Allegheny College
Computer And Information Science Department

Senior Thesis

NFL Draft Assist

by

Vital Joseph II

ALLEGHENY COLLEGE

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

Project Supervisor: **Emily Graber**
Co-Supervisor: **Janyl Jumadinova**

Abstract

NFL Draft Assist is a decision support tool designed to assist in NFL draft process decisions and analysis. It uses predictive modeling to evaluate college players entering the National Football League (NFL) draft and predict which ones will succeed at the next level. That evaluation process incorporates a variety of metrics like, college production, physical body measurements, combine statistics and statistical techniques like linear models, monte carlo cross validation score thresholding, and clustering. What was found is that those metrics don't just add up in a straightforward way to determine a player's potential. It takes a combination of statistical analysis, manual feature engineering and a deep understanding of what really correlates to success in the NFL to get a accurate read on a player's chances of success. That's where the real value of this project lies. The model that performed the best, Linear Threshold Weighted, achieved an 80% accuracy rate in classifying current NFL players by how successful they were likely to be before the NFL draft and compared that to how well they are doing in the NFL right now. This accuracy shows that a data-driven approach might be useful to the scouting process. What it also reveals are the problems in the way teams currently draft players. You see biases toward athleticism over actual performance, and the way certain positions and schools get overvalued or undervalued. That's where this tool comes in to play. NFL Draft Assist is a web application built with Flask that lets users visualize players predicted draft outcomes for upcoming NFL draft, as well visualizations to see how well the models perform on current NFL players. This tool is a resource for scouts, media professionals and analysts to make more informed decisions and talking points regarding NFL draft prospects. By integrating predictive analytics, like NFL Draft Assist, into the traditional scouting process, teams can understand potential bias and make informed draft decisions. This is cutting edge in an industry where data is increasingly driving decisions about NFL talent.

Table of contents

1 Introduction **6**
- 1.1 Background . . . 6
- 1.2 Problem . . . 7
- 1.3 Solution . . . 8
- 1.4 Goals . . . 9
- 1.5 Current State of the Art . . . 11
- 1.6 Qualifications . . . 12
- 1.7 Ethical Implications . . . 13

2 Related work **14**
- 2.1 What is the NFL Draft and How Important is it? . . . 14
- 2.2 What are key predictors of success for NFL draft prospects? . . . 15
- 2.3 Normalization in predictive modeling . . . 16
  - 2.3.1 Importance using normalization for NFL performance metrics . . . 16
  - 2.3.2 Normalization techniques used or considered when modeling 17
  - 2.3.3 Benefits of using normalization techniques in predictive modeling . . . 17
- 2.4 Overfiiting . . . 18
- 2.5 Predictive modeling and Validation techniques . . . 18
- 2.6 Feature Engineering: Balancing Bias and Predictive Value . . . 19
  - 2.6.1 The Value of Feature Engineering . . . 19
  - 2.6.2 Bias in Feature Engineering . . . 20
- 2.7 Challenges in Predicting NFL Player Success . . . 20
  - 2.7.1 Scheme Fit and System Variability . . . 20
  - 2.7.2 Player Peaks, Declines, and Developmental Curves . . . 21
  - 2.7.3 External Factors Impacting Success . . . 21
  - 2.7.4 Addressing These Challenges in the NFL Draft Assist Tool 21

3 Method of approach **22**
- 3.1 Introductions to Methods . . . 22
- 3.2 Data Collection and Preprocessing . . . 23
  - 3.2.1 NFL Performance Data (2019–2022) . . . 23
  - 3.2.2 College Statistics (2019–2022) . . . 25
  - 3.2.3 NFL Combine Measurements (2019–2022) . . . 26
- 3.3 Data Cleaning and Merging . . . 28
- 3.4 Modeling Approach . . . 30
  - 3.4.1 Monte Carlo Cross Validation . . . 31
  - 3.4.2 Train Test Split and Parameters . . . 31
  - 3.4.3 Models . . . 32
  - 3.4.4 Feature Engineering . . . 33
  - 3.4.5 Categorization Methods . . . 34
- 3.5 Web Application Approach . . . 37
  - 3.5.1 Technical Implementation . . . 37

3.5.2 UI/UX For modeled data 38
3.5.3 Interactive UI/UX for New Data 41

4 Experiments **42**
4.1 Experimental Design 43
4.1.1 Data Preparation and Selection 43
4.1.2 Training and Testing 43
4.1.3 Prediction and Evaluation Metrics 43
4.2 Evaluation 44
4.2.1 Classification Report Comparison 44
4.2.2 Evaluation of the Linear Threshold (LT) Model 45
4.2.3 Evaluation of the Linear Threshold Weighted (LTW) Model 47
4.2.4 Evaluation of the Linear Cluster (LC) Model 50
4.2.5 Evaluation of the Linear Cluster Weighted (LCW) Model 52
4.3 Threats to Validity/Limitations 55
4.4 Summary of Results 56
4.4.1 Results 56

5 Conclusion **58**
5.1 Future Work 59
5.2 Future Ethical Implications and Recommendations 60
5.2.1 Bias and Fairness 60
5.2.2 Privacy and Data Usage 60
5.2.3 Transparency and Interpretability 60
5.2.4 Impact on Player Careers 60
5.2.5 Recommendations for Responsible Use 60

List of Figures

1 Model Overview 23
2 Table 44
3 lt1 46
4 lt2 46
5 lt3 47
6 ltw1 48
7 ltw2 49
8 ltw3 49
9 lc1 51
10 lc2 51
11 lc3 52
12 lcw1 53
13 lcw2 54
14 lcw3 54

List of Tables

1 Introduction

1.1 Background

The National Football League (NFL) draft is a pivotal process for all 32 NFL teams. It plays a important role in shaping teams' futures by providing a structured method for acquiring young talent directly from college football, serving as a critical tool for team-building. The draft helps ensure competitive balance in the league by giving lower performing teams the opportunity to select top prospects, encouraging talent distribution.

The history of the NFL draft, where teams find promising young talent is a story of how that process has become much more sophisticated over the decades. In the early days, teams pretty much relied on who they knew, what they'd heard and what little game footage they had. Evaluations were informal. Coaches from colleges would recommend players, and teams would take their word, rather than doing their own thorough research. That changed as the NFL got more competitive. Teams started building dedicated scouting departments. That led to using more principle approaches to assess talent coming in. The 1960s and 1970s saw the emergence of legendary scouts like Gil Brandt. He revolutionized scouting by using psychological tests, numbers and a systematic way of looking at players. That laid the groundwork for the scouting processes we see today. His idea that data and structure could improve draft picks, rather than just relying on gut feelings was revolutionary at the time[Iannuzzi [11]][1]. The creation of the NFL Scouting Combine in 1982 took that process and made it official. The NFL scouting combine standardized athletic testing and medical evaluations, giving teams a much clearer picture of their potential draft picks[2]. Despite the improvements made in the 80's, advances in scouting had yet to eliminate biases. Even with more detailed reports, teams still led heavily on the eye test and traditional football instincts, that often lead to problems in identifying talent. That changed with the rise of analytics in the 2000s and 2010s. Data and numbers started to quantify performance and cut down on biases. But even now, some scouts resist that shift. That resistance and the debate over where traditional expertise ends and statistical analysis begins, continues to shape the draft process today[29].

For years NFL teams relied almost entirely on their scouting departments and traditional methods to evaluate talent. As I mentioned earlier, that reliance on scouting and old-school evaluation remained largely unchanged until the NFL Combine came along and standardized athletic testing. But even with that standardization, the interpretation of combine results was mostly subjective. Teams focused on isolated measurable traits like 40-yard dash times, bench press reps and vertical jumps without fully considering how those traits translated to on-field success.

As player salaries increased and the game evolved, teams started looking for ways to evaluate talent more efficiently. Over the past two decades, some teams have started to slowly incorporate more data-driven methods into their draft process, using statistical analysis to supplement traditional scouting. While

there have been examples of teams finding value through analytics, data-driven scouting is still not league-wide. Many teams are just scratching the surface of predictive modeling and there's still a lot of room for growth in how data and machine learning can be used to support traditional scouting.

Some teams have established analytics departments to support draft strategy[4], but data is still used as a supplement rather than a foundation of the evaluation process. The collaboration between scouts and analysts is growing but a fully integrated data-driven approach to the NFL Draft is still a work in progress. Much of the NFL's talent evaluation process still relies heavily on subjective assessments rather than the potential of available predictive modeling and efficiency metrics.

Drafting also serves as a cost efficient way for teams to fill their rosters, as rookie contracts are more affordable than veteran free agent deals, making it a key factor in salary cap management. A well structured salary cap is one of the most important factors in the NFL draft process. The NFL's hard salary cap, where teams have a very real limit on what they can spend on player contracts each season, means teams have to be smart about how they manage their finances. Effective salary cap management is key to maintaining that financial flexibility and getting the most out of talent acquisitions. Poor management of the salary cap can result in significant "dead money" from released players, limit a team's ability to re-sign key contributors, and force undesirable roster decisions. The draft plays a huge role in this balancing these team building actions. Rookie contracts, especially for non-first-round picks, are much more affordable than those for veteran free agents. That means teams that draft well can build a strong foundation of young talent on cost-controlled contracts. That gives teams the resources to focus on other areas, like keeping their star players or signing high-impact free agents. By incorporating data and analytics to their draft strategy, teams can find undervalued prospects who deliver high performance at a lower cost. That optimizes salary cap usage and keeps them competitive for the long haul.

Furthermore, teams invest heavily in scouting and analytics to avoid drafting "busts," as draft success is essential for long-term success. These "busts" (players who don't meet expectations) can blow up a team's plans, waste resources and disrupt their trajectory for years. Ultimately, the NFL Draft is a cornerstone of the league, player development, and the overall long-term success for NFL teams.

1.2 Problem

Scouts are a big part of the NFL draft process. Their expertise in evaluating physical skills, on-field intelligence and character traits of NFL draft prospects is the foundation of draft decisions. Scouts watch games, watch film, interview and evaluate on field performance and intangible traits of prospects. Their grades impact how prospects are ranked and that is very significant in a competitive league with big money on the line. But even the most experienced scouts have limitations, cognitive biases and difficulty in analyzing large amounts of data

from the college prospects. These same limitations and biases applies to media professionals who cover the NFL draft on a yearly basis. Media professionals and analysts have a big impact on the public and team perception of NFL draft prospects. Sports writers, TV analysts and draft experts produce mock drafts, player rankings and scouting reports that influence fan expectations and sometimes even team decisions. Their opinions, based on film study, talking to scouts and looking at statistics, are part of the bigger draft conversation. But media narratives can over-hype or undervalue prospects based on incomplete data or trends and create biases in public opinion. And with the draft getting so much coverage, there's another layer of complexity as media can boost or hurt a prospect's draft stock based on off-field rumors or other subjective factors. I plan to addresses these limitations from scouts and media professionals by providing a data-driven perspective to complement the scout's qualitative grades, resulting in a balanced approach to player evaluation.

Traditional NFL draft scouting evaluation processes rely heavily on subjective assessments of player potential. Scouts watch prospects in live games, watch game film, interview coaches, teammates and the players themselves. They look at physical traits like size, speed and strength and intangibles like leadership, work ethic and football IQ. Players also participate in pre-draft events like the NFL Combine and Pro Days where they show off their athletic ability through drills and testing. This process has predicted many great NFL players but it's also prone to many biases. One bias is recency bias where scouts overvalue recent performance instead of looking at a player's entire college career. Confirmation bias can also play a role where scouts look for evidence that supports their pre-conceived opinion of a player instead of objectively evaluating all the data. Positional bias can also occur where certain positions get more attention and draft value than others. Intangibles like leadership and character are also hard to quantify so scouts will interpret them differently. These biases can lead to highly skilled players getting overlooked and less talented players getting overvalued based on anecdotal evidence instead of data-driven analysis.

1.3 Solution

To address the limitations of traditional scouting and media evaluations I created a data driven predictive modeling tool called NFL Draft Assist. This tool uses statistical modeling to objectively evaluate draft prospects based on college production, athletic testing and physical measurements. By combining historical data with predictive analytics the solution aims to help make smarter and less biased draft decisions.

In the tool, linear regression is widely used during the modeling process. Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It is widely used for predictive analysis and can be particularly useful in understanding the impact of various factors on an outcome, which is why it plays a crucial role in the my tool. Linear regression based modeling is fundamental to predictive analytics in sports, especially when it comes to how various pre-draft metrics contribute to

a player's success in the NFL. By using historical data from past drafts, linear regression helps figure out which variables like, college production, combine performance, physical measurements, matter most for NFL success. It also allows for the assessment of the relationships of player variables, making more objective evaluations of a player. Linear modeling to predict draft success helps to mitigate the biases that come from purely subjective scouting reports and gives a more balanced view of the draft prospects.

Ultimately, this study aims to present four predictive linear models to predict how NFL draft prospects will perform in the NFL using a mix of different key metrics, as well as different feature values for these metrics to determine how wll they correlate with actual success in the NFL. The goal is to help scouts and analysts make better draft decisions based on data.

1.4 Goals

The ultimate goal is to revolutionize the draft process for NFL teams by providing a decision support tool that enhances player selection strategies and overall roster building. This tool aims to help teams identify undervalued prospects while avoiding the overvaluation of prospects who might become busts. By improving the accuracy of projecting star prospects from the draft, it minimizes the risk of underperforming picks, ultimately saving teams valuable time and resources while maximizing their draft capital. A key aspect I want my tool to be able to identify, is undervalued players that might not show promise from your traditional metrics but overall possess the attributes needed to succeed at a NFL level. And vice versa, I dont want this tool to just look at statistics as a direct correlation to success in the NFL, because it is not. By integrating a wide range of metrics, like college statistics, combine metrics, conference strength, and scout evaluations, my tool should generate a comprehensive evaluation for every player, hopefully unveiling hidden gems and overvalued prospects or future busts. Another motivation for this project is to maximize my tools accessibility and user friendliness by appealing to NFL teams, scouts, and media professionals that cover the NFL Draft yearly. My plan is to provide clear, interpretable outputs that simplify complex data into more football-based knowledge. This will help scouts and media access insights more efficiently and effectively. By implementing intuitive dashboards and visualizations, this tool can provide key attributes and insights to prospects for the targeted audience to analyze and use as a support tool for making final decisions and judgments on different prospects. By demonstrating the importance of machine learning, statistical analysis, and the use of a wide range of different metrics in improving player evaluations, this tool hopes to inspire further innovations for sports analytics and predictive modeling. Furthermore, my project is aimed to bring raw data and decision making together, allowing teams to make more informed draft choices. A number of different metrics are used in the modelling process to generate a comprehensive evaluation of each player's potential success in the NFL. Some of these metrics include NFL combine data, college career statistics, scout evaluations, college conference strength, and more. By looking at all these

different metrics we get a more complete picture of a player's potential, and can possibly reduce the uncertainty and risk of high stakes drafting.

