Boosting Draft Accuracy: A Two-Stage Classifier–Regressor Approach for NFL Wide Receiver Prospect Evaluation

Aadi Patangi

Amador Valley High School

ABSTRACT

The NFL draft is an opportunity for teams to draft new players, address positional needs, and strengthen their roster for the upcoming season. Teams often trade players or compensation packages to secure certain prospects, which can be critical for team success. Wide receivers were selected as the focus of this study because the position is heavily influenced by objective data — such as receiving statistics and combine metrics like speed, agility, and explosiveness — making it well-suited for predictive modeling using machine learning. This study presents a two-stage machine learning approach to first predict whether a wide receiver (WR) invited to the NFL Scouting Combine will be drafted, and if so, at which overall position. Using physical testing data from the NFL combine, college production statistics, and historical draft result training data from 2000 to 2024, we construct a Gradient Boosting Classifier to predict draft likelihood followed by a CatBoost Regressor to estimate draft position for those predicted to be selected. This approach provides NFL teams and scouts with a reliable estimate of whether and when a combine-invited wide receiver will be drafted, helping them make more informed decisions and strategically position themselves to select desired prospects. In validation, our classifier reached 89.2% accuracy ($F_1 = 0.936$), and our regressor yielded a 49.2-pick MAE ($\rho = 0.626$), demonstrating robust predictive performance.

Keywords: predictive analytics, NFL draft, wide receivers, sports analytics, machine learning

Introduction

In recent years, the wide receiver has emerged as a crucial position for the success of teams in the National Football League (NFL). NFL teams strive to have star wide receivers on their roster, which can often transform a team's offense into a high-powered, game-changing threat that can stretch the field and exploit defensive mismatches. Wide receivers not only need to run various routes but also have the ability to jump and catch passes and occasionally block. While teams can sign wide receivers through free agency, many teams prefer to draft core receivers out of college. Many of the college football prospects each year, including wide receivers, are invited to the NFL Combine if they receive an invitation from the Player Selection Committee based on college performance. Although only select players are invited to the combine, a strong showing can significantly elevate a player's draft stock, while a weak performance can just as easily cause it to fall. Thus, it's possible to quantify and predict whether a player gets drafted or not and if so, at what pick based on their college performance and their performance at the combine if they got invited.

Past studies have used machine learning (ML) models to forecast binary football-related outcomes. Primarily, Clark et al. (2013) constructed a logistic regression model using field goal play-by-play data from the 2000-2011 NFL seasons to identify influential factors for field goal success and found that psychological factors have no statistically significant effects on the probability of making a field goal whereas environmental factors such as temperature, precipitation, and wind speed have a far more significant impact. Many predictor variables used in the study such as weather indicators have a high degree of correlation and field goals themselves are random events affected by human performance and this randomness is impossible to capture with a model. Additionally, a majority of data consists of high-success probability scenarios as most field goals are likely attempted if there's a high chance of making them. A study by Gifford & Bayrak (2023) used all regular season game data from the 2002-2018 seasons and constructed a logistic regression model and a decision tree model to identify the most important predictors of NFL game outcomes. They achieved 83% validation accuracy with logistic regression in predicting NFL game outcomes and found that turnovers lost on offense, turnovers forced on defense, total yards on offense, and total yards allowed on defense were the most significant indicators. However, the study doesn't account for statistics that are derived from causation such as aggressive vs. conservative play style and doesn't account for psychological factors affecting gameplay such as confidence and shifts in momentum. A similar study by Baker & Kwartler (2015) constructed two logistic regression models to predict the outcome of a play—whether a pass or run was executed—using play-by-play data from the 2000–2012 seasons for the Cleveland Browns and Pittsburgh Steelers. The model achieved 66.4% accuracy in predicting Cleveland's plays and 66.9% accurate for Pittsburgh, representing only a slight improvement over random guessing. The study did not employ a validation set and all data points were used to train the model, which slightly flaws its accuracy. Additionally, team-based models might not be the strongest indicator of play-calling compared to team personnel or coaching staffs, and the study doesn't account for weather conditions which can affect play-calling. It is a promising study that can inform the strategies of teams and anticipate play selection.

