Samuel Thomas

Predicting NFL Draft Outcomes from NFL Combine Data

Introduction

Every year, NFL teams select college football players via the NFL Draft. One aspect of the scouting process is the NFL Draft Combine, an annual event where college players are invited to showcase their physical and mental skills for NFL personnel. The Combine is made up of position-specific skill drills, mental tests, and interviews. However, the NFL Draft Combine is most famous for its 6 standard drills that showcase the physical capabilities of every player. These drills are the 40-yard dash, 225lb bench press, vertical jump, broad jump, 20-yard shuttle, and 3-cone drill. Combined, these drills determine the speed, strength, and agility of every athlete who participates at the NFL Combine. Further, these drills can help or hurt a player's possibility of being drafted by a team.

The dataset that will be analyzed is NFL Combine data from 2000 to 2017. This data contains the Name, Position, Height, Weight, 40 Yard-dash time, Vertical jump, Reps of 225lb bench, Broad jump length, 3 Cone time, Shuttle time, Year, Performance_ID, AV, Team Drafted to, Round drafted in, and Pick drafted for each of the 5882 players attending the NFL Combine between these years.

The goal of this analysis is to predict whether a player will be drafted by an NFL team based on their Position, Height, Weight, and drill performance.

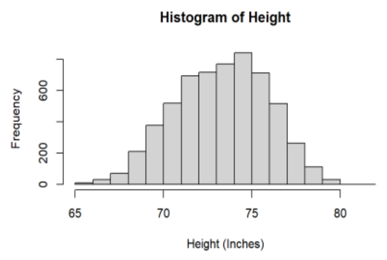
Exploratory Data Analysis and Data Cleaning

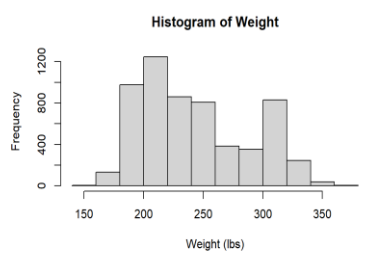
Position

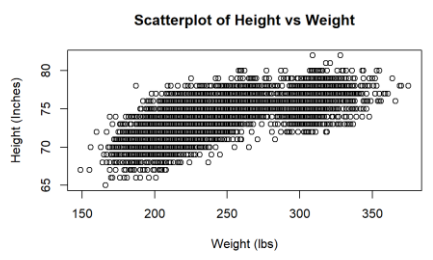
First, it appears that in 2018, the positions were reclassified to being more general. Hence, a SS or FS is classified as an S in 2018, whereas any other year they would be classified as a SS or FS. Unfortunately, this means that almost all of the 2018 players had a different classification compared to the rest of the dataset. For this reason, I decided to only use players from 2000 – 2017. Next, I reclassified 2 NT to DT since a NT is a DT. The final position counts are as follows:

C	**CB**	**DE**	**DT**	**FB**	**FS**	**ILB**	**K**	**LS**
162	589	471	442	116	229	261	81	19
OG	**OLB**	**OT**	**P**	**QB**	**RB**	**SS**	**TE**	**WR**
365	411	437	113	331	509	213	320	813

