IST 707 Final Project - Predicting NFL Playoff Eligibility

Tesslyn Knapp, Joshua McCleary, Bobby Schaible

12/11/2020

Introduction

The National Football League, or the NFL for short, was not always the exciting and lucrative sports league we know and love today. Formed in 1918, the league came about right after football's most vulnerable time. Lack of real protection and enduring vicious hits in the early 20th century had led to a number of player deaths, particularly at the collegiate level. The carnage was so troublesome that president Theodore Roosevelt, a staunch fan of the game, acknowledged that the sport needed reform if it was to continue. The saving grace for the sport was the introduction of the forward pass.

Forward passing revolutionized the safety and excitement of a quickly dying sport and presented an opportunity for wealthy businessmen to capitalize. Using the quickly growing pool of collegiate players, the burgeoning league began to build its player and fan-bases. Weathering a World War, multiple franchise changes, and a league merger, the game of American Football ingrained itself into the American psyche. With such a vested interest in the sport, many have looked for more innovative ways to study and improve it. From this initiative, sports analytics was born.

Sports analytics provides a wide range of knowledge regarding specific teams and the sport under analysis. For the NFL, this data is used for recovery timetables, drafting new talent, evaluating current talent, and making effective game plans. All of these uses translate to hundreds of millions of dollars for both the players and franchises because their profit margins are linked to how well games are played and how often superstars appear in them. This is especially true now that fantasy football has gained enormous popularity within the NFL fan-base. Fantasy football had the unintended consequence of shifting the focus away from teams and their playoff chances to individual players and how their production influences the fantasy owner's playoff hopes. When prize money is added to the mix, the statistics and analyses created by the data scientist or sourced from apps like ESPN or the Dominator app become invaluable.

Analysis and Models

About the Data

Importing Data

This dataset is comprised of NFL data from ‘nfl_data.csv’. It has 669 observations of 63 variables. There are 32 teams featured in the dataset playing 16 games per season spanning from years 1998-2019.

```
# Install necessary packages
library(viridis)
library(ggplot2)
library(arules)
library(e1071)
library(caret)
library(dplyr)
library(tidyr)
library(ggplot2)
library(rpart)
library(rpart.plot)
library(rattle)
library(class)
library(randomForest)
library(FactoMineR)
library(ggcorrplot)
library(arulesViz)
library(grid)
library(gridExtra)
library(RCurl)

# Importing data from GitHub
nfl_url <- getURL('https://raw.githubusercontent.com/ttknapp/ist707finalproje
ct/main/nfl_data.csv')
nfl <- read.csv(text=nfl_url, header = TRUE, stringsAsFactors = FALSE)
```

Preliminary Data Cleaning

The team_code column was separated into ‘team’ and ‘year’ columns and the dataset was reordered.

```
# Check data types
#str(nfl)

# Data Cleaning
nfl$year <- substring(nfl$team_code, 4) # grabs everything from team_code sta
rting at 4th character
nfl$team <- substring(nfl$team_code, 1, 3) # grabs first three characters fro
m team_code
nfl <- nfl[,c(1, 64, 65, 2:63)] # moves newly created columns to front of dat
```

```r
aset
head(nfl)
```

