\_vars if i not in y]

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LogisticRegression

logreg = LogisticRegression()

rfe = RFE(logreg, 20)

rfe = rfe.fit(os\_data\_X, os\_data\_y.values.ravel())

print(rfe.support\_)

print(rfe.ranking\_)

[ True True True True True True True True True True True True

True True True True True True True True]

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

C:\Users\Spencer\Anaconda3\lib\site-packages\sklearn\linear\_model\logistic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.

FutureWarning)

In [320]:

#Defines features used to build model as decided by the Recursive Feature Elimination

cols=['Number of Shot Assists\_0.0', 'Number of Shot Assists\_1.0',

'Number of Shot Assists\_2.0', 'Number of Shot Assists\_3.0',

'Number of Shot Assists\_4.0', 'Shot Danger\_High', 'Shot Danger\_Low',

'Shot Danger\_Mid', 'Shot Quality\_Clean', 'Shot Quality\_Rebound',

'Shot Quality\_Redirection', 'Shot Quality\_Rush',

'Shot Quality\_Transition', 'Shot Type\_Backhand', 'Shot Type\_One-Timer',

'Shot Type\_Slap', 'Shot Type\_Snap', 'Shot Type\_Wrap', 'Shot Type\_Wrist']

X=os\_data\_X[cols]

y=os\_data\_y['y']

In [321]:

import statsmodels.api as sm

#Creates logistic model

logit\_model=sm.Logit(y,X)

#Fits model using training data/test data

result=logit\_model.fit()

#Summarizes the results of running the model

print(result.summary2())

#print(result.mle\_retvals)

Warning: Maximum number of iterations has been exceeded.

Current function value: 0.499892

Iterations: 35

Results: Logit

=======================================================================================

Model: Logit Pseudo R-squared: 0.279

Dependent Variable: y AIC: 5424.8345

Date: 2019-04-03 15:04 BIC: 5550.0812

No. Observations: 5388 Log-Likelihood: -2693.4

Df Model: 18 LL-Null: -3734.7

Df Residuals: 5369 LLR p-value: 0.0000

Converged: 0.0000 Scale: 1.0000

No. Iterations: 35.0000

---------------------------------------------------------------------------------------

Coef. Std.Err. z P>|z| [0.025 0.975]

---------------------------------------------------------------------------------------

Number of Shot Assists\_0.0 -1.1449 0.1257 -9.1102 0.0000 -1.3912 -0.8986

Number of Shot Assists\_1.0 -0.0841 0.1082 -0.7773 0.4370 -0.2962 0.1280

Number of Shot Assists\_2.0 -0.1708 0.1417 -1.2051 0.2282 -0.4485 0.1070

Number of Shot Assists\_3.0 -1.7058 0.4207 -4.0547 0.0001 -2.5304 -0.8813

Number of Shot Assists\_4.0 -3.7151 1.0348 -3.5903 0.0003 -5.7432 -1.6870

Shot Danger\_High 1.7934 0.1157 15.5012 0.0000 1.5666 2.0201

Shot Danger\_Low -0.6969 0.1323 -5.2687 0.0000 -0.9562 -0.4377

Shot Danger\_Mid -0.0322 0.1177 -0.2733 0.7846 -0.2628 0.1985

Shot Quality\_Clean -0.0963 0.1219 -0.7899 0.4296 -0.3352 0.1426

Shot Quality\_Rebound 1.7765 0.1419 12.5191 0.0000 1.4984 2.0547

Shot Quality\_Redirection 0.7983 0.2139 3.7328 0.0002 0.3791 1.2175

Shot Quality\_Rush -0.0081 0.1161 -0.0697 0.9444 -0.2356 0.2194

Shot Quality\_Transition -0.3170 0.1436 -2.2074 0.0273 -0.5984 -0.0355

Shot Type\_Backhand -1.3306 0.1595 -8.3450 0.0000 -1.6432 -1.0181

Shot Type\_One-Timer 0.3076 0.1643 1.8719 0.0612 -0.0145 0.6297

Shot Type\_Slap -0.1487 0.1525 -0.9753 0.3294 -0.4475 0.1501

Shot Type\_Snap -1.1533 0.1270 -9.0814 0.0000 -1.4022 -0.9044

Shot Type\_Wrap -29.6043 198655.5679 -0.0001 0.9999 -389387.3627 389328.1540

Shot Type\_Wrist -0.3324 0.1062 -3.1312 0.0017 -0.5405 -0.1243

=======================================================================================

C:\Users\Spencer\Anaconda3\lib\site-packages\statsmodels\base\model.py:508: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle\_retvals

"Check mle\_retvals", ConvergenceWarning)

In [322]:

#Logistic regression using reorganized data

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3,

random\_state=0)

logreg = LogisticRegression()

#refit model

logreg.fit(X\_train, y\_train)

C:\Users\Spencer\Anaconda3\lib\site-packages\sklearn\linear\_model\logistic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.

