Predicting Draft Round For College Football Players

Chris Kourkoulakos
Brown University Class of 2025
GitHub Repo

Introduction

This project uses college football player statistics to predict the round of the NFL draft a player was selected in, or if they went undrafted. The NFL draft is crucial for team improvement, as it helps select new talent and provides trade collateral. Accurately predicting draft rounds enables teams to optimize their strategies and make better trade decisions.

The data is taken from Stathead, Pro Football Reference's premium database service, with college football player stats dating back to 1956. Six total datasets – one for passing, rushing, and receiving statistics for drafted and undrafted players – were merged to create the final dataset. The datasets focused on the top 1,000 – 5,000 players by touchdowns in each category. Many players only appeared in one dataset so after merging, any missing values were set to zero since any minimal statistics they might have had would be inconsequential for determining a player's draft round. For example, a receiver that threw one touchdown pass on a trick play would not appear in the top 1,000 passers so after merging, their passing statistics would be null, then replaced with zero. On the other hand, if a dual-threat quarterback appears in the rushing dataset, he would retain his rushing data after merging.

Given the NFL's changing draft formats, the data was filtered only to include players in the current 7-round draft structure, which began in 1994. After filtering, we were left with 8,825 players. For this project, Round 0 represents undrafted players.

Exploratory Data Analysis (EDA)

One of the most important findings from our EDA was that the final dataset was imbalanced.

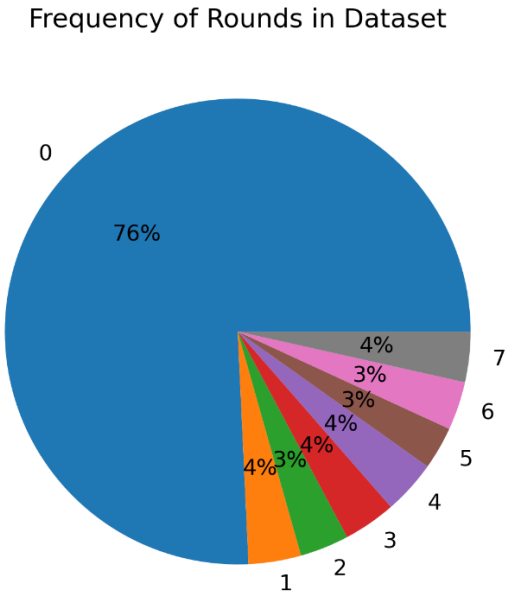

Figure 1 - Pie chart showing proportion of data in each class; indicates imbalanced dataset

This gives us an indication of how we should split our data. We see that the majority class is Round 0 and the rest of the data is fairly evenly split across all rounds. Similarly, we can also examine how our data is structured by round and position.

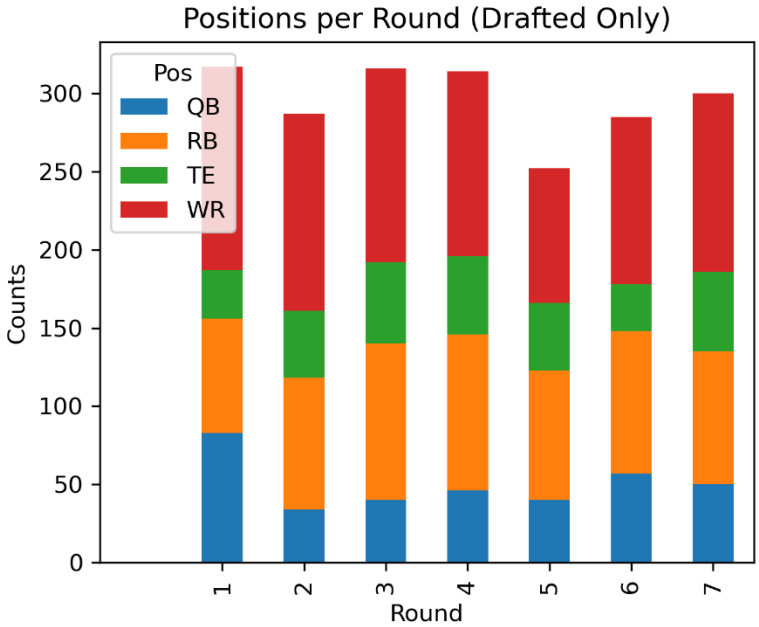

Figure 2- Bar graph depicting count of position by draft round, excluding undrafted players

For each draft round, receivers comprised the largest position group. This trend was also true for undrafted players (graph not shown). We can further explore our target variable with some of our numerical features.