Additionally, ML models have also been used for regression tasks in football, not involving binary outcomes and rather predicting the numerical values of a variable. A study by Taylor (n.d.) utilized both a shallow convolutional neural network (CNN) and transfer learning with the VGG19 image model to predict the offensive play call and predict the yardage outcome of a play using play-by-play data along with images of the field before the snap of each play for plays that only resulted in a yardage outcome (avoiding penalties, punts etc.). Using a cost function of mean absolute yards, both models with tuned hyperparameters struggle to predict yardage outcomes with an improved performance versus guessing the median of all training data for all plays, and the image data also contributed no significant value. However, both the shallow CNN and VGG19 transfer learning had improved accuracy (both 0.606 accuracy) versus the benchmark (guess all plays are pass) for predicting play call but preformed slightly worse than a simpler random forest model without image data (0.614 accuracy). A similar study by Teich et al. (2016) also predicted yardage outcome of a play as well as well as the progress metric which calculates a scaled score between 0 and one based on the current down, achieved yards, and yards to go as a measure of success. For example, achieving a 1st down receives a score of 1, and achieving yards but not reaching a 1st down gets a decimal which is smaller on later downs, and not reaching a 1st down on a 3rd or 4th down receives a 0. All the machine learning methods used to predict yards gained in a play resulted in a Mean Average Error (MAE) greater than 5 and an RMSE greater than 8, which is a high error margin considering that 10 yards is a 1st down. However, all ML models on the progress metric only had a MAE of 0.15 or less and a RMSE of 0.25 or less. While regression tasks in the NFL are inherently difficult, using a scaled and context-aware metric like progress makes prediction more meaningful and can limit prediction errors.

Very few papers actually predicted the order of the NFL draft. A study by Mulholland & Jensen (2014) used very similar predictor variables to this study to predict the draft order of tight ends using a linear regression model and a decision tree model. For the decision tree, college yards and the 40-yard dash seemed to be the most important predictor variables while the linear regression chose the 40-yard dash and the bench press as the most important predictor variables. Although the study achieved an RMSE of 56.52 using decision trees, the model only predicts draft placement on tight ends who are known to be drafted. Although they briefly explored a logistic regression model to predict whether a tight end is drafted, that analysis lacks depth and doesn't mention any coefficients or model diagnostics. Although they state that 40-yard dash time was the most significant predictor and that four of six combine measures were selected in the final model, the absence of detailed metrics prevents a thorough evaluation of the model's predictive power. Although 40-yard dash is identified as the most significant predictor, the lack of quantitative details limits the interpretability of the model. A more thorough analysis—such as reporting feature weights or applying cross-validation—would enhance the practical value of their findings. Finally, Dhar (n.d.) constructed a linear regression model and a recursive partitioning regression tree (CART) model to predict draft order for wide receivers. Since the study uses only drafted receivers from 1999-2008 for training and also discards receivers without college data available online, only 266 training data points were used. Surprisingly, the linear regression model that combined college and combine variables with a R^2 of 0.302 had a 40-yard dash coefficient of 227.5, suggesting that a 0.1 second decrease in the 40-yard dash results in a 22.75 slot improvement in predicted draft placement. The primary split in the CART was total college receiving yards at a threshold of 1627 yards with players above being "over-achievers" and players below being "under-achievers". However, running a fast 40-yard dash could improve projected draft position for "under-achievers" confirming the NFL bias toward 40-yard dash dash times. The study never uses any testing data, and all reported metrics came from the training data. Thus, no prior research has addressed predicting whether a wide receiver invited to the combine will be drafted, or estimating their draft position with proper evaluation on unseen data.

