Andrew Chen, Christina Deng, Tyler Lee, Chaewon Park, Thy Pham

Statistical Analysis on NFL Performance

Introduction

The dataset we are using for this project, titled "NFL Combine - Performance Data (2009 - 2019)," is from Kaggle, where the values of this dataset were collected from the NFL Combine results from 2009 to 2019. This data contains information from 3476 athletes, and 18 variables, some of which are the year, player, age, school, height, weight, 40-yard dash time in seconds, vertical jump height in cm, bench press reps, and broad jump in cm. This dataset will be used to address our driving questions of:

1. In the year 2017, does player type, vertical jump, and height affect the distance of the player's broad jump?
2. How do physical attributes and performance metrics predict whether a player will be drafted or not into the NFL?

We are specifically interested in investigating the relationship between the various predictors and the broad jump in the year 2017 because this year is the most recent in the data set with a sizable amount of points, over 100, for a robust test. Furthermore, we decided to only investigate 1 year to avoid any possible time trends we would not be able to account for in a linear regression test. This way, our findings are precise for finding the relationship between player type, vertical jump, and height on the broad jump.

The NFL dataset contains the measurements of top college football players, capturing their physical abilities and the draft outcome of their performance. Since there are many variables capturing both physical attributes and performance metrics, we are interested in how physical attributes and performance metrics can influence the likelihood of a player being drafted into the NFL. Specifically, height and weight, which we will consider as physical attributes, while bench press reps, sprint for 10, 20, and 40 yards, vertical and broad jump, and shuttle, which will be considered performance metrics that can affect the chance of getting drafted.

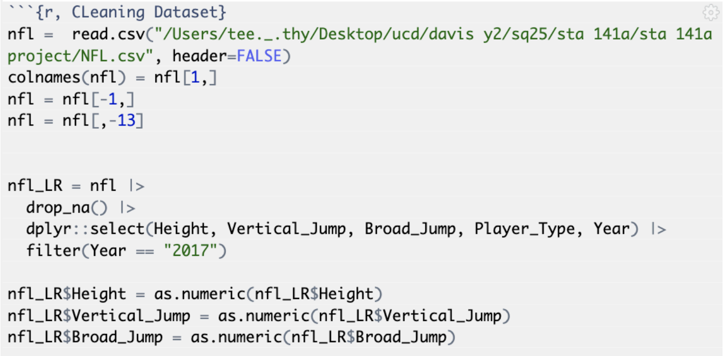
For the first question, we looked at how player type, vertical jump, and height affect a player's broad jump distance in cm, specifically on data from 2017, using a linear regression model. The broad jump, like the 40-yard dash, is used to measure an athlete's athleticism, which is a key part of athletic performance in American football.

For the second question, we will use logistic regression, performing variable selection and excluding certain variables that may not be relevant for the analysis. For instance, a player's name has no statistical significance in affecting the chance for the draft result, as it is treated as an observation key rather than having any statistical value.

We were interested in this topic to see if we could optimize player training to manage their workouts with a limited amount of time. If certain traits like vertical jump or height turn out to be strong predictors of how well someone performs, it could help coaches design specialized workouts for the athletes. For example, if vertical jump is closely correlated to broad jump distance, coaches might want to focus more on lower-body training.

Part I: “In the year 2017, does player type, vertical jump, and height affect the distance of the player's broad jump?”

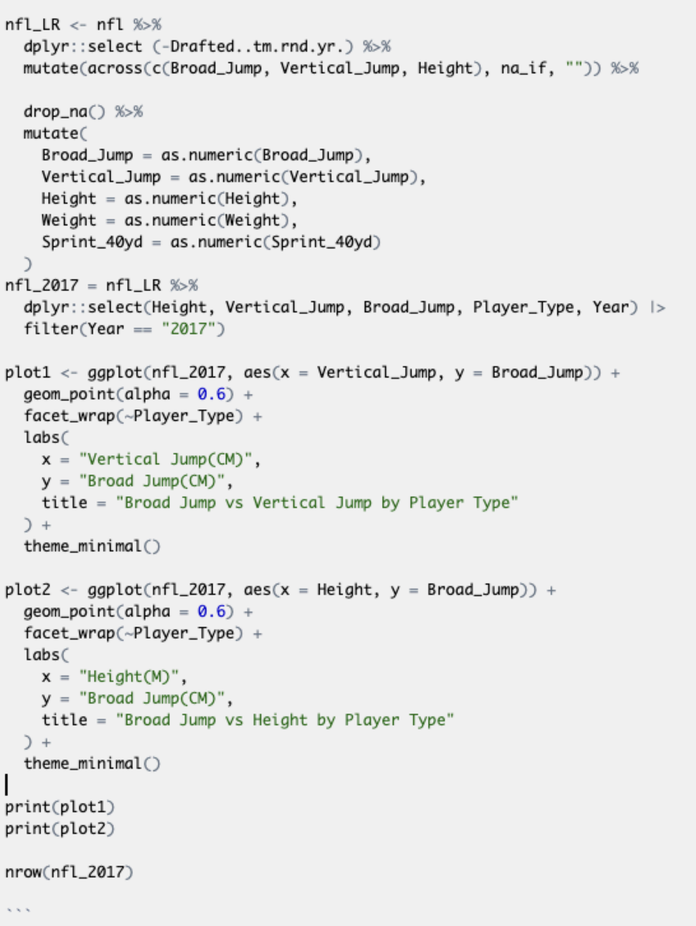
Data Cleaning



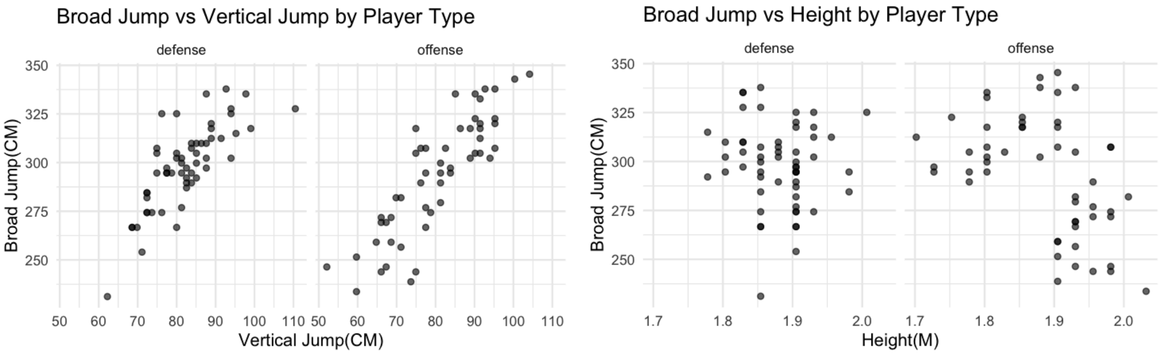
~~~
```{r, CLeaning Dataset}
nfl =  read.csv("/Users/tee._.thy/Desktop/ucd/davis y2/sq25/sta 141a/sta 141a
project/NFL.csv", header=FALSE)
colnames(nfl) = nfl[1,]
nfl = nfl[-1,]
nfl = nfl[,-13]