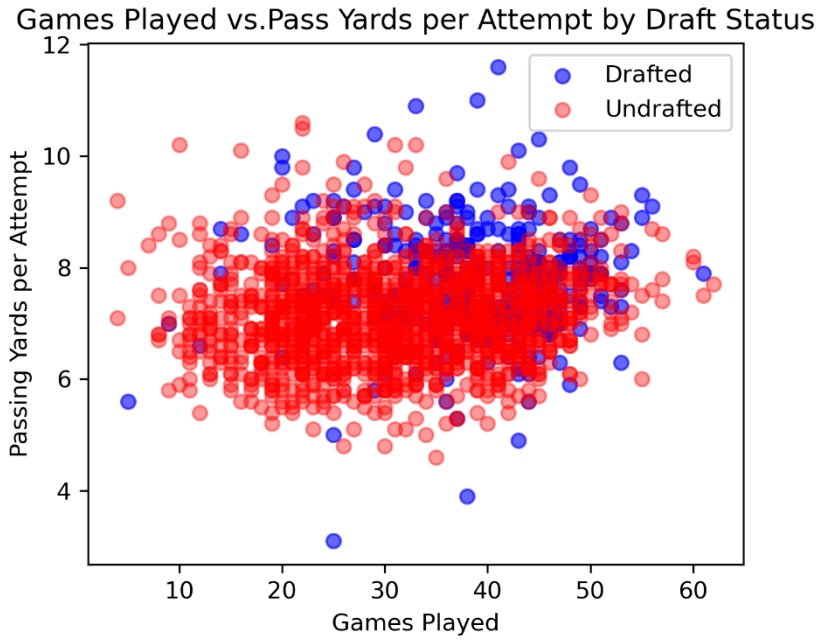

Figure 3- Scatterplot of games played and passing yards per attempt for drafted and undrafted players

There was no clear decision boundary between drafted and undrafted players by games played and passing yards per attempt; however players that played more games and had higher passing yards per attempt tended to be more frequently drafted than undrafted. This makes sense with our prior knowledge about the context. We can also examine if there are trends over time for our statistical categories.

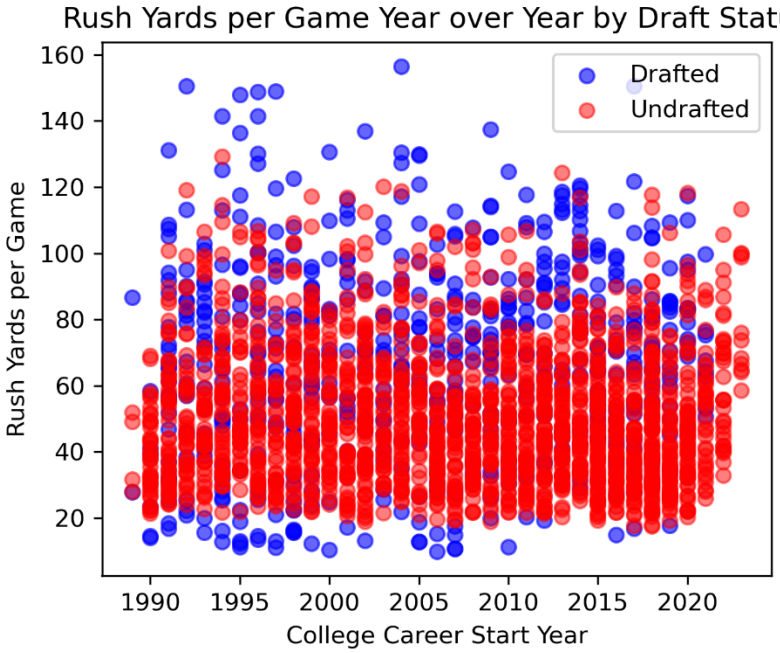

Figure 4- Scatterplot of rush yards per game over time for drafted and undrafted players

There is no significant trend between rush yards per game over time. The variability decreased slightly over time but there does not appear to be a clear difference in the relationship between drafted and undrafted players. Generally, players that had higher rush yards per game tended to be drafted more often than not, which is again what we would expect in this context. Finally we can examine a feature in the last major statistical category: receiving.