FutureWarning)

Out[322]:

LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True,

intercept\_scaling=1, max\_iter=100, multi\_class='warn',

n\_jobs=None, penalty='l2', random\_state=None, solver='warn',

tol=0.0001, verbose=0, warm\_start=False)

In [323]:

#Gives the accuracy of the predictivity of the model

y\_pred = logreg.predict(X\_test)

print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(

logreg.score(X\_test, y\_test)))

Accuracy of logistic regression classifier on test set: 0.87

In [324]:

#Confusion Matrix for model. This gives two arrays.

#The top array tells us the number of correct predictions, the bottom

#tells us the number of incorrect predictions

from sklearn.metrics import confusion\_matrix

confusion\_matrix = confusion\_matrix(y\_test, y\_pred)

print(confusion\_matrix)

[[684 115]

[ 92 726]]

In [325]:

#Computes Precision, recall, F-measure, and support values for model

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))

precision recall f1-score support

0 0.88 0.86 0.87 799

1 0.86 0.89 0.88 818

micro avg 0.87 0.87 0.87 1617

macro avg 0.87 0.87 0.87 1617

weighted avg 0.87 0.87 0.87 1617

In [326]:

#Creates ROC (Receiving Operator Characteristic) Curve for model. A good classifier (the blue line) stays as far away from that

#of a purely random classifier (the dotted line), preferably in the top left of the plot

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

logit\_roc\_auc = roc\_auc\_score(y\_test, logreg.predict(X\_test))

fpr, tpr, thresholds = roc\_curve(y\_test, logreg.predict\_proba(X\_test)[:,1])

plt.figure()

plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit\_roc\_auc)

plt.plot([0, 1], [0, 1],'r--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

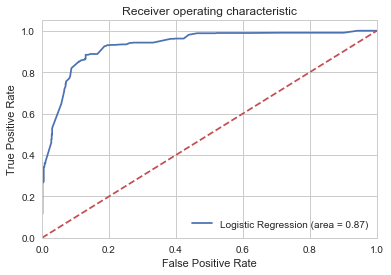
plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic')

plt.legend(loc="lower right")

plt.savefig('Log\_ROC')

plt.show()

****

**Appendix C: Decisions Made**

Several key moments occurred throughout the coding process of this project that helped to shape the overall end-product. Once the decision was made to utilize a logistical model, the data needed to be altered in order to be read into the program properly. This is the reason that a second program was written, utilizing the same exact data separation and grouping process as the original program, but redefining the shot outcomes in binary terms, meaning any shot attempt which resulted in a post, a save, a missed net, a block, or a deflection was re-categorized as resulting in no goal. This is because the model used necessitated a response variable with an either-or result to properly relate the other categories to the result. Once this was completed, the data needed to be properly balanced. It was decided that the process of Synthetic Minority Oversampling Technique, or SMOTE, would be implemented in order to develop an even, more reliable dataset for the model to be trained accurately. This meant splitting the data into randomly generated groups – one to train the model, and one to test the model. At this point, the test data was set aside and left untouched until the time came to test the validity of the model itself. SMOTE was used on the training set.

When the model was built using the entirety of the training set after the use of SMOTE, the result was a model which projected 100% accuracy, a statistical anomaly, and almost impossible to achieve with an organic dataset. In order to determine the root cause of the issue, the data was then subjected to Recursive Feature Elimination. This determines the viability of each feature of the model using the previously defined variables and lists which of the variables is trustworthy when used in the logistical model. This revealed that shot attempts on which there were 5 or more shot assists did not occur with enough frequency to be reliably used to predict future outcomes. These were then excised from the model entirely.

When writing the code used to generate the plots used in this study, a decision was made to utilize horizontal bar graphs, as they provided the most logical visual representation of the multivariate nature of the data. The plots were regenerated more than 10 times in order to ensure that the annotations and text present were to a large enough scale which a read could comprehend the context for the plot.

**Appendix D: Tables**

Table 2.1: Shot Danger vs Shot Outcome

|  |  |  |  |
| --- | --- | --- | --- |
| **Shot Danger** | **Shot Outcome** | **Totals** | **Percentages** |
| **High** | **Blocked** | 122 | 11.71% |
| **Goal** | 103 | 9.88% |
| **Miss** | 133 | 12.76% |
| **Post** | 8 | 0.77% |
| **Save** | 676 | 64.88% |
| **Low** | **Blocked** | 646 | 39.15% |
| **Deflection** | 64 | 3.88% |
| **Goal** | 26 | 1.58% |
| **Miss** | 312 | 18.91% |
| **Post** | 3 | 0.18% |
| **Save** | 599 | 36.30% |
| **Mid** | **Blocked** | 390 | 29.91% |
| **Deflection** | 7 | 0.54% |
| **Goal** | 29 | 2.22% |
| **Miss** | 183 | 14.03% |
| **Post** | 7 | 0.54% |
| **Save** | 688 | 52.76% |