Methods

Data

Using the nfl_data_py Python library, we compiled NFL Combine data for all wide receivers who attended between 2000 and 2024. Data for the 2021 season was not included in the study, as it came from college pro days and not from the invitation-only NFL combine. In addition to combine metrics, the dataset included each player's height, weight, college, draft year, and draft pick (if applicable). The draft year and draft pick fields were NaN for undrafted players. College football receiving statistics were compiled from the 1999 to 2023 through the Sports Reference website. The chosen fields were total receptions, total yards, touchdowns and yards per game. This left us with 1152 data points out of which 170 (14.6%) data points did not have any recorded college stats on the Sports Reference website.

Variable	Description
calculated_forty	Draft class–ranked 40-yard dash time (lower raw time ranked higher)
calculated_bench	Draft class–ranked bench press repetitions (higher repetitions ranked higher)
calculated_vertical	Draft class–ranked vertical jump (higher jump ranked higher)
calculated_broad_jump	Draft class–ranked broad jump (higher jump ranked higher)
calculated_cone	Draft class–ranked 3-cone drill time (lower time ranked higher)
calculated_shuttle	Draft class–ranked shuttle run time (lower time ranked higher)
REC	Number of receptions in final college season
YDS	Total receiving yards in final college season
Y_R	Yards per reception in final college season
TD	Receiving touchdowns in final college season
Y_G	Receiving yards per game in final college season
ht_inches	Player height in inches
wt	Player weight in pounds
conf_ACC	Indicator variable for Atlantic Coast Conference
conf_Big 12	Indicator variable for Big 12 Conference
conf_Big Ten	Indicator variable for Big Ten Conference
conf_Other	Indicator variable for non–Power Five conferences or independent programs
conf_Pac-12	Indicator variable for Pac-12 Conference
conf_SEC	Indicator variable for Southeastern Conference

Table 1. Feature names and descriptions

Data Preparation

Before the creation of our models, the data was cleaned to remove all NaN values and dummy variables were created. The dataset included a draft_ovr field, which specified the overall draft position for each player if they were drafted and was otherwise missing (NaN). We created a binary classification target variable, is_drafted, by assigning a value of 1 to players with a non-missing draft_ovr and 0 to those with a missing value. We used the IterativeImputer from the scikit-learn library, which fits a Bayesian Ridge regression model to estimate and iteratively fill in missing values. Each variable with missing data is modeled as a function of the other variables, and the imputation process is repeated for several iterations to improve accuracy. We created one imputer for power and explosiveness metrics (e.g., height, weight, bench press) and a separate one for speed/agility metrics (e.g., 40-yard dash, 3-cone drill, shuttle run). While the 40-yard dash has high feature importance, only 63 (5.47%) players were missing it and they were all from early years. The rest of the features we imputed had minimal feature importance compared to the 40-yard dash. Since college reception stats had a high feature importance and there's no logical way to impute it, we decided to remove all records without college stats leaving us with 982 records being used in the study. Additionally, for players that played multiple years in college before attending the combine, only their most recent year of college before attending the combine was kept in the study to ensure consistency. For all metrics except height and weight, we applied feature ranking to standardize values and ensure consistency across different draft years. This approach accounts for evolving athletic standards over time and reduces model variance as different features have different scales. For example, a player who recorded the fastest 40-yard dash in his draft class received a score of 1 in the calculated_forty feature. This ranked value was used in place of the raw time to reflect the fact that wide receivers have become increasingly athletic over the years, and prospects are ultimately evaluated relative to others in their draft class—not across eras. The logic accounted for the fact that a greater value in some features is beneficial while a smaller numerical value in other features is better. If two prospects tie, then they both receive that ranking. Height and weight were not ranked due to there not being a census on the optimal ranges. Additionally, height and weight are subjective and highly dependent on play style and scheme fit, so we thus avoided misleading assumptions into the model. College reception stats were also not ranked because we removed many prospects who attended the combine but did not have college receiving statistics available on Sports Reference, as doing so would result in inflated rankings for those with data available that ignore prospects without data available and introduce bias into the model. We created the dummy variables conf_ACC, conf_Big 12, conf_Big Ten, conf_Pac-12, conf_SEC, and conf_Other which are a 0 if a player does not attend a school in that conference and a 1 if they attend a school in that conference. Table 1 lists all feature names and their labels that were used in the study. Table 2 lists the statistics for all the features used in this study.