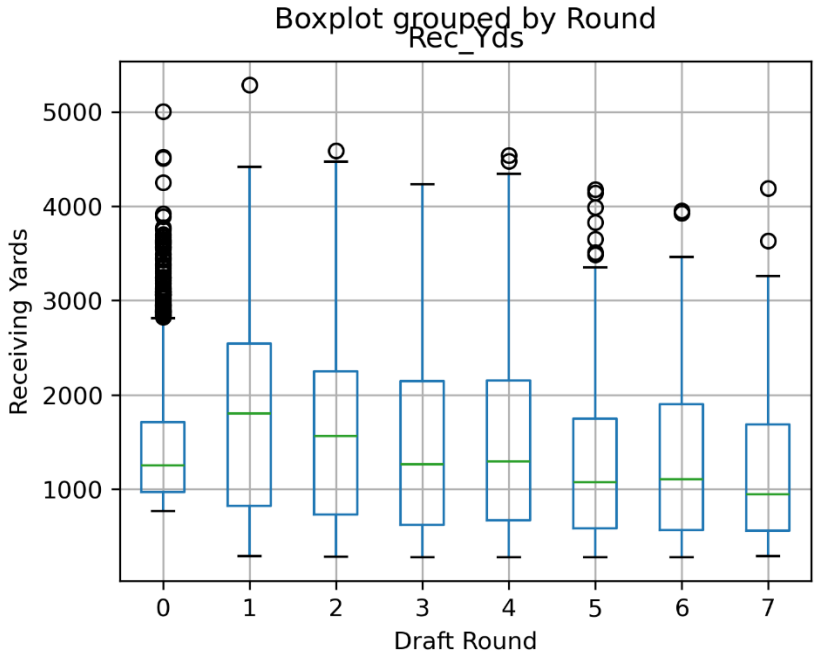

Figure 5- Boxplot of receiving yards in each draft round

Overall the distribution of receiving yards in each round was fairly similar. Undrafted players had lower variability but many outliers on the upper end. First-round picks had the highest median receiving yards and contained the maximum value, which is consistent with our background expectations. The distribution of rounds two through seven are fairly similar, so using receiving yards alone to predict draft round is likely not very predictive.

Our EDA reveals some general differences between drafted and undrafted players across various features. While no single feature clearly predicts draft round, the combination of all features in our dataset should allow the models to better identify trends and make more accurate predictions than random guessing or always selecting the majority class.

Methods

Since our data is imbalanced, we used a stratified split. We created an 80/20 split of training and testing data. On the training data, we applied StratifiedKFold cross validation with 4 splits, across three random states to account for uncertainty. With nearly 9,000 observations, four splits seemed to provide a good balance between capturing general trends of the data and minimizing computational load. We preprocessed the data using a one-hot encoder (position and year variables), standard scaler (non-proportion continuous variables), minmax scaler (percentage variables), and multilabelbinarizer (team variable). After preprocessing, we applied the standard scaler to all features to ensure a mean of 0 and a standard deviation of 1. Finally, we applied four ML classification algorithms: XGBoost, Random Forest, Support Vector Classifier (SVC), and K Nearest Neighbors (KNN). To evaluate our models' performance, we used accuracy. Accuracy, though typically not ideal for imbalanced datasets, is suitable here due to the problem's context. Misclassifying a player's draft round—whether above or below their true round—has similar consequences, unlike scenarios like cancer detection, where false negatives are far costlier than false positives. Since incorrect predictions are roughly equally costly in our case, we prioritized correct predictions.

The table below displays the hyperparameters we tuned. We used log-space when necessary to apply our models on different orders of magnitude for appropriate hyperparameters.