nfl_LR = nfl |>
  drop_na() |>
  dplyr::select(Height, Vertical_Jump, Broad_Jump, Player_Type, Year) |>
  filter(Year == "2017")

nfl_LR$Height = as.numeric(nfl_LR$Height)
nfl_LR$Vertical_Jump = as.numeric(nfl_LR$Vertical_Jump)
nfl_LR$Broad_Jump = as.numeric(nfl_LR$Broad_Jump)
~~~

Before we start, we must clean and organize the data. When the data set was loaded into R, the column names were not properly assigned to the data and were instead made as the first row, so the first thing we did was fix this. Upon inspecting our data, we noticed that it contained many missing values. Looking at the data set, many of the missing values in the Drafted..tm.rnd.yr column was solely since the player was not drafted, so removing the NAs in this column would skew our data. Since we had no interest in this variable, we just decided to remove this variable from our data set before continuing to remove missing values. After removing the remaining NA points, we were left with a dataset containing 1179 rows. Afterwards, we wanted to narrow down our data set to only select variables we were interested in, so we subsetted the data to only include the Player Type (offense or defense), Broad jump, Vertical Jump, and Height numbers in centimeters, leaving us with 140 observations. R automatically classified these variables as characters, so we changed them to numeric. Furthermore, since our interest is in the data from 2017, we filtered the data to only include data from 2017.

Exploratory Data Analysis



~~~
nfl_LR <- nfl %>%
  dplyr::select (-Drafted..tm.rnd.yr.) %>%
  mutate(across(c(Broad_Jump, Vertical_Jump, Height), na_if, "")) %>%

  drop_na() %>%
  mutate(
    Broad_Jump = as.numeric(Broad_Jump),
    Vertical_Jump = as.numeric(Vertical_Jump),
    Height = as.numeric(Height),
    Weight = as.numeric(Weight),
    Sprint_40yd = as.numeric(Sprint_40yd)
  )
nfl_2017 = nfl_LR %>%
  dplyr::select(Height, Vertical_Jump, Broad_Jump, Player_Type, Year) |>
  filter(Year == "2017")

plot1 <- ggplot(nfl_2017, aes(x = Vertical_Jump, y = Broad_Jump)) +
  geom_point(alpha = 0.6) +
  facet_wrap(~Player_Type) +
  labs(
    x = "Vertical Jump(CM)",
    y = "Broad Jump(CM)",
    title = "Broad Jump vs Vertical Jump by Player Type"
  ) +
  theme_minimal()

plot2 <- ggplot(nfl_2017, aes(x = Height, y = Broad_Jump)) +
  geom_point(alpha = 0.6) +
  facet_wrap(~Player_Type) +
  labs(
    x = "Height(M)",
    y = "Broad Jump(CM)",
    title = "Broad Jump vs Height by Player Type"
  ) +
  theme_minimal()

print(plot1)
print(plot2)

nrow(nfl_2017)

```
~~~



The scatter plot titled “Broad Jump vs Vertical Jump by Player Type” compares the relationship between vertical and broad jump performances among defensive and offensive player results. In the graph, we see a positive linear relationship in both groups, indicating that athletes who jump higher vertically also tend to perform better in the broad jump. This is expected, as both tests measure lower-body strength. However, the strength and consistency of this relationship differ between the two player types. Offensive players show a tighter, more linear trend, suggesting that their jump performance is more normal and potentially influenced by how they train for their position. In defensive players, we see greater variance, with a wider spread in broad jump distances for similar vertical jump values. While both groups cover similar ranges in jump performance, the stronger trend among offensive players suggests a more predictable relationship between the two variables. These can be important for tailoring training programs and identifying athletic strengths or deficiencies in a player's statistics.

The scatter plot titled “Broad Jump vs Height by Player Type” shows the relationship between player height and broad jump distance for both defensive and offensive NFL combine athletes. We see that there does not appear to be a strong linear correlation between height and broad jump performance in either player group. The data points are widely scattered across the height range, with no clear upward or downward trend.

In the defensive player group, most athletes fall between 1.80 m and 1.95 m in height, and their broad jump performances are spread fairly evenly from 250 cm to 340 cm. This suggests that among defensive players, height alone is not a reliable predictor of broad jump ability. Similarly, for offensive players, the majority also cluster around the same height range, but their broad jump outcomes show a slightly more noticeable grouping, with some players that are between 1.90 m and 2.00 m having both high and low jump results. This inconsistency shows that other factors may play a more important role in determining broad jump performance than height alone.

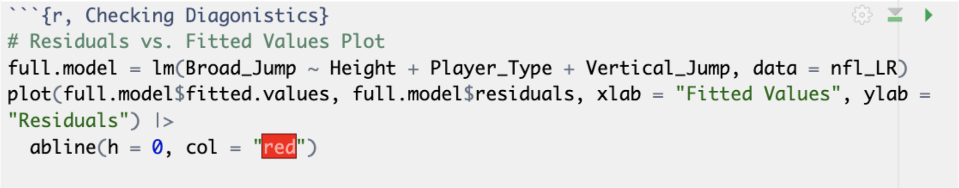
Overall, the plots seem to suggest that height does not have a strong, direct effect on broad jump results. To solidify our findings, we will be creating a linear model. These findings suggest that coaches should not significantly change players' training styles to meet the averages.

Linear Regression

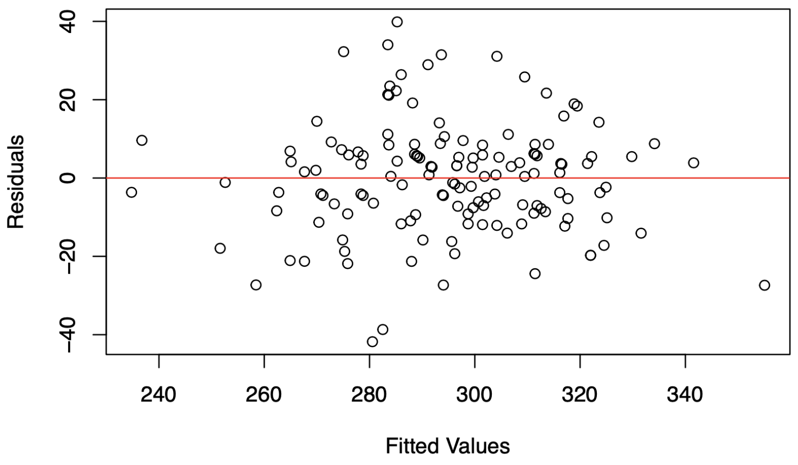
To answer our first question about the relationship between player type, vertical jump, height, and broad jump in the year 2017, we will be using linear regression, a supervised learning method. We are using the same aforementioned cleaned and subsetted dataset with 140 rows from the EDA.