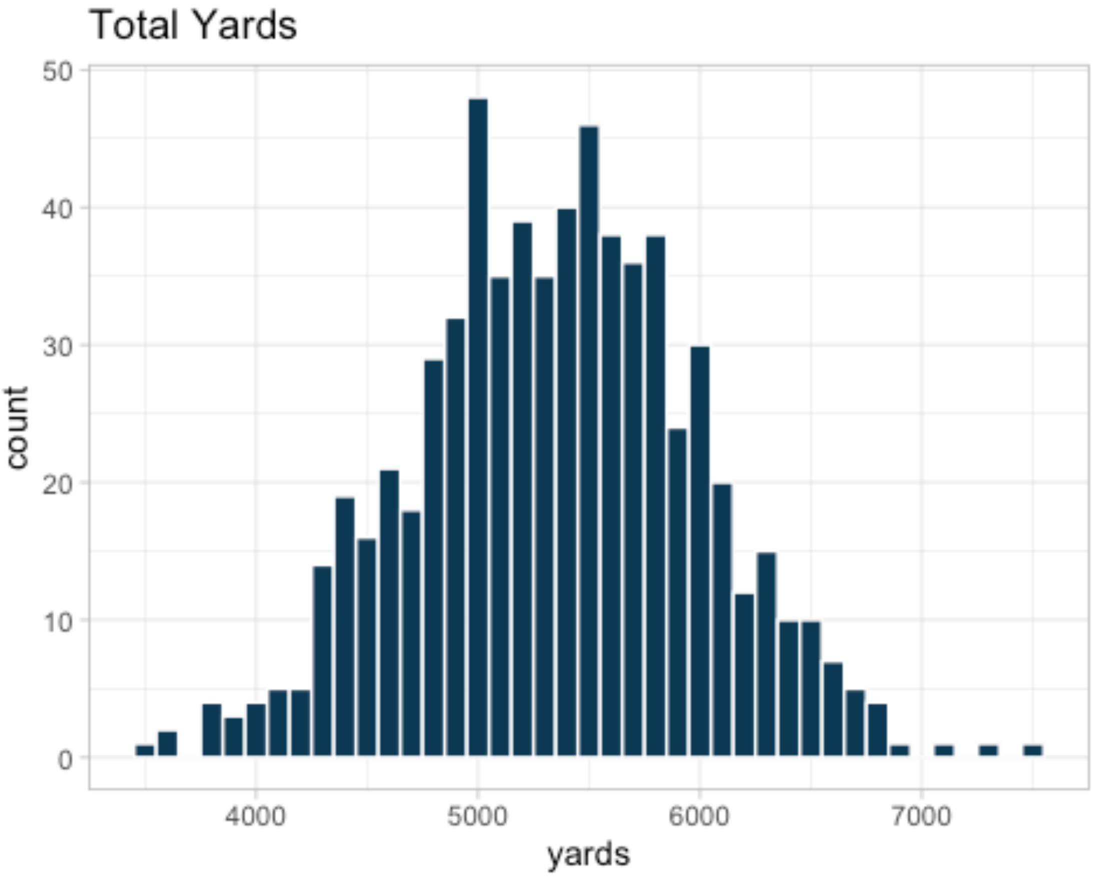
```
##   team_code year team wins losses  PF yards plays yards.play TO Fumbles.Lo
st
## 1   nwe2019 2019  nwe   12      4 420  5664  1095        5.2 15
6
## 2   buf2019 2019  buf   10      6 314  5283  1018        5.2 19
7
## 3   nyj2019 2019  nyj    7      9 276  4368   956        4.6 25
9
## 4   mia2019 2019  mia    5     11 306  4960  1022        4.9 26
8
## 5   rav2019 2019  rav   14      2 531  6521  1064        6.1 15
7
## 6   pit2019 2019  pit    8      8 289  4428   937        4.7 30
11
##   X1st.downs completions pass.attempts pass.yards pass.td int
## 1        338         378           620       3961      25   9
## 2        314         299           513       3229      21  12
## 3        253         323           521       3111      19  16
## 4        315         371           615       3804      22  18
## 5        386         289           440       3225      37   8
## 6        265         315           510       2981      18  19
##   net.yards.per.pass.att pass.1st.downs   X rush.yards rush.td
## 1                    6.1            197 447       1703      17
## 2                    5.8            162 465       2054      13
## 3                    5.4            162 383       1257       6
## 4                    5.7            210 349       1156      10
## 5                    6.9            171 596       3296      21
## 6                    5.5            147 395       1447       7
##   rush.yards.per.att rush.1st.downs penalties pen.yards pen.1st.downs
## 1                3.8            110        94       828            31
## 2                4.4            120       117       927            32
## 3                3.3             61       115      1105            30
## 4                3.3             64        92       769            41
## 5                5.5            188       109       867            27
## 6                3.7             75       111       893            43
##   number.drives score.percentage turnover.percentage avg.start
## 1           185             36.8                 7.6  Own 32.5
## 2           183             30.6                10.4  Own 28.9
## 3           183             23.0                11.5  Own 26.4
## 4           180             30.6                13.3  Own 27.6
## 5           163             52.1                 8.6  Own 28.9
## 6           185             28.6                15.7  Own 29.4
##   avg.time.per.drive avg.plays.per.drive avg.yards.per.drive
## 1               2:40                5.97                29.9
## 2               2:34                5.62                28.6
## 3               2:28                5.23                23.1
## 4               2:33                5.78                27.1
## 5               3:22                6.61                39.3
## 6               2:28                5.17                23.2
##   avg.points.per.drive opp.PF opp.yards opp.plays opp.yards.play opp.TO
## 1                 1.99    225      4414       948            4.7     36
## 2                 1.66    259      4772       985            4.8     23
## 3                 1.21    359      5170      1037            5.0     21
## 4                 1.63    494      6364      1053            6.0     16
## 5                 2.96    282      4809       921            5.2     25
## 6                 1.38    303      4866      1030            4.7     38
##   opp.Fumbles.Lost opp.1st.downs opp.completions opp.pass.attempts
## 1               11           261             303               536
## 2                9           295             348               553
## 3                9           302             363               585
## 4                3           361             344               545
## 5               12           276             318               544
## 6               18           304             314               514
##   opp.pass.yards opp.pass.td opp.int opp.net.yards.per.pass.att
## 1           2886          13      25                        5.0
## 2           3123          15      14                        5.2
## 3           3779          25      12                        6.1
## 4           4198          39      13                        7.4
## 5           3315          15      13                        5.7
## 6           3113          23      20                        5.5
##   opp.pass.1st.downs opp.rush.att opp.rush.yards opp.rush.td
## 1                150          365           1528           7
## 2                169          388           1649          12
## 3                187          417           1391          12
## 4                214          485           2166          15
## 5                163          340           1494          12
## 6                164          462           1753           7
##   opp.rush.yards.per.att opp.rush.1st.downs opp.penalties opp.pen.yards
## 1                    4.2                 72           107           920
## 2                    4.3                 93            94           815
## 3                    3.3                 75           107           902
## 4                    4.5                113           111          1084
## 5                    4.4                 74            97           795
## 6                    3.8                110           115          1118
##   opp.pen.1st.downs opp.number.drives opp.score.percentage
## 1                39               191                 19.4
## 2                33               178                 23.6
## 3                40               189                 34.4
## 4                34               182                 45.6
## 5                39               164                 32.9
## 6                30               184                 29.9
##   opp.turnover.percentage opp.avg.start opp.avg.time.per.drive
## 1                    17.3      Own 24.8                   2:20
## 2                    12.4      Own 27.7                   2:39
## 3                    10.1      Own 31.4                   2:35
## 4                     8.8      Own 30.2                   2:46
## 5                    14.6      Own 27.3                   2:27
## 6                        19.0      Own 30.5                   2:41
##   opp.avg.plays.per.drive opp.avg.yards.per.drive opp.avg.points.per.drive
## 1                     5.0                    22.8                     1.00
## 2                     5.6                    26.2                     1.29
## 3                     5.6                    26.7                     1.81
## 4                     5.9                    34.3                     2.53
## 5                     5.7                    28.8                     1.65
## 6                     5.7                    25.7                     1.55
```