Variable	Role	Mean	Standard Dev.	Non-missing	Missing	Min.	Median	Max.	Skewness	Kurtosis
calculated_forty	INPUT	23.32	14.09	982	0	1	22	63	0.23	-0.89

calculated_bench	INPUT	23.46	13.91	982	0	1	23	63	0.22	-0.86
calculated_vertical	INPUT	23.37	14.01	982	0	1	23	63	0.22	-0.88
calculated_broad_jump	INPUT	23.32	13.91	982	0	1	23	62	0.24	-0.86
calculated_cone	INPUT	23.49	13.94	982	0	1	23	62	0.26	-0.85
calculated_shuttle	INPUT	23.51	13.85	982	0	1	23	63	0.22	-0.84
REC	INPUT	56.9	25.06	982	0	1	55	158	0.42	0.12
YDS	INPUT	826.4	360.4	982	0	3	799.5	2060	0.27	-0.15
Y_R	INPUT	14.73	3.15	982	0	3	14.35	29.6	0.7	1.88
TD	INPUT	6.73	4.16	982	0	0	6	25	0.74	0.62
Y_G	INPUT	70.17	27.57	982	0	0.3	68.2	187.3	0.23	0.05
ht_inches	INPUT	72.67	2.25	982	0	65	73	78	-0.23	-0.23
wt	INPUT	200.99	15.04	982	0	149	201	247	-0.12	-0.07
conf_ACC	INPUT	0.1	0.31	982	0	0	0	1	2.58	4.65
conf_Big 12	INPUT	0.1	0.3	982	0	0	0	1	2.63	4.93
conf_Big Ten	INPUT	0.13	0.33	982	0	0	0	1	2.26	3.13
conf_Other	INPUT	0.36	0.48	982	0	0	0	1	0.59	-1.65
conf_Pac-12	INPUT	0.12	0.33	982	0	0	0	1	2.28	3.19
conf_SEC	INPUT	0.19	0.39	982	0	0	0	1	1.62	0.62

Table 2. Feature statistics

Model Creation

In order to predict draft likelihood and draft position for those predicted to be drafted, we built two models, a classification model and a regression model. For both models, we initially used LazyPredict to test multiple models for their baseline performance, and those with the best results were chosen. We then performed hyperparameter tuning with a grid search through the GridSearchCV class within scikit-learn. The final classifier selected was a Gradient Boosted Decision Trees (GBDT) model, which significantly outperformed baseline models such as logistic regression and Support Vector Machines (SVM). A GBDT classifier combines multiple decision trees into a single model, where each tree is trained to be maximally correlated with the negative gradient of the loss function, associated with the entire ensemble (Natekin & Knoll, 2013). The tree recursively partitions data to determine which traits a player should have in order to have the best chance to be drafted. Trees are added sequentially, with each one trained to correct the errors made by the ensemble learned so far. We use the default cross-entropy loss, which optimizes probabilistic classification and is also used in logistic regression. To determine the success of the classification model, we looked at validation accuracy with classifying players as drafter or not. We also evaluated the F1 score which helps us understand how well the model balances precision (how many predicted drafted players were correct) and recall (how many players were captured as drafted) (Vakili et al., 2020). This penalizes the model if it is biased towards drafted players and fails to perform on the minority undrafted players in the data. The equation is shown below where TP = true positives, FP = false positives, and FN = false negatives.