Diagnostics (Assumptions + outliers)

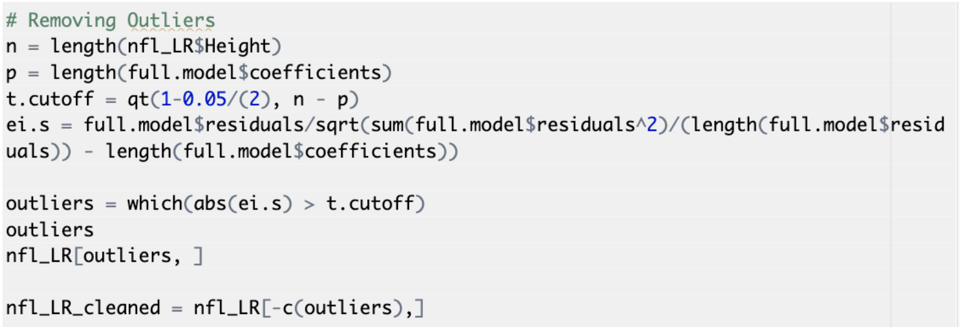
Now with our data cleaned and ready, we must check that the assumptions necessary for linear regression are met. These assumptions include, the errors are independent, normally distributed, and have constant variance. This requires us to fit a full model consisting of all the variables we are interested in, Broad Jump as our response variable, and the Player Type, Vertical Jump, and Height as our predictors. Using the residuals and fitted values from this model, we first created a fitted value vs. residuals plot to get some insight into the assumptions.



~~~
```{r, Checking Diagionistics}
# Residuals vs. Fitted Values Plot
full.model = lm(Broad_Jump ~ Height + Player_Type + Vertical_Jump, data = nfl_LR)
plot(full.model$fitted.values, full.model$residuals, xlab = "Fitted Values", ylab =
"Residuals") |>
  abline(h = 0, col = "red")
~~~

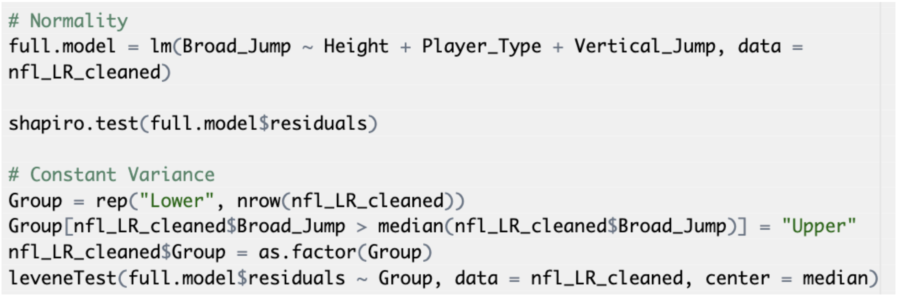


From this plot, we can see the data points being randomly scattered around 0 with no distinct pattern, indicating that a linear relationship is an appropriate model fit for these variables. The vertical spread of the data seems consistent throughout, with no strong funnel shape, suggesting there is approximately constant variance. Furthermore, it does seem to contain outliers. For example, most of the points seem to have residuals of less than -20, which makes the two points with a fitted value of 280 and residuals of -40 unusually large and possible outliers to remove.



```
# Removing Outliers
n = length(nfl_LR$Height)
p = length(full.model$coefficients)
t.cutoff = qt(1-0.05/(2), n - p)
ei.s = full.model$residuals/sqrt(sum(full.model$residuals^2)/(length(full.model$resid
uals)) - length(full.model$coefficients))

outliers = which(abs(ei.s) > t.cutoff)
outliers
nfl_LR[outliers, ]

nfl_LR_cleaned = nfl_LR[-c(outliers),]
```

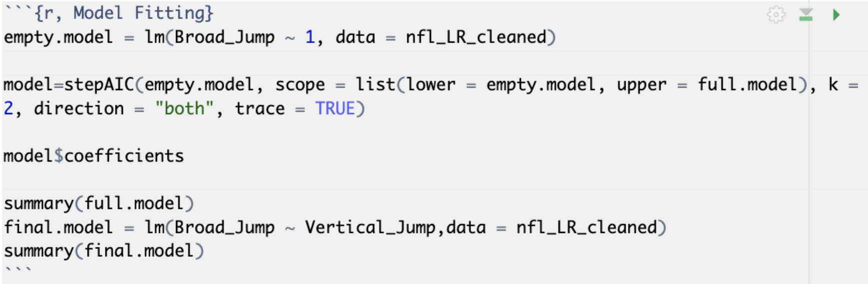
We then used the studentized outliers method to determine a cutoff for outliers; we found 8 outliers, corresponding to the rows 46, 64, 66, 67, 76, 104, 117, and 133. Having these outliers in our data will skew the regression line, making it an inaccurate representation of the true relationship between the variables, so we removed them, leaving us with a total of 132 rows of data.



```
# Normality
full.model = lm(Broad_Jump ~ Height + Player_Type + Vertical_Jump, data =
nfl_LR_cleaned)

shapiro.test(full.model$residuals)