Exploratory Data Analysis

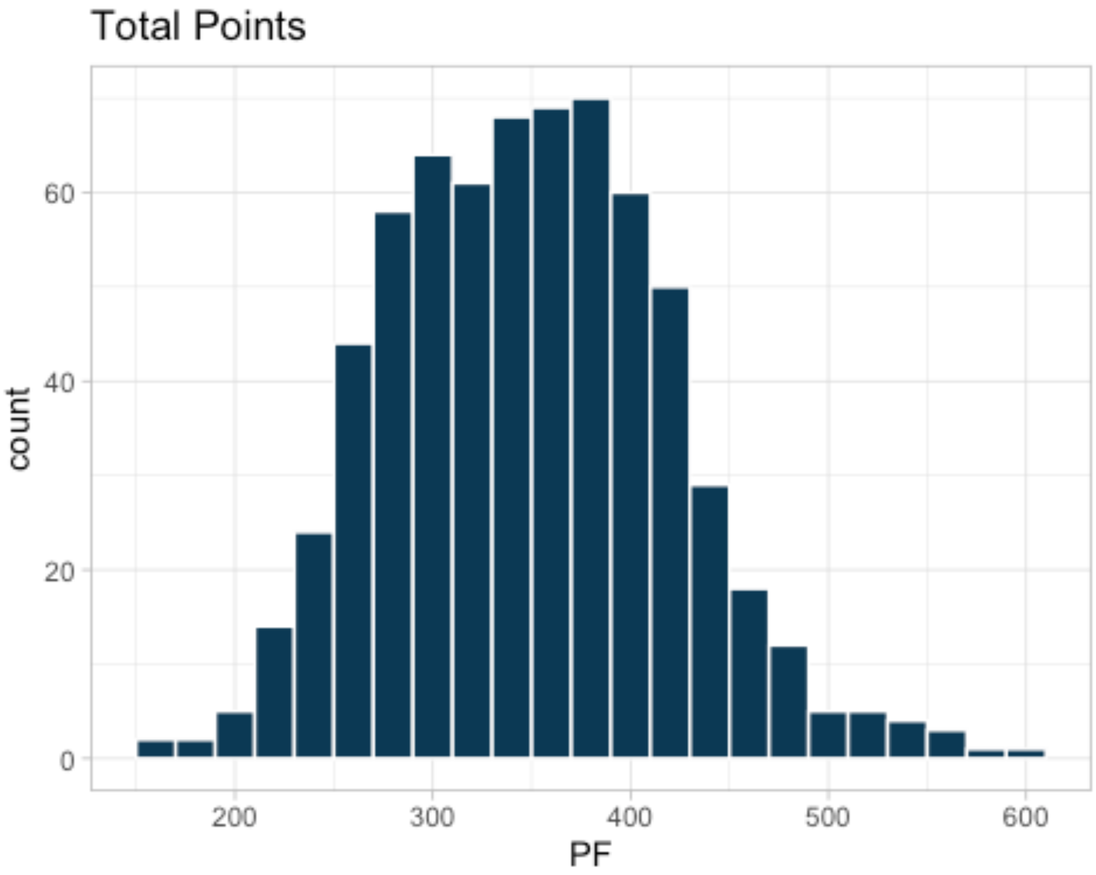
Histograms of Total NFL Yards, Points, and Wins

In order to get a better understanding of the variable distributions, histograms were created for total NFL yards, points, and wins.

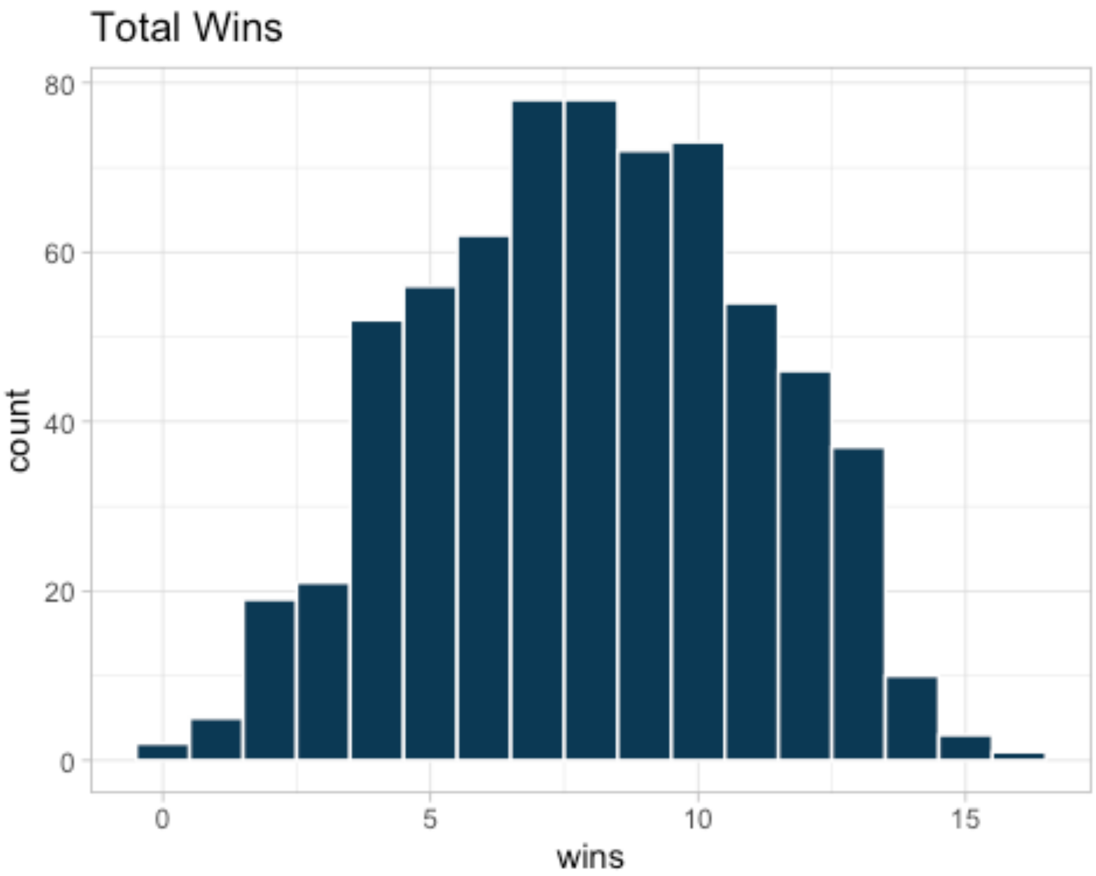
```
# Histograms of Total NFL Yards, Points, and Wins
# Total yards histogram
ggplot(nfl, aes(x = yards)) + theme_light() + geom_histogram(binwidth = 100,
color = "white", fill = "#0b3954") +
  ggtitle("Total Yards")
```



```
# Total points scored histogram
ggplot(nfl, aes(x = PF)) + theme_light() + geom_histogram(binwidth = 20, colo
r = "white", fill = "#0b3954") +
  ggtitle("Total Points")
```



```
# Total wins histogram
ggplot(nfl, aes(x = wins)) + theme_light() + geom_histogram(binwidth =1, colo
r = "white", fill = "#0b3954") +
  ggtitle("Total Wins")
```



```
tapply(nfl$yards, list(nfl$year), mean)

##      1999     2000     2001     2002     2003     2004     2005     2006 
## 5100.258 5110.806 5081.290 5253.750 5092.781 5234.562 5054.719 5153.812 
##      2007     2008     2009     2010     2011     2012     2013     2014 
## 5203.500 5235.844 5362.188 5376.781 5549.344 5555.219 5575.656 5570.281 
##      2015     2016     2017     2018     2019 
## 5642.719 5606.312 5345.000 5635.562 5565.844
```