Another key goal behind this tool is the need to address historical draft mistakes that have cost teams significantly, in terms of time, money, and player development. Teams have wasted time, money and player development on players that are overvalued based on speed, size and agility. Players who didn't shine in combine drills or came from smaller schools have gone on to succeed in the NFL in ways no one anticipated. This tool will help teams avoid those mistakes by looking at the whole picture more comprehensively by providing insights using historical performance data, player traits and advanced metrics. As teams increasingly turn to data science and predictive analytics to get ahead of the game, the use of predictive models in the draft process becomes more and more important. Too often, scouting and analytics are seen as separate processes. My project aims to change that. Predictive modeling can give scouts an extra layer of analysis to work with, helping teams identify real talent more accurately and reduce the biases that come with traditional scouting. That expertise and those numbers can work together to make teams better at what they do.

Another big part of this project is identifying "sleeper" or prospects who may not have the most impressive college statistics or combine numbers but can still play at the NFL level. Players from non-Power 5 conferences or who may have faced lesser competition might be overlooked by traditional scouts even though they have a unique skill or have shown significant growth during their college career. Through my model I want to highlight these players so teams aren't missing out on potential gems because they don't fit the mold of a top-ranked draft prospect. It is also important not to over-hype these prospects as well being that, the level of competition is a lot lesser, but if they show elite production, it suggests there is legitimate talent worth taking a calculated risk on.

Furthermore, the financial impact of making the wrong draft decision is another motivating factor. Those picks can leave a team with "dead money" on the salary cap and roster spots that could have been filled by players who actually produce. With the salary cap constraints NFL teams face, making better draft decisions can have a lasting impact on your financial flexibility and your chances of success.

That's where NFL Draft Assist proves valuable. By improving the accuracy of player projections, teams can use their resources more effectively. They can keep their salary cap flexible and get the most out of their draft capital. By combining subjective scouting reports with objective data analysis, we can give teams the insights they need to make better decisions. With that kind of information at their fingertips, teams can optimize their draft strategies, reduce the uncertainty that comes with evaluating players and make better choices.

Ultimately, this project aims to create a more balanced, more comprehensive approach to evaluating NFL prospects. One that teams, scouts and media can all use to improve the draft process.

1.5 Current State of the Art

The current state of art in NFL Draft scouting tools and predictive modeling has advanced significantly, leveraging advanced analytics, machine learning, and scouting insights to improve decision making more and more each year. There are a lot of NFL mock draft databases and platforms that use different scout analyses and opinions to create big boards, full mock drafts for all 32 NFL teams, and prospect rankings that reflect the general mainstream media's opinion of the college prospects per year. Some mainstream platforms that do this consist of, The NFL Mock Draft Database, PFF (Pro Football Focus), MockOut, and The Draft Network, this is just to name a few though. Advanced metrics is also an area that has significantly grown in the past few years that provide very detail oriented information about on the field metrics and statistics. Some of these metrics include route success rate for receivers, man to man and zone coverage grade for defensive backs, top speed in mph when carrying the ball, and many other metrics that are very telling and provide a lot of useful insight for scouts, teams, and media to understand players overall skill level. Your basic statistics dont always give the full picture as to how good or valuable you are as a player, there is a lot of moving parts that goes on in between the whistles that doesn't get accounted for but is very important to the play.

Predictive modeling has also advanced in the past years. Different teams and researcher have used different types of machine learning like decision trees and random forests modeling. These models are usually used to evaluate different metrics for prospects like combine statistics or college statistics. A decision tree model can make a prediction based on splitting data on metrics and choosing higher or lower foreach step of the tree until it reaches a category destination. Decision trees can be very useful for a number of reasons, one is that they can handle missing data, which is important because not very often will you be guaranteed data completeness for your datasets, especially working with large datasets. Unlike other models, you dont need really large datasets to get the most accurate results. Certain analyses wont have a large datasets/s you can work with all the time. All sorts of different databases can be used in decision trees as well. Decision trees can handle both numerical and categorical data and dont need scaling or normalization. Some weaknesses of decision trees include bias toward more dominant data, the ability of decision trees to determine the most important features by splitting the dataset on the variables that provide the highest information gain can cause unintentional bias towards data that just appears more. Decision trees can also be prone to overfitting and can grow overly complex and fit noise in the training data, leading to poor generalization on unseen data. Next Gen statistics use decision tree modeling for their tool, they take in combine metrics and and body measurements to produce a size and athletic score and also use college statistics to produce a production score. The three scores then are used to produce a final draft score[**?**].

Random Forests is another model that uses the same type of metrics as decision tree modeling does and is similar in structure. Random forest consists of many single decision trees that uses subsets of the data and the features. It

averages the prediction results of each of these different trees to improve accuracy and generalization. Currently, no widely recognized NFL draft prediction tools explicitly advertise the use of random forests in their algorithms. Neural Networks are another machine learning model that acts similar to that of the human brains cognitive learning structure. They are designed to recognize patterns and relationships in data, making them a useful tool to use for complex non-linear interactions and correlations. SnapLogic have used a neural network architecture with multiple layers, including input and hidden layers, to analyze player data for predicting NFL Draft outcomes[8].

1.6 Qualifications

The project also allows me to apply my data science skills in a real-world situation, providing the opportunity to address complex problems using data-driven insights, which aligns perfectly with my passion for data science, sports analytics and predictive modeling. I am well equipped to build and refine models that use different types of data, ensuring they are both accurate and scalable. My expertise in programming languages like R and Python allow me to perform advanced data manipulation, statistical analysis, and machine learning implementations. Furthermore, I have some skills in creating dynamic data visualizations and user friendly dashboards in different tools and libraries like Tableau, gplot2, and Matplotlib. Both my technical expertise and application skills motivate me to design a comprehensive decision support tool that meets the needs of NFL teams, scouts, and analysts, bridging the gap between raw data and strategic decision making.

My passion to improve the NFL draft process stems from my deep understanding of what it takes to succeed at a high level, thanks to my personal background in football. I have been playing football for my whole life since I was 8 up until playing football here at Allegheny College. I have been a three year starter here at Allegheny at the defensive back position, finishing up my last year by receiving ALL-PAC (Presidential Athletic Conference) honors both academically and athletically. As the years goes on and on you start to to pick up what different attributes, whether that be intangible or physical, make a good football player. Personally I have played the positions, of running back, wide receiver, and defensive back throughout my career, and have extensive knowledge on the skills associated with those positions, as well as the quarterback position. The story is not always told in the statistics, so it is important to be able to understand position specific information as much as possible. Ultimately, my point is that my background knowledge associated with the basis around this project is top tier and I plan to leverage all of what I know for the betterment of the tool. This is supported by my solid grasp of the metrics and statistics involved in the draft process, allowing me to integrate these factors effectively.

1.7 Ethical Implications

The ethical considerations for implementing data-driven programs spread across several areas, like data collection, algorithmic bias, privacy, and social equity. These areas are important because data-driven programs have heavy influence on decision making processes and outcomes that can have a huge impact on individuals, communities and organizations. When it comes to statistical data, there is not much bias if any at all for same statistics and metrics but, categorical data does tend to have some bias. Categorical data that I have integrated like conference strength, can hold some bias, such as undervaluing or over valuing certain conferences. I limited bias as much as possible by using a ranking system based on an Elo rating of each conference's combined winning percentage, including their record against out-of-conference teams. This method favors direct match-ups between conferences but still doesn't fully account for unbalanced schedules or limited cross-conference games.

Another ethical issue I run into when implementing my model is the problem of unintentionally undervaluing prospects with missing data or skewed data from possible injury or varying reasons. Currently looking at ways to minimize this risk by implementing data imputation techniques and models capable of handling incomplete datasets. By integrating historical data, social imbalances might also arise if models develop systemic inequities by relying a lot on that data. This can prioritize past trends from prospects in the past instead of more current ones. To combat this, the tool includes a wide range of metrics and emphasizes focusing on understanding context and specific situations instead of relying only on fixed, score based evaluations. Furthermore, information privacy, accuracy, and reliability is an important implication that should not be considered lightly. While most of the data used in this tool is publicly available, there is still a risk of integrating data that uses sensitive or private information.

Insights like injury history or personal information might be included in scout evaluations or other qualitative data sources. Making sure that all data is sourced properly and follows privacy regulations is very important. Also, the reliability of outcomes associated with this toll is directly related to the accuracy of input data. Inaccurate data for prospects can lead to false and misleading predictions and can negatively impact a prospects draft stock or judgement about them. To address this issue, the tool uses prioritizes blending subjective and objective data points while quantifying qualitative inputs wherever possible to reduce human error and subjectivity.

2 Related work

This chapter provides an in-depth exploration of existing studies and research that form the foundation for the NFL Draft Assist tool. By examining the historical, theoretical, and technical aspects of the NFL Draft process and predictive modeling, this chapter highlights the importance and challenges of using data-driven approaches to enhance decision making in sports analytics. The subsections below offer a detailed discussion of the draft's significance, predictors of draft success, normalization techniques, and predictive modeling methodologies, while also touching upon ethical considerations related to bias and data usage.

2.1 What is the NFL Draft and How Important is it?

The NFL Draft is a way for teams to build and acquire young talent in professional football. Established in 1936 the draft was created to ensure competitive balance in the league. Bert Bell the founder of the system and then owner of the Philadelphia Eagles wanted to address the disparity in team success so he created a reverse order draft also called a snake order. This allowed teams with worse records to pick first from the pool of eligible college players so they could rebuild and stay competitive[22]. The draft allows all teams, currently 32 teams, to get one pick every round. There are a total of 7 rounds, each team is given 10 minutes to call in and confirm their pick. The draft style is standard, meaning if you were to pick first in the first round then you would also pick 1st in the second round, and this will apply to all 7 rounds. Teams are also allowed to trade their picks for other picks or players on an opposing team[17]. The rules and structure of the draft have changed over the years but still the focus is on redistributing talent for parity. Teams make picks using strategic decision making to improve their roster while considering their cap space and long-term goals. In the NFL, cap space is the amount of money a team has available to spend on player salaries for a given year. The salary cap is a limit set by the league on how much a team can spend on its players' contracts. The purpose of the salary cap is to promote competitive balance by keeping rich teams from out bidding their opponents for talent. Teams use their cap space in different ways based on their strategy, roster needs and future plans. One way is to sign free agents to fill holes in their roster. If a team has a lot of cap space they can offer big contracts to top free agents and get better talent on the team. Another way is to restructure contracts which allows teams to create more cap space for the current year. For example they can convert a player's salary into a signing bonus and spread the bonus out over multiple years to reduce the player's cap hit for the current year while keeping the value of the contract in the future. Cap space is also important for signing rookie players from the NFL Draft as their contracts although based on their draft position still count against the cap. Teams also use cap space to extend contracts for key players they want to keep for the long-term so they don't have to let them enter free agency. Lastly teams have to account for "dead money" when a player is released or traded which is any remaining guaranteed money from their contract that still counts

against the cap. Teams plan for this by allocating cap space for it when making roster decisions.[23]

While this system has brought many benefits to the league it's not without its flaws. Many studies show problems in how talent is evaluated and selected and some say the draft doesn't always achieve its goal of equal competition[24]. In addition to being a necessary team-building process the NFL Draft has become a cultural and financial phenomenon. Its popularity is a sign of the broader interest in sports analytics and team strategy. For example the 2016 NFL Draft had 8.2 million viewers on opening night[28]. The draft is not just about who gets picked but about what it means for the team's future. Analysts, scouts, fans and executives pour over the details of player profiles, college statistics and combine numbers to try to speculate on the potential impact of each selection. From a business perspective the draft is a key part of the NFL's model. Teams invest big time in scouting, data analysis and training programs to find the best talent. A good draft pick can enhance the team's performance, increase ticket sales and merchandise revenue. A bad draft pick especially in the first round can have big financial and operational implications as teams are locked into contracts with players who may not perform. That's why every decision is so important in the draft process where every choice has long-term implications for the team's success and profitability. Despite being designed as an equalizer the draft has not always produced the desired competitive balance. Historical data shows that first round picks often perform better than later round picks but the variability within each draft class indicates problems in scouting and decision making[28] For example, at skill positions like running back and wide receiver, speed is highly valued in the draft process, yet it does not always correlate with long-term NFL success. Additionally, the economic pressure to win now means teams often prioritize physical attributes over long-term potential or intangibles like leadership and adaptability. Fixing these problems is a big challenge for the NFL as they continue to refine their draft process and metrics. So ultimately, NFL Draft is a necessary but not perfect system where sports strategy, economics and entertainment collide. It works at competitive balance but there's still a lot of work to be done.

2.2 What are key predictors of success for NFL draft prospects?

Determining what makes NFL draft prospects successful is a hotly debated topic. The NFL Scouting Combine is a big part of the evaluation process where players are tested on many physical and mental attributes. Standardized drills like the 40 yard dash time, bench press repetitions and three cone drill time are trying to quantify a player's athletic ability[9]. But research shows that these metrics alone are not enough to predict success. For example, Meil found that while some Combine results correlated with draft position, they were inconsistent at forecasting long-term performance especially across positions[14]. Tucker et al. further supported this by looking at the combine results and draft outcomes. They found that certain metrics like speed and jumping ability were

informative for positions like wide receiver and defensive back but had little correlation to production for positions that required more refined skills like quarterback or offensive line[27]. This highlights the need for a more comprehensive evaluation process that combines athletic performance with intangible attributes like game intelligence, adaptability and leadership. Collegiate performance has also been studied. Players who did well in college have better football IQ and experience which is key to navigating the pro game. But Brower warns against over relying on raw numbers, college systems and competition can inflate or obscure a player's true talent. Brower also says specific Combine numbers like the 40 yard dash are overvalued and teams will prioritize physical traits over skills that matter more in the pros[3]. Another big factor in draft evaluations is positional variability. Some positions like linebacker or safety are undervalued in favor of flashier roles like quarterback or wide receiver. Von Eicken points out this imbalance and says problems in evaluating positional value lead to suboptimal draft outcomes. These problems are further compounded by biases like overvaluing college programs or specific physical trait[28]. Advanced analytics offer ways to refine draft evaluations. Metrics from platforms like Pro Football Focus (PFF) give more detailed information on player performance and capture things traditional statistics miss. For example, analytics can account for the context of a player's contributions like blocking efficiency for offensive lineman (OL) or coverage ability for defensive back (DB). Incorporating these tools into the process can help eliminate biases and improve accuracy. In short, it's all about the whole package. Physical performance matters but must be balanced with qualitative and advanced analytics to get the full picture of a player. Fix the problems and biases and you can make better decisions that fit short term needs and long-term strategy.

The next sections will describe background pertaining to the technical methods of my analysis.

2.3 Normalization in predictive modeling

Normalization is a key part of data preparation and processing for both data analysis predictive modeling to make things comparable across different datasets and features. For the NFL Draft Assist tool, normalization is crucial in making metrics from different sources (college statistics, physical measurements, combine data) comparable. These metrics are on completely different scales (e.g. 40 yard dash times in seconds vs career receiving yards in thousands) so normalization is key to making fair and meaningful comparisons.