Height and Weight

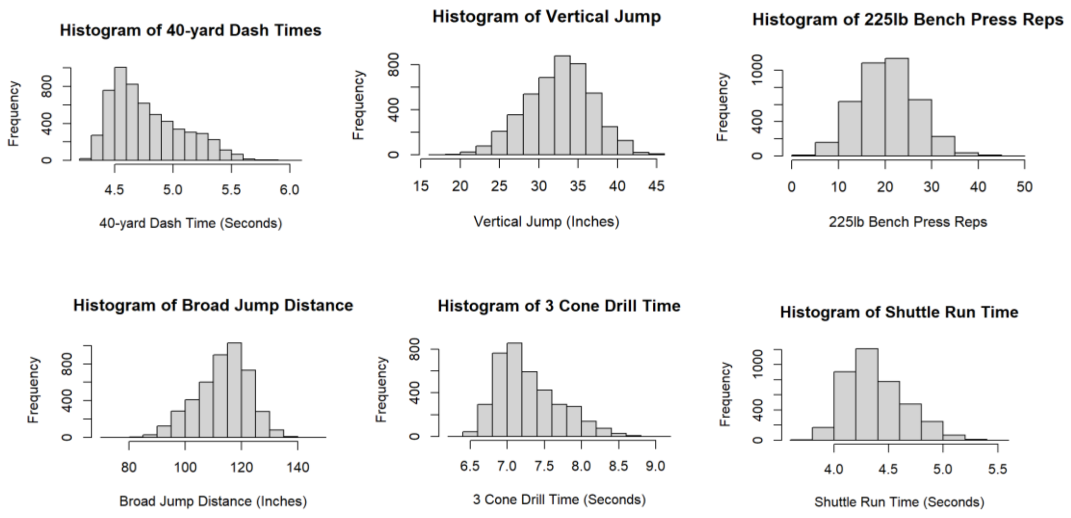






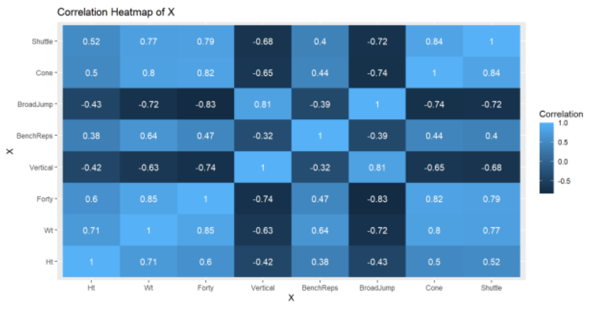
Height is almost normally distributed with mean 73.77, but weight has a bimodal distribution with peaks at 200 and 300. Also, there appears to be a strong relationship between height and weight.

Drills



Most drills have an approximately normal distribution. Some drills have a slightly skewed distribution. This is mainly due to outliers from underperformers. That being said, it is not unexpected since there are certain positions that are not expected to perform well at certain drills. For example, an offensive guard or defensive tackle is not expected to run a fast 40-yard dash since success at their position does not rely on speed as much as a cornerback or wide receiver.

Another, possibly unsurprising, finding is that there is quite a bit of correlation between drills, height, and weight. The correlation matrix is given as follows:

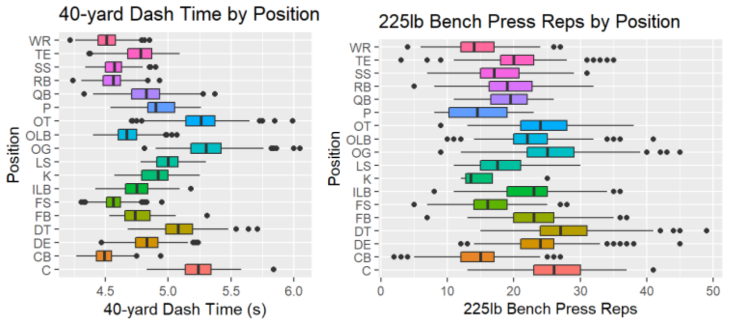


For this reason, I believe that models which tend to approximate correlated data better, such as ridge regression, will perform the best.

Missing Values

There are many missing data values for each drill. Many players decide to skip some drills since they either have an established result from a previous showcase or believe completing the drill at the combine will hurt their draft stock. Because of the quantity of missing values, it is necessary to impute these values to maintain enough data for fitting models.

I decided to impute the median performance of their position group. This was done because position group is a good representation for performance since each position requires unique skillsets that are reflected by these drills. Attached are two boxplots of drills grouped by each position. The boxplots broken up by position appears to cover the data well without a ton of outliers.



IsDrafted

The response is binary with 1 representing the player was drafted and 0 representing the player was not drafted. Out of 5881 players, 3738 were drafted and 2144 were not.

Modeling

After splitting our data via an 80-20 train-test split, the proportion of players who were Undrafted in our test set was 0.359 vs Drafted of 0.641. The baseline error rate if we just chose every player to be drafted is 0.359. Unless otherwise noted, all models were created with Height, Weight, Imputed versions of the Combine Drills, and Position as the explanatory variables, and IsDrafted as the response.

Logistic Regression

To start, two logit model were created. The first model included Position as a factor, while the second did not include Position. The reason Position was not included in the second model is because logistic regression may have difficulties with the increase in dimensionality when including Position as a factor. With position, there are 25 covariates, but there are only 9 without. This was also done for LDA, QDA, and KNN.

The model with Position gave a train error of 0.3118 and test error rate of 0.3091. The model without Position gave a train error of 0.3250 and test error rate of 0.3023.

LDA

Two LDA models were created. The first model included Position as a set of indicator variables. The training error was 0.3144 while the test error rate was 0.3100. The second model without Position came out with a training error of 0.3237, and a test error rate of 0.3049.

QDA

The first model including Position produced a training error rate of 0.3384, and a testing error rate of 0.3433. The second model without Position produced a train error rate of 0.3220 and test error rate of 0.3091. These two error rates are slightly higher error rates than those given by LDA, respectively.

KNN

The first KNN model included position, and after trying k from 1 to 100, k = 39 had the best test error rate of 0.3296. Also, the train error was 0.3263. The other KNN model without position yielded the lowest test error rate of 0.3364 at k = 38 while also having a train error of 0.3243.

Ridge Regression

A ridge regression model was then created with 10-fold cross-validation on lambda, which came out to 0.006. After cross-validation, and the training error rate was 0.3214, but the testing error rate was 0.3091. The covariates with the highest magnitude are Weight and Forty_Imputed.