It seems that most teams gain about 4500-6000 yards per season (and therefore between 280-375 yards per game). The maximum number of yards gained in one season is 7474 yards by the New Orleans Saints in 2011.

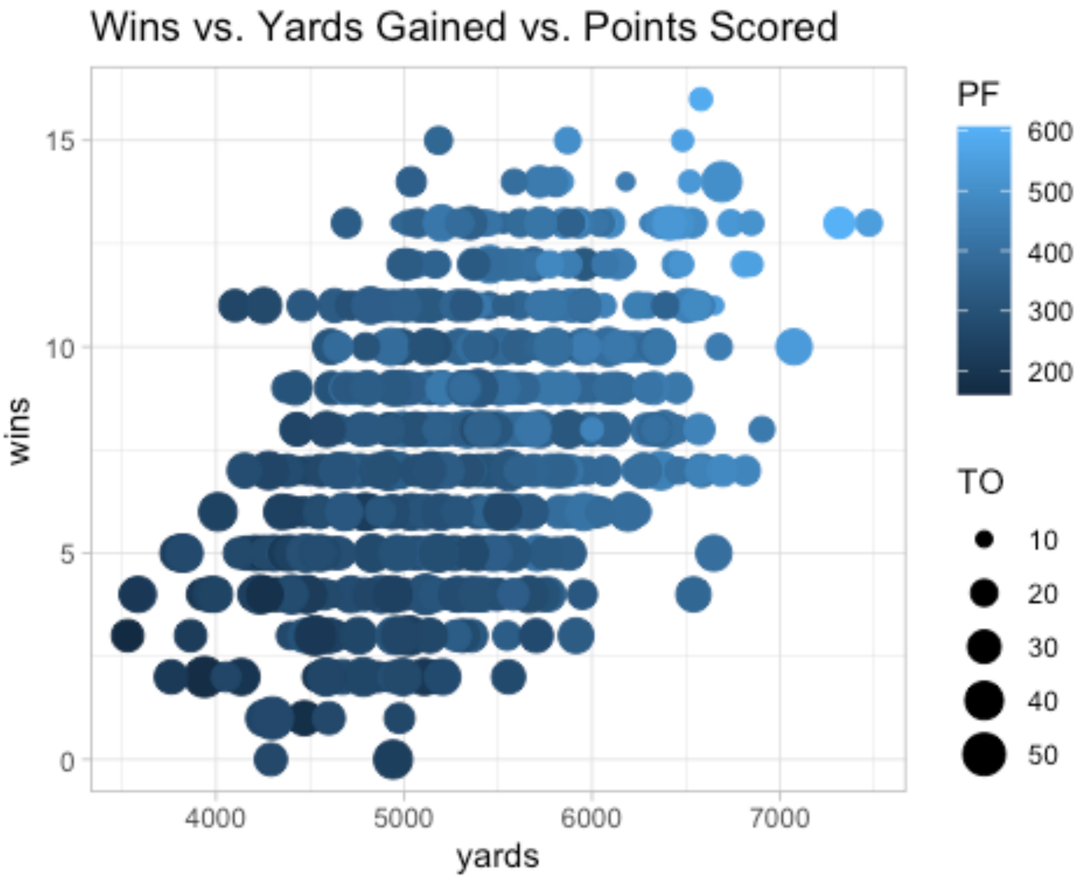
Most teams score between 250-400 points per season (and therefore averaging about 12-15 points per game). The maximum amount of points scored in one season is 606 by the Denver Broncos in 2013.

Most teams average about 7 or fewer wins per season with a maximum of 16 wins.

Comparing Yards, Wins, Passes For, and Turnovers for the Season

Additionally, it was interesting to analyze yards, wins, passes for, and turnovers.

```
# Plot comparing Yards, Wins, Passes For, and Turnovers for the season
ggplot(nfl, aes(x = yards, y = wins, color = PF)) + geom_point(aes(size = TO)
) +
  theme_light() + ggtitle("Wins vs. Yards Gained vs. Points Scored")
```



Teams with more yards gained and more points scored usually win more games. Turnovers are more variable, but generally fewer turnovers trend with more points-for and wins. Also, teams with greater than 20 turnovers and scoring less than 400 points in a season typically have fewer than 10 wins.