2.3.1 Importance using normalization for NFL performance metrics

- In the context of NFL draft analytics, normalization solves for:
  - Data variability across sources: Metrics from the NFL Combine like speed and strength are measured on a physical scale, while college performance data like touchdowns and catches are related to game

outcomes that depend on context and competition level. Without normalization, models might overvalue certain metrics just because of their numerical size.

- Comparing players across conferences: College football has different levels of competition. You have the SEC (Southeastern Conference), that has been widely suggested as the best conference for many years all the way down to a conference like the MAC (The Mid-America Conference) which is considered to be on the less competitive side. Normalizing metrics given conference strength ensures fair comparisons, especially when evaluating players from weaker conferences who may have inflated raw statistics.
- Career Length: Players with longer college careers have more impressive aggregate statistics and can skew evaluations. Normalization techniques (per game, per season averages) help to mitigate these biases and give a clearer picture of player production and consistency.

2.3.2 Normalization techniques used or considered when modeling

- Several normalization techniques are either used or was considered in the NFL Draft Assist tool, each for different types of data:
  - Min-Max Scaling: Rescales data to a fixed range [0,1] so all metrics contribute equally to the model. This is useful for metrics like weight and speed which are naturally bounded.
  - Z-Score Standardization: Centers data around the mean with a standard deviation of 1, good for metrics with a normal distribution like height and weight. This prevents outliers from dominating the model.
  - Log Transformation: Good for metrics with highly skewed distributions (career touchdowns, interceptions). Reduces skewness and highlights proportional differences between players.
  - Tanh Normalization: Combines robustness to outliers with bounded output, good for datasets with extreme values or noisy metrics like scout grades.

These techniques match what Dalwinder Singh and Birmohan Singh found in their thesis, normalization is necessary to improve performance of classification and regression models especially when dealing with a wide variety of datasets with different types of measurements[25].

2.3.3 Benefits of using normalization techniques in predictive modeling

- Normalization has been shown to:

- Improve Model Accuracy: As Singh et al. noted, normalization ensures all features are weighted equally, improves generalizability and reduces overfitting[25].
- Faster Computation: By standardizing the data, normalization speeds up model training especially for algorithms like gradient boosting or support vector machines.
- Across Players: In the NFL, normalization shows a player's true value, accounting for variations in team systems and competition levels.

2.4 Overfiiting

Overfitting happens when a model learns not only the patterns in the data but also the noise and outliers. This implies that the model works great on the training data but terrible on new or future data. Overfitting is a big problem because it kills generalization, which is what we're trying to do with predictive modeling. It's especially common in complex models with many parameters, especially when trained on small or imbalanced datasets[5].

Several strategies can mitigate overfitting and improve model performance. Validation techniques like repeated k-fold and cross validation are great for evaluating model performance across multiple training and validation splits and reduce bias and ensure robustness[5]. Regularization techniques like L1 (lasso) and L2 (ridge) penalize complex models by constraining parameter magnitudes while simplifying models by reducing features or using less complex algorithms can also improve generalizability. Early stopping in iterative training processes like gradient descent can prevent overfitting by stopping training when validation performance stops improving. Hyperparameter tuning done through nested cross-validation can ensure optimal performance without overfitting by systematically testing and selecting the best hyperparameters. Data augmentation which increases the diversity of training data using synthetic examples or transformations can reduce reliance on noise and outliers.

Charilaou and Battat stress the need to use nested k-fold and cross validation for hyperparameter tuning to avoid optimistic biases in performance evaluation. They advise against evaluating and tuning models on the same dataset split which often leads to overfitting. For smaller datasets, stratified k-folds are recommended to ensure balanced representation of outcomes in training and testing folds. This reduces overfitting even more and makes predictive metrics more valid[5]. By addressing overfitting, predictive models can get higher accuracy and more robust reliability. This makes them more effective and trustworthy for applications such as NFL draft analytics and other complex prediction systems

2.5 Predictive modeling and Validation techniques

The NFL Draft Assist tool is built around predictive modeling so you can evaluate players using their data. Linear regression and categorization methods

capture the many facets of NFL draft prospects and provide insight. Linear regression is the foundation for understanding relationships between independent variables and dependent outcomes. It's interpretable so you can see quantifiably how specific metrics, like 40 yard dash time impact success at different positions. You can also customize linear regression to fit the needs of specific position groups so you get more precise and relevant predictions. However, as Julius Ogunleye notes, linear regression is simple and may struggle to capture complex non-linear interactions; therefore, incorporating decision tree modeling is a promising direction to explore in future versions of this project.[16]. Decision trees help model non-linear relationships and interactions between variables. For example, a player's vertical jump combined with arm length is more predictive for tight ends than either metric alone. Random forests take this further by aggregating multiple decision trees, improving accuracy and reducing overfitting, for a robust way to handle complex datasets with many variables[16]. We should explore these methods in the future.

Additionally, techniques like cross-validation, already employed in the NFL Draft Assist tool, help fine-tune models and validate predictions by repeatedly splitting the data into training and testing sets. Approaches such as leave-one-out and k-fold cross-validation are particularly effective at reducing overfitting as well[31].

Furthermore, predictive modeling, more specifically the NFL Draft Assist tool is going to be able to be used in many different applications. Before draft day teams can use the NFL Draft Assist tool to find the best-value prospects that balance immediate impact with long-term potential. Data driven evaluations complement traditional scouting so biases and cognitive limitations can be counteracted in human evaluations. Post draft, the tool allows for retrospective evaluations so that past successes and failures can be examined and scouting and drafting can be refined going forward. By combining normalization with robust models, the NFL Draft Assist tool is at the leading edge of sports analytics and gives you actionable information to reduce the uncertainty and risk of high-stakes drafting decisions.

2.6 Feature Engineering: Balancing Bias and Predictive Value

Feature engineering is a key part of predictive modeling where you select, transform or create new variables to improve model performance. In the context of NFL Draft analytics, feature engineering is crucial in extracting insights from raw data and improving model predictiveness. But it also introduces bias that must be managed to ensure fair and accurate evaluations.

2.6.1 The Value of Feature Engineering

Feature engineering allows you to refine raw statistics into more informative metrics that better capture a player's abilities and potential. For example instead of just using total career receiving yards, a model might introduce efficiency

metrics like yards per route run or contested catch percentage[10]. Instead of using raw 40 yard dash time, models might use speed score, a weight-adjusted speed metric that provides a more context aware measure of athleticism[26].

Feature interactions are also important. Certain traits only work in combination with others. For example a quarterback's arm strength alone may not be a strong predictor of success, but when combined with decision making metrics and pocket awareness it could be a more complete evaluation[7].

Another important aspect of feature engineering is applying weighted scores to metrics based on their correlation with NFL success. Not all statistics are created equal when it comes to predicting success at the professional level. That's why manually weighting to metrics that have historically been closely tied to NFL success, makes your models more accurate. You can refine those weights using machine learning techniques to figure out which metrics really matter and which don't[10]. That way, your key performance indicators get the emphasis they deserve, and less predictive variables don't drag your models down.

2.6.2 Bias in Feature Engineering

Feature engineering when done properly, has many benefits. But that same process can also introduce biases if it's not handled with care. One reason for that is the way historical biases in scouting evaluations can skew certain metrics. Height and weight have historically been overvalued in predicting quarterback success. That means smaller, highly skilled quarterbacks might get underestimated as a result[12]. College statistics can also be misleading if you don't account for the level of competition. That can lead to some pretty misleading conclusions about a player's true talent[21].

To avoid that bias, you need to use fairness, aware modeling techniques. One way to do that is to reweight features based on what they actually contribute to predictive accuracy, not just what they've traditionally been important for in scouting narratives. Techniques like Principal Component Analysis (PCA) and just overall specified knowledge can help you figure out which features really matter for accuracy[7]. That way, you can avoid overfitting to those historically biased metrics.

2.7 Challenges in Predicting NFL Player Success

NFL player success is tough to predict because so many factors come into play. That's true even for the most advanced models. Football is a dynamic game where context, scheme fit and individual development all play significant roles[15].

2.7.1 Scheme Fit and System Variability

One of the biggest challenges in predicting success is finding the right scheme fit. For example, a quarterback who thrives in a spread offense with quick reads may struggle in a pro style system that demands full field progressions[12]. That's

because some teams coaching philosophies prioritize certain physical traits over technical skills. This can happen on the defensive side of the ball as well, a defensive player's effectiveness can depend just as much on whether they're playing in a 4-3 or 3-4 scheme[18].

2.7.2 Player Peaks, Declines, and Developmental Curves

Players develop at different rates. Some enter the league with high level skills but plateau early. Others develop gradually and reach their prime years later. Injuries, coaching changes and personal motivation all affect a player's career trajectory[15]. And then there are external factors like playing time, positional competition and changes in the scheme. Those can either accelerate or hinder development and make long-term outcomes highly uncertain. That variability in development, combined with the impact of scheme fit and external factors, makes it hard to project long-term success with certainty[26]. And that's why predicting NFL player success remains one of the toughest challenges in sports.

2.7.3 External Factors Impacting Success

External factors like team culture, coaching, personal discipline, all have just as much to do with a player's NFL success as their physical and technical abilities. You see that highly talented players can fail due to off field issues. Conversely, players with less impressive college resumes can succeed because of their work ethic and adaptability[18]. Those intangible factors are hard to put a number on, but they're what really determine a player's true potential. Media pressure, contract situations and locker room dynamics can warp an athlete's mental approach to the game and their ability to perform at that level over time as well[21].

2.7.4 Addressing These Challenges in the NFL Draft Assist Tool

To account for those uncertainties, the NFL Draft Assist tool looks at multiple layers of analysis. That includes historical trends and metrics that take the context of a player's situation into account. By acknowledging the inherent unpredictability of NFL careers, the tool aims to provide insights that complement traditional scouting rather than replace it. That way, predictions are grounded in both data and real-world considerations.

3 Method of approach

3.1 Introductions to Methods

Like previously stated in above sections, The NFL Draft is a vital part of team-building. Teams use scouting analysis to identify players with the most potential to succeed, but there are problems in the draft process like biases towards athleticism over other measures, overvaluing specific positions groups over others, as well as overvaluing players from big name colleges. This study will investigate problems using a python framework and machine learning techniques that will give insight into the relationship between pre-draft metrics and professional success. These techniques include, linear modeling, Monte Carlo Cross Validation (MCCV), KMeans clustering and manual threshold engineering for classification, and feature engineering weights for different metrics.

To give some insight on these different techniques, machine learning is training algorithms on historical data to find patterns and make predictions on new unseen data. It allows researchers to model complex relationships between features and targets, which in this case are pre-draft metrics and future NFL performance. By feeding the model a large set of examples it can learn which features are most predictive of success. The goal is to minimize prediction error while avoiding overfitting so the model performs well not just on training data but on new players as well. This allows for more objective, repeatable evaluations than just scouting alone.

Monte Carlo Cross Validation is used to train and test the predictive model on random subsets of the data so the results aren't skewed by a particular dataset split. This will allow us to look at variability in player success predictions and determine the reliability of different pre-draft metrics. KMeans Clustering is a machine learning algorithm that will group players by performance outcomes to give us an independent validation of the MCCV categorizations. Manual threshold engineering and feature engineering weights are both techniques that are based upon personal general knowledge of different pre-draft metrics and how well they correlate with success in the NFL. These two techniques are used to not solely just rely on the integrated models and their weights/classification methods.

Using historical college data, player measurers, NFL Combine data, and professional statistics from the NFL, this study will develop a data-driven framework to evaluate draft efficiency. Using my personal knowledge of current trends, I predict production measures will be more of a indicator of success over athleticism measures and body measures. Athleticism tends to be the most glaring area of bias, as well as school size. I think its not smart to just rely just on tradition scouting processes like mock draft and film tape. It would be beneficial to incorporate data-driven processes for evaluation to get more insight on prospects. The findings from this study will show us the key predictors of success, the areas of systemic bias in scouting and recommend ways to improve the decision-making process in the draft. These findings may be useful to judge player performance trajectories and show how to optimize the talent evaluation

process.

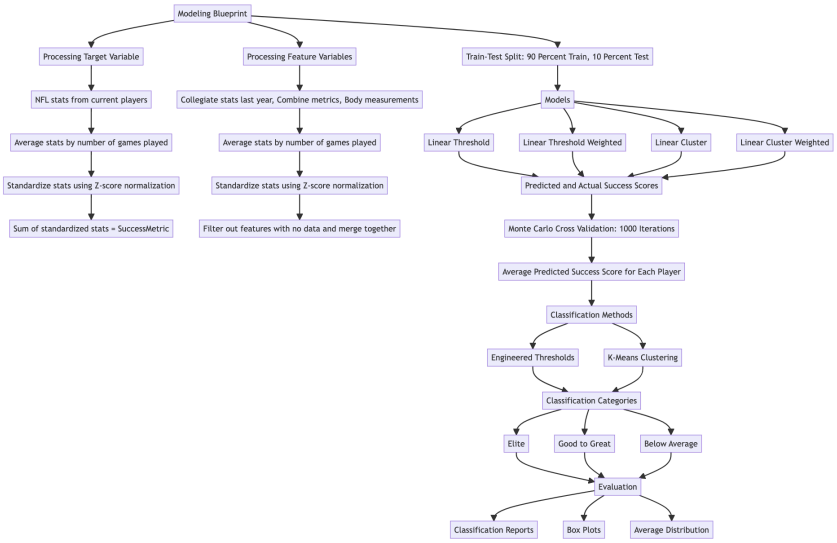


Figure 1: Model Overview

3.2 Data Collection and Preprocessing

The datasets used in this study were sourced from multiple historical records, public databases:

3.2.1 NFL Performance Data (2019–2022)

```
Rnd,Pick,Tm,Player,Pos,Age,To,AP1,PB,St,wAV,DrAV,G,nflRec,nflYds,nflTD,nflYears
1,25,BAL,Marquise Brown,WR,22,2024,0,0,4,34,24,74,322,3735,28,6
1,32,NWE,N'Keal Harry,WR,21,2023,0,0,1,7,6,49,64,714,5,5
2,36,SFO,Deebo Samuel,WR,23,2024,1,1,5,47,47,81,334,4792,22,6
2,51,TEN,A.J. Brown,WR,22,2024,0,3,6,66,26,90,446,7026,49,6
```

This dataset has post-draft performance statistics for NFL players[20] so we can measure how well each player did in their career based on multiple metrics. This dataset is used to get the actual success score or dependent variable used in the regression analysis. The research will analyze these success score metrics to see if pre-draft evaluations will predict NFL success.

There are 17 important metrics in the NFL performance dataset:

- Rnd (Round) & Pick (Overall): Where was a player drafted? We can see if players drafted in earlier rounds outperform those drafted later.