$$\text{F1 Score} = 2 \cdot \frac{\left(\frac{\text{TP}}{\text{TP} + \text{FP}}\right) \cdot \left(\frac{\text{TP}}{\text{TP} + \text{FN}}\right)}{\left(\frac{\text{TP}}{\text{TP} + \text{FP}}\right) + \left(\frac{\text{TP}}{\text{TP} + \text{FN}}\right)}$$

For our regression model, we used a CatBoost regressor to predict the draft order for prospects predicted to be drafted. CatBoost outperforms other boosting implementations such as XGBoost and LightGBM due to its use of ordered boosting, which prevents target leakage and prediction shift during training by ensuring that each model update is based only on past data in a permutation of the training set. In addition, CatBoost handles categorical variables natively through ordered target statistics, which convert categorical features into numeric values using historical data, thereby avoiding leakage while supporting high-cardinality features. For each tree split, CatBoost also constructs combinations of categorical features already used in previous splits, allowing the model to capture higher-order interactions between variables (Prokhorenkova et al., 2019). We optimized the CatBoost model using the Mean Absolute Error (MAE) loss function, which represents the average number of draft picks the model is off by from the actual draft outcome.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

We also used R^2 to assess the model and it explains the proportion of variance in draft position explained by the model. An R^2 of 0 would indicate that our model is no better than predicting the mean draft pick for all players, while an R^2 of 1 indicates perfect predictions. Since we use a nonlinear regressor (GBDT), R^2 still reflects the proportion of variance explained, while MAE provides additional context on prediction error. Finally, we decided to also use Spearman's rank correlation coefficient as it provided the correlation coefficient between the rankings of draftable players versus their order of being drafted, and thus, is not as affected by the uneven distribution of picks each round. The Spearman coefficient ranges from -1 to 1 and is used to measure any monotonic relationship between variables that do not follow an approximately normal distribution (Rovetta, 2020). For example, in 2020, while our model predicted Justin Jefferson to be drafted at 21.77 (retain decimals to evaluate model), he ended up being drafted at 22. This would make the MAE 0.23 while Justin Jefferson would contribute 0.000724 to the spearman coefficient as he was predicted to be the first WR drafted by the model but ended up being the 5th WR taken in the 2020 draft while 51 total WRs attended the combine according to this formula:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}$$

Model Training and Evaluation

We used Leave-One-Season-Out Cross-Validation (LOSOCV), in which each season (2020, 2022, 2023, and 2024) was held out once as a test set while the remaining seasons excluding 2021 were used for training. In each fold, both the classifier and regressor were retrained using the same hyperparameters. Results were reported separately for each season to assess year-by-year performance. This cross-validation strategy was chosen to account for changes in prospects and drafting styles over time. It provided more consistent performance estimates compared to standard train-test splits or splitting by seasons. The classifier was trained on all players in the training data. We plotted validation accuracy against the number of estimators for the classifier to identify the optimal number of boosting stages. However, to ensure proper evaluation of the regression model, the regressor was trained only on players who were actually drafted in the training set. This avoided penalizing the regressor for classifier errors and ensured that evaluation metrics such as Mean Absolute Error (MAE) and R^2 reflected only cases where a draft position existed. Figure 1 shows how the model operates on unseen data. First, raw wide receiver data is cleaned and transformed through feature engineering, then passed into a binary classifier to predict draft status. If the player is classified as drafted, the data is further processed by a regression model to estimate their draft position.