Table 2.2: Shot Type vs Shot Outcome

|  |  |  |  |
| --- | --- | --- | --- |
| **Shot Type** | **Shot Outcome** | **Totals** | **Percentages** |
| **Backhand** | **Blocked** | 38 | 13.97% |
| **Goal** | 17 | 6.25% |
| **Miss** | 40 | 14.71% |
| **Post** | 1 | 0.37% |
| **Save** | 176 | 64.71% |
| **One-Timer** | **Blocked** | 57 | 19.39% |
| **Deflection** | 4 | 1.36% |
| **Goal** | 26 | 8.84% |
| **Miss** | 78 | 26.53% |
| **Post** | 3 | 1.02% |
| **Save** | 126 | 42.86% |
| **Slap** | **Blocked** | 200 | 42.92% |
| **Deflection** | 11 | 2.36% |
| **Goal** | 12 | 2.58% |
| **Miss** | 80 | 17.17% |
| **Post** | 1 | 0.21% |
| **Save** | 162 | 34.76% |
| **Snap** | **Blocked** | 328 | 29.60% |
| **Deflection** | 21 | 1.90% |
| **Goal** | 36 | 3.25% |
| **Miss** | 158 | 14.26% |
| **Post** | 9 | 0.81% |
| **Save** | 556 | 50.18% |
| **Wrap** | **Blocked** | 11 | 18.33% |
| **Miss** | 5 | 8.33% |
| **Save** | 44 | 73.33% |
| **Wrist** | **Blocked** | 524 | 29.18% |
| **Deflection** | 35 | 1.95% |
| **Goal** | 67 | 3.73% |
| **Miss** | 267 | 14.87% |
| **Post** | 4 | 0.22% |
| **Save** | 899 | 50.06% |

Table 2.3: Shot Quality vs Shot Outcome

|  |  |  |  |
| --- | --- | --- | --- |
| **Shot Quality** | **Shot Outcome** | **Totals** | **Percentages** |
| **Clean** | **Blocked** | 636 | 38.11% |
| **Deflection** | 58 | 3.48% |
| **Goal** | 31 | 1.86% |
| **Miss** | 254 | 15.22% |
| **Post** | 5 | 0.30% |
| **Save** | 685 | 41.04% |
| **Rebound** | **Blocked** | 89 | 23.00% |
| **Deflection** | 1 | 0.26% |
| **Goal** | 42 | 10.85% |
| **Miss** | 49 | 12.66% |
| **Post** | 2 | 0.52% |
| **Save** | 204 | 52.71% |
| **Redirection** | **Blocked** | 6 | 7.79% |
| **Goal** | 13 | 16.88% |
| **Miss** | 28 | 36.36% |
| **Save** | 30 | 38.96% |
| **Rush** | **Blocked** | 280 | 20.63% |
| **Deflection** | 2 | 0.15% |
| **Goal** | 45 | 3.32% |
| **Miss** | 197 | 14.52% |
| **Post** | 7 | 0.52% |
| **Save** | 826 | 60.87% |
| **Transition** | **Blocked** | 147 | 29.05% |
| **Deflection** | 10 | 1.98% |
| **Goal** | 27 | 5.34% |
| **Miss** | 100 | 19.76% |
| **Post** | 4 | 0.79% |
| **Save** | 218 | 43.08% |

Table 2.4: Number of Shot Assists vs Shot Outcome

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Shot Assists** | **Shot Outcome** | **Totals** | **Percentages** |
| **0.0** | **Blocked** | 519 | 24.93% |
| **Deflection** | 14 | 0.67% |
| **Goal** | 79 | 3.79% |
| **Miss** | 316 | 15.18% |
| **Post** | 8 | 0.38% |
| **Save** | 1146 | 55.04% |
| **1.0** | **Blocked** | 401 | 32.42% |
| **Deflection** | 36 | 2.91% |
| **Goal** | 47 | 3.80% |
| **Miss** | 185 | 14.96% |
| **Post** | 7 | 0.57% |
| **Save** | 561 | 45.35% |
| **2.0** | **Blocked** | 143 | 33.97% |
| **Deflection** | 16 | 3.80% |
| **Goal** | 19 | 4.51% |
| **Miss** | 79 | 18.76% |
| **Post** | 2 | 0.48% |
| **Save** | 162 | 38.48% |
| **3.0** | **Blocked** | 43 | 38.05% |
| **Deflection** | 1 | 0.88% |
| **Goal** | 4 | 3.54% |
| **Miss** | 26 | 23.01% |
| **Post** | 1 | 0.88% |
| **Save** | 38 | 33.63% |
| **4.0** | **Blocked** | 25 | 38.46% |
| **Deflection** | 2 | 3.08% |
| **Goal** | 1 | 1.54% |
| **Miss** | 10 | 15.38% |
| **Save** | 27 | 41.54% |
| **5.0** | **Blocked** | 11 | 33.33% |
| **Deflection** | 1 | 3.03% |
| **Goal** | 4 | 12.12% |
| **Miss** | 4 | 12.12% |
| **Save** | 13 | 39.39% |
| **6.0** | **Blocked** | 6 | 30.00% |
| **Deflection** | 1 | 5.00% |
| **Miss** | 4 | 20.00% |
| **Save** | 9 | 45.00% |
| **7.0** | **Blocked** | 7 | 63.64% |
| **Goal** | 1 | 9.09% |
| **Miss** | 1 | 9.09% |
| **Save** | 2 | 18.18% |
| **8.0** | **Blocked** | 1 | 16.67% |
| **Goal** | 1 | 16.67% |
| **Miss** | 1 | 16.67% |
| **Save** | 3 | 50.00% |
| **9.0** | **Miss** | 1 | 50.00% |
| **Save** | 1 | 50.00% |
| **10.0** | **Goal** | 1 | 50.00% |
| **Save** | 1 | 50.00% |
| **11.0** | **Blocked** | 1 | 100.00% |
| **13.0** | **Blocked** | 1 | 100.00% |
| **14.0** | **Miss** | 1 | 100.00% |