- Pos (Position): Helps to identify position value and trends in scouting for specific roles.
- Age: The age a player enters the NFL could impact longevity and development trajectory.
  - Mean: 22.10
  - Range: [21.00, 25.00]
- To (Last Year): The final year of the player's career helps to measure career longevity and sustainability of success.
  - Mean: 2022.86
  - Range: [2018.00, 2024.00]
- AP1 (First Team All Pro): How many times a player was selected to the All-Pro team. A measure of elite success; players with this are considered top in the league at there position.
  - Mean: 0.04
  - Range: [0.00, 1.00]
- PB (Pro Bowl): Another measure of success; how a player is viewed by peers, coaches and fans.
  - Mean: 0.17
  - Range: [0.00, 3.00]
- St (Years as Primary Starter): Consistency and did the player develop into a long-term contributor.
  - Mean: 1.14
  - Range: [0.00, 7.00]
- wAV (Weighted Career AV): A cumulative measure of career contributions.
  - Mean: 10.88
  - Range: [0.00, 66.00]
- DrAV (Weighted AV for Drafting Team): How well did a player's value translate to the team that drafted them? This is key to evaluating draft efficiency.
  - Mean: 9.78
  - Range: [0.00, 51.00]
- G (Games Played): Basic availability metric to see how often a player was healthy and used.
  - Mean: 40.95
  - Range: [1.00, 114.00]
- Receiving Metrics (nflRec, nflYds, nflTD, nflYears): Receiving statistics are key for skill position players like wide receivers and tight ends. One of the bigger indicators of production in the sport for these positions.

  - nflRec:
    * Mean: 103.87
    * Range: [0.00, 558.00]
  - nflYds:
    * Mean: 1354.68
    * Range: [0.00, 7531.00]
  - nflTD:
    * Mean: 7.88
    * Range: [0.00, 49.00]
  - nflYears:
    * Mean: 3.83
    * Range: [1.00, 7.00]

3.2.2 College Statistics (2019–2022)

```
Player,Team,Conf,Class,Pos,G,Rec,Yds,Y/R,TD,Y/G,HSrank
Marquise Brown,Oklahoma,Big 12,JR,WR,14,75,1318,17.6,10,94.1,3
N'Keal Harry,Arizona State,Pac-12,JR,WR,12,73,1088,14.9,9,90.7,5
Deebo Samuel,South Carolina,SEC,SR,WR,12,62,882,14.2,11,73.5,3
A.J. Brown,Ole Miss,SEC,JR,WR,12,85,1320,15.5,6,110.0,4
```

This dataset has a player's college level performance so we can see how they performed before entering the NFL. This is used to measure overall college production and how well college performance predicts to NFL success.

Metrics and their Importance:

- Player, Team, Conf (Conference), Class, Pos (Position): So we can put performance into context with the competition and positional demand.
- Conf (Conference):
  - Mean: 6.68
  - Range: [1.00, 10.00]
- G (Games Played): Durability and experience at the college level.
  - Mean: 12.14
  - Range: [8.00, 15.00]
- Receiving statistics: Core statistics for skill position players to measure production.
  - Rec (Receptions):
    * Mean: 63.65
    * Range: [21.00, 117.00]
  - Yds (Receiving Yards):
    * Mean: 960.20

        * Range: [285.00, 1856.00]
    - TD (Touchdowns):
        * Mean: 8.46
        * Range: [0.00, 23.00]
    - Y/R (Yards per Reception): Measures efficiency per catch, big play ability.
        * Mean: 15.25
        * Range: [8.80, 23.10]
    - Y/G (Yards per Game): Accounts for consistency across different game samples.
        * Mean: 78.40
        * Range: [2.00, 149.10]
- HSrank: Number of ranking stars each player was classified by in high-school (Higher is better)
    - Mean: 3.25
    - Range: [0.00, 5.00]

3.2.3 NFL Combine Measurements (2019–2022)

```
Player,40yd,Vert(in),Bench,Broad(in),3cone,Shuttle
Marquise Brown,4.27,n/a,n/a,n/a,n/a,n/a
N'Keal Harry,4.53,38.5,27,122,n/a,n/a
Deebo Samuel,4.48,39.0,15,122,7.03,4.14
A.J. Brown,4.49,36.5,19,120,n/a,4.35
```

This dataset includes physical and athletic testing results from the NFL Combine where prospects go through standardized drills. These metrics are heavily weighted in scouting but their actual correlation to NFL success can vary[14]. My modeling strategies will look into if athletic testing results are overvalued or if they are heavy indicators for professional success.

Metrics and their Importance:

- 40yd Dash(sec) (Speed Test): Important for skill positions like WRs, CBs, and RBs where speed is key.
    - Mean: 4.47
    - Range: [4.26, 4.73]
- Vert (Vertical Jump) & Broad (Broad Jump): Measures lower body explosiveness which is important for positions that require jumping ability like WRs and DBs.
    - Vert(in):
        * Mean: 35.51

        * Range: [29.00, 44.50]
    - Broad(in):
        * Mean: 124.39
        * Range: [112.00, 140.00]

- Bench Press(repetitions): Tests upper body strength often used for linemen but less predictive for skill positions, repetitions of 225lbs.
    - Mean: 14.98
    - Range: [6.00, 27.00]
- 3 Cone Drill & Shuttle: Tests agility and change of direction speed which is important for positions that require quick movements like slot receivers, defensive backs.
    - 3 Cone(sec):
        * Mean: 6.99
        * Range: [6.56, 7.40]
    - Shuttle(sec):
        * Mean: 4.26
        * Range: [3.99, 4.62]
- Body Measurements (Height, Weight, Hand Size, Arm Length, etc.)

```
Player,Height(in),Weight,Hand(in),Arm(in),Wingspan(in)
Marquise Brown,69.375,166,9,30.5,71.75
N'Keal Harry,74.375,228,9.5,33,78.25
Deebo Samuel,71.25,214,10,31.375,75.125
A.J. Brown,72.5,226,9.75,32.875,78
```

This dataset includes physical body dimensions like height, weight, and reach. NFL teams have historical preferences for certain body types at different positions[13]. Incorporating these metrics will show if these physical measurements really matter or just lead to bias in scouting by comparing models with different weighted values of these metrics.

Metrics and their Importance:

- Height: Catch radius, higher capability, etc.
    - Mean: 72.83
    - Range: [68.00, 77.50]
- Weight: Injury reduction, muscle mass.
    - Mean: 201.32
    - Range: [155.00, 238.00]
- Hand Size: Impact on ball security and catching ability, especially for QBs and WRs.

  - Mean: 9.42
  - Range: [7.88, 10.75]

- Arm Length & Wingspan: Advantages in blocking, tackling and catching.
  - Arm(in):
    * Mean: 31.82
    * Range: [28.50, 35.25]
  - Wingspan(in):
    * Mean: 76.20
    * Range: [70.00, 83.88]

3.3 Data Cleaning and Merging

Before modeling, data was pre-processed to clean and fill in missing values. This was done using a Python framework, mostly pandas for data manipulation and cleaning and numpy for numerical operations and missing data imputation. These tools allowed for efficient filtering, normalizing and structuring of the data to get it ready for machine learning.

For the NFL performance dataset, steps in the preprocessing phase included standardization, where NFL career statistics were divided by the number of professional years (nflYears) to calculate per-year averages. Certain metrics were also z-score normalized to ensure comparability across different scales[25]. Missing values in the dataset were handled by representing them as `np.nan` as shown in the code blocks below, allowing for consistent processing using Python libraries. Additionally, some values required contextual interpretation to ensure that their meaning was accurately captured and appropriately represented within the dataset.

For the rest of the data, missing values across the datasets were handled by replacing them with np.nan. This ensures that missing data points are properly recognized for further cleaning or exclusion during preprocessing.

```
combined_data_combine.fillna(np.nan, inplace=True)
```

Conferences were ranked numerically based on historical strength to create a schedule strength feature. Unlisted conferences were set to the lowest ranking so everything is consistent across the dataset.[19]

```
conference_rankings = {
    'SEC': 10,
    'Big Ten': 9,
    'Big 12': 8,
    'Pac-12': 7,
    'ACC': 6,
    'AAC': 5,
    'MWC': 4,
```

```
    'C-USA': 3,
    'Ind': 2,
    'CAA': 1
}

# Map conference names to their rankings, default to 1 for any
# conference not listed
college_statistics['ConfRank'] = college_statistics['Conf'].map(
    conference_rankings).fillna(1)
```

The various datasets — college statistics, measurements, and combine results — were merged on the Player field which is the unique identifier. This ensures features and target variables line up for modeling.

```
# Merge datasets on Player column
data = college_statistics.merge(measurements, on="Player").merge(
    combine_statistics, on="Player")

# Ensure that the players in the target and features match
merged_data = data.merge(nfl_statistics[['Player', 'SuccessMetric']],
    on='Player', how='inner')
```

40 yard dash times were inverted so faster is better. Players who played in 7 or fewer games in their final college season were excluded to focus on more reliable samples.

```
# Faster is better
merged_data['40yd_inv'] = 1 / merged_data['40yd']

# Exclude players with 7 or fewer games played in college
merged_data = merged_data[merged_data['G'] > 7].reset_index(
    drop=True)
```

All numerical features were standardized with StandardScaler. This allows features which had different units and ranges to be comparable on the same scale.

```
# Standardize the averaged metrics (normalize by
# standard deviation)
nfl_statistics[normalized] = scaler.fit_transform(nfl_statistics[
    [metric + '_avg' for metric in nfl_metrics_to_normalize]])

# Normalize the feature metrics (standard deviation normalization)
features_normalized = scaler.fit_transform(features)
```

Finally all datasets were merged again on the Player field after normalization and cleaning. The final merged dataset ensures all modeling features and success metrics are associated with the correct player.

```
# Merge datasets on Player column
data = college_statistics.merge(measurements, on="Player").merge(
    combine_statistics, on="Player")

# Ensure that the players in the target and features match
merged_data = data.merge(nfl_statistics[['Player', 'SuccessMetric']],
    on='Player', how='inner')
```

A key feature, SuccessMetric, was engineered to represent an NFL player's performance and was then used as the target variable for modeling. It was the sum of z-scored values for career average reception statistics, accolade statistics(e.g, AP1, pro bowl, starter) and wAV for each player. The statistics were avergaed by the number of games played for each player.

Below is the formula used to calculate the Success Score (Target Variable) of each current NFL player:

$$SuccessMetric = Z(\text{nflYds/G}) + Z(\text{nflTDs/G}) + Z(\text{nflRec/G}) + Z(\text{AP1/G}) + Z(\text{PB/G}) + Z(\text{St/G}) + Z(\text{wAV})$$

where $Z(x)$ represents the z score normalization of feature (x), defined as:

$$Z(x) = \frac{x - \mu_x}{\sigma_x}$$

with (μ_x) as the mean and (σ_x) as the standard deviation of the respective metric across all players.

3.4 Modeling Approach

Linear Regression is used as the primary modeling technique to estimate the success scores of NFL draft prospects based on their college statistics, combine metrics and body measurements. Linear regression is used to predict a continuous dependent variable based on one or more independent variables. In this case we want to model the relationship between the features (metrics) and the target variable (SuccessMetric or success scores for players) and make predictions on new data. Linear Regression is implemented using the `LinearRegression` class from the Python machine learning library `scikit-learn`. This library provides a simple interface to apply regression and includes functionality to fit, predict and evaluate models. The LinearRegression class in `scikit-learn` uses ordinary least squares (OLS) to find the best fitting line through the data.

To evaluate the model we use Monte Carlo Cross Validation (MCCV) by splitting the dataset into training and testing sets multiple times.

3.4.1 Monte Carlo Cross Validation

To get a accurate evaluation and avoid overfitting, Monte Carlo Cross Validation (MCCV) is used. MCCV involves splitting the dataset into training and testing subsets, training a model on the training set, and evaluating it on the test set. For this project, 1000 iterations of MCCV were performed, each time generating a new random split of the data. In every iteration, a Linear Regression model was trained on a unique subset and used to predict player success scores based on pre-draft features.

Using this repeated random sampling approach has several advantages over a single train-test split. It exposes the model to many data partitions, so performance metrics can be averaged across many different scenarios. This gives a more reliable estimate of the model's ability to generalize to new players. By averaging results over 1000 runs, the evaluation is less sensitive to any particular random split of the data, so the conclusions drawn from the model's performance are stronger.

Each iteration of MCCV ties into the train-test split and parameter setup below, where the data is split, the model is trained and the predictions are validated.

3.4.2 Train Test Split and Parameters

For every iteration of the MCCV, the dataset is split into training and testing sets using the `train_test_split` function from `scikit-learn`. This function randomly partitions the dataset into two parts, one for training the model and another for testing its performance. The parameters used are:

- `features`: The independent variables (predictors) used to train the model. These are going to be college statistics, physical measurements and combine metrics for a player.
- `target`: The dependent variable (response variable) the model is trying to predict. In this case it's the NFL success score for a player.
- `test_size`: The proportion of the dataset used for the test set. For this model test_size=0.1, a value of 0.1 means 10% of the data is used for testing and 90% for training. This ensures a large test set for model generalization across different Monte Carlo iterations.
- `random_state`: A variable to ensure reproducibility of the train test split by setting a fixed random seed for each iteration. In MCCV, random_state=i (i changes with each iteration) to get different train test splits across multiple runs to reduce bias and improve the strength of the evaluation.

The `train_test_split` function returns four variables:

- `X_train`: The features used for training the model. It's 90% of the dataset in each iteration.

- `X_test`: The features used for testing. It's 10% of the dataset and is used to see how well the trained model generalizes to new data.
- `y_train`: The target values corresponding to X_train. These are the actual success scores the model learns from during training.
- `y_test`: The target values corresponding to X_test. These are the actual success scores used to compare with the model's predictions to measure accuracy and performance.

This is done for each iteration in MCCV, so the model is made on multiple versions of the dataset. By training and testing on different subsets the approach gives a better evaluation of the model's prediction and reduces overfitting.

After setting up the framework with Linear Regression, Monte Carlo Cross Validation and train-test splitting, the next step was to develop and apply specific models to predict and categorize player success. Four models were created based on how the predictions were turned into player categories and how the models handled weighting. Each model is a slightly different way of mapping success scores into categories, which I'll explain below.

3.4.3 Models

Four models were built to predict and categorize NFL prospects based on pre-draft data: Linear Threshold, Linear Threshold Weighted, Linear Cluster, and Linear Cluster Weighted. Each model used the same underlying prediction method, Linear Regression with Monte Carlo Cross Validation, but differed in how they categorized players and whether or not they used manually engineered feature weights.

Linear Threshold used a standard OLS regression without custom feature weights. Player success scores were categorized into tiers (Elite, Good to Great, Below Average) based on fixed, manually chosen thresholds.

Linear Threshold Weighted applied custom-designed feature weights to the linear regression, but still used the same fixed thresholds.

Linear Cluster used unweighted OLS predictions but instead of manual thresholds, it grouped players using KMeans clustering on the predicted success scores. This allowed natural groupings in the data to determine player tiers.

Linear Cluster Weighted combined both feature engineering and dynamic categorization. It used the manually weighted feature regression and grouped players using KMeans clustering, blending guided feature importance with data-driven categorization.