# Constant Variance
Group = rep("Lower", nrow(nfl_LR_cleaned))
Group[nfl_LR_cleaned$Broad_Jump > median(nfl_LR_cleaned$Broad_Jump)] = "Upper"
nfl_LR_cleaned$Group = as.factor(Group)
leveneTest(full.model$residuals ~ Group, data = nfl_LR_cleaned, center = median)
```

To make concrete conclusions for normality and constant variance, we will use hypothesis tests with a significance level of 0.05. This requires us to refit a full model consisting of all the variables we are interested in, Broad Jump as our response variable, and the Player Type, Vertical Jump, and Height as our predictors, with the cleaned data. Using the errors from this model to test if the residuals pass the Shapiro-Wilks test, for normality, and the Levene Test, for constant variance, we find p-values of 0.258 and 0.5936, respectively. Since these two p-values are larger than our significance level, we fail to reject the null hypothesis and affirm that they pass the assumptions needed to conduct a linear regression model.

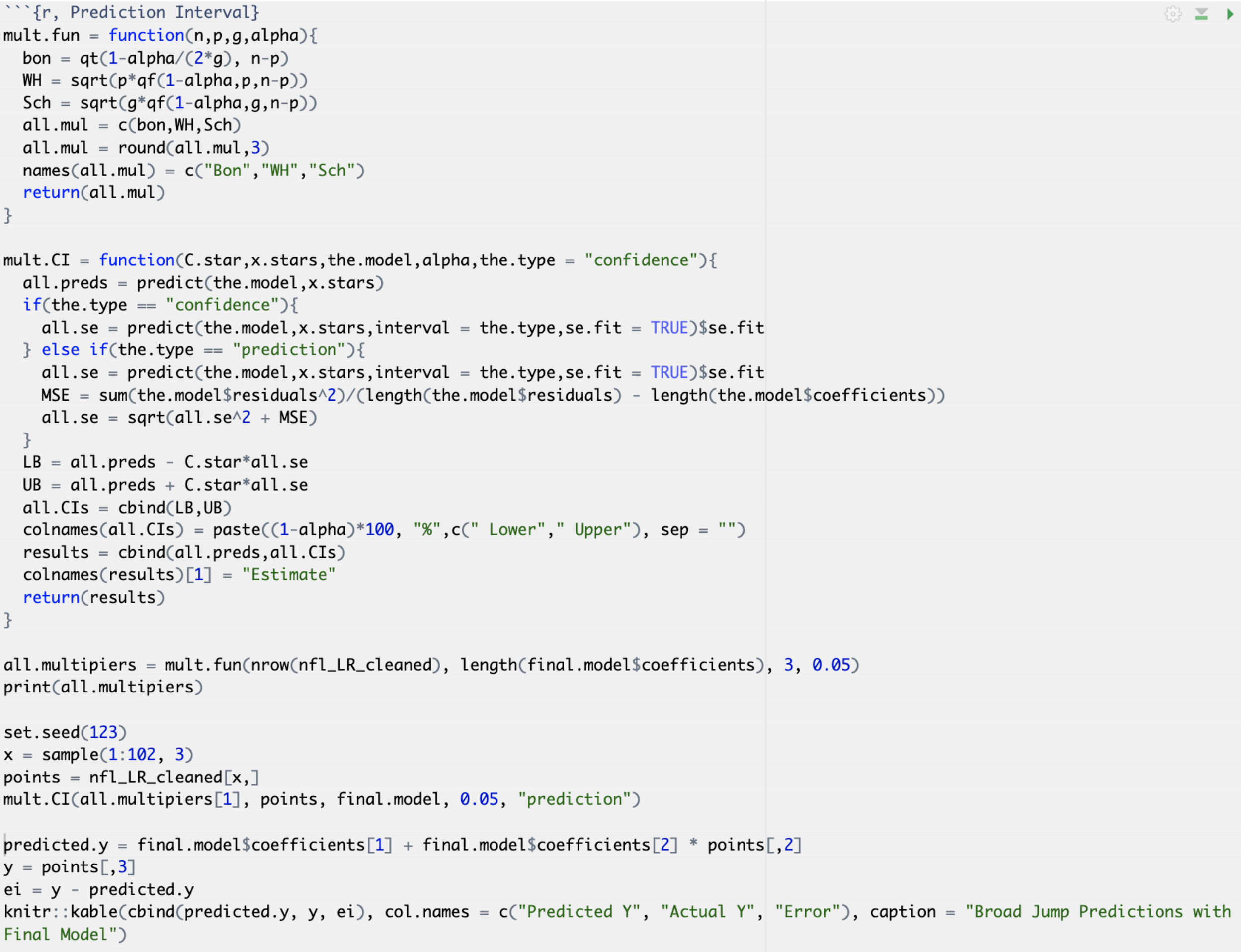
Methodology



~~~
```{r, Model Fitting}
empty.model = lm(Broad_Jump ~ 1, data = nfl_LR_cleaned)

model=stepAIC(empty.model, scope = list(lower = empty.model, upper = full.model), k =
2, direction = "both", trace = TRUE)

model$coefficients

summary(full.model)
final.model = lm(Broad_Jump ~ Vertical_Jump,data = nfl_LR_cleaned)
summary(final.model)
```
~~~

To pick our final model to predict the Broad Jump of a player in centimeters, we considered Player Type, Vertical Jump, and Height as our predictors. We decided to use the method of Forward-Backward selection, which adds predictors to the model one by one and compares the AIC for each model, keeping the ones that decrease the AIC the most at each step, stopping when the AIC no longer decreases. Then, taking this final model found using this Forward method, we undergo the Backward selection part, which removes the predictors one at a time, removing the one that leads to the smallest change in AIC, also stopping when AIC no longer decreases. We choose to use this method since it balances the underfitting of only using Forward Selection and the overfitting caused by the Backward Selection. From this, we get that the only relevant predictor is Vertical Jump. Now fitting a final model containing only the predictor Vertical Jump and the response variable Broad Jump, it has an intercept of 130.11 and a slope of 2.03. This means that the distance of the Broad Jump in cm increases by 2.03 cm for every 1 cm increase in the Vertical Jump distance. This final model has an adjusted R^2 of 0.7551, telling us that this model explains 75.51% of the variability in our data. Since this proportion is relatively high, it suggests our model is a good fit even though it only contains one predictor. Looking at the adjusted R^2 for the full model containing all the predictors, we see it has a value of 0.7516. This further supports our findings that Vertical Jump is the only important model since removing those variables increased the adjusted R^2. Thus, this demonstrates that Player type and Height are not significant for the prediction of Broad Jump. Normally, we would also check for multicollinearity to eliminate or combine any correlated predictors. Yet since we only have one predictor in our final model, there is no chance for multicollinearity; we do not need to check it.

Analysis & Interpretation



~~~
```{r, Prediction Interval}
mult.fun = function(n,p,g,alpha){
  bon = qt(1-alpha/(2*g), n-p)
  WH = sqrt(p*qf(1-alpha,p,n-p))
  Sch = sqrt(g*qf(1-alpha,g,n-p))
  all.mul = c(bon,WH,Sch)
  all.mul = round(all.mul,3)
  names(all.mul) = c("Bon","WH","Sch")
  return(all.mul)
}

mult.CI = function(C.star,x.stars,the.model,alpha,the.type = "confidence"){
  all.preds = predict(the.model,x.stars)
  if(the.type == "confidence"){
    all.se = predict(the.model,x.stars,interval = the.type,se.fit = TRUE)$se.fit
  } else if(the.type == "prediction"){
    all.se = predict(the.model,x.stars,interval = the.type,se.fit = TRUE)$se.fit
    MSE = sum(the.model$residuals^2)/(length(the.model$residuals) - length(the.model$coefficients))
    all.se = sqrt(all.se^2 + MSE)
  }
  LB = all.preds - C.star*all.se
  UB = all.preds + C.star*all.se
  all.CIs = cbind(LB,UB)
  colnames(all.CIs) = paste((1-alpha)*100, "%",c(" Lower"," Upper"), sep = "")
  results = cbind(all.preds,all.CIs)
  colnames(results)[1] = "Estimate"
  return(results)
}

all.multipiers = mult.fun(nrow(nfl_LR_cleaned), length(final.model$coefficients), 3, 0.05)
print(all.multipiers)

set.seed(123)
x = sample(1:102, 3)
points = nfl_LR_cleaned[x,]
mult.CI(all.multipiers[1], points, final.model, 0.05, "prediction")

predicted.y = final.model$coefficients[1] + final.model$coefficients[2] * points[,2]
y = points[,3]
ei = y - predicted.y
knitr::kable(cbind(predicted.y, y, ei), col.names = c("Predicted Y", "Actual Y", "Error"), caption = "Broad Jump Predictions with
Final Model")
~~~