Table 2.5: Shot Danger and Shot Type vs Shot Outcome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shot Danger** | **Shot Type** | **Shot Outcome** | **Totals** | **Percentages** |
| **High** | **Backhand** | **Blocked** | 27 | 0.120535714 |
| **Goal** | 16 | 0.071428571 |
| **Miss** | 33 | 0.147321429 |
| **Post** | 1 | 0.004464286 |
| **Save** | 147 | 0.65625 |
| **One-Timer** | **Blocked** | 14 | 0.107692308 |
| **Goal** | 18 | 0.138461538 |
| **Miss** | 25 | 0.192307692 |
| **Post** | 2 | 0.015384615 |
| **Save** | 71 | 0.546153846 |
| **Slap** | **Blocked** | 2 | 0.285714286 |
| **Miss** | 1 | 0.142857143 |
| **Save** | 4 | 0.571428571 |
| **Snap** | **Blocked** | 31 | 0.122529644 |
| **Goal** | 27 | 0.106719368 |
| **Miss** | 30 | 0.118577075 |
| **Post** | 4 | 0.015810277 |
| **Save** | 161 | 0.636363636 |
| **Wrap** | **Blocked** | 11 | 0.183333333 |
| **Miss** | 5 | 0.083333333 |
| **Save** | 44 | 0.733333333 |
| **Wrist** | **Blocked** | 37 | 0.100543478 |
| **Goal** | 42 | 0.114130435 |
| **Miss** | 39 | 0.105978261 |
| **Post** | 1 | 0.002717391 |
| **Save** | 249 | 0.676630435 |
| **Low** | **Backhand** | **Blocked** | 1 | 0.25 |
| **Miss** | 2 | 0.5 |
| **Save** | 1 | 0.25 |
| **One-Timer** | **Blocked** | 28 | 0.329411765 |
| **Deflection** | 3 | 0.035294118 |
| **Goal** | 2 | 0.023529412 |
| **Miss** | 24 | 0.282352941 |
| **Save** | 28 | 0.329411765 |
| **Slap** | **Blocked** | 161 | 0.442307692 |
| **Deflection** | 11 | 0.03021978 |
| **Goal** | 7 | 0.019230769 |
| **Miss** | 67 | 0.184065934 |
| **Post** | 1 | 0.002747253 |
| **Save** | 117 | 0.321428571 |
| **Snap** | **Blocked** | 172 | 0.390909091 |
| **Deflection** | 19 | 0.043181818 |
| **Goal** | 2 | 0.004545455 |
| **Miss** | 79 | 0.179545455 |
| **Save** | 168 | 0.381818182 |
| **Wrist** | **Blocked** | 284 | 0.375165125 |
| **Deflection** | 31 | 0.040951123 |
| **Goal** | 15 | 0.019815059 |
| **Miss** | 140 | 0.184940555 |
| **Post** | 2 | 0.002642008 |
| **Save** | 285 | 0.376486129 |
| **Mid** | **Backhand** | **Blocked** | 10 | 0.227272727 |
| **Goal** | 1 | 0.022727273 |
| **Miss** | 5 | 0.113636364 |
| **Save** | 28 | 0.636363636 |
| **One-Timer** | **Blocked** | 15 | 0.189873418 |
| **Deflection** | 1 | 0.012658228 |
| **Goal** | 6 | 0.075949367 |
| **Miss** | 29 | 0.367088608 |
| **Post** | 1 | 0.012658228 |
| **Save** | 27 | 0.341772152 |
| **Slap** | **Blocked** | 37 | 0.389473684 |
| **Goal** | 5 | 0.052631579 |
| **Miss** | 12 | 0.126315789 |
| **Save** | 41 | 0.431578947 |
| **Snap** | **Blocked** | 125 | 0.301204819 |
| **Deflection** | 2 | 0.004819277 |
| **Goal** | 7 | 0.01686747 |
| **Miss** | 49 | 0.118072289 |
| **Post** | 5 | 0.012048193 |
| **Save** | 227 | 0.546987952 |
| **Wrist** | **Blocked** | 203 | 0.302533532 |
| **Deflection** | 4 | 0.005961252 |
| **Goal** | 10 | 0.01490313 |
| **Miss** | 88 | 0.131147541 |
| **Post** | 1 | 0.001490313 |
| **Save** | 365 | 0.543964232 |