By building these four models, we explored how feature weighting and different categorization methods affect NFL draft prospect evaluation.

Throughout the MCCV process predicted success scores, actual scores and their corresponding categories were saved to evaluate prediction consistency and accuracy across multiple data splits.

3.4.4 Feature Engineering

Feature engineering played a crucial role in ensuring that the model accurately captured the relationships between pre-draft metrics and NFL success. Instead of treating all features equally, manual weighting was applied to emphasize the relative importance of certain metrics in predicting success. This approach allowed for a different interpretation of how specific attributes contribute to a player's transition from college football to the professional level compared to not using wieghted features and relying on OLS to learn weights for each feature.

3.4.4.1 Weighted Metrics for Predictive Modeling As an alternative to OLS, custom weights were assigned to each feature that reflect how much they may contribute to NFL success based on personal intangibles and knowledge. I looked at what historically correlates best for players in the NFL as well as leveraged my personal knowledge and gave those metrics more influence in the model by using larger coefficents. That way, the really important statistics, like receptions, yards and touchdowns, have much more significance when determining success values. In this way, we can guide the model to focus on the hypothesized high impact statistics and focus less on the others.

Weights were applied directly to the model like this:

```
    # Define own custom weights
    custom_weights = np.array([1, 1, 1, .5, 1, .25, 4, .25,
        .1, .1, .1, .1, .1, .5])

    # Manually set the coefficients
    model.coef_ = custom_weights
```

Terminal Ouput shows which weights correlate to the specfic features

```
=== Engineered Model Coefficients ===

Rec: 1.000
Yds: 1.000
Y/R: 1.000
TD: 0.500
Y/G: 1.000
G: 0.250
ConfRank: 4.000
40yd_inv: 0.250
Height(in): 0.100
Weight: 0.100
Hand(in): 0.100
Arm(in): 0.100
Wingspan(in): 0.100
HSrank: 0.500
```

3.4.4.2 Weighted SuccessMetric Calculation When engineering the target variable to reflect the importance of different NFL performance statistics, different NFL statistics were combined. These included yards, touchdowns, receptions and accolade statistics like All-Pro selections and Pro Bowl appearances. Each of those components were given a specific weight based on how much they actually correlated with evaluating a player's overall professional success.

That weight was determined by how much each statistic has historically correlated in evaluating a player's impact in the NFL. Receiving yards and touchdowns got higher weights because they're directly tied to a team's offense and a player's longevity in the league[6]. Pro Bowl selections, Number of years started, and weighted Approximate Value (wAV) were given lower weights to account for other contributions that might not be as directly measurable like player progression or decline within a players full career. All-Pro team selections were weighted higher because this is a direct reflection of being elite, or beingn regarded as best in the position, and not commonly given out. So I weighted this metric higher to hold more significance.

```
# Define weights for SuccessMetric calculation
success_metric_weights = {
    'nflYds_avg_normalized': .6,
    'nflTD_avg_normalized': .5,
    'nflRec_avg_normalized': .6,
    'AP1_avg_normalized': .6,
    'St_avg_normalized': .4,
    'PB_avg_normalized': .4,
    'wAV_normalized': .4
}
```

By incorporating personal and reliable knowledge into both feature selection and target variable construction, this approach to feature engineering made it possible to start thinking about possible limitations of OLS modeling for these kinds of data. The goal was to produce real-world scouting insights and performance expectations, so validating the OLS strategy and understanding its strengthsand limits is essential.

After generating player success predictions with Linear Regression and feature importance through engineering, the next step was to turn those continuous success scores into something actionable. Instead of just evaluating model performance with raw error metrics, players were grouped into tiers to mimic how NFL teams scout, rank and differentiate prospects. This section outlines the tiering strategies used to turn success predictions into meaningful groups.

3.4.5 Categorization Methods

Here are the methods used, manual thresholding and KMeans clustering, to categorize NFL prospects based on their predicted and actual success scores: manual threshold-based categorization and clustering-based categorization. Both

methods looked at player performance and potential based on pre-draft data and predicted success in the NFL but used different ways to turn numerical success metrics into player categories.

3.4.5.1 Need for Categorization While traditional model evaluation metrics like Mean Squared Error (MSE) or R-squared are useful for understanding prediction accuracy, they aren't as intuitive when applied to real-world player evaluation. NFL teams, scouts and decision makers don't evaluate prospects solely on continuous numerical success scores. Instead, players are often grouped into qualitative tiers like, "elite prospects", "mid-round contributors" or "developmental players", to guide drafting and roster decisions.

So grouping players into performance groups based on their predicted and actual success scores was a key part of this project. Grouping allows for clearer comparison between model expectations and real-world outcomes. It also reflects how scouts and media proffesionals think about prospect value rather than just pure numeric prediction error. Both manual thresholding and clustering were used to explore different ways of forming these groups and understanding player trajectories beyond what summary statistics like MSE can tell you.

3.4.5.2 Manual Thresholding The manual threshold-based categorization method used a simple, rule-based system to categorize players. In each iteration the model predicted player success scores based on the pre-draft features. To turn those continuous success scores into categorical labels the `categorize_player` function was used. This function assigned players to one of three categories; Elite, Good to Great, and Below Average, based on specific success score thresholds. Players with scores of 7.7 or higher were labeled Elite, players with a score lower than 7.7 but higher than 3 were labeled Good to Great, and players with scores below 3 were labeled Below Average. This approach ensured that each player's predicted and actual performance was mapped directly to a category so we could compare expected and real-world outcomes.

```
# Define categorize_player function
def categorize_player(success_metric):
    if success_metric >= 7.7:
        return "Elite"
    elif success_metric >= 3:
        return "Good to Great"
    else:
        return "Below Average"
```

3.4.5.3 Kmeans Clustering On the other hand the clustering-based categorization method used `scikit-learn` to create dynamic clusters of players based on their success scores. Similar to the threshold method, the MCCV framework with 1000 iterations and Linear Regression was used to generate

predicted success scores. But instead of using fixed thresholds, the KMeans clustering algorithm was used to define player categories based on the natural groupings in the data. The averaged out predicted success score and actual success score for each player were collected seperately, then KMeans clustering was applied with three clusters, matching the three categories used in the threshold models[30]. The clustering process was run 100 times (n_init=1000) to ensure the resulting clusters were stable. After clustering the mean success score within each cluster was calculated and the clusters were ranked from highest to lowest. Those ranked clusters were then mapped to the established categories with the top cluster labeled "Elite", middle cluster named "Good to Great", and the bottom named "Below Average". The functions `cluster_to_category` and `actual_cluster_to_category` did this mapping, translating the cluster labels into interpretable categories for each player.

```
# Perform K-Means clustering on average predicted success scores
avg_success_array = np.array(list(
    player_avg_success_scores.values())).reshape(-1, 1)
kmeans = KMeans(n_clusters=3, random_state=42, n_init=100)
kmeans.fit(avg_success_array)

# Sort clusters based on average success score
cluster_avg_scores = pd.DataFrame({
    "Cluster": kmeans.labels_,
    "Score": list(player_avg_success_scores.values())
}).groupby("Cluster")["Score"].mean().sort_values(ascending=False)

# Map sorted clusters to categories
cluster_to_category = {
    cluster_avg_scores.index[0]: "Elite",
    cluster_avg_scores.index[1]: "Good to Great",
    cluster_avg_scores.index[2]: "Below Average"
}

# Assign predicted categories based on clustering
player_predicted_categories = {player: cluster_to_category[
    kmeans.labels_[i]] for i, player in enumerate(
        player_avg_success_scores.keys())}

# Assign actual categories based on clustering the target variable
actual_success_array = np.array(list(
    player_actual_scores.values())).reshape(-1, 1)
kmeans_actual = KMeans(n_clusters=3, random_state=42, n_init=100)
kmeans_actual.fit(actual_success_array)

actual_cluster_avg_scores = pd.DataFrame({
    "Cluster": kmeans_actual.labels_,
```

```
    "Score": list(player_actual_scores.values())
}).groupby("Cluster")["Score"].mean().sort_values(ascending=False)

actual_cluster_to_category = {
    actual_cluster_avg_scores.index[0]: "Elite",
    actual_cluster_avg_scores.index[1]: "Good to Great",
    actual_cluster_avg_scores.index[2]: "Below Average"
}

player_actual_categories = {player: actual_cluster_to_category[
    kmeans_actual.labels_[i]] for i, player in enumerate(
        player_actual_scores.keys())}
```

After developing the prediction models and categorization methods, next was building a platform to visualize the results. That's where a web application came in. To make the outputs more accessible and interactive, a web app was implemented that shows players' predicted success rates and category assignments in a way that's easy for anyone to understand and use.

3.5 Web Application Approach

The web app was built using `Flask`, a lightweight Python web framework, to provide an interactive platform to display and compare the results of all four linear-based models. This app allows users to view player success score predictions, actual success score outcomes and categorization correctness through a structured and visually friendly interface.

3.5.1 Technical Implementation

The system follows a model-view-controller (MVC) architecture, where the model is the statistical analysis scripts, which would consider all of my code that produces the output when run in terminal. MVC was a good choice because it separates the data processing, user interface and control logic so the system is easier to develop, debug and extend. Compared to more tightly coupled designs like, scripted-based and template-only web apps, MVC allows the backend statistical models and the frontend web pages to evolve independently without breaking functionality. This separation also matches the goal of keeping a clear flow between player prediction data and its presentation to the user. The view is all of the HTML templates that produces the specific pages for each model on the web app. And the controller is handled by Flask routes in `app.py` which manage navigation and data flow between backend and frontend. To make backend processing and frontend experience smooth, the Flask app imports and structures model data into dictionaries so it can be easily rendered in the HTML templates. Flask routes `@app.route` handle navigation and data flow so users can switch between model results. To suppress model import con-

sole outputs, a NullWriter class was added to temporarily redirect standard output before restoring normal Flask server output.

3.5.2 UI/UX For modeled data

The backend processes and formats the prediction data before serving it to the frontend. All models generate player success predictions based on pre-draft metrics and produce actual success scores, predicted success scores and performance categories. This structured data is extracted and formatted in Flask and made available to the frontend through API endpoints (/threshold_data, /threshold_weighted_data, etc.). The Flask app maps players to their predicted and actual success scores, as well as their predicted and actual categories so users see how well the models perform on different players.

```
@app.route("/threshold_data")
def threshold_data():
    return jsonify(linear_threshold_data)

@app.route("/threshold_weighted_data")
def threshold_weighted_data():
    return jsonify(linear_threshold_weighted_data)

@app.route("/cluster_data")
def cluster_data():
    return jsonify(linear_cluster_data)

@app.route("/cluster_weighted_data")
def cluster_weighted_data():
    return jsonify(linear_cluster_weighted_data)
```

```
# Convert data into a structured format for the webpage
linear_threshold_data = [
    {
        "name": player,

        "actual_category":
            linear_threshold.player_actual_categories.get(
                player, "Unknown"),

        "predicted_category":
            linear_threshold.player_predicted_categories.get(
                player, "Unknown"),

        "actual_score":
            linear_threshold.target[
                linear_threshold.merged_data[
```

```
                    'Player'] == player].values[
                        0] if player in linear_threshold.
                            merged_data[
                                'Player'].values else None,

        "predicted_score":
            linear_threshold.player_avg_success_scores.get(
                player, None),

        "match":
            linear_threshold.player_actual_categories.get(
                player) == linear_threshold.
                    player_predicted_categories.get(
                        player)
    }
    for player in lt_player_list
]
```

The frontend was enhanced to provide interactive visualizations in addition to tabular data. The homepage is the central hub with links to each model's results and each results page has a "Back to Home" button. Each model's results page now has a bar chart using Chart.js to display actual vs. predicted success scores for each player. Also a emoji-coded matching system was added where correctly predicted categories are set as green check and mismatches are red x's. This allows users to quickly see how well the models did.

```
# Suppress output from imported modules
class NullWriter:
    def write(self, _): pass
    def flush(self): pass

sys.stdout = NullWriter()
```

```
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/linear_threshold")
def lt():
    return render_template(
        "linear_threshold.html",
        players=linear_threshold_data)

@app.route("/linear_threshold_weighted")
def ltw():
```

```
    return render_template(
        "linear_threshold_weighted.html",
            players=linear_threshold_weighted_data)

@app.route("/linear_cluster")
def lc():
    return render_template("linear_cluster.html",
        players=linear_cluster_data)

@app.route("/linear_cluster_weighted")
def lcw():
    return render_template(
        "linear_cluster_weighted.html",
            players=linear_cluster_weighted_data)
```

```
// data
const playerNames = playersSubset.map(
    player => player.name);

const actualScores = playersSubset.map(
    player => player.actual_score);

const predictedScores = playersSubset.map(
    player => player.predicted_score);

const matchColors = playersSubset.map(
    player => player.match ? ' ' : ' ');

const predictedCategories = playersSubset.map(
    player => player.predicted_category);

// chart
new Chart(canvas.getContext('2d'), {
    type: 'bar',
    data: {
        labels: playerNames,
        datasets: [
            {
                label: 'Actual Score',
                data: actualScores,
                backgroundColor: 'blue'
            },
            {
                label: 'Predicted Score',
```

```
                data: predictedScores,
                backgroundColor: 'orange'
            }
        ]
    },
```

3.5.3 Interactive UI/UX for New Data

Overall the app is a intuitive, interactive and user-friendly user interface to visualize the performance of all models on NFL draft prospects, as well as use insights of models to support decision making when it comes to making judgements or evaluating NFL draft prospects. The purpose of the web app is to be a data visualization tool that connects the statistical analysis and human interpretation. By making the model outputs visual, the app allows the user to check model accuracy by looking at the emojis, find gaps in the draft process by asseesing the accuracy of the different models, and see where the model and actual player performance diverge by looking at the graphs and emojis. This tool is ment to be a resource for scouts/media proffesionals/coaches and help to make data-driven decisions and get inisight into how a players pre-draft metrics may or may not be indictive of professional success.

4 Experiments

This chapter describes the experiments, the process of running the models and how they were evaluated. The main goal of these experiments is to test different linear regression based models to see how well they can predict the success of wide receivers transitioning from college football to the NFL. By using historical data, the models were trained to recognize patterns in player statistics and use them to predict future NFL performance.

The data used to train and test the models includes a bunch of different features: college statistics, physical measurements (height, weight, wingspan, etc.), and NFL Combine results (40-yard dash, bench press, vertical jump, agility drills). By including multiple features, the models were designed to capture multiple aspects of player potential.

After running the models, the predictions were compared to actual player outcomes using multiple methods. A key part of this was categorizing players into three different success levels:

- Elite: Players that performed at an extremely high level in the NFL, often becoming Pro Bowlers or getting selected to the All Pro team.
- Good to Great: Players that have solid or above-average NFL careers but didn't become elite.
- Below Average: Players that didn't have significant success in the NFL or had short careers.

Each model gave a predicted success score to each player and then we used those scores to see if the model put the player in the correct category. I assessed the accuracy and reliability of those predictions using two main methods: classification reports and box plots.