Data Cleaning

In order to predict playoff eligibility, a new column was created containing all of the teams that made the playoffs between 1998-2019.

```
# Create new data frame containing all playoff teams since 1999
playoffs <- c("sfo2019", "gnb2019", "nor2019", "phi2019", "sea2019", "min2019
"
              , "rav2019", "kan2019", "nwe2019", "htx2019", "buf2019", "oti20
19"
              , "nor2018", "ram2018", "chi2018", "dal2018", "sea2018", "phi20
18"
              , "kan2018", "nwe2018", "htx2018", "rav2018", "sdg2018", "clt20
```

```
18"
             , "phi2017", "min2017", "ram2017", "nor2017", "car2017", "atl20
17"
             , "nwe2017", "pit2017", "jax2017", "kan2017", "oti2017", "buf20
17"
             , "dal2016", "atl2016", "sea2016", "gnb2016", "nyg2016", "det20
16"
             , "nwe2016", "kan2016", "pit2016", "htx2016", "rai2016", "mia20
16"
             , "car2015", "crd2015", "min2015", "was2015", "gnb2015", "sea20
15"
             , "den2015", "nwe2015", "cin2015", "htx2015", "kan2015", "pit20
15"
             , "sea2014", "gnb2014", "dal2014", "car2014", "crd2014", "det20
14"
             , "nwe2014", "den2014", "pit2014", "clt2014", "cin2014", "rav20
14"
             , "sea2013", "car2013", "phi2013", "gnb2013", "sfo2013", "nor20
13"
             , "den2013", "nwe2013", "cin2013", "clt2013", "kan2013", "sdf20
13"
             , "atl2012", "sfo2012", "gnb2012", "was2012", "sea2012", "min20
12"
             , "den2012", "nwe2012", "htx2012", "rav2012", "clt2012", "cin20
12"
             , "gnb2011", "sfo2011", "nor2011", "nyg2011", "atl2011", "det20
11"
             , "nwe2011", "rav2011", "htx2011", "den2011", "pit2011", "cin20
11"
             , "atl2010", "chi2010", "phi2010", "sea2010", "nor2010", "gnb20
10"
             , "nwe2010", "pit2010", "clt2010", "kan2010", "rav2010", "nyj20
10"
             , "nor2009", "min2009", "dal2009", "crd2009", "gnb2009", "phi20
09"
             , "clt2009", "sdg2009", "nwe2009", "cin2009", "nyj2009", "rav20
09"
             , "nyg2008", "car2008", "min2008", "crd2008", "atl2008", "phi20
08"
             , "oti2008", "pit2008", "mia2008", "sdg2008", "clt2008", "rav20
08"
             , "dal2007", "gnb2007", "sfo2007", "tam2007", "nyg2007", "was20
07"
             , "nwe2007", "clt2007", "sdg2007", "pit2007", "jax2007", "oti20
07"
             , "chi2006", "nor2006", "phi2006", "sea2006", "dal2006", "nyg20
06"
             , "sdg2006", "rav2006", "clt2006", "nwe2006", "nyj2006", "kan20
06"
             , "sea2005", "chi2005", "tam2005", "nyg2005", "car2005", "was20
```

```
05"
              , "clt2005", "den2005", "cin2005", "nwe2005", "jax2005", "pit20
05"
              , "phi2004", "atl2004", "gnb2004", "sea2004", "ram2004", "min20
04"
              , "pit2004", "nwe2004", "clt2004", "sdg2004", "nyj2004", "den20
04"
              , "phi2003", "ram2003", "car2003", "gnb2003", "sea2003", "dal20
03"
              , "nwe2003", "kan2003", "clt2003", "rav2003", "oti2003", "den20
03"
              , "phi2002", "tam2002", "gnb2002", "sfo2002", "nyg2002", "atl20
02"
              , "rai2002", "oti2002", "pit2002", "nyj2002", "clt2002", "cle20
02"
              , "ram2001", "chi2001", "phi2001", "gnb2001", "sfo2001", "tam20
01"
              , "pit2001", "nwe2001", "rai2001", "mia2001", "rav2001", "nyj20
01"
              , "nyg2000", "min2000", "nor2000", "phi2000", "tam2000", "ram20
00"
              , "oti2000", "rai2000", "mia2000", "rav2000", "den2000", "clt20
00"
              , "ram1999", "tam1999", "was1999", "min1999", "dal1999", "det19
99"
              , "jax1999", "clt1999", "sea1999", "oti1999", "buf1999", "mia19
99")

# Add playoff teams as a new column in the NFL dataset
nfl$playoffs <- nfl$team_code %in% playoffs
nfl$playoffs <- ifelse(nfl$playoffs == "TRUE", 1, 0)
tapply(nfl$playoffs, list(nfl$year), sum) #check to see if all 12 playoff tea
ms were coded properly
```

```
## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013
2014
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   11
12
## 2015 2016 2017 2018 2019
##   12   12   12   12   12
```

```
nfl$playoffs <- as.factor(nfl$playoffs) #change playoffs column to factor dat
a type
#str(nfl)
data.frame(colnames(nfl))
```