So, our final model only consists of the variable Vertical Jump as a significant predictor for Broad Jump.

$$\hat{Y} = 130.11 + 2.03X_{Vertical\_Jump}$$

Then we made 95% prediction intervals for the Broad Jump to test our model on how it may predict random values from the training data. The intervals were quite large, about 57 centimeters in range.

```
   Estimate 95% Lower 95% Upper
31 315.9940  287.2390  344.7489
84 321.1575  292.3408  349.9741
52 295.3400  266.6977  323.9823
```

A prediction interval tells us that if we sampled the data many times, 95% of all intervals will contain the true value of Broad Jump at a specific Vertical Jump value.

We then used our final linear model to get exact prediction estimates of Broad Jump for three random points in our data to compare to the actual Broad Jump values from the training dataset. Organizing these values into a table, we get:

Table 1: Broad Jump Predictions with Final Model

Predicted Y	Actual Y	Error
315.9940	312.42	-3.573973
321.1575	327.66	6.502534
295.3400	289.56	-5.780001

The errors for points were fairly small at about 3.5 to 6.5 centimeters (about 5 centimeters of average error). Having some error in our predictions is typically normal since the true model consists of an error term that is always present. Furthermore, the relationship between Broad Jump and Vertical Jump may not be perfectly linear, as our model suggests, but relatively linear, creating some error in our predictions. Overall, our model tends to overpredict the distance of the broad jump, indicated by the fact that 2 out of 3 errors are from the predicted value being greater than the actual value.

Part I Conclusion

We find our results to be reasonable and not too far from our initial expectations. We expected height to be a significant variable in predicting the broad jump because we believed longer legs may help achieve a longer jump distance. Yet when only vertical jump proved significant, we were not that shocked either because vertical jump should be the most indicative of broad jump due to the similarities between them; if you can jump high, then you can probably jump far as well.

A limitation in our prediction evaluation is that we used training points in our regression line and not new actual points, which means we do not know how well our model can be generalized to unseen data. Another limitation of our linear regression model is that it's based specifically on evaluating these 3 predictors in 2017, so it may not be as relevant now in 2025, and there could be more significant factors to broad jump that we did not evaluate or were not included in this dataset.

In application to the real world, if aspiring NFL players wanted to know what attributes mattered the most in their broad jump score, they could use their vertical score as a predictor. This could also guide their training, as practicing the vertical jump may also double as training for the broad jump.

Part II: “How do physical attributes and performance metrics predict whether a player will be drafted or not into the NFL?”

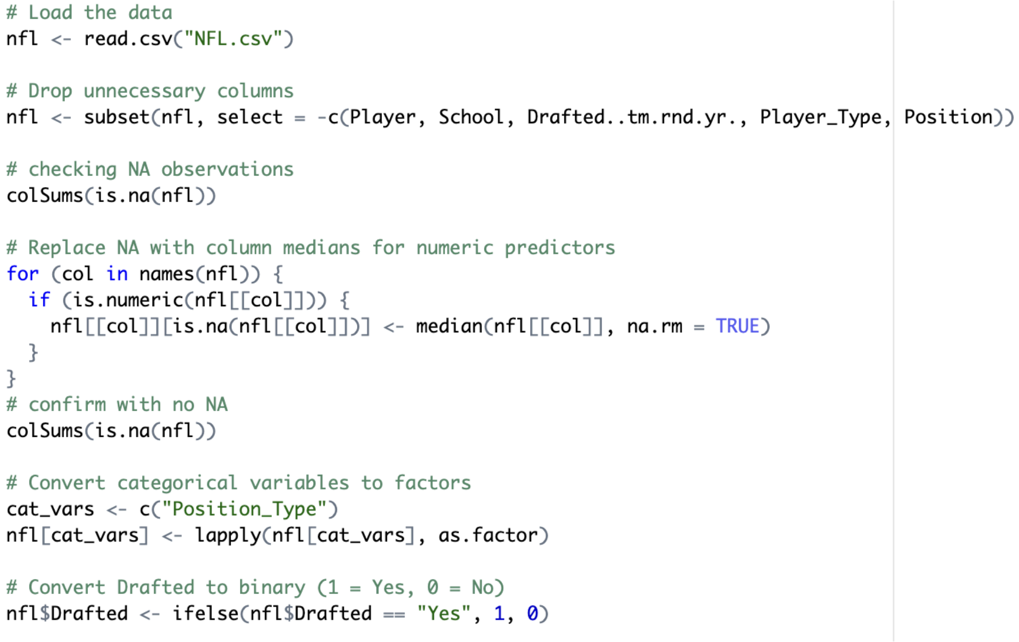
Data Cleaning

We began by removing several irrelevant columns from the dataset, including \`Player\`, \`School\`, \`Drafted..tm.rnd.yr.\`, \`Player_Type\`, and \`Position\`. We noticed that player type and position are highly correlated with position type. This raised a concern of multicollinearity, which can make unreliable coefficients for the model variables that ultimately lead to not fully capturing individual effects on the draft chance. Thus, we decided to only keep one of the three variables that have similar characteristics.

For the linear regression, we simply removed all observations with missing values. In contrast, for logistic regression, our main focus was on predicting draft status. We had concerns about class imbalance and thus wanted more reliable data with statistical significance. To address missing values in the numeric predictors, we implemented a for loop to replace all NA entries with the median of each respective column. This method preserves the distribution of each variable while ensuring that all 3476 observations can be used in the model. We double-checked that no NA values remained in the dataset.

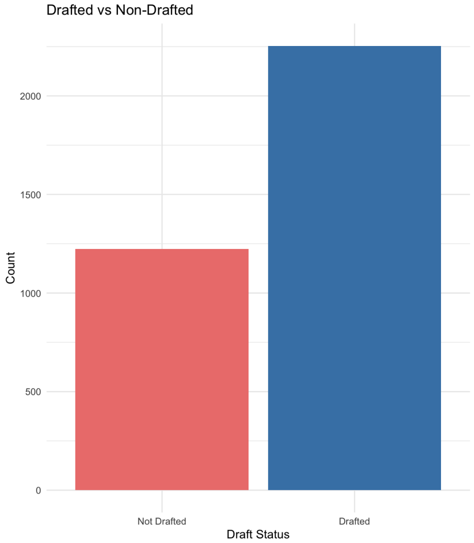
To ensure correct model fitting, we converted the categorical variable `Position_Type` into a factor, allowing R to treat it as a qualitative predictor.