Classification reports gave a breakdown of how well each model did in each category. Those reports included key metrics: precision, recall and F1-score.

- Precision measures how often the model's predictions were correct for a given category.
- Recall measures how well the model identified all players that truly belonged to a category.
- F1-score balances precision and recall to give an overall performance measure.

Box plots were used to visually compare the actual and predicted success scores. If the model did well, the predicted scores should match the actual scores. If the predictions were off, there would be big differences between actual and predicted scores. Since multiple models were run, each model was evaluated individually before comparing them directly. The results of each model were noted and their pros and cons were looked at. By comparing the models we

could see which approach did the best. These experiments provide a way to systematically test different linear regression models for NFL success. Results will be used later to determine which factors matter most for wide receiver performance at pro level.

4.1 Experimental Design

4.1.1 Data Preparation and Selection

Before I ran the experiments I cleaned and filtered the data to make it more reliable. I excluded players with incomplete data, very limited college playing time, or skewed data from injury. This way the models were trained and tested on players with a real career path rather than outliers or players with very little impact. I also standardized all the numerical features so that scale differences wouldn't affect the model's performance.

4.1.2 Training and Testing

To avoid bias in model evaluation I used Monte Carlo cross validation as the primary method to train and test the models. This splits the data into training and testing sets, randomly selecting different data points each time. By using multiple random splits this way I ensure the results aren't dependent on a single dataset split which increases the overall reliability and stability of the evaluation. For each experiment I trained a model on a subset of the data and then tested it on unseen data. I compared the model's predictions to the actual success scores of the players. I did this multiple times to minimize the randomness and get average performance measures.

4.1.3 Prediction and Evaluation Metrics

Once the models were trained to predict success for each player, the accuracy and effectiveness of those predictions were evaluated with numbers and visuals.

4.1.3.1 Model Comparison After evaluating each model separately, a comparison was done to see which model performed best at predicting NFL success. The following was considered:

- Consistency across multiple test sets
- Ability to classify players into success categories
- Stability as shown by cross validation results
- Alignment between actual and predicted scores in visual analysis

This comparison showed the strengths and weaknesses of different models and which factors were most important in predicting wide receiver success at the NFL level.

4.1.3.2 Fair and Consistent Evaluation

To ensure fairness and consistency, all models were tested under the same conditions with the same data, same training method and same evaluation metrics. By keeping the experiment the same, I could make meaningful comparisons without bias or unfair advantages. This will be used to choose the best approach for predicting NFL wide receiver performance and to inform future models which features matter most.

4.2 Evaluation

4.2.1 Classification Report Comparison

Classification Report Comparison

Model	Category	Precision (%)	Recall (%)	F1-score (%)	Support
Linear Threshold	Below Average	80.73	86.27	83.41	102
	Elite	66.67	14.29	23.53	14
	Good to Great	10.00	12.50	11.11	16
	Overall Accuracy	**69.70**	-	-	-
Linear Threshold Weighted	Below Average	96.81	80.53	87.92	113
	Elite	50.00	75.00	60.00	4
	Good to Great	34.38	73.33	46.81	15
	Overall Accuracy	**79.55**	-	-	-
Linear Cluster	Below Average	73.91	38.20	50.37	89
	Elite	14.81	66.67	24.24	6
	Good to Great	27.12	43.24	33.33	37
	Overall Accuracy	**40.91**	-	-	-
Linear Cluster Weighted	Below Average	77.50	35.23	48.44	88
	Elite	14.29	100.00	25.00	5
	Good to Great	22.81	33.33	27.08	39
	Overall Accuracy	**37.12**	-	-	-

Figure 2: Table

4.2.2 Evaluation of the Linear Threshold (LT) Model

The LT model had an overall accuracy of 69.70% which is the second best so far, better than Linear Cluster (LC) and Linear Cluster Weighted (LCW) but behind Linear Threshold Weighted (LTW).

Pros:

- Good at predicting Below Average players
- The model had 80.73% precision for this group, meaning when it predicted a player as Below Average it was usually correct.
- 86.27% recall for Below Average players, meaning it found most of the Below Average players in the dataset.

Cons:

- Struggled with Elite and Good to Great players
- Only 14.29% of actual Elite players were correctly identified, many top players were misclassified.
- Did even worse with Good to Great players, only 12.50% were correctly identified.
- Low precision and recall for these groups means the model had trouble separating good players from average or below average players.

Box Plot Insights:

- The box plots show predicted success scores for Below Average players were aligned with actual scores.
- But there was more inconsistency in the predicted scores for Elite and Good to Great players which explains the poor performance in these groups.

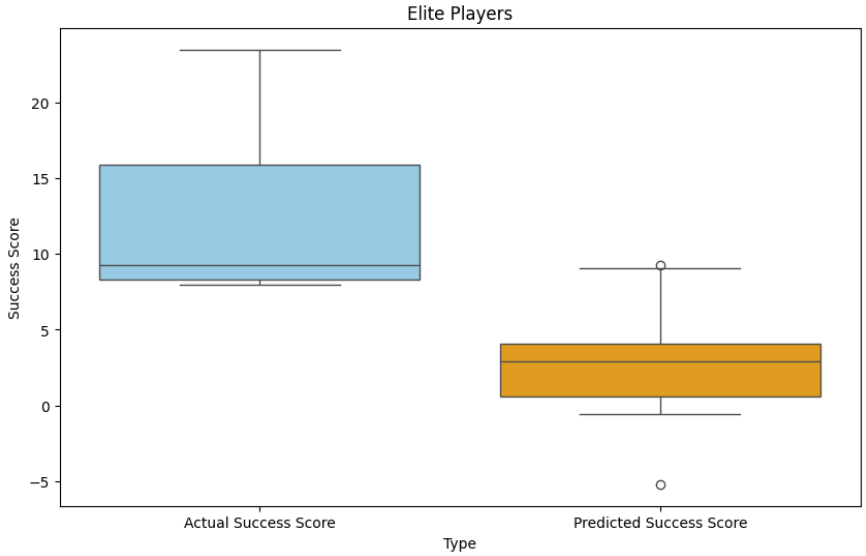


Figure 3: lt1

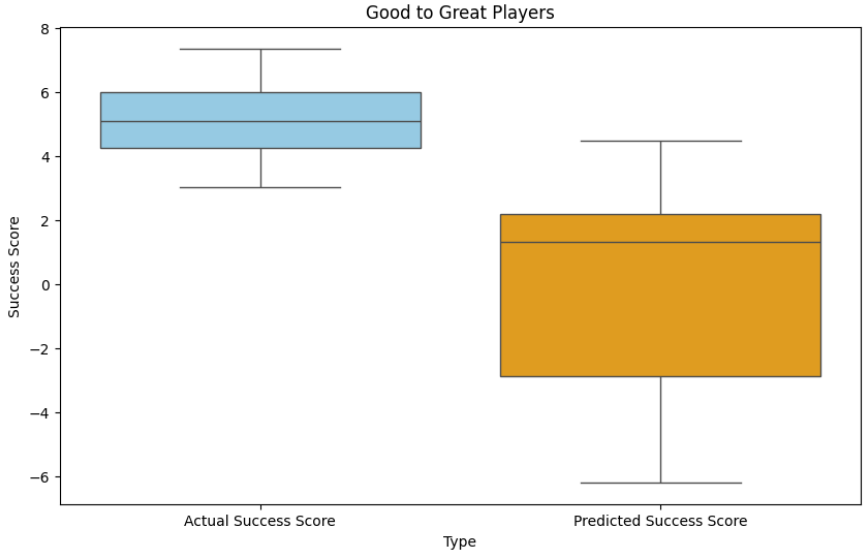


Figure 4: lt2

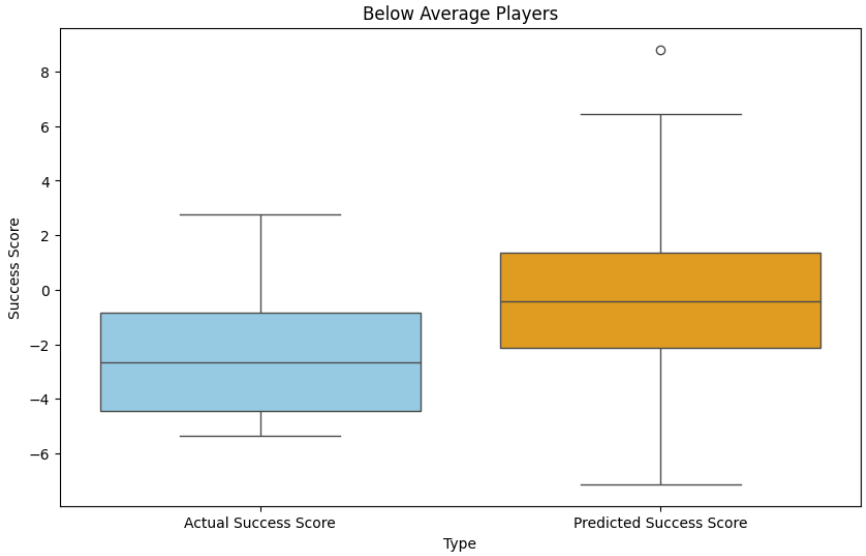


Figure 5: lt3

4.2.2.0.1 Final Verdict The LT model was good at Below Average players but struggled with higher tier players. Since it's the second best overall, it's not bad but needs to improve on Elite and Good to Great prospects. Maybe adjust feature importance or test other classification techniques to improve accuracy for top groups.

4.2.3 Evaluation of the Linear Threshold Weighted (LTW) Model

LTW model scored 79.55%, the best so far. Better than LT, LC and LCW models.

Pros:

- High Accuracy for Below Average Players Precision: When classifying Below Average players, it was right 97% of the time.
- Recall: 80.53%, Model got 80% of actual Below Average players right.
- Very good at identifying weaker NFL prospects like LT model but better precision.
- Improved Recall for Elite Players Recall: 75.00%, Model got 75% of actual Elite players right.
- Much better at finding top tier talent, less Elite players misclassified as lower tiers.

Cons:

- Low Precision for Elite and Good to Great Players Elite Precision: 50.00%, When the model classified Elite players, it was right only 50% of the time.
- Good to Great Precision: 34.38%, Model often misclassified players in this group.
- Inconsistent classification for Good to Great players, Recall: 73.33%. Model found many Good to Great players but low precision (34.38%) means it also misclassified many players in this group.
- So not super reliable for Good to Great NFL prospects.

Box Plot Insights:

- The box plots show Below Average category has aligned predicted and actual scores. So it's good at predicting this category.
- For Elite and Good to Great players, the predicted success scores are kind of scattered. So it's hard to separate high level talent.

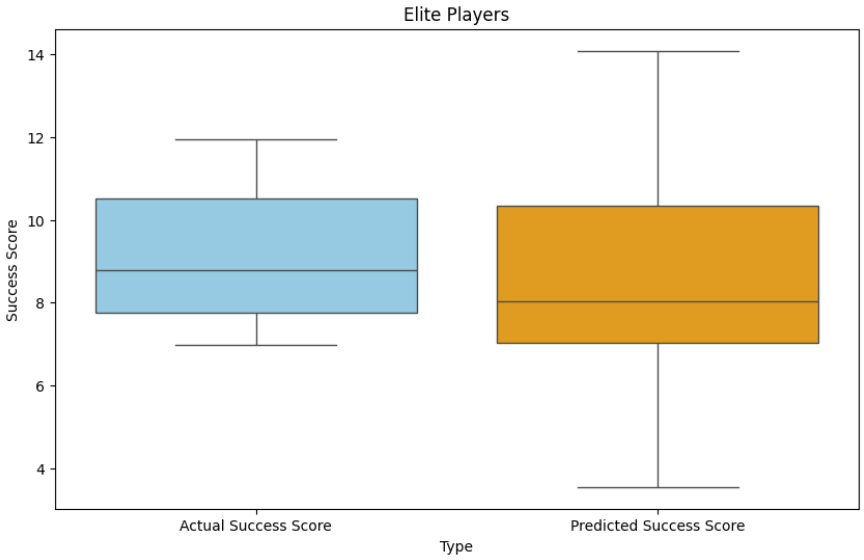


Figure 6: ltw1

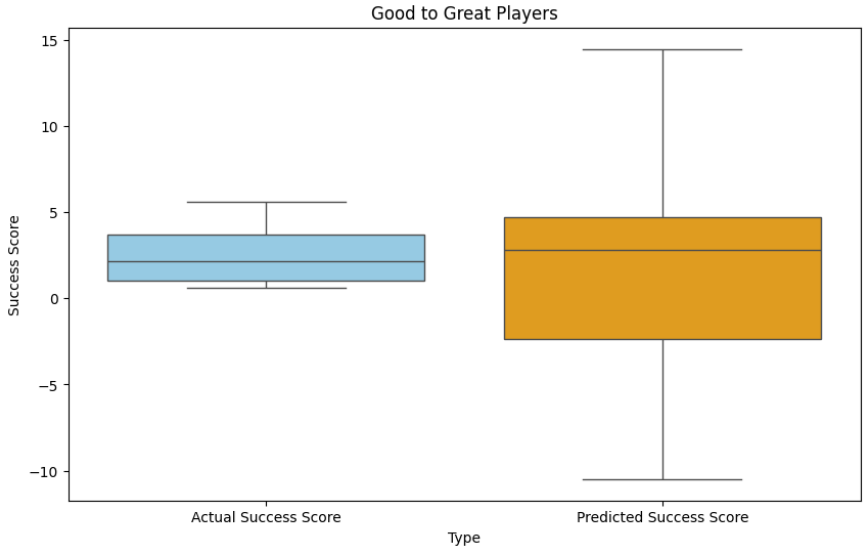


Figure 7: ltw2

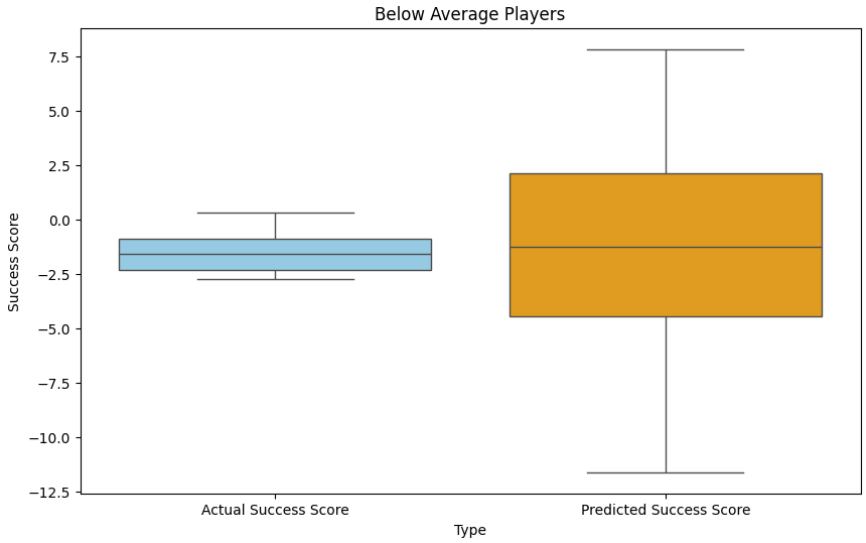


Figure 8: ltw3

4.2.3.1 Final Verdict LTW is the best model so far with 79.55% accuracy, 10.85% better than LT (69.70%). It's good at predicting Below Average players and much better at Elite players. Fewer top talents are missed, but not great precision for Good to Great and Elite players. By far the best appraoch to use

for accuracy though.