Model	Hyperparameter Values
XGBoost	learning_rate: [0.03] reg_alpha: [0.001, 0.01, 0.1, 1, 10] reg_lambda: [0.001, 0.01, 0.1, 1, 10] max_depth:[1, 3, 10, 20]
Random Forest	max_depth: [1, 3, 10, 50] max_features: [0.5, 0.75, 1.0]
SVC	gamma: [0.001, 0.01, 0.1, 1, 10] C: [0.01, 0.1, 1, 10, 100]
KNN	n_neighbors: [1, 3, 5, 10, 15] weights: ['uniform', 'distance'] metric: ['minkowski', 'manhattan', chebyshev']

Table 1 – Table of trained hyperparameter values for each algorithm

The table below shows the results of the best hyperparameters for each random state, as well as the mean validation score across all folds.

Model	Random State	Best Hyperparameters	Mean Validation Score
XGBoost	1	learning_rate: 0.03 reg_alpha: 0.1 reg_lambda: 1 max_depth: 10	0.77152975
XGBoost	2	learning_rate: 0.03 reg_alpha: 0.01 reg_lambda: 10 max_depth: 20	0.77025496
XGBoost	3	learning_rate: 0.03 reg_alpha: 1 reg_lambda: 10 max_depth: 10	0.77365439
Random Forest	1	max_depth: 10 max_features: 0.5	0.77223796
Random Forest	2	max_depth: 10 max_features: 0.5	0.77082153
Random Forest	3	max_depth: 10 max_features: 0.5	0.77223796
SVC	1	gamma: 0.001 C: 10	0.76898017
SVC	2	gamma: 0.001 C: 10	0.76742210
SVC	3	gamma: 0.001 C: 10	0.76813031
KNN	1	n_neighbors: 10 weights: ‘uniform’ metric: ‘manhattan’	0.76033994
KNN	2	n_neighbors: 10 weights: ‘uniform’ metric: ‘manhattan’	0.75934844
KNN	3	n_neighbors: 10 weights: ‘uniform’ metric: ‘manhattan’	0.75949008

Table 2 – Table of best hyperparameter combinations for each algorithm in each random state

Results

The baseline accuracy for this dataset was about 75.74%, meaning that if we were to predict every player as undrafted, we would be correct about 75.74% of the time. All models performed better than the baseline, but only marginally. The mean and standard deviations of the test accuracy for each of our best models across the three random states are depicted in the table and figures below.

Model	Mean Test Accuracy	Standard Deviation Test Accuracy
XGBoost	0.76883853	0.00487384
Random Forest	0.77110482	0.00166794
SVC	0.76694995	0.00463375
KNN	0.75939566	0.00070664

Table 3 – Table of mean and standard deviation accuracy on test set for each algorithm across all random states

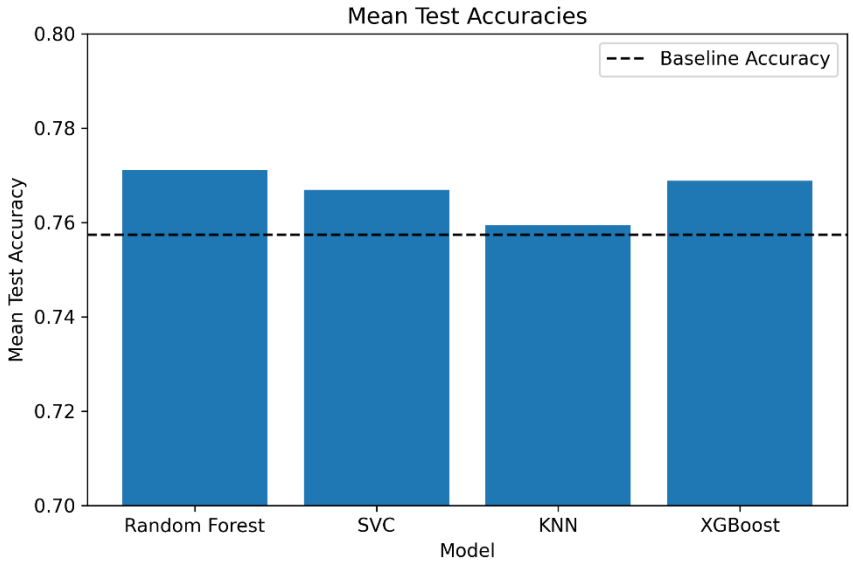

Figure 6- Mean test accuracy for each model across all random states, plotted relative to baseline accuracy