Table 2.6: Shot Danger and Shot Quality vs Shot Outcome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shot Danger** | **Shot Quality** | **Shot Outcome** | **Totals** | **Percentages** |
| **High** | **Clean** | **Blocked** | 25 | 0.11627907 |
| **Goal** | 7 | 0.03255814 |
| **Miss** | 28 | 0.130232558 |
| **Post** | 2 | 0.009302326 |
| **Save** | 153 | 0.711627907 |
| **Rebound** | **Blocked** | 35 | 0.134615385 |
| **Goal** | 39 | 0.15 |
| **Miss** | 27 | 0.103846154 |
| **Post** | 2 | 0.007692308 |
| **Save** | 157 | 0.603846154 |
| **Redirection** | **Blocked** | 4 | 0.071428571 |
| **Goal** | 10 | 0.178571429 |
| **Miss** | 18 | 0.321428571 |
| **Save** | 24 | 0.428571429 |
| **Rush** | **Blocked** | 29 | 0.092063492 |
| **Goal** | 29 | 0.092063492 |
| **Miss** | 31 | 0.098412698 |
| **Post** | 2 | 0.006349206 |
| **Save** | 224 | 0.711111111 |
| **Transition** | **Blocked** | 29 | 0.147959184 |
| **Goal** | 18 | 0.091836735 |
| **Miss** | 29 | 0.147959184 |
| **Post** | 2 | 0.010204082 |
| **Save** | 118 | 0.602040816 |
| **Low** | **Clean** | **Blocked** | 455 | 0.445641528 |
| **Deflection** | 53 | 0.051909892 |
| **Goal** | 17 | 0.016650343 |
| **Miss** | 175 | 0.171400588 |
| **Post** | 1 | 0.000979432 |
| **Save** | 320 | 0.313418217 |
| **Rebound** | **Blocked** | 27 | 0.5 |
| **Deflection** | 1 | 0.018518519 |
| **Goal** | 1 | 0.018518519 |
| **Miss** | 12 | 0.222222222 |
| **Save** | 13 | 0.240740741 |
| **Redirection** | **Goal** | 1 | 0.5 |
| **Save** | 1 | 0.5 |
| **Rush** | **Blocked** | 96 | 0.237623762 |
| **Deflection** | 1 | 0.002475248 |
| **Goal** | 4 | 0.00990099 |
| **Miss** | 85 | 0.21039604 |
| **Post** | 2 | 0.004950495 |
| **Save** | 216 | 0.534653465 |
| **Transition** | **Blocked** | 68 | 0.402366864 |
| **Deflection** | 9 | 0.053254438 |
| **Goal** | 3 | 0.017751479 |
| **Miss** | 40 | 0.236686391 |
| **Save** | 49 | 0.289940828 |
| **Mid** | **Clean** | **Blocked** | 156 | 0.360277136 |
| **Deflection** | 5 | 0.011547344 |
| **Goal** | 7 | 0.016166282 |
| **Miss** | 51 | 0.11778291 |
| **Post** | 2 | 0.004618938 |
| **Save** | 212 | 0.48960739 |
| **Rebound** | **Blocked** | 27 | 0.369863014 |
| **Goal** | 2 | 0.02739726 |
| **Miss** | 10 | 0.136986301 |
| **Save** | 34 | 0.465753425 |
| **Redirection** | **Blocked** | 2 | 0.105263158 |
| **Goal** | 2 | 0.105263158 |
| **Miss** | 10 | 0.526315789 |
| **Save** | 5 | 0.263157895 |
| **Rush** | **Blocked** | 155 | 0.242946708 |
| **Deflection** | 1 | 0.001567398 |
| **Goal** | 12 | 0.018808777 |
| **Miss** | 81 | 0.126959248 |
| **Post** | 3 | 0.004702194 |
| **Save** | 386 | 0.605015674 |
| **Transition** | **Blocked** | 50 | 0.354609929 |
| **Deflection** | 1 | 0.007092199 |
| **Goal** | 6 | 0.042553191 |
| **Miss** | 31 | 0.219858156 |
| **Post** | 2 | 0.014184397 |
| **Save** | 51 | 0.361702128 |