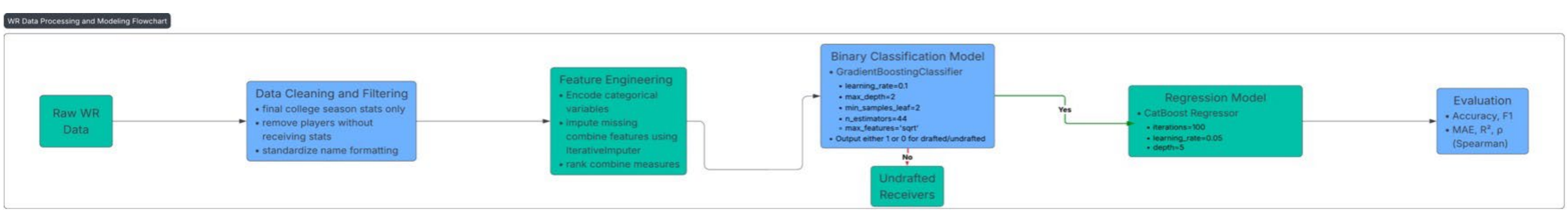

Figure 1. Model Flowchart

Results

The results of the study are summarized in Table 3. For both the classifier and the regressor, 2024 was the best year of performance for both models while 2023 was the worst year for the classifier, and 2020 was the worst year for the regressor based on the MAE, R^2, and Spearman coefficient. According to PFF, the 2023 draft class did not feature a standout receiver and none of the top three prospects were nearly as developed as those from previous years (Monson, 2023). Notably, Jaxon Smith-Njigba, widely regarded as the second-best receiver in the class, was predicted to go undrafted by our classifier. This likely occurred because he did not run the 40-yard dash at the NFL Combine—his unofficial Pro Day time was not included in the dataset—and because he missed most of the 2022 season with a hamstring injury, recording only 5 receptions for 43 yards. As our model uses only final-season statistics for consistency, this limited his feature data considerably. Additionally, the imputed 40-yard dash was beneficial and prevented dumping lots of early-season data but was not perfect even though it was based on other speed/agility metrics. No solid evidence was found for the weak regressor performance in 2020. However, the strong performance for both models in 2024 is likely due to the 2024 draft class being one of the strongest in the last decade, but predicting draft position has more confounding variables compared to predicting whether a player is drafted or not (Sikkema, 2024). While the regressor occasionally misranks individual players compared to other mock drafts, it consistently succeeds in distinguishing between early- and late-round prospects. For instance, the model's top predicted wide receiver prospects in order were Brian Thomas, Xavier Legette, Malik Nabers, Rome Odunze, Troy Franklin, and Marvin Harrison Jr. Another interesting player was George Pickens, who tore his ACL in 2021 and missed most of his junior season, and finished with only 5 receptions for 107 yards after returning for the latter part of the season. He was predicted to be drafted at 115.76 by our model and was ranked as the 21st best receiver by our model, but he ended up being picked at 52 by the Pittsburgh Steelers as the 11th WR taken in his class. This discrepancy highlights several limitations in using purely performance and combine data for draft prediction. For Pickens, both behavior concerns and his ACL tear contributed to a wide range of team evaluations. According to Bachar (n.d.), at least one NFL team removed Pickens from their draft board due to concerns about his behavior during a pre-draft visit. Despite these concerns, the Steelers were willing to draft him and overlook his injury history and other pre-draft concerns, valuing his high upside as a prospect. While the model penalized him for lack of recent production, it couldn't account for team-specific evaluations of his character, which can vary from team to team.

Year	Classifier Accuracy	Classifier F1 Score	MAE on Drafted	R² on Drafted	Spearman ρ
2020	0.8431	0.8824	66.80	-0.0037	0.3006
2022	0.8056	0.8727	52.26	0.1635	0.4200
2023	0.6042	0.6667	55.15	0.1303	0.3562
2024	0.8919	0.9355	49.18	0.3233	0.6262
Avg	**0.7862**	**0.8393**	**55.85**	**0.1533**	**0.4257**