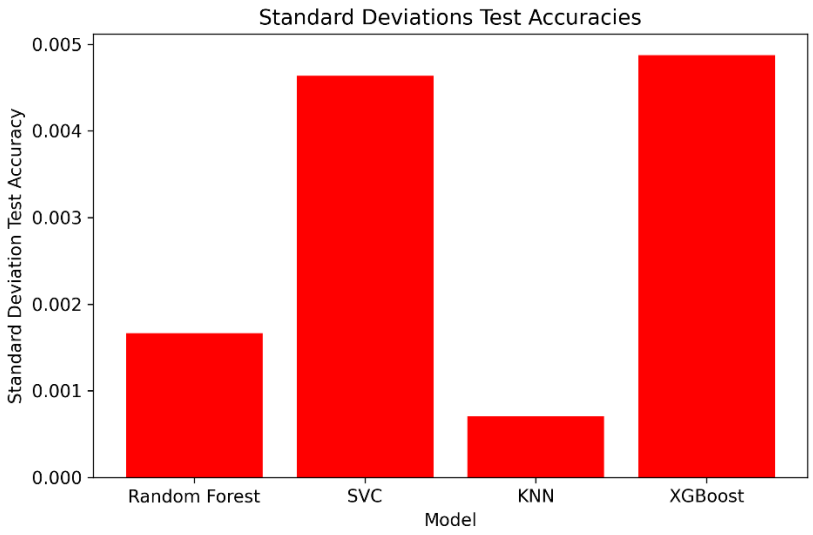

Figure 7- Standard deviation test accuracy for each model across all random states

Random Forest classification performed the best on our dataset, although it did not display very strong predictive power; the mean test score was less than 2% better than the baseline. This means that the Random Forest model was marginally better at predicting what round college football athletes would be drafted in than simply predicting all players as going undrafted. We can use feature importance to examine which variables the model deemed were the most critical in making predictions.

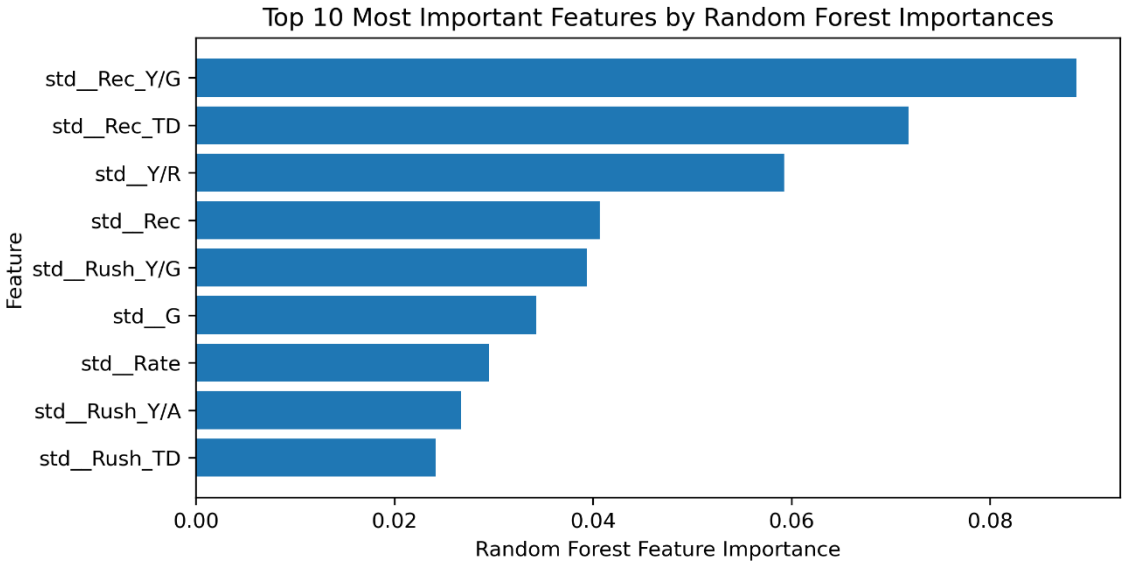

Figure 8- Most important features by Random Forest's built-in feature importance metric