4.2.4 Evaluation of the Linear Cluster (LC) Model

LC model had an overall accuracy of 40.91%. It did worst than LT and LTW models especially in classifying Below Average and Good to Great players.

Pros:

- Okay at Identifying Elite Players
- 66.67% recall, Found two-thirds of actual Elite players in the dataset.
- Better than LT (14.29%) but not as good as LTW (75.00%).
- Good precision for Below Average players
- 73.91% precision when classifying player as Below Average.
- Recall for Below Average players was only 38.20% meaning it missed a lot of actual Below Average players.

Cons:

- Low Overall Accuracy
- 40.91% accuracy is much lower than LT (69.70%) and LTW (79.55%).
- Failed to classify players across all categories.
- Misclassifications in Good to Great Category
- Low precision for Good to Great players (27.12%), Many players predicted to be Good to Great were actually in a different category.
- Recall (43.24%) was also low, meaning it missed more than half of actual Good to Great players.
- This means the model can't distinguish between average and high talent.
- Inconsistent Predictions for Below Average Players
- Precision was decent (73.91%) but recall was very low (38.20%) meaning it missed many actual Below Average players.
- Doesn't consistently identify weaker prospects.

Box Plot Insights:

- Elite players had a wide range of predicted scores so the model was inconsistent in its classification.
- Good to Great players had a lot of variation so that's why the model struggled to classify them.

- Below Average players had spread out predictions so it's hard to trust the model's decisions for this group.

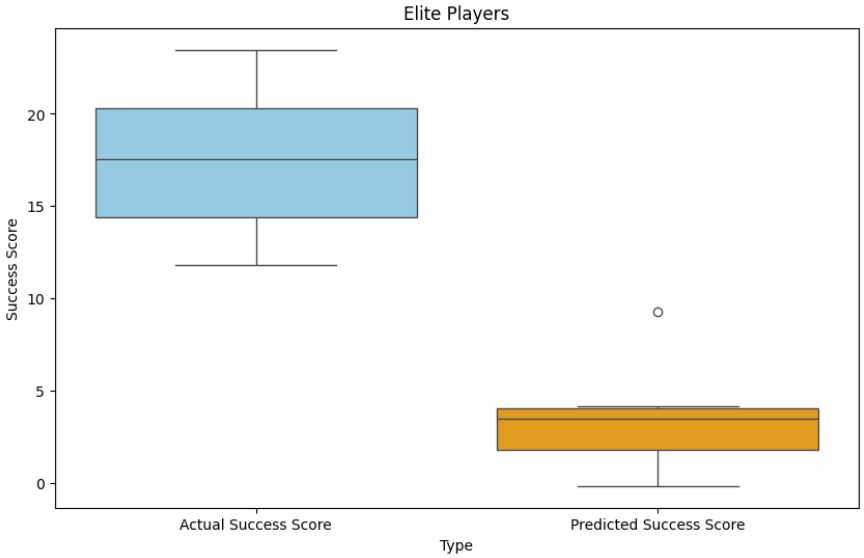


Figure 9: lc1

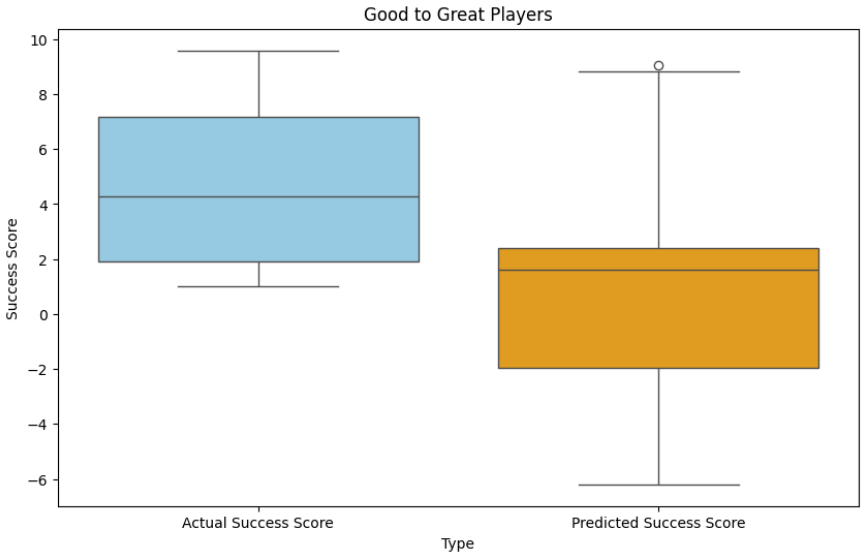


Figure 10: lc2

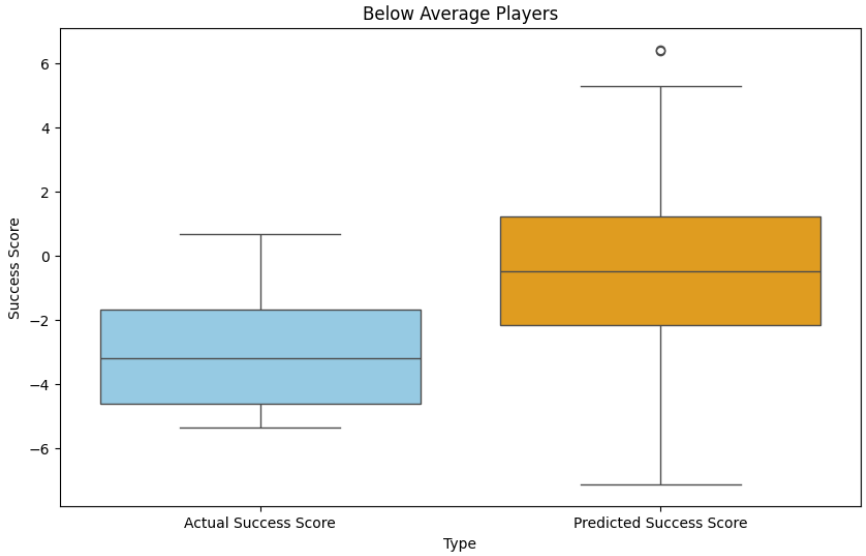


Figure 11: lc3

4.2.4.1 Final Verdict LC model had the 2nd lowest accuracy (40.91%). Overall, LC model is not reliable enough to predict NFL success and need to be improved before it can be useful.

4.2.5 Evaluation of the Linear Cluster Weighted (LCW) Model

The Linear Cluster Weighted (LCW) model had an overall accuracy of 37.12% which is the worst among all the models.

Pros:

- Really good Elite player detection
- 100% recall for Elite players, the model picked all actual Elite players. This means no Elite players were missed, but at a significant precision cost
- Moderate Precision for Below Average Players (77.50%). The model was correct about 78% of the time when predicting Below Average players

Cons:

- Poor Overall Performance
- Very low overall accuracy of 37.12%. Couldn't classify players consistently across categories
- Many players mislabeled as Good to Great, 33.33% recall. Missed many actual Good to Great players

- Inconsistent Below Average Player Identification
- Despite 77.50% precision, very low recall (35.23%) Missed about 65% of actual Below Average players

Box Plot Insights:

- Elite players actual and predicted scores have a large variation Indicates inconsistent predictions for top tier talent
- Good to great players actual and predicted scores have a wide spread, means the model struggles to distinguish this player category
- Below average players predicted scores have more spread than actual scores Means the model can't predict this category consistently

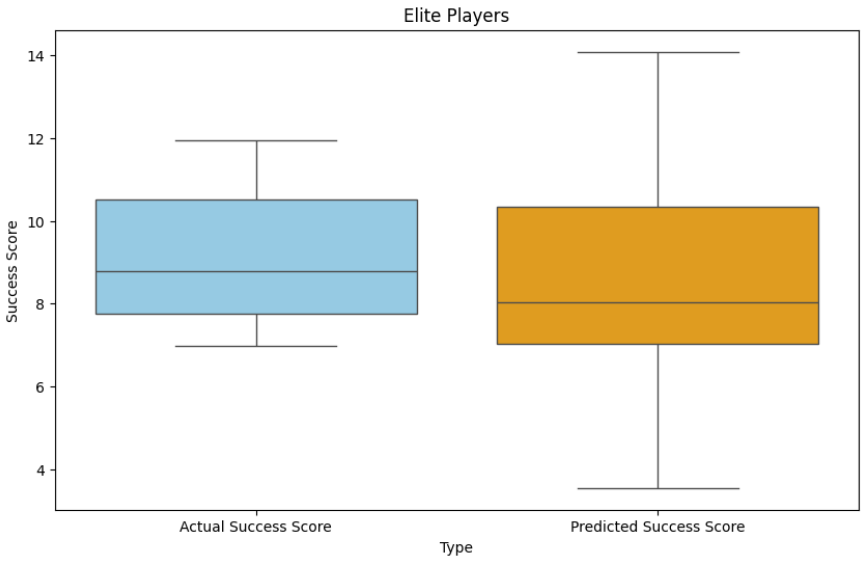


Figure 12: lcw1

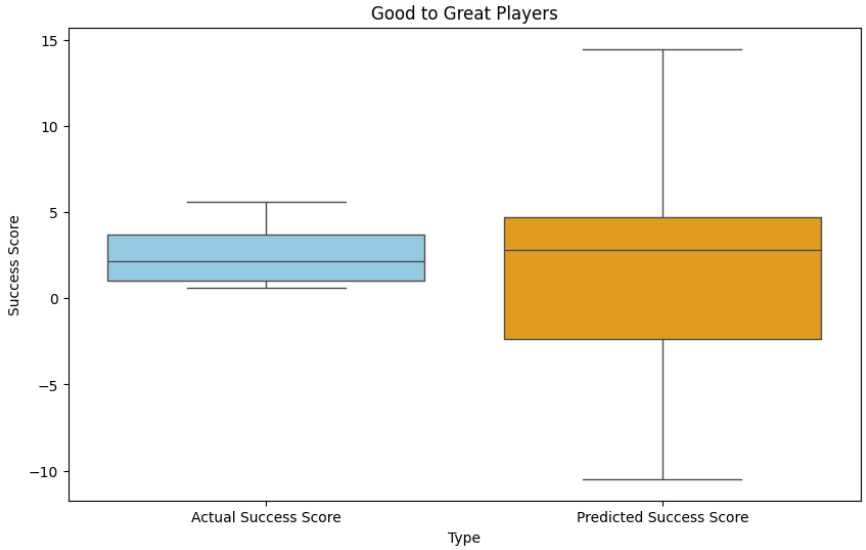


Figure 13: lcw2

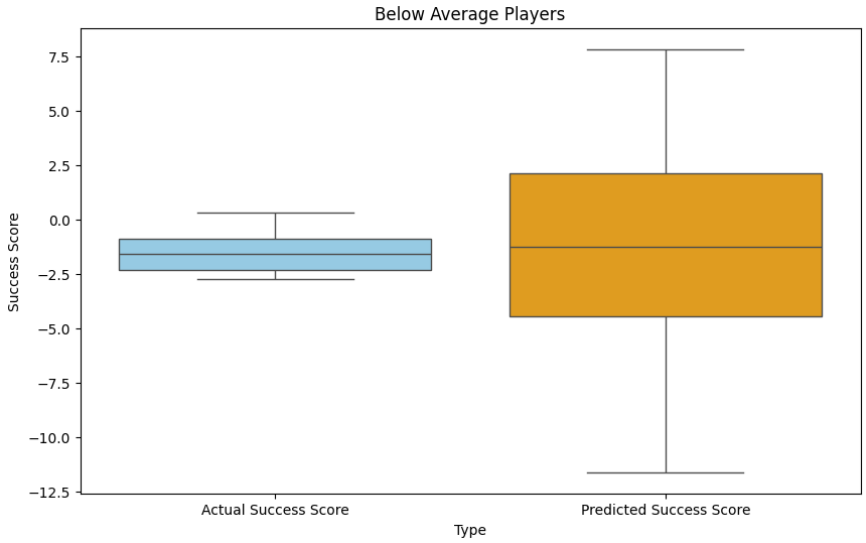


Figure 14: lcw3

4.2.5.1 Final Verdict The Linear Cluster Weighted (LCW) model is the worst model as far as acurracy. High recall for Elite players can't compensate for the poor overall accuracy and inconsistent performance across other categories. A lot of work is needed to make this model useful for NFL player success

prediction.

4.3 Threats to Validity/Limitations

While the models successfully achieve their intended purpose, there are some threats to validity that must be acknowledged.

- Data Quality and Availability: The model is only as good as the input data, including college statistics, body measurments and combine metrics. Missing or inconsistent data will introduce bias and limit accuracy.
- Feature Selection Bias: The chosen features may not capture all the factors that influence NFL success. Unaccounted for variables like psychological traits, team fit or injuries could impact player performance but are hard to measure.
- Limited Categorization: We only have three success categories, simplicity is good but precision suffers. Broader classification could help with identification and precision.
- Target Variable Interpretation: Raw NFL data needed to be transformed to define success categories. Since statistics like number of pro bowls, all team selections, or even years started don't fully capture success, some subjectivity was introduced when interpreting NFL success based on NFL statistics. Unpredictable situations like player peak, regression, and injury make it hard to directly correlate the statistics mentioned with success in the NFL. A player could be really good for 1 or 2 years and then have a big drop off in terms of production for the next 3 years. So then you would have to find a good middle between the two scenarios to accuratley interpret that players' success.
- Feature Metric Limitations: When gathering data for predicting NFL success for prospects, college data from last year of college was collected for each player. This tends to be the best performaing season for most players, but for some players, data can be skewed because of injury or other extreme circumstances like player opting out due to covid. For combine statistics not every prospect particpates in all the drills, so there a a lot of missing values in the body combines statistics datset that doesnt allow me to use some tests/drills.
- Dataset Skew: The dataset has way more Below Average and Good to Great players than elite ones. This could make the model favor predicting mediocrity over true top talent.
- Overfitting to historical data – Historical data is valuable but the model may be too tuned to past trends and not generalize well to future draft classes especially as the NFL evolves in play style, scouting and team strategies.

- Sample Size Limitations – Even with a big dataset, there are sample size constraints especially for players with shorter careers or from smaller programs and could skew predictions.
- Model Assumptions & Technique Limitations: Linear regression assumes simple relationships between variables which may not fully capture the complexity of NFL player success. More advanced techniques like decision trees or neural networks could model those non-linear interactions better.
- Engineered Feature Weights (LTW & LCW Models): Linear modeles with feature engineering performed better especially LTW. However manually adjusting these weights introduces bias and may not generalize well across different draft classes.