Table 2.7: Shot Quality and Shot Type vs Shot Outcome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shot Quality** | **Shot Type** | **Shot Outcome** | **Totals** | **Percentages** |
| **Clean** | **Backhand** | **Blocked** | 11 | 0.15942029 |
| **Goal** | 3 | 0.043478261 |
| **Miss** | 10 | 0.144927536 |
| **Save** | 45 | 0.652173913 |
| **One-Timer** | **Miss** | 1 | 0.333333333 |
| **Save** | 2 | 0.666666667 |
| **Slap** | **Blocked** | 135 | 0.460750853 |
| **Deflection** | 10 | 0.034129693 |
| **Goal** | 8 | 0.027303754 |
| **Miss** | 45 | 0.153583618 |
| **Save** | 95 | 0.324232082 |
| **Snap** | **Blocked** | 200 | 0.421052632 |
| **Deflection** | 15 | 0.031578947 |
| **Goal** | 3 | 0.006315789 |
| **Miss** | 68 | 0.143157895 |
| **Post** | 3 | 0.006315789 |
| **Save** | 186 | 0.391578947 |
| **Wrap** | **Blocked** | 4 | 0.125 |
| **Miss** | 2 | 0.0625 |
| **Save** | 26 | 0.8125 |
| **Wrist** | **Blocked** | 286 | 0.358845671 |
| **Deflection** | 33 | 0.04140527 |
| **Goal** | 17 | 0.021329987 |
| **Miss** | 128 | 0.160602258 |
| **Post** | 2 | 0.00250941 |
| **Save** | 331 | 0.415307403 |
| **Rebound** | **Backhand** | **Blocked** | 12 | 0.184615385 |
| **Goal** | 8 | 0.123076923 |
| **Miss** | 5 | 0.076923077 |
| **Post** | 1 | 0.015384615 |
| **Save** | 39 | 0.6 |
| **One-Timer** | **Blocked** | 1 | 0.25 |
| **Save** | 3 | 0.75 |
| **Slap** | **Blocked** | 19 | 0.452380952 |
| **Goal** | 1 | 0.023809524 |
| **Miss** | 8 | 0.19047619 |
| **Save** | 14 | 0.333333333 |
| **Snap** | **Blocked** | 26 | 0.185714286 |
| **Deflection** | 1 | 0.007142857 |
| **Goal** | 18 | 0.128571429 |
| **Miss** | 19 | 0.135714286 |
| **Post** | 1 | 0.007142857 |
| **Save** | 75 | 0.535714286 |
| **Wrap** | **Blocked** | 4 | 0.266666667 |
| **Miss** | 2 | 0.133333333 |
| **Save** | 9 | 0.6 |
| **Wrist** | **Blocked** | 27 | 0.223140496 |
| **Goal** | 15 | 0.123966942 |
| **Miss** | 15 | 0.123966942 |
| **Save** | 64 | 0.52892562 |
| **Redirection** | **Backhand** | **Blocked** | 2 | 0.166666667 |
| **Goal** | 1 | 0.083333333 |
| **Miss** | 5 | 0.416666667 |
| **Save** | 4 | 0.333333333 |
| **One-Timer** | **Blocked** | 3 | 0.081081081 |
| **Goal** | 5 | 0.135135135 |
| **Miss** | 15 | 0.405405405 |
| **Save** | 14 | 0.378378378 |
| **Slap** | **Save** | 1 | 1 |
| **Snap** | **Goal** | 2 | 0.153846154 |
| **Miss** | 6 | 0.461538462 |
| **Save** | 5 | 0.384615385 |
| **Wrist** | **Blocked** | 1 | 0.071428571 |
| **Goal** | 5 | 0.357142857 |
| **Miss** | 2 | 0.142857143 |
| **Save** | 6 | 0.428571429 |
| **Rush** | **Backhand** | **Blocked** | 11 | 0.1 |
| **Goal** | 4 | 0.036363636 |
| **Miss** | 16 | 0.145454545 |
| **Save** | 79 | 0.718181818 |
| **One-Timer** | **Blocked** | 5 | 0.277777778 |
| **Goal** | 4 | 0.222222222 |
| **Miss** | 4 | 0.222222222 |
| **Save** | 5 | 0.277777778 |
| **Slap** | **Blocked** | 21 | 0.2625 |
| **Goal** | 2 | 0.025 |
| **Miss** | 17 | 0.2125 |
| **Post** | 1 | 0.0125 |
| **Save** | 39 | 0.4875 |
| **Snap** | **Blocked** | 79 | 0.194581281 |
| **Deflection** | 1 | 0.002463054 |
| **Goal** | 11 | 0.027093596 |
| **Miss** | 58 | 0.142857143 |
| **Post** | 4 | 0.009852217 |
| **Save** | 253 | 0.623152709 |
| **Wrap** | **Blocked** | 2 | 0.2 |
| **Miss** | 1 | 0.1 |
| **Save** | 7 | 0.7 |
| **Wrist** | **Blocked** | 162 | 0.22100955 |
| **Deflection** | 1 | 0.001364256 |
| **Goal** | 24 | 0.032742156 |
| **Miss** | 101 | 0.137789905 |
| **Post** | 2 | 0.002728513 |
| **Save** | 443 | 0.604365621 |
| **Transition** | **Backhand** | **Blocked** | 2 | 0.125 |
| **Goal** | 1 | 0.0625 |
| **Miss** | 4 | 0.25 |
| **Save** | 9 | 0.5625 |
| **One-Timer** | **Blocked** | 48 | 0.206896552 |
| **Deflection** | 4 | 0.017241379 |
| **Goal** | 17 | 0.073275862 |
| **Miss** | 58 | 0.25 |
| **Post** | 3 | 0.012931034 |
| **Save** | 102 | 0.439655172 |
| **Slap** | **Blocked** | 25 | 0.5 |
| **Deflection** | 1 | 0.02 |
| **Goal** | 1 | 0.02 |
| **Miss** | 10 | 0.2 |
| **Save** | 13 | 0.26 |
| **Snap** | **Blocked** | 23 | 0.310810811 |
| **Deflection** | 4 | 0.054054054 |
| **Goal** | 2 | 0.027027027 |
| **Miss** | 7 | 0.094594595 |
| **Post** | 1 | 0.013513514 |
| **Save** | 37 | 0.5 |
| **Wrap** | **Blocked** | 1 | 0.333333333 |
| **Save** | 2 | 0.666666667 |
| **Wrist** | **Blocked** | 48 | 0.366412214 |
| **Deflection** | 1 | 0.007633588 |
| **Goal** | 6 | 0.045801527 |
| **Miss** | 21 | 0.160305344 |
| **Save** | 55 | 0.419847328 |