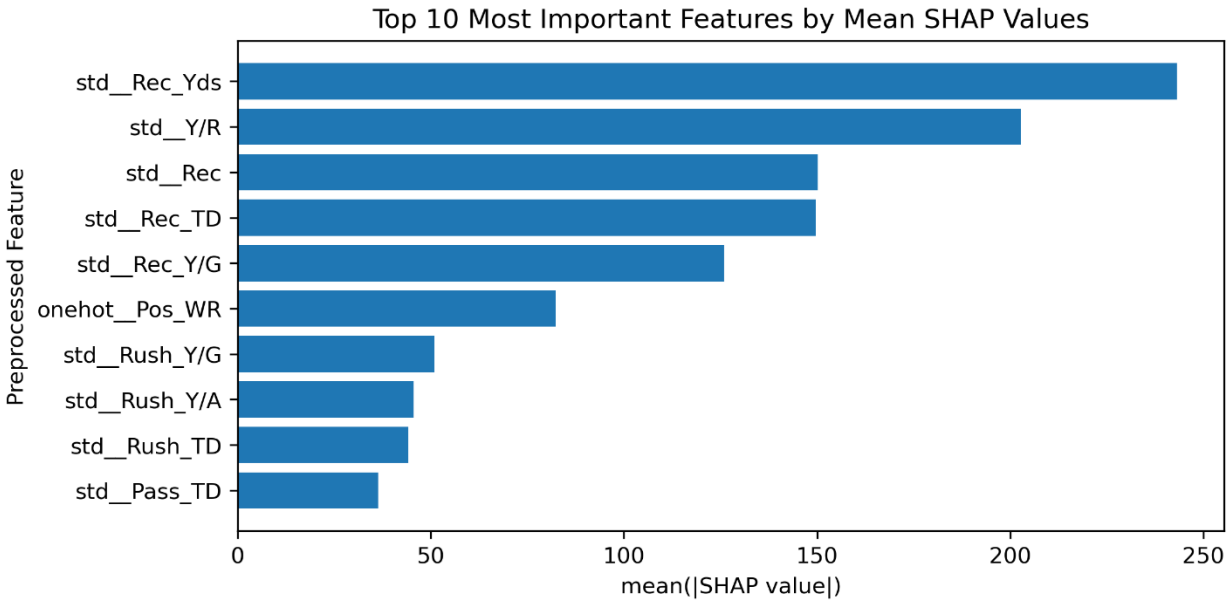

Figure 9 - Most important features by mean SHAP values

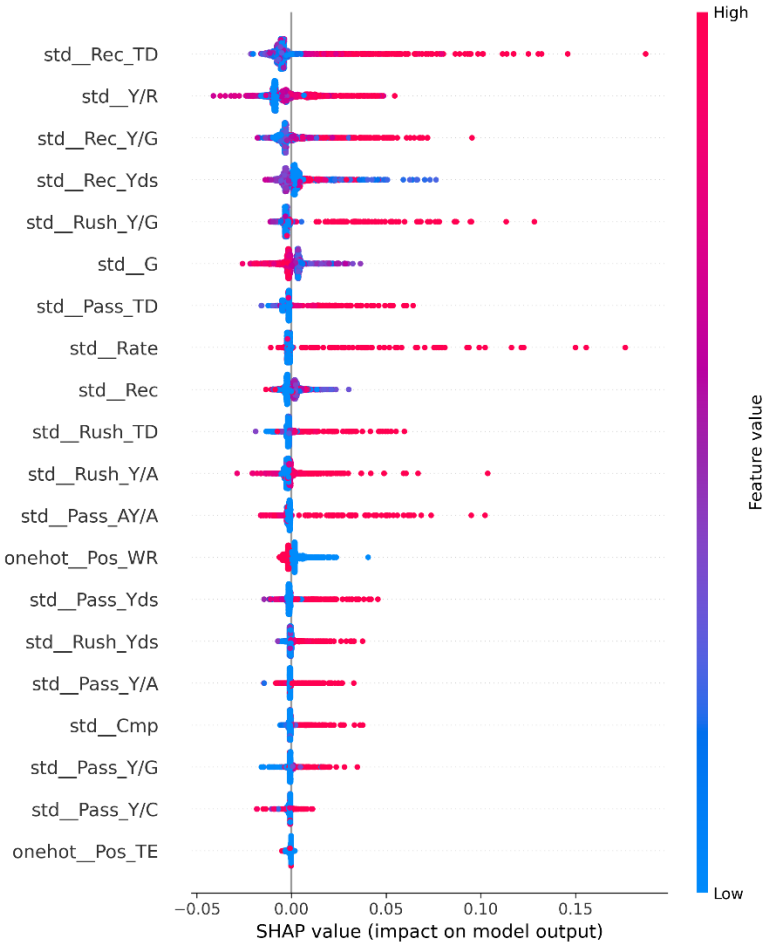

Figure 10 - Global SHAP value for each feature for predicting Round 1

The model seemed to place a strong emphasis on receiving metrics. The top four features by Random Forest feature importance and the top five features by mean SHAP values were all receiving statistics, highlighting the model's reliance on receiving data in its predictions. Global SHAP values for Round 1 predictions reinforced this finding, with higher SHAP values for players excelling in receiving touchdowns and receiving yards per game, while average receiving performers had SHAP values near zero. We also see that across all the feature importance metrics, the model relied more on continuous data than categorical data. There are some position variables listed, however the vast majority of features displayed are numerical. We do not see any team variables performing among the most important features, which is an interesting finding. One might think that of two players with similar statistics, the player on the better team might be drafted higher. Given the figures above, we do not see substantial evidence to support this claim.

We can further examine the model by looking at the confusion matrix.