Addressing these threats through better feature selection, alternative modeling techniques and refined target variable definitions could improve the model's robustness and prediction power.

4.4 Summary of Results

Four linear regression based models were tested to predict success of wide receivers from college football to the NFL. These models, Linear Threshold (LT), Linear Threshold Weighted (LTW), Linear Cluster (LC), and Linear Cluster Weighted (LCW), were evaluated by accuracy, classification report and box plot analysis.

4.4.1 Results

- Best Model: Linear Threshold Weighted (LTW)
  - Accuracy: 79.55% (highest)
  - Verdict: Best model overall, good at Below Average and Elite. Needs work on Good to Great.
- Runner Up: Linear Threshold (LT)
  - Accuracy: 69.70%
  - Verdict: Good at Below Average but struggles with top end talent.
- Poor: Linear Cluster (LC)
  - Accuracy: 40.91%
  - Verdict: Inconsistent across all categories, not recommended for NFL success prediction.
- Worst: Linear Cluster Weighted (LCW)
  - Accuracy: 37.12% (lowest)
  - Verdict: High recall for Elite but overall accuracy is poor. Needs a lot of work, not recommened for NFL success prediction.

The Linear Threshold Weighted (LTW) model was the best. It had the highest accuracy and was good at predicting Elite and Below Average talent, but needs work on Good to Great talent. The Linear Threshold (LT) model was second, good at predicting Below Average talent but not good at predicting top end talent. The Linear Cluster (LC) and Linear Cluster Weighted (LCW) models has low overall accuracy and inconsistent at predicting across all categories, so they are not recommened for NFL success prediction.

In terms of feature weighting, the engineered coefficients in the LTW model performed better than the OLS coefficients. The manual assignment of weights allowed for more football context and intuition to be applied to the modeling. For example, conference strength (ConfRank) was heavily weighted because production against tougher competition matters a lot when projecting NFL success. Core performance metrics like receptions, receiving yards, yards per reception, and yards per game were highly weighted. Touchdowns and high school ranking were moderately weighted, while hand size, arm length, and wingspan were weighted much lower because while measurables are useful they shouldn't trump proven on field production and competitive context. By balancing these features based on domain knowledge rather than just OLS regression outputs, the LTW model performed the best.

5 Conclusion

This study presents four predictive linear models to predict how NFL draft prospects will perform in the NFL using a mix of current NFL statistics, combine results, college statistics, and body measurments. The goal is to help scouts and analysts make better draft decisions based on data.

The results show that the metrics used in modeling does indeed matter. The best model predicted 80% of the players correctly, which is far from perfect but way better than no data at all. For industry standard, I would say theat both the linear threshold weighted model accuracy (80%) and the linear threshold model accuracy (70%) both prove to be good accuracy in my regard. When dealing with this type of problem with predicting player success and trying to categorize the player into categories of talent is not an very easy sitaution to predict. In the future, I would like collect more data to train models, as well as provide more refined classification groups, that can pinpoint certain types of players better and lower the broadness of a player's predicted talent level.

One of the main challenges I encountered is that the relationship between the metrics used to predict how well college players would do in the NFL may not have been exactly linear to the NFL statistics used as a target/basis for these players. Testing this with historical data has helped me pinpoint this challenge, but based on the overall accuracy of my LTW model, both feature engineering and manual thresholding for classification have offsett this challenge, especially manual thresholding.

Even with challenges and limitations, this project proves a data-driven approach to evaluating NFL talent can be successful, especialy whne the models are featured egnineered. Being able to predict a player's success with the backing of raw statistics and data instead of just using traditonal scouting is a important step to revolutionizing the NFL draft process. As analytics becomes more mainstream in pro sports, models like this can support traditional scouting, making it less subjective and increasing the chances of drafting good players.

Another important reflection is how well this aligns with the original goals and objectives I set out. The original purpose was to create a decision support tool that would help NFL teams, scouts and media make smarter, data-driven draft decisions and identify undervalued talent that traditional scouting would miss. Throughout this study I was able to build models that combined college statistics, combine metrics, body measurements and scouting information to create a predictive system that supports this vision. The success of the linear threshold weighted model and the linear threshold model in particular shows that a data-driven approach can complement and strengthen traditional scouting methods and reduce subjectivity and bias in the player evaluation process.

Additionally, this was a real-world application of my data science, sports analytics and predictive modeling skills which was another big goal I set out to achieve. By tackling a complexive imperfect problem with a structured model driven approach, I was able to show how statistical and machine learning can be applied to sports decision making. Looking forward, the foundation built here will support further innovation in predictive sports analytics and will encourage

a more balanced mix of raw data and traditional football knowledge, as well as push the evolution of the NFL draft process forward.

5.1 Future Work

While this study has successfully demonstrated the potential of data-driven scouting for NFL draft prospects, there are several areas for future improvement and expansion.

One big area is to broaden the scope beyond just wide receivers. Expanding the model to include multiple positions like, quarterbacks, running backs, defensive players, etc, would give a more complete picture of what predicts success across the entire draft. Since different positions require different skills and attributes, refining the model for position-specific evaluations will make it more accurate and applicable.

Another area of improvement is to use more advanced machine learning techniques. This study was limited to linear regression models, but using decision trees, random forests, gradient boosting, and neural networks could help capture the complex and non-linear relationships between college performance, combine metrics and NFL success. These methods could uncover patterns that linear models might miss.

Collecting more data and richer data will be important too. Future versions of the model should include position-based advanced performance metrics like player speed, route efficiency and separation metrics for recievers and many other metrics for other specific positions. Contextual team-based data like scheme fit and coaching influence could help refine predictions as well. Adding qualitative factors like psychological assessments, leadership qualities and injury risk models would be very complex to implement, but would give an even more complete picture of a player's potential.

Another big improvement would be to make the player classification system more detailed. Right now the model sorts players into 3 broad categories but future versions should have more specific groups that capture different player types. This would give a more precise evaluation of their potential in the NFL rather than making general predictions about their talent level.

Validation will also be key going forward. Testing on upcoming new draft classes and tracking real time player performance will help gauge its accuracy and fine tune the approach. Updating the model with new data will keep it current as the NFL changes in style of play, talent evaluation and strategy.

Furthermore, real-time draft decision tools could be a game changer. A system where NFL teams can use predictive analytics in real time during the draft, adjust their strategy based on live data and model output could be a big improvement. AI-driven recommendations in the war room, room where the front office of teams meet to pick their draft pick, could complement traditional scouting and give teams an edge.

By continuously refining the model, expanding its scope, and leveraging more advanced analytics, this research has the potential to transform NFL scouting

and draft strategy, minimizing risk and maximizing the chances of selecting impact players.

5.2 Future Ethical Implications and Recommendations

As predictive analytics becomes more important in NFL scouting, we need to address the ethical implications to ensure fair and responsible use of these models.

5.2.1 Bias and Fairness

One of the biggest ethical issues is data bias. The model is trained on historical data which may reflect the biases in scouting, playing opportunities and player evaluations. If not managed properly, predictive models could perpetuate existing inequalities. Future work should focus on reducing these biases by having diverse datasets and exploring fairness aware machine learning techniques.

5.2.2 Privacy and Data Usage

Another big issue is the responsible collection and usage of player data. Advanced tracking systems, biometric data and psychological evaluations can improve model accuracy but also raise privacy concerns. Players should have control over their data and any implementation of these models should follow ethical data standards and relevant regulations.

5.2.3 Transparency and Interpretability

If NFL teams or organizations use AI to guide predictive analytics in their draft decisions, transparency will be key. Scouts, coaches and players should know how the predictions are made when using a deep learning AI. Improving model explainability with explainable AI will help stakeholders trust and use the insights more comprehensively.

5.2.4 Impact on Player Careers

Over reliance on predictive models could unfairly impact a player's draft stock. If a model says a player is below average, teams might pass on them without fully considering context, development potential or unmeasured attributes like work ethic and adaptability. Its important to make sure scouts are aware of this possibility.

5.2.5 Recommendations for Responsible Use

- Testing and Bias Mitigation – Run fairness tests to check for bias in data and models. Adjust as needed to ensure fair evaluations.
- Player Data Protection – Set guidelines for ethical data collection and compliance with privacy regulations.

- Interpret AI – Build models that provide transparency so teams and players know why predictions are made.
- Comprehensive Decision Making – Encourage NFL teams to use analytics as one of many tools not the only tool to make decisions.

By addressing these ethical concerns moving forward, predictive analytics can be used responsibly to enhance NFL scouting while ensuring fairness, transparency, and respect for all players' careers.

References

[1] 2020. Gold Jacket Spotlight: Gil Brandt, NFL Scout and Innovator. https://www.profootballhof.com/news/2020/12/gold-jacket-spotlight-gil-brandt-nfl-scout-and-innovator/

[2] 2025. Looking back on when Indy claimed the NFL Combine. *Axios Indianapolis* (2025). https://www.axios.com/local/indianapolis/2025/02/26/nfl-combine-history

[3] Owen Brower. 2020. Applying Analytics to the NFL Draft: Athletic Performance Measures as a Predictor of Future Success. https://jstor.org/stable/community.36513983 Accessed: 2024-12-12.

[4] Bernd Buchmasser. 2025. NFL Scouting Combine, explained: How does it work, and what does it mean for the Patriots? *Pats Pulpit* (2025). https://www.patspulpit.com/2025/2/25/24372460/nfl-combine-2025-schedule-workouts-position-groups-how-to-watch-patriots

[5] Panagiotis Charilaou and Rami Battat. 2022. Machine learning models and over-fitting considerations. *World Journal of Gastroenterology* 28, 5 (2022), 605–607. https://doi.org/10.3748/wjg.v28.i5.605

[6] E. Cherry, J. Witmondt, and F Woodlief. 2017. Year to Year Statistical Correlation in the NFL. https://wsb.wharton.upenn.edu/wp-content/uploads/2022/01/2017_NFL_Cherry_Yr2Yr-1.pdf Wharton Sports Business Initiative.

[7] Brad Congelio. 2020. Advanced Model Creation with NFL Data. https://bradcongelio.com/nfl-analytics-with-r-book/05-nfl-analytics-advanced-methods.html

[8] V. Dimitrieviski. 2023. AI-Generated Predictions for the 2023 NFL Draft. https://www.snaplogic.com/blog/ai-generated-predictions-2023-nfl-draft. Accessed: 2024-12-12.

[9] T. Drovetto. 2023. How the NFL Combine Works: The Drills. Seahawks Home. https://www.seahawks.com/news/how-the-nfl-combine-works-the-drills-151331 Accessed: 2024-12-12.

[10] Brock Grassy. 2017. Applying Machine Learning to Predict the NFL Draft. https://www.brockgrassy.com/blog/applying-machine-learning-to-predict-the-nfl-draft/

[11] Sam Iannuzzi. 2022. Brandt, Computer Helped Revolutionize Scouting. *Dallas Cowboys Official Website* (2022). https://www.dallascowboys.com/news/brandt-computer-helped-revolutionize-scouting

[12] Purple Insider. 2024. Why is it so hard to predict which college QBs will succeed in the NFL? https://www.purpleinsider.football/p/why-is-it-so-hard-to-predict-which

[13] Robert Mays. 2019. The Eight Most-Telling NFL Team Draft Tendencies. *The Ringer* (2019). https://www.theringer.com/2019/04/18/nfl-draft/most-telling-nfl-team-draft-tendencies-patriots-vikings-steelers-giants

[14] A. J. Meil. 2018. Predicting Success Using the NFL Scouting Combine. https://scholarworks.calstate.edu/downloads/1z40kt64t. Accessed: 2024-12-12.

[15] The Draft Network. 2024. There's A Continuity Problem In Quarterback Development. https://www.thedraftnetwork.com/2024/11/13/nfl-cfb-quarterback-development-problem

[16] Julius Ogunleye. 2022. *Predictive Data Analysis Using Linear Regression and Random Forest.* https://doi.org/10.5772/intechopen.107818

[17] NFL Football Operations. 2024. The rules of the NFL draft. https://operations.nfl.com/journey-to-the-nfl/the-nfl-draft/the-rules-of-the-draft/. Accessed: 2024-12-12.

[18] NFL Football Operations. 2025. Getting Into the Game. https://operations.nfl.com/journey-to-the-nfl/the-next-generation-of-nfl-stars/getting-into-the-game/

[19] Sports Reference. 2025. Conference Index. https://www.sports-reference.com/cfb/conferences/

[20] Sports Reference. 2025. NFL Draft. https://www.pro-football-reference.com/years/2019/draft.htm

[21] MIT Sloan Management Review. 2018. Is It Possible to Judge Individual Talent in the NFL? https://sloanreview.mit.edu/audio/is-it-possible-to-judge-individual-talent-in-the-nfl/

[22] Z. Reynolds, T. Bonds, S. Thompson, and C. W. Lecrom. 2015. An Evaluation of the NFL Draft as a Predictor of Team Success. *Journal of Applied Sports Management* 7, 3 (2015), 74–86.

[23] D. Robinson. 2024. How does the NFL salary cap work? NFL salary cap explained. *Pro Football Network* (August 1 2024). https://www.profootballnetwork.com/how-does-nfl-salary-cap-work/

[24] C. Ryan. 2024. Does the NFL draft actually promote competitive balance? *Vanderbilt Business School* (October 9 2024). https://business.vanderbilt.edu/news/2024/10/09/does-the-nfl-draft-actually-promote-competitive-balance

[25] Dalwinder Singh and Birmohan Singh. 2020. Investigating the impact of data normalization on classification performance. *Applied Soft Computing* 97 (2020), 105524. https://doi.org/10.1016/j.asoc.2019.105524

[26] Brian Szekely, Christian Sinnott, Savannah Halow, and Gregory Ryan. 2023. NFL Career Success as Predicted by NFL Scouting Combine. https://arxiv.org/abs/2303.05774

[27] R. Tucker and W. Black. 2021. The National Football League Combine: Do Performance Measures Predict Draft Status Among NFL Draftees. *The Sport Journal* 27 (2021).

[28] Aryan von Eicken. 2017. An Analysis of the Efficiency of the NFL Draft. *Economics Student Theses and Capstone Projects* 41 (2017). https://creativematter.skidmore.edu/econ_studt_schol/41

[29] Jenny Vrentas. 2016. Gil Brandt: The Draft Guru Who's Seen It All. *Sports Illustrated* (2016). https://www.si.com/nfl/2016/04/26/gil-brandt-nfl-draft-history

[30] Haizhou Wang and Mingzhou Song. 2011. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal* 3, 2 (2011), 29–33. https://journal.r-project.org/archive/2011/RJ-2011-015/RJ-2011-015.pdf

[31] Luke A. Yates, Zach Aandahl, Shane A. Richards, and Barry W. Brook. 2022. Cross validation for model selection: A review with examples from ecology. , e1557 pages. https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/ecm.1557 Accessed: 2024-12-12.