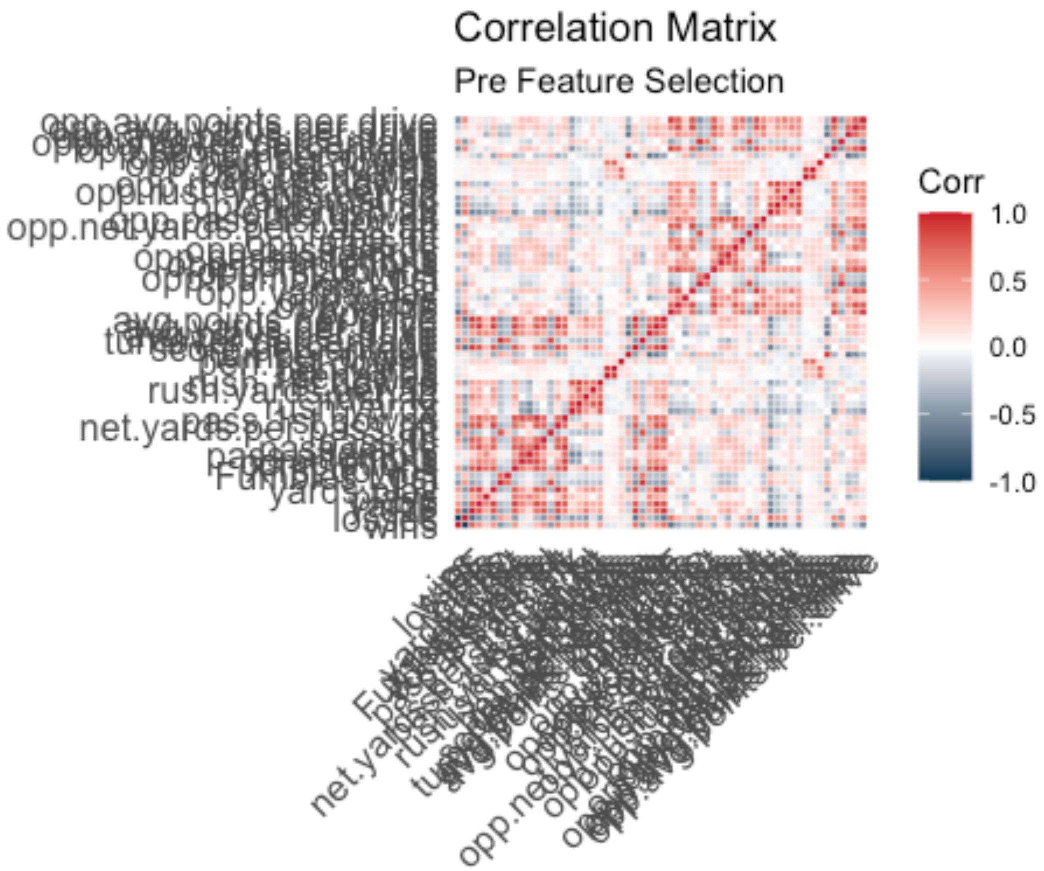
```
##                   colnames.nfl.
## 1                     team_code
## 2                          year
## 3                          team
## 4                        wins
## 5                      losses
## 6                          PF
## 7                       yards
## 8                       plays
## 9                  yards.play
## 10                         TO
## 11               Fumbles.Lost
## 12                 X1st.downs
## 13                completions
## 14              pass.attempts
## 15                 pass.yards
## 16                    pass.td
## 17                        int
## 18     net.yards.per.pass.att
## 19             pass.1st.downs
## 20                          X
## 21                 rush.yards
## 22                    rush.td
## 23         rush.yards.per.att
## 24             rush.1st.downs
## 25                  penalties
## 26                  pen.yards
## 27              pen.1st.downs
## 28              number.drives
## 29           score.percentage
## 30        turnover.percentage
## 31                  avg.start
## 32         avg.time.per.drive
## 33        avg.plays.per.drive
## 34        avg.yards.per.drive
## 35       avg.points.per.drive
## 36                     opp.PF
## 37                  opp.yards
## 38                  opp.plays
## 39             opp.yards.play
## 40                     opp.TO
## 41           opp.Fumbles.Lost
## 42              opp.1st.downs
## 43            opp.completions
## 44          opp.pass.attempts
## 45             opp.pass.yards
## 46                opp.pass.td
## 47                    opp.int
## 48 opp.net.yards.per.pass.att
## 49         opp.pass.1st.downs
## 50               opp.rush.att
## 51             opp.rush.yards
## 52                opp.rush.td
## 53     opp.rush.yards.per.att
## 54         opp.rush.1st.downs
## 55              opp.penalties
## 56              opp.pen.yards
## 57          opp.pen.1st.downs
## 58          opp.number.drives
## 59       opp.score.percentage
## 60    opp.turnover.percentage
## 61              opp.avg.start
## 62     opp.avg.time.per.drive
## 63    opp.avg.plays.per.drive
## 64    opp.avg.yards.per.drive
## 65   opp.avg.points.per.drive
## 66                   playoffs

newNFL <- nfl[,-c(1:3,31:32,61:62)] #Remove non-numerical features
#str(newNFL)
```

Teams that made the playoffs were given a value of 1 and teams that did not make the playoffs were given a value of 0.

Model Pre-Processing

In order to reduce the likelihood of overfitting the model and skewing the data, analysis was done on the correlation of variables to decide which variables, if any, to remove for future playoff prediction models.

```
# Find most correlated variables in order to remove for further analysis
set.seed(4321)
correlationMatrix <- cor(newNFL[,-59])
ggcorrplot(correlationMatrix, outline.color = "white", colors = c("#0b3954",
"white", "#C81D25")) + ggtitle("Correlation Matrix", subtitle = "Pre Feature
Selection")
```