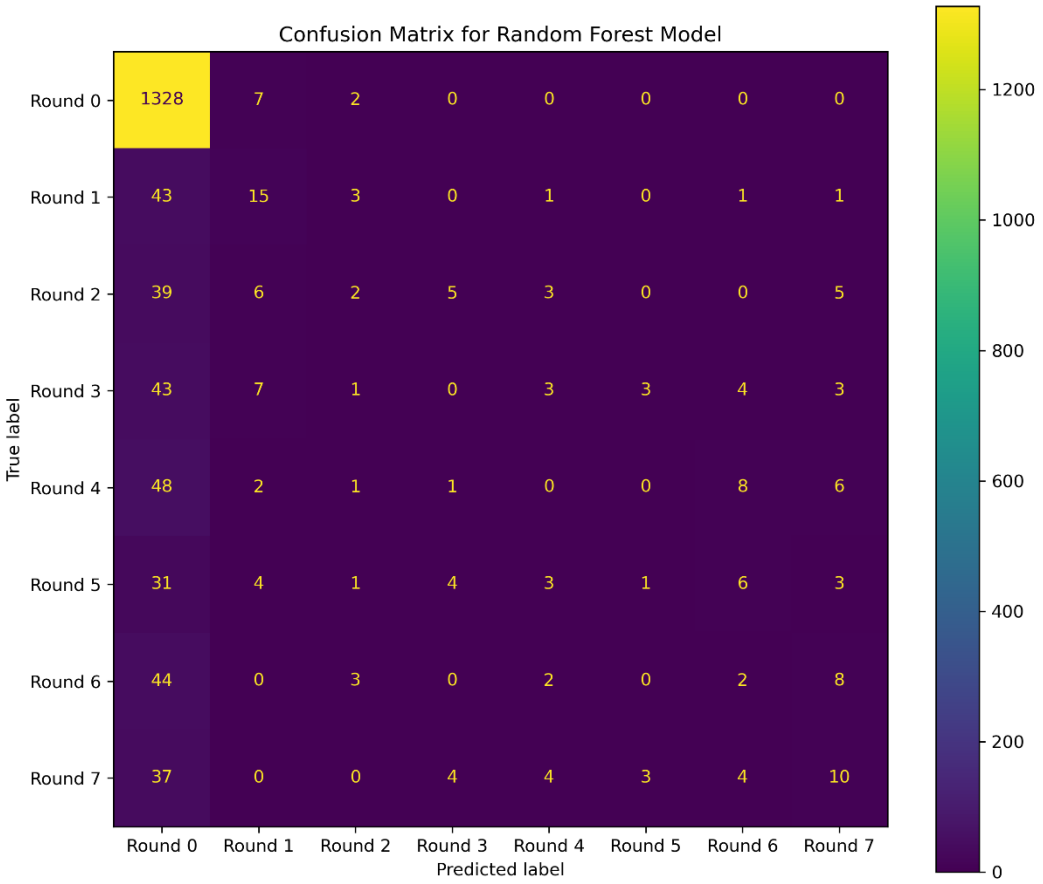

Figure 11- Confusion matrix for Random Forest Model

As expected with an imbalanced dataset, the model accurately predicts many undrafted players, but most errors involve falsely predicting players as undrafted. Following undrafted players, the model performs best with first-round picks, likely because these players excel across multiple statistics, making them easier for the model to identify. Notably, the model failed to correctly predict any third or fourth round selections and made few accurate predictions for other middle rounds. Individual points offer further insights into the model's behavior.

We can examine the model even further by looking at individual points.

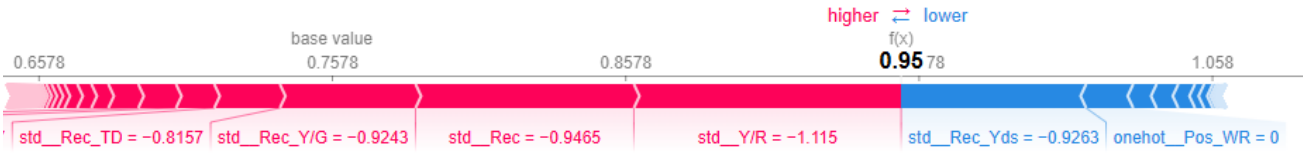

Figure 12- Force plot for an undrafted player

The force plot above depicts the model's predicted probability for a particular undrafted player. As seen in Figure 1, the baseline prediction for undrafted players was roughly 76%. For this specific player, the

model predicts a 95% chance of going undrafted. This prediction was pushed upwards by many receiving statistics such as yards per reception, receptions, and receiving yards per game, however the predicted probability was slightly offset by the receiving yards and position variables. Interestingly, this player is not a receiver, despite receiving statistics being the key predictors for their draft outcome. For models with performance close to the baseline, local interpretability may offer limited insights. Examining a first-round draftee could provide a more meaningful context.

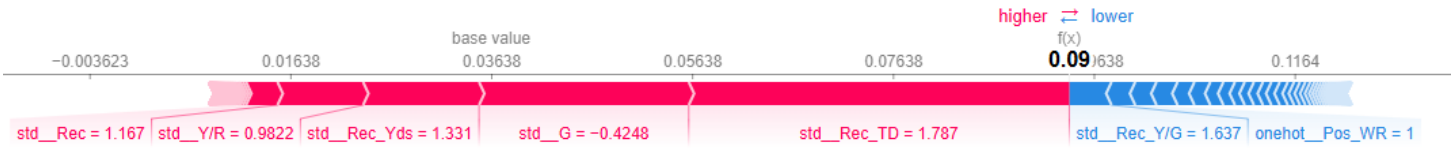

Figure 13- Force plot for a player drafted in Round 1