Table 3. Model results

Discussion and conclusion

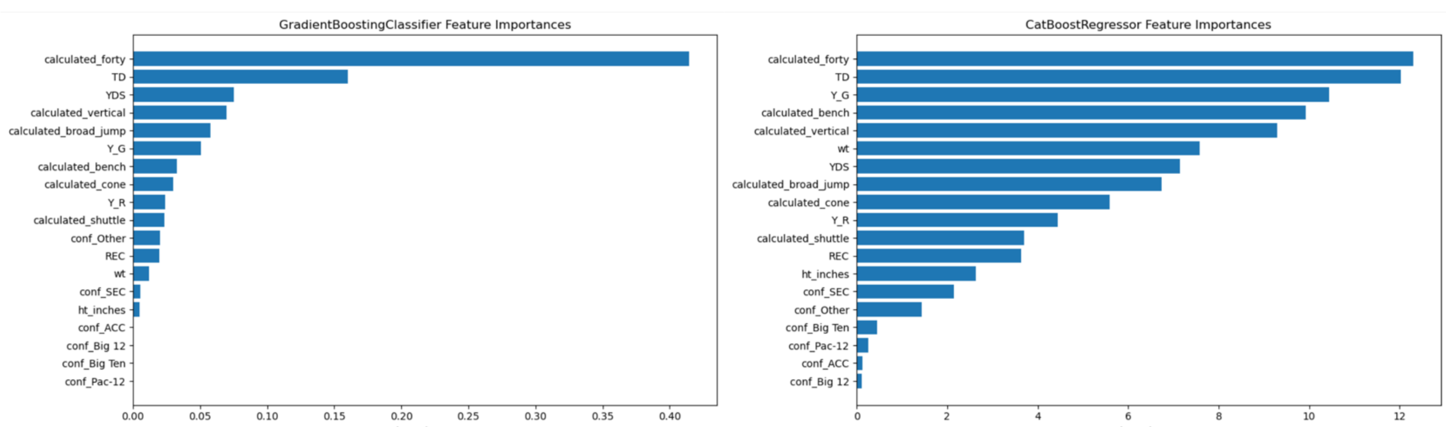

Figure 2. Feature Importance

Our two-stage modeling approach showed that predicting whether a WR gets drafted or not with a high level of accuracy is possible and predicting the exact draft slot is also possible, but much harder to do and is easily affected by outlying variables. The classifier achieved high accuracy across seasons, especially in 2024, indicating strong ability to identify whether a WR would be drafted. The regressor being able to predict the predict the overall draft position of a player within 49.18 picks indicate that for prospects with no anomalies in college or in the combine, predicting their draft round within 2 rounds is possible. Further tuning of the data, such as removing players who were injured in their final season, and more data for more recent years with less missing data could lead both models to become significantly more accurate and can offer an can offer an unbiased, statistical ranking of WRs, free from media or player hype. According to Figure 2, the 40-yard dash was the most important predictor for both models, followed by college touchdowns. Both models' reliance on college statistics and combine measures and not being able to quantify features like injury history, pre-draft interviews, and play style/team fit make the model weaker and show the problems associated with a purely-data approach. Regression is also a much harder task as it is impacted by team needs, position depth, and off-field behavior, which is unquantifiable. For this reason, the Spearman coefficient tells a different story: while the predicted draft positions may not always be numerically close, the overall ranking of receivers produced by our regressor was similar to the true draft order. Our average Spearman coefficient of 0.4257 suggests a moderate correlation between the actual draft order of WRs and the order predicted by our model. Unlike MAE which penalizes large errors and punishes predictions off by a few rounds, the Spearman shows relative rankings of receivers and is useful when picks aren't as important. Teams in need of a receiver want baseline rankings for prospects, and can further do research on prospects based on the non-biased rankings of the model. They can also use the regressor's predicted draft slot for a player to gauge where they end up being drafted, and can use this to trade up in order to secure the player. Prospects looking to boost their stock can also refer to the feature importance rankings to identify which areas of performance matter most and where to focus their improvement. Ideally, reducing the MAE to within one round (MAE ≤ 32), or even lower, would make the predictions more practical. For example, misclassifying a WR3 as the WR6 might yield an MAE of 50 or 60, but the relative ranking of the prospect is still informative—especially considering that 50+ wide receivers typically attend the combine. This level of approximation can still provide meaningful value to NFL front offices during draft evaluations. Potential areas of improvement for the model include using Natural Language Processing (NLP) to gain sentiment analysis on prospects, although this would only really be effective for the top prospects and could overshadow those from smaller schools. Using multiple years of college reception data for players who have it can also be useful, especially for players that ended up being injured or had minimal production in their final season of college football. Additionally, grading each NFL team on their need for wide receivers in a given draft class based on roster depth could improve the regressor's ability to predict actual pick order by accounting for team-specific drafting behavior. This approach can be applied to other positions in the NFL but may be more difficult due to the lesser importance given to the combine and greater emphasis on intangibles for positions other than WR. It can also be applied for predicting draft outcomes in other sports. Still, this methodology holds promise for broader draft analysis in football and other sports.