Table 3.1: Shot Danger vs Shot Outcome Binary

|  |  |  |  |
| --- | --- | --- | --- |
| **Shot Danger** | **Shot Outcome** | **Totals** | **Percentages** |
| **High** | **Goal** | 103 | 9.88% |
| **No Goal** | 939 | 90.12% |
| **Low** | **Goal** | 26 | 1.58% |
| **No Goal** | 1624 | 98.42% |
| **Mid** | **Goal** | 29 | 2.22% |
| **No Goal** | 1275 | 97.78% |

Table 3.2: Shot Type vs Shot Outcome Binary

|  |  |  |  |
| --- | --- | --- | --- |
| **Shot Type** | **Shot Outcome** | **Totals** | **Percentages** |
| **Backhand** | **Goal** | 17 | 6.25% |
| **No Goal** | 255 | 93.75% |
| **One-Timer** | **Goal** | 26 | 8.84% |
| **No Goal** | 268 | 91.16% |
| **Slap** | **Goal** | 12 | 2.58% |
| **No Goal** | 454 | 97.42% |
| **Snap** | **Goal** | 36 | 3.25% |
| **No Goal** | 1072 | 96.75% |
| **Wrap** | **No Goal** | 60 | 100.00% |
| **Wrist** | **Goal** | 67 | 3.73% |
| **No Goal** | 1729 | 96.27% |

Table 3.3: Shot Quality vs Shot Outcome Binary

|  |  |  |  |
| --- | --- | --- | --- |
| **Shot Quality** | **Shot Outcome** | **Totals** | **Percentages** |
| **Clean** | **Goal** | 31 | 1.86% |
| **No Goal** | 1638 | 98.14% |
| **Rebound** | **Goal** | 42 | 10.85% |
| **No Goal** | 345 | 89.15% |
| **Redirection** | **Goal** | 13 | 16.88% |
| **No Goal** | 64 | 83.12% |
| **Rush** | **Goal** | 45 | 3.32% |
| **No Goal** | 1312 | 96.68% |
| **Transition** | **Goal** | 27 | 5.34% |
| **No Goal** | 479 | 94.66% |

Table 3.4: Number of Shot Assists vs Shot Outcome Binary

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Shot Assists** | **Shot Outcome** | **Totals** | **Percentages** |
| **0.0** | **Goal** | 79 | 3.79% |
| **No Goal** | 2003 | 96.21% |
| **1.0** | **Goal** | 47 | 3.80% |
| **No Goal** | 1190 | 96.20% |
| **2.0** | **Goal** | 19 | 4.51% |
| **No Goal** | 402 | 95.49% |
| **3.0** | **Goal** | 4 | 3.54% |
| **No Goal** | 109 | 96.46% |
| **4.0** | **Goal** | 1 | 1.54% |
| **No Goal** | 64 | 98.46% |
| **5.0** | **Goal** | 4 | 12.12% |
| **No Goal** | 29 | 87.88% |
| **6.0** | **No Goal** | 20 | 100.00% |
| **7.0** | **Goal** | 1 | 9.09% |
| **No Goal** | 10 | 90.91% |
| **8.0** | **Goal** | 1 | 16.67% |
| **No Goal** | 5 | 83.33% |
| **9.0** | **No Goal** | 2 | 100.00% |
| **10.0** | **Goal** | 1 | 50.00% |
| **No Goal** | 1 | 50.00% |
| **11.0** | **No Goal** | 1 | 100.00% |
| **13.0** | **No Goal** | 1 | 100.00% |
| **14.0** | **No Goal** | 1 | 100.00% |

Table 3.5: Shot Danger and Shot Type vs Shot Outcome Binary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shot Danger** | **Shot Type** | **Shot Outcome** | **Totals** | **Percentages** |
| **High** | **Backhand** | **Goal** | 16 | 7.14% |
| **No Goal** | 208 | 92.86% |
| **One-Timer** | **Goal** | 18 | 13.85% |
| **No Goal** | 112 | 86.15% |
| **Slap** | **No Goal** | 7 | 100.00% |
| **Snap** | **Goal** | 27 | 10.67% |
| **No Goal** | 226 | 89.33% |
| **Wrap** | **No Goal** | 60 | 100.00% |
| **Wrist** | **Goal** | 42 | 11.41% |
| **No Goal** | 326 | 88.59% |
| **Low** | **Backhand** | **No Goal** | 4 | 100.00% |
| **One-Timer** | **Goal** | 2 | 2.35% |
| **No Goal** | 83 | 97.65% |
| **Slap** | **Goal** | 7 | 1.92% |
| **No Goal** | 357 | 98.08% |
| **Snap** | **Goal** | 2 | 0.45% |
| **No Goal** | 438 | 99.55% |
| **Wrist** | **Goal** | 15 | 1.98% |
| **No Goal** | 742 | 98.02% |
| **Mid** | **Backhand** | **Goal** | 1 | 2.27% |
| **No Goal** | 43 | 97.73% |
| **One-Timer** | **Goal** | 6 | 7.59% |
| **No Goal** | 73 | 92.41% |
| **Slap** | **Goal** | 5 | 5.26% |
| **No Goal** | 90 | 94.74% |
| **Snap** | **Goal** | 7 | 1.69% |
| **No Goal** | 408 | 98.31% |
| **Wrist** | **Goal** | 10 | 1.49% |
| **No Goal** | 661 | 98.51% |