Figure 1 shows that the baseline probability for a player to be drafted in the first round was about 4%. The model gave this particular player a 9% chance. We can see that this player was a receiver and performed about one to two standard deviations above average for multiple receiving metrics. In particular, their receiving touchdowns gave them a large boost in probability, however intriguingly their receiving yards per game worked against their predicted probability, despite being above average.

Outlook

Overall, the final models showed only slight improvement over baseline accuracy and therefore would not be practical for real-world use. To enhance performance, we could have examined a feature correlation matrix to remove highly correlated features. Another limitation was the scope of hyperparameter tuning; expanding the range could improve results, though at a cost to computing efficiency. The model identified receiving statistics as key predictors, which aligns with Figure 2 showing receivers as the most common position group in each round. Expanding data for other positions and reducing the imbalance between undrafted and drafted players could improve predictive accuracy. Additionally, incorporating NFL Combine results and physical attributes like height and weight might further strengthen the model.

References

Data is taken from Stathead's College Football Player Season Finder. The datasets themselves cannot be linked via the web as they are behind a paywall, however they are available for download in the data folder of the GitHub repository. This is in accordance with Stathead's terms of use as we only share a small sample of data from the Stathead database and are not using it to create a competing database or abuse their sharing policies.

GitHub Repository: https://github.com/ckourkou/DATA-1030-Final-Project.git