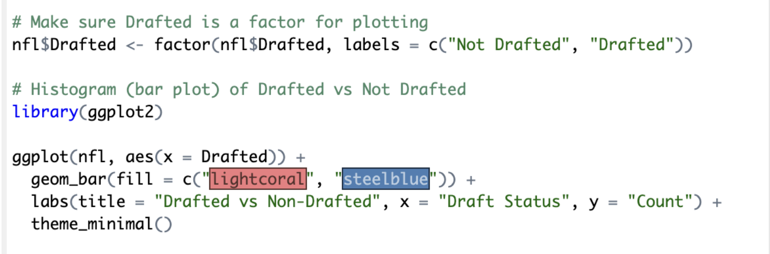
Lastly, we converted the target variable Drafted into a binary numeric variable, where 1 indicates a player was drafted and 0 indicates they were not. This transformation was necessary to fit a logistic regression model, which requires the dependent variable to be binary.



```
# Load the data
nfl <- read.csv("NFL.csv")

# Drop unnecessary columns
nfl <- subset(nfl, select = -c(Player, School, Drafted..tm.rnd.yr., Player_Type, Position))

# checking NA observations
colSums(is.na(nfl))

# Replace NA with column medians for numeric predictors
for (col in names(nfl)) {
  if (is.numeric(nfl[[col]])) {
    nfl[[col]][is.na(nfl[[col]])] <- median(nfl[[col]], na.rm = TRUE)
  }
}
# confirm with no NA
colSums(is.na(nfl))

# Convert categorical variables to factors
cat_vars <- c("Position_Type")
nfl[cat_vars] <- lapply(nfl[cat_vars], as.factor)

# Convert Drafted to binary (1 = Yes, 0 = No)
nfl$Drafted <- ifelse(nfl$Drafted == "Yes", 1, 0)
```

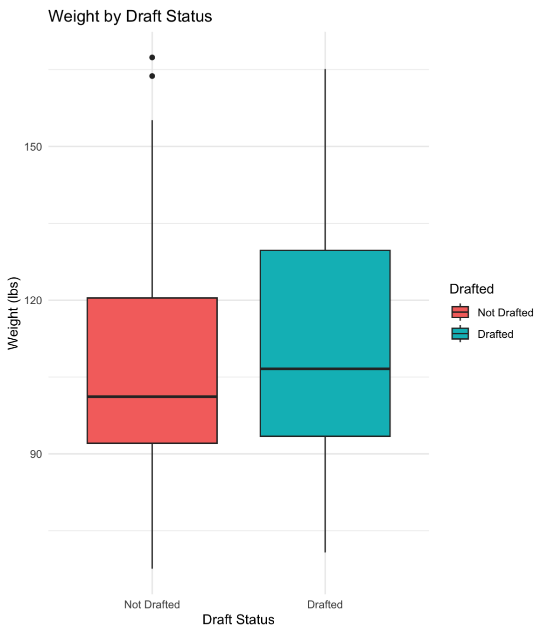
Exploratory Data Analysis

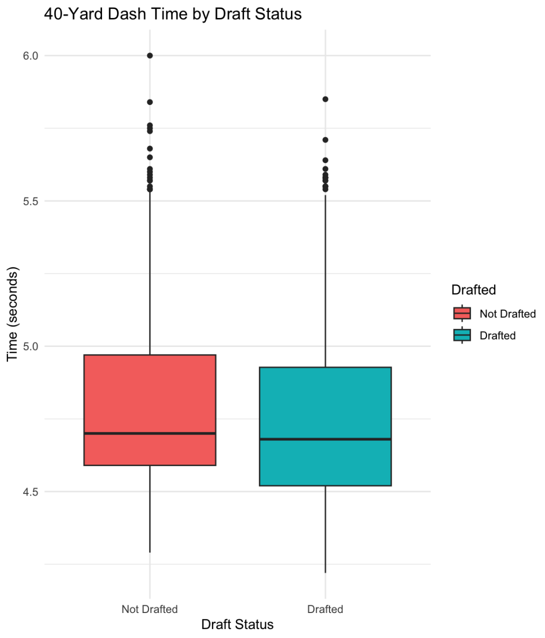


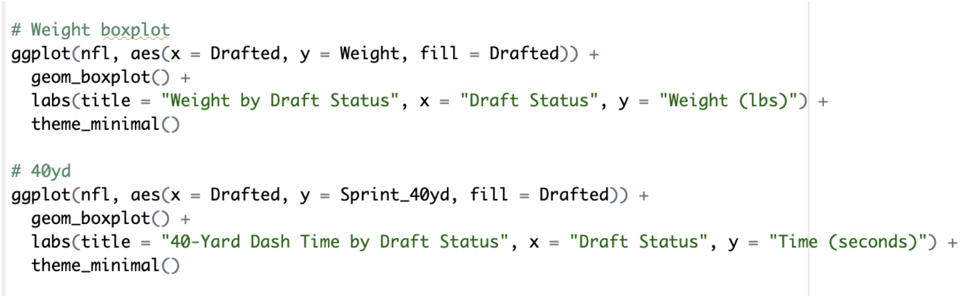


```
# Make sure Drafted is a factor for plotting
nfl$Drafted <- factor(nfl$Drafted, labels = c("Not Drafted", "Drafted"))

# Histogram (bar plot) of Drafted vs Not Drafted
library(ggplot2)

ggplot(nfl, aes(x = Drafted)) +
  geom_bar(fill = c("lightcoral", "steelblue")) +
  labs(title = "Drafted vs Non-Drafted", x = "Draft Status", y = "Count") +
  theme_minimal()
```

The bar chart provides the distribution of players who were drafted versus those who were not drafted. It shows that a larger number of athletes in the dataset were drafted compared to those who were not. In detail, the “Drafted” category has more than 2000 athletes, while the “Non Drafted” category has around 1250 athletes, which is less than ½ the athletes from the drafted category. The dataset may be slightly imbalanced toward drafted players, which could potentially create bias in our model selection and performance.







```
# Weight boxplot
ggplot(nfl, aes(x = Drafted, y = Weight, fill = Drafted)) +
  geom_boxplot() +
  labs(title = "Weight by Draft Status", x = "Draft Status", y = "Weight (lbs)") +
  theme_minimal()

# 40yd
ggplot(nfl, aes(x = Drafted, y = Sprint_40yd, fill = Drafted)) +
  geom_boxplot() +
  labs(title = "40-Yard Dash Time by Draft Status", x = "Draft Status", y = "Time (seconds)") +
  theme_minimal()
```

The first box plot compares player weights based on draft status. We can see that, on average, drafted players have slightly heavier weights than those who were not drafted, with a higher median weight and wider interquartile range. This suggests that weight can play an important role in NFL draft selection. Similarly, the second boxplot shows the 40-yard dash times for drafted versus non-drafted players. Drafted players tend to have slightly faster dash times, though the difference is minimal than the difference for weight. We noticed many outliers for 40-yard dash times for both drafted and non-drafted players. This could have biased the distribution, making it difficult to distinguish drafted versus non-drafted players.