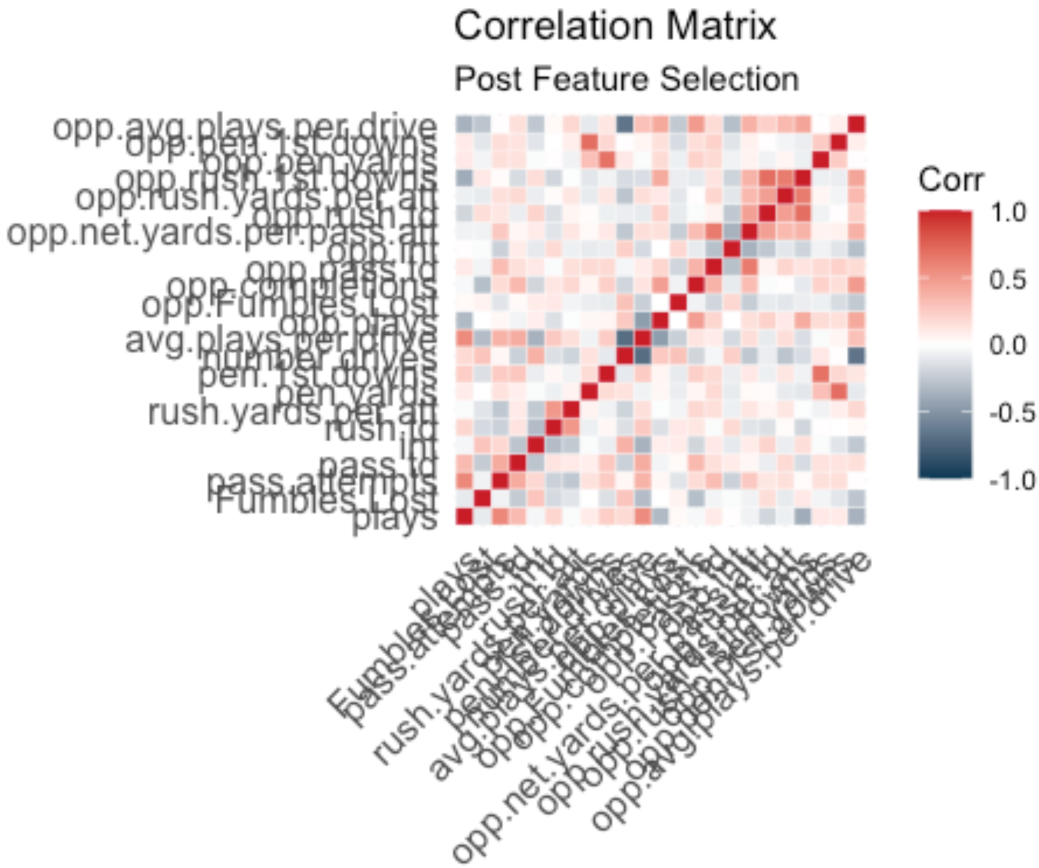
```
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff = 0.75) #Identify highly correlated features
colnames(newNFL[,highlyCorrelated]) #View highly correlated features
```

```
##  [1] "avg.yards.per.drive"      "avg.points.per.drive"
##  [3] "losses"                   "score.percentage"
##  [5] "opp.avg.yards.per.drive"  "X1st.downs"
##  [7] "yards"                    "PF"
##  [9] "yards.play"               "opp.rush.att"
## [11] "opp.avg.points.per.drive" "opp.score.percentage"
## [13] "opp.1st.downs"            "pass.yards"
## [15] "net.yards.per.pass.att"   "opp.PF"
## [17] "opp.yards.play"           "opp.yards"
## [19] "pass.1st.downs"           "opp.pass.yards"
## [21] "completions"              "opp.pass.1st.downs"
## [23] "X"                        "opp.rush.yards"
## [25] "TO"                       "rush.1st.downs"
## [27] "opp.number.drives"        "opp.pass.attempts"
## [29] "turnover.percentage"      "rush.yards"
## [31] "opp.TO"                   "opp.turnover.percentage"
## [33] "opp.penalties"            "penalties"
```

```
cleanNFL<- newNFL[,-highlyCorrelated] #Remove highly correlated features
colnames(cleanNFL)
```

```
##  [1] "wins"                          "plays"
##  [3] "Fumbles.Lost"                  "pass.attempts"
##  [5] "pass.td"                       "int"
##  [7] "rush.td"                       "rush.yards.per.att"
##  [9] "pen.yards"                     "pen.1st.downs"
## [11] "number.drives"                 "avg.plays.per.drive"
## [13] "opp.plays"                     "opp.Fumbles.Lost"
## [15] "opp.completions"               "opp.pass.td"
## [17] "opp.int"                       "opp.net.yards.per.pass.att"
## [19] "opp.rush.td"                   "opp.rush.yards.per.att"
## [21] "opp.rush.1st.downs"            "opp.pen.yards"
## [23] "opp.pen.1st.downs"             "opp.avg.plays.per.drive"
## [25] "playoffs"
```

```
cleanNFL <- cleanNFL[,-1] #Remove wins as a feature
updatedCorMatrix <- cor(cleanNFL[,-24])
ggcorrplot(updatedCorMatrix, outline.color = "white", colors = c("#0b3954", "
white", "#C81D25")) + ggtitle("Correlation Matrix", subtitle = "Post Feature
Selection")
```