Acknowledgements

I would like to acknowledge Prasad Sharanabasawa for his invaluable opinions and guidance throughout the course of this project.

References

Bachar, Z. (n.d.). *George Pickens Reportedly Was Removed from Multiple NFL Teams' Draft Boards in 2022*. Retrieved June 28, 2025, from https://bleacherreport.com/articles/25194649-george-pickens-reportedly-was-removed-multiple-nfl-teams-draft-boards-2022

Baker, R. E., & Kwartler, T. (2015). Sport Analytics: Using Open Source Logistic Regression Software to Classify Upcoming Play Type in the NFL. *Journal of Applied Sport Management*, *7*(2). https://doi.org/10.7290/jasm07c14e

Clark, T. K., Johnson, A. W., & Stimpson, A. J. (2013). *Going for Three: Predicting the Likelihood of Field Goal Success with Logistic Regression*.

Dhar, A. (n.d.). *Drafting NFL Wide Receivers: Hit or Miss*.

Gifford, M., & Bayrak, T. (2023). A predictive analytics model for forecasting outcomes in the National Football League games using decision tree and logistic regression. *Decision Analytics Journal*, *8*, 100296. https://doi.org/10.1016/j.dajour.2023.100296

Monson, S. (2023, April 4). *2023 NFL Draft Rankings: Sam Monson's top 15 wide receivers, including USC's Jordan Addison at No. 1*. PFF. https://www.pff.com/news/draft-2023-nfl-draft-wide-receiver-rankings-sam-monson-jordan-addison-jaxon-smith-njigba-quentin-johnston

Mulholland, J., & Jensen, S. T. (2014). Predicting the draft and career success of tight ends in the National Football League. *Journal of Quantitative Analysis in Sports*, *10*(4). https://doi.org/10.1515/jqas-2013-0134

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*. https://doi.org/10.3389/fnbot.2013.00021

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2019). *CatBoost: Unbiased boosting with categorical features* (No. arXiv:1706.09516). arXiv. https://doi.org/10.48550/arXiv.1706.09516

Rovetta, A. (2020). Raiders of the Lost Correlation: A Guide on Using Pearson and Spearman Coefficients to Detect Hidden Correlations in Medical Sciences. *Cureus*. https://doi.org/10.7759/cureus.11794

Sikkema, T. (2024, April 22). *2024 NFL Draft Position Rankings: Wide receivers*. PFF. https://www.pff.com/news/draft-2024-nfl-draft-position-rankings-wide-receivers

Taylor, C. (n.d.). *Deep Learning for In-Game NFL Predictions*.

Teich, B., Lutz, R., & Kassarnig, V. (2016). *NFL Play Prediction* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1601.00574

Vakili, M., Ghamsari, M., & Rezaei, M. (2020). *Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification* (No. arXiv:2001.09636). arXiv. https://doi.org/10.48550/arXiv.2001.09636