Logistic regression

Here, to answer our second question, "How do physical attributes and performance metrics predict whether a player will be drafted or not into the NFL?", we will use a logistic regression model to answer it.

Methodology

Logistic regression is appropriate here because our response variable `Drafted` is a categorical variable with binary categories, where 1 = drafted, 0 = not drafted. The model was fitted using the glm() function in R, with the binomial family. This modeling approach estimates the effect of each predictor on the log-odds of being drafted, enabling us to: determine which characteristics are statistically significant predictors, interpret the direction and magnitude of each effect, and assess the overall ability of the model to classify players correctly.

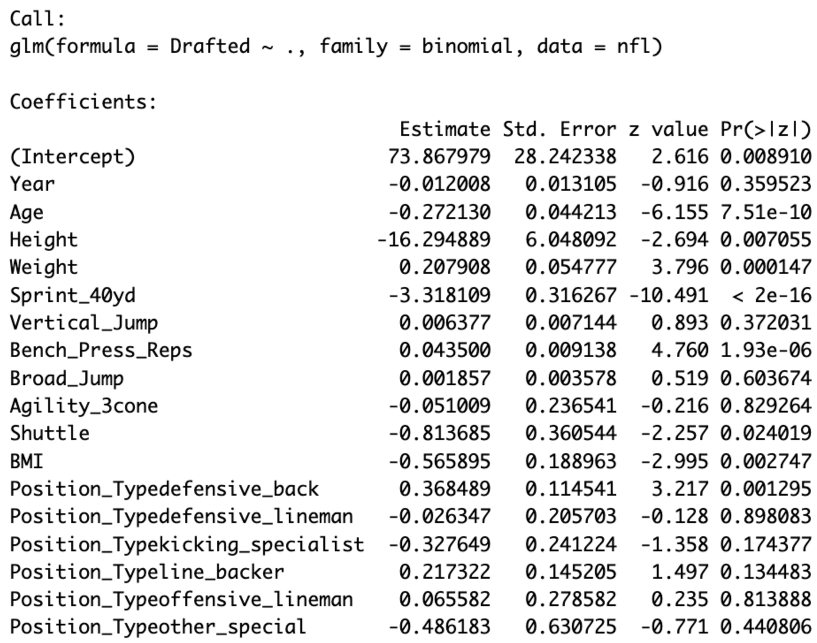
We later evaluated the model's predictive performance using accuracy, a confusion matrix, and the area under the ROC curve, which gives a comprehensive view of how well the model distinguishes between drafted and undrafted players.

Analysis & Interpretation

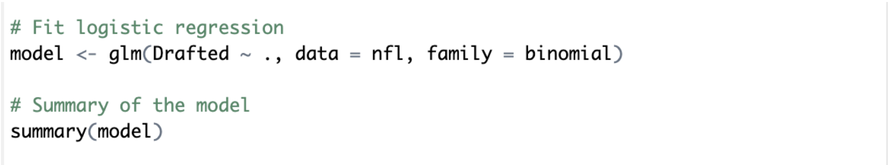
The logistic regression output provides coefficient estimates, standard errors, z-values, and p-values for each predictor. We used the standard that variables with $p < 0.05$ are statistically significant predictors. Based on the model, several variables were found to be statistically significant predictors of whether a player is drafted:

- \`Sprint_40yd\` was highly significant with a negative coefficient. This suggests that faster sprint times increase the likelihood of being drafted, as expected.
- Age was also highly significant and negatively associated with draft probability, indicating that younger players are more likely to be drafted.
- Weight had a positive and significant effect, implying that heavier players have slightly higher odds of being drafted.
- Height had a negative effect, suggesting that greater height slightly decreases draft probability.
- \`Bench_Press_Reps\` was positively and significantly associated with draft status, suggesting that upper-body strength is an important variable.
- Shuttle time had a negative and significant impact, indicating that faster shuttle times are associated with a higher chance of being drafted.
- BMI showed a significant negative effect, implying that higher BMI may negatively impact the performance of a player.
- Among position types, being a defensive back significantly increased the chance of being drafted relative to the other groups.

Other variables, including \`Vertical_Jump\`, \`Broad_Jump\`, and \`Agility_3cone\`, did not have statistically significant effects in this model. Therefore, we do not include those variables.

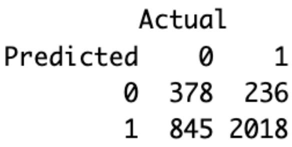


```
Call:
glm(formula = Drafted ~ ., family = binomial, data = nfl)

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                   73.867979  28.242338   2.616 0.008910
Year                          -0.012008   0.013105  -0.916 0.359523
Age                           -0.272130   0.044213  -6.155 7.51e-10
Height                       -16.294889   6.048092  -2.694 0.007055
Weight                         0.207908   0.054777   3.796 0.000147
Sprint_40yd                   -3.318109   0.316267 -10.491  < 2e-16
Vertical_Jump                  0.006377   0.007144   0.893 0.372031
Bench_Press_Reps               0.043500   0.009138   4.760 1.93e-06
Broad_Jump                     0.001857   0.003578   0.519 0.603674
Agility_3cone                 -0.051009   0.236541  -0.216 0.829264
Shuttle                       -0.813685   0.360544  -2.257 0.024019
BMI                           -0.565895   0.188963  -2.995 0.002747
Position_Typedefensive_back    0.368489   0.114541   3.217 0.001295
Position_Typedefensive_lineman -0.026347   0.205703  -0.128 0.898083
Position_Typekicking_specialist -0.327649  0.241224  -1.358 0.174377
Position_Typeline_backer       0.217322   0.145205   1.497 0.134483
Position_Typeoffensive_lineman 0.065582   0.278582   0.235 0.813888
Position_Typeother_special    -0.486183   0.630725  -0.771 0.440806
```

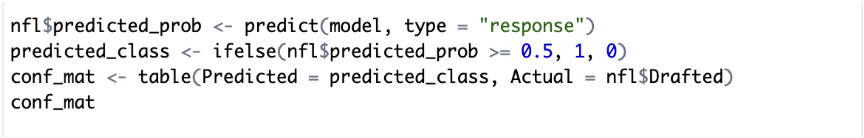


```r
# Fit logistic regression
model <- glm(Drafted ~ ., data = nfl, family = binomial)

# Summary of the model
summary(model)
```