Table 3.6: Shot Danger and Shot Quality vs Shot Outcome Binary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shot Danger** | **Shot Quality** | **Shot Outcome** | **Totals** | **Percentages** |
| **High** | **Clean** | **Goal** | 7 | 3.26% |
| **No Goal** | 208 | 96.74% |
| **Rebound** | **Goal** | 39 | 15.00% |
| **No Goal** | 221 | 85.00% |
| **Redirection** | **Goal** | 10 | 17.86% |
| **No Goal** | 46 | 82.14% |
| **Rush** | **Goal** | 29 | 9.21% |
| **No Goal** | 286 | 90.79% |
| **Transition** | **Goal** | 18 | 9.18% |
| **No Goal** | 178 | 90.82% |
| **Low** | **Clean** | **Goal** | 17 | 1.67% |
| **No Goal** | 1004 | 98.33% |
| **Rebound** | **Goal** | 1 | 1.85% |
| **No Goal** | 53 | 98.15% |
| **Redirection** | **Goal** | 1 | 50.00% |
| **No Goal** | 1 | 50.00% |
| **Rush** | **Goal** | 4 | 0.99% |
| **No Goal** | 400 | 99.01% |
| **Transition** | **Goal** | 3 | 1.78% |
| **No Goal** | 166 | 98.22% |
| **Mid** | **Clean** | **Goal** | 7 | 1.62% |
| **No Goal** | 426 | 98.38% |
| **Rebound** | **Goal** | 2 | 2.74% |
| **No Goal** | 71 | 97.26% |
| **Redirection** | **Goal** | 2 | 10.53% |
| **No Goal** | 17 | 89.47% |
| **Rush** | **Goal** | 12 | 1.88% |
| **No Goal** | 626 | 98.12% |
| **Transition** | **Goal** | 6 | 4.26% |
| **No Goal** | 135 | 95.74% |

Table 3.7: Shot Quality and Shot Type vs Shot Outcome Binary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shot Quality** | **Shot Type** | **Shot Outcome** | **Totals** | **Percentages** |
| **Clean** | **Backhand** | **Goal** | 3 | 4.35% |
| **No Goal** | 66 | 95.65% |
| **One-Timer** | **No Goal** | 3 | 100.00% |
| **Slap** | **Goal** | 8 | 2.73% |
| **No Goal** | 285 | 97.27% |
| **Snap** | **Goal** | 3 | 0.63% |
| **No Goal** | 472 | 99.37% |
| **Wrap** | **No Goal** | 32 | 100.00% |
| **Wrist** | **Goal** | 17 | 2.13% |
| **No Goal** | 780 | 97.87% |
| **Rebound** | **Backhand** | **Goal** | 8 | 12.31% |
| **No Goal** | 57 | 87.69% |
| **One-Timer** | **No Goal** | 4 | 100.00% |
| **Slap** | **Goal** | 1 | 2.38% |
| **No Goal** | 41 | 97.62% |
| **Snap** | **Goal** | 18 | 12.86% |
| **No Goal** | 122 | 87.14% |
| **Wrap** | **No Goal** | 15 | 100.00% |
| **Wrist** | **Goal** | 15 | 12.40% |
| **No Goal** | 106 | 87.60% |
| **Redirection** | **Backhand** | **Goal** | 1 | 8.33% |
| **No Goal** | 11 | 91.67% |
| **One-Timer** | **Goal** | 5 | 13.51% |
| **No Goal** | 32 | 86.49% |
| **Slap** | **No Goal** | 1 | 100.00% |
| **Snap** | **Goal** | 2 | 15.38% |
| **No Goal** | 11 | 84.62% |
| **Wrist** | **Goal** | 5 | 35.71% |
| **No Goal** | 9 | 64.29% |
| **Rush** | **Backhand** | **Goal** | 4 | 3.64% |
| **No Goal** | 106 | 96.36% |
| **One-Timer** | **Goal** | 4 | 22.22% |
| **No Goal** | 14 | 77.78% |
| **Slap** | **Goal** | 2 | 2.50% |
| **No Goal** | 78 | 97.50% |
| **Snap** | **Goal** | 11 | 2.71% |
| **No Goal** | 395 | 97.29% |
| **Wrap** | **No Goal** | 10 | 100.00% |
| **Wrist** | **Goal** | 24 | 3.27% |
| **No Goal** | 709 | 96.73% |
| **Transition** | **Backhand** | **Goal** | 1 | 6.25% |
| **No Goal** | 15 | 93.75% |
| **One-Timer** | **Goal** | 17 | 7.33% |
| **No Goal** | 215 | 92.67% |
| **Slap** | **Goal** | 1 | 2.00% |
| **No Goal** | 49 | 98.00% |
| **Snap** | **Goal** | 2 | 2.70% |
| **No Goal** | 72 | 97.30% |
| **Wrap** | **No Goal** | 3 | 100.00% |
| **Wrist** | **Goal** | 6 | 4.58% |
| **No Goal** | 125 | 95.42% |