Lasso Regression

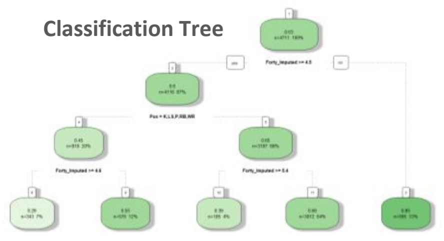
A lasso regression model was also created on all covariates with 10-fold cross-validation on lambda, which yielded 0.0002. The training error rate was 0.3212, and the test error rate was 0.3091. Interestingly, this is the same error as the ridge model. The only classification differences between the two models are 4 misclassifications. For ridge, 4 players were misclassified as drafted when they were not drafted, and for lasso vice versa. Also, the lasso kept all the covariates.

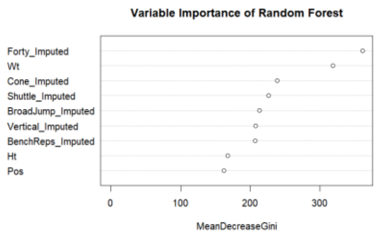
Tree

A classification tree was fit on all covariates. The tree was then pruned based on 10-fold cross-validation. The tree with depth of 5 gave the lowest training error rate of 0.3262. Then, the testing error was 0.3228. Here, the two covariates that determine the outcome are Forty_Imputed and Position.

Random Forest

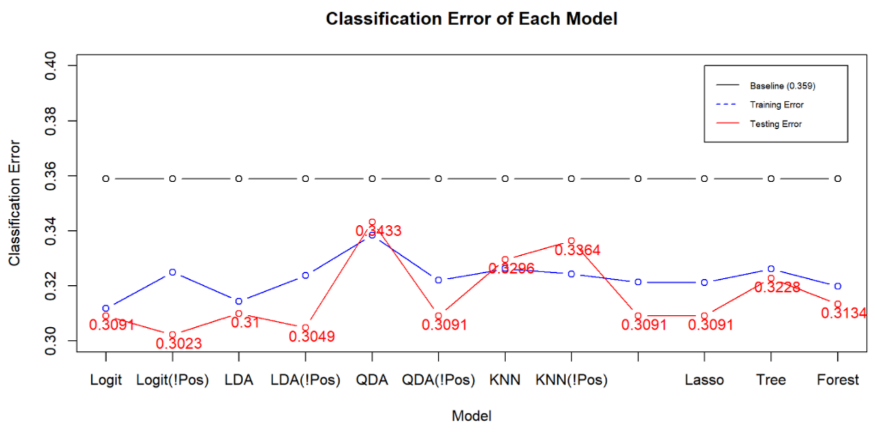
Next was a random forest. Cross-validation yielded an optimal value of 2 for M. Then, after fitting a new random forest with m = 2 and 2000 trees, there was a training error rate of 0.3199, and a testing error rate of 0.3134. This time, the two most important variables were Forty_Imputed and Weight while Position is the least important variable.





Summary

Below is a graph showing the Baseline Error, Training Error, and Testing Error for every model that was created. Note that missing label on the x-axis is the Ridge model. Also, models labeled with (!Pos) indicate that they did not include position as a covariate.



The best performing model was the Logit(!Pos) model. Other good performers were Logit, LDA, LDA (!Pos), QDA (!Pos), Ridge, and Lasso. I believe that the normality of the covariates helped the Logit, LDA, and QDA models perform well. Also, the Ridge regression likely performed well due to the high correlation in the data. These models beat the baseline by about 0.05, a 13.9% improvement. Additionally, there is no evidence of overfitting on the training data.

The Classification Tree, QDA, and both KNN models all performed relatively poorly. For QDA and KNN, I believe that both the dimensionality and noise of the data hurt predictive performance.

Another important takeaway is that almost every model indicated Weight and 40-yard Dash as the most important covariates. On the flip side, Position generally was the least important indicator or hurt certain methods due to the dimensionality increase.

Lastly, most of the error produced by the models is Type 1 error. There were many false positives, or players that were predicted to be drafted when they were not. On the other hand, Type 2 error was relatively low. Below is the confusion matrix from the Logit(!Pos) model:

Actual / Predicted	0	1
0	136	285
1	69	681

Next Steps

Other models that can be implemented with this data include Support Vector Machines and Binary Principal Component Regression. I especially believe that Binary PCR would predict the data well because of its correlated nature.

Additionally, improving the imputation method for missing drill data has the possibility to improve the predictive performance of every model. A better manner may be using other highly correlated covariates to impute values rather than position. For example, imputing missing 40-yard dash values with Weight (Correlation = 0.85) or Broad Jump (Correlation = - 0.83) may improve performance.

Next, recategorizing positions to more general categories will help mitigate possible information loss from not including them in models and dimensionality issues from including them. Examples of this would be reclassifying C, OG, OT as just Offensive Linemen.

An extension of these predictions is to predict the exact round a player gets drafted to. There are 7 rounds in the NFL Draft. Predicting the round players we already have predicted to be drafted would add value to these models. Another extension is predicting NFL playing success based on Drill performance. This would require finding more data but would yield interesting results.