To evaluate the model's predictive performance, we classified players as drafted or not drafted using a probability threshold of 0.5 and compared the predictions against actual outcomes. The resulting confusion matrix was:



```
         Actual
Predicted    0    1
        0  378  236
        1  845 2018
```

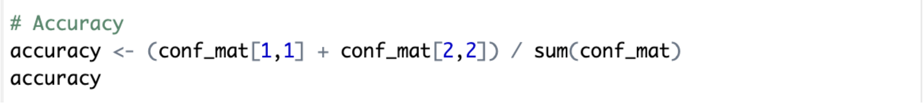


```r
nfl$predicted_prob <- predict(model, type = "response")
predicted_class <- ifelse(nfl$predicted_prob >= 0.5, 1, 0)
conf_mat <- table(Predicted = predicted_class, Actual = nfl$Drafted)
conf_mat
```

From the confusion matrix, we observed:

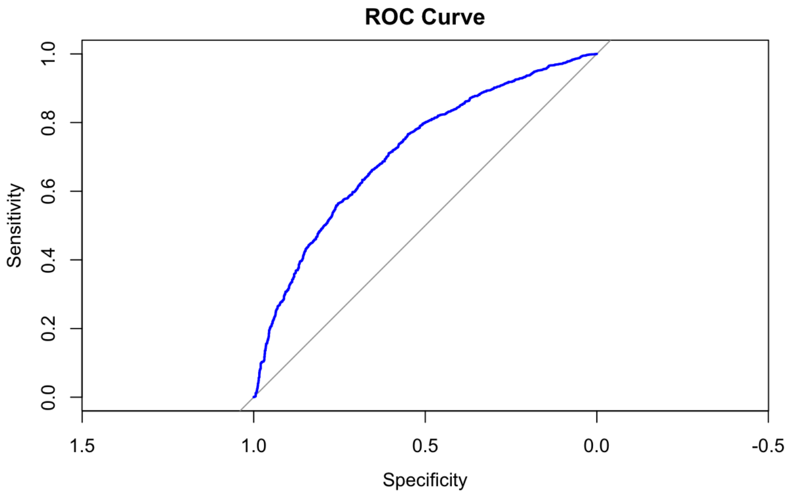
- True Positives (drafted and predicted drafted): 2018
- True Negatives (not drafted and predicted not drafted): 378
- False Positives (predicted drafted but not drafted): 845
- False Negatives (predicted not drafted but actually drafted): 236

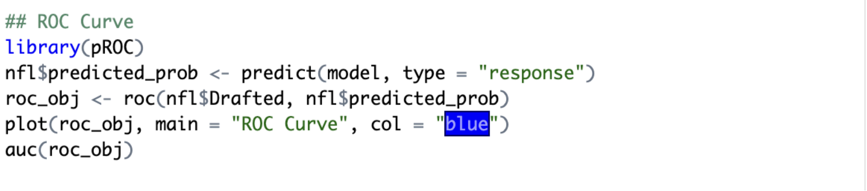
Therefore, the model's overall accuracy was 0.6891. The model correctly classified approximately 68.9% of all players. While this may not seem very high and accurate, it is still moderately strong given the complexity of the draft process. In detail, being drafted into the NFL can depend on numerous factors beyond just physical and performance metrics. For example, injuries, off-field behaviors, and NFL team needs all can significantly affect the draft status, which are not captured in our dataset. Similarly, the dataset had a class imbalance where drafted players were nearly twice as numerous as undrafted players. This imbalance can potentially raise a concern of biasing the model that favors the majority class, which is drafted players. Overall, these reasons can support that the model still achieved moderately strong accuracy, capturing meaningful patterns in the data while still navigating such limitations.



```
# Accuracy
accuracy <- (conf_mat[1,1] + conf_mat[2,2]) / sum(conf_mat)
accuracy
```

To further assess the model's ability to discriminate between drafted and undrafted players across different probability thresholds, we plotted a Receiver Operating Characteristic (ROC) curve. The resulting area under the curve (AUC) was 0.7138, indicating that the model has a good level of discriminatory power. In practical terms, this means that given a randomly selected pair of one drafted and one undrafted player, the model has approximately a 71.4% chance of assigning a higher predicted draft probability to the drafted player.





```
## ROC Curve
library(pROC)
nfl$predicted_prob <- predict(model, type = "response")
roc_obj <- roc(nfl$Drafted, nfl$predicted_prob)
plot(roc_obj, main = "ROC Curve", col = "blue")
auc(roc_obj)
```

Part II Conclusion

Our analysis examined which measurable player characteristics and athletic performance metrics influence the likelihood of being drafted into the NFL. Using logistic regression, we modeled the binary outcome of draft status based on a range of predictors, including sprint time, strength, agility, body composition, and position type.

Our results show that several factors significantly affect draft probability:

- Faster sprint times, younger age, greater upper-body strength (measured by bench press reps), and lower BMI were all associated with a higher likelihood of being drafted.
- Certain position types, such as defensive backs, also showed a statistically significant increase in draft probability.

Model evaluation metrics support the validity of these findings:

- The logistic regression model achieved an accuracy of approximately 68.9%.
- The ROC curve showed an AUC of 0.7138, indicating good classification performance.

These insights may be useful to scouts, coaches, or players themselves aiming to understand how specific attributes influence draft outcomes.

Report Conclusion

Through linear regression, we were able to find which variable(s) out of player type, vertical jump, and height affect the distance of a player's broad jump, and make prediction estimates. From the linear regression test, we came to a final model that only shows vertical jump as significant in predicting broad jump distances. Hence, we made prediction intervals and used the model to create estimates to test our model's accuracy. There was an average prediction error of about 5 centimeters, and mostly over-predicting. Since the margin of error is small, we conclude our model is fairly accurate.

Through logistic regression, we were able to find which variables out of various physical attributes and performance metrics predict whether or not a player may get drafted into the NFL. After fitting a logistic regression model, we found the following variables to be significant in predicting the NFL draft: sprint times, age, upper-body strength (measured by bench press reps), BMI, and position type. Consequently, the final prediction model, including all these variables, proved to be satisfactory with a model accuracy of 68.9% and an AUC of 0.7138 from an ROC curve, demonstrating good classification accuracy.

With our findings, we can train aspiring NFL players by focusing on the most relevant variables that determine whether you are drafted to the NFL and how to increase the performance of certain tests, like the broad jump.

Contributions

Introduction, Part I EDA– Tyler Lee
Linear Regression (Part I), Report Conclusion– Thy Pham, Christina Deng
Logistic Regression (Part II)-- Andrew Chen, Chaewon Park