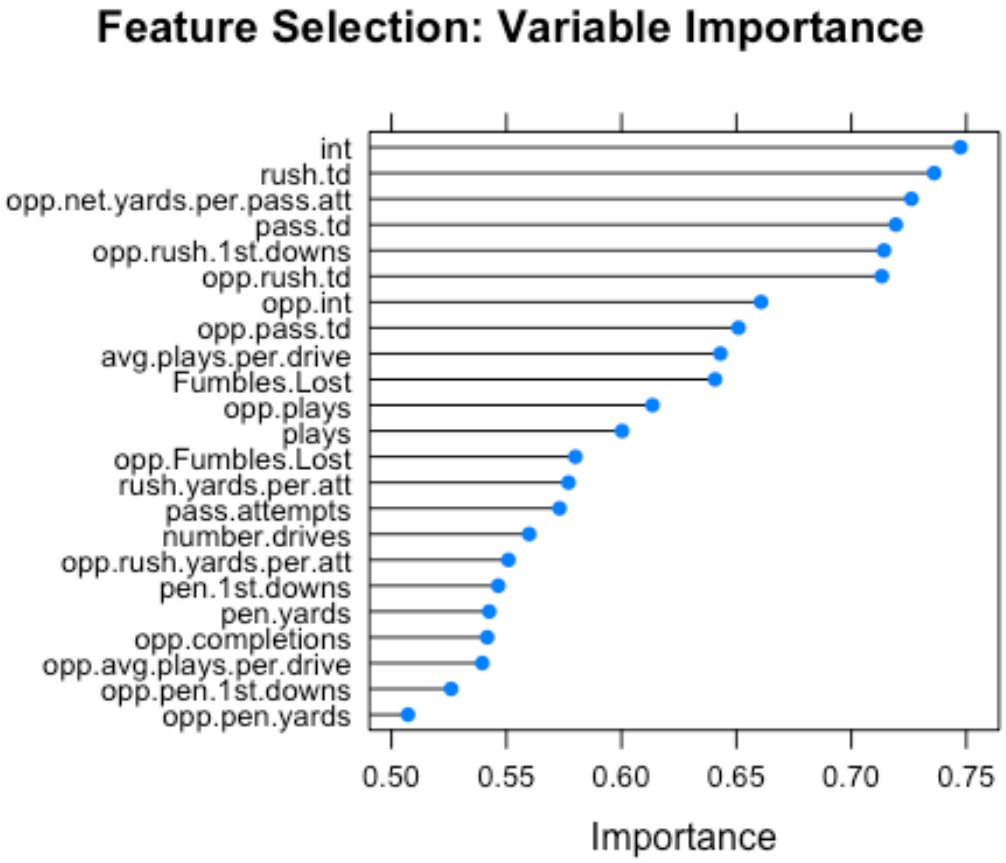


The most highly correlated features were removed from the data frame. Wins did not appear among the most highly correlated features, but they were still removed because losses appeared under the most highly correlated features.

The data was reduced from 66 variables to 24 variables.

Next, the variables were ranked individually by importance in playoff eligibility.

```
# Visualize variable importance
control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
model <- train(playoffs~., data = cleanNFL, method = "lvq", preProcess = "sca
le", trControl = control)
importance <- varImp(model, scale = FALSE)
plot(importance, main = "Feature Selection: Variable Importance")
```



Data Discretization

The data needed to be discretized for playoff prediction modeling. The important variables for playoff appearance were discretized into 4 bins by frequency.

```
# Discretize NFL data
discreteNFL <- discretizeDF(cleanNFL, default = list(method = "frequency", br
eaks = 4))
head(discreteNFL)
```

```
##                  plays Fumbles.Lost pass.attempts pass.td     int rush.td
## 1 [1.04e+03,1.19e+03]        [2,8)     [582,740] [22,27)  [2,12) [16,32]
## 2 [1.01e+03,1.04e+03)        [2,8)     [501,540) [18,22) [12,15) [13,16)
## 3           [859,982)       [8,10)     [501,540) [18,22) [15,19)   [2,9)
## 4 [1.01e+03,1.04e+03)       [8,10)     [582,740] [22,27) [15,19)  [9,13)
## 5 [1.04e+03,1.19e+03]        [2,8)     [358,501) [27,55]  [2,12) [16,32]
## 6           [859,982)      [10,13)     [501,540) [18,22) [19,32]   [2,9)
##   rush.yards.per.att      pen.yards pen.1st.downs number.drives
## 1            [3,3.9)      [765,859)       [26,32)     [179,187)
## 2          [4.4,5.5]      [859,948)       [32,50]     [179,187)
## 3            [3,3.9) [948,1.36e+03]       [26,32)     [179,187)
## 4            [3,3.9)      [765,859)       [32,50]     [179,187)
## 5          [4.4,5.5]      [859,948)       [26,32)     [150,179)
## 6            [3,3.9)      [859,948)       [32,50]     [179,187)
##   avg.plays.per.drive           opp.plays opp.Fumbles.Lost opp.completions
## 1         [5.85,6.67]           [882,981)          [11,13)       [302,328)
## 2          [5.6,5.85)      [981,1.01e+03)           [8,11)       [328,352)
## 3         [4.56,5.37) [1.01e+03,1.04e+03)           [8,11)       [352,425]
## 4          [5.6,5.85) [1.04e+03,1.16e+03]            [2,8)       [328,352)
## 5         [5.85,6.67]           [882,981)          [11,13)       [302,328)
## 6         [4.56,5.37) [1.01e+03,1.04e+03)          [13,26]       [302,328)
##   opp.pass.td opp.int opp.net.yards.per.pass.att opp.rush.td
## 1      [6,19) [19,33]                  [4.3,5.7)      [3,10)
## 2      [6,19) [12,15)                  [4.3,5.7)     [12,16)
## 3     [23,27) [12,15)                  [6.1,6.6)     [12,16)
## 4     [27,45] [12,15)                  [6.6,7.9]     [12,16)
## 5      [6,19) [12,15)                  [5.7,6.1)     [12,16)
## 6     [23,27) [19,33]                  [4.3,5.7)      [3,10)
##   opp.rush.yards.per.att opp.rush.1st.downs  opp.pen.yards opp.pen.1st.dow
ns
## 1              [4.2,4.5)            [53,84)      [853,948)           [32,5
8]
## 2              [4.2,4.5)            [84,96)      [763,853)           [32,5
8]
## 3              [2.7,3.9)            [53,84)      [853,948)           [32,5
8]
## 4              [4.5,5.3]          [108,161] [948,1.21e+03]           [32,5
8]
## 5              [4.2,4.5)            [53,84)      [763,853)           [32,5
8]
## 6              [2.7,3.9)          [108,161] [948,1.21e+03]           [27,3
2)
##   opp.avg.plays.per.drive playoffs
## 1               [4.8,5.4)        1
## 2               [5.6,5.8)        1
## 3               [5.6,5.8)        0
## 4               [5.8,6.7]        0
## 5               [5.6,5.8)        1
## 6               [5.6,5.8)        0
```

```
#str(discreteNFL)
```

Next, a sample of 80% of the data was created in order to assign as training data. The remaining 20% of the data was assigned as testing data.

```
# Derive sample of total data and separate into testing and training data
set.seed(4321)
split <- sample(nrow(discreteNFL), nrow(discreteNFL)*.80, replace = FALSE)
nflTrain <- discreteNFL[split,]
nflTest <- discreteNFL[-split,]
nrow(nflTrain)

## [1] 535

nrow(nflTest)

## [1] 134
```

There are 535 teams in the training data and 134 teams in the testing data.

Finally, two different data frames were created to include playoff results and exclude playoff results to test model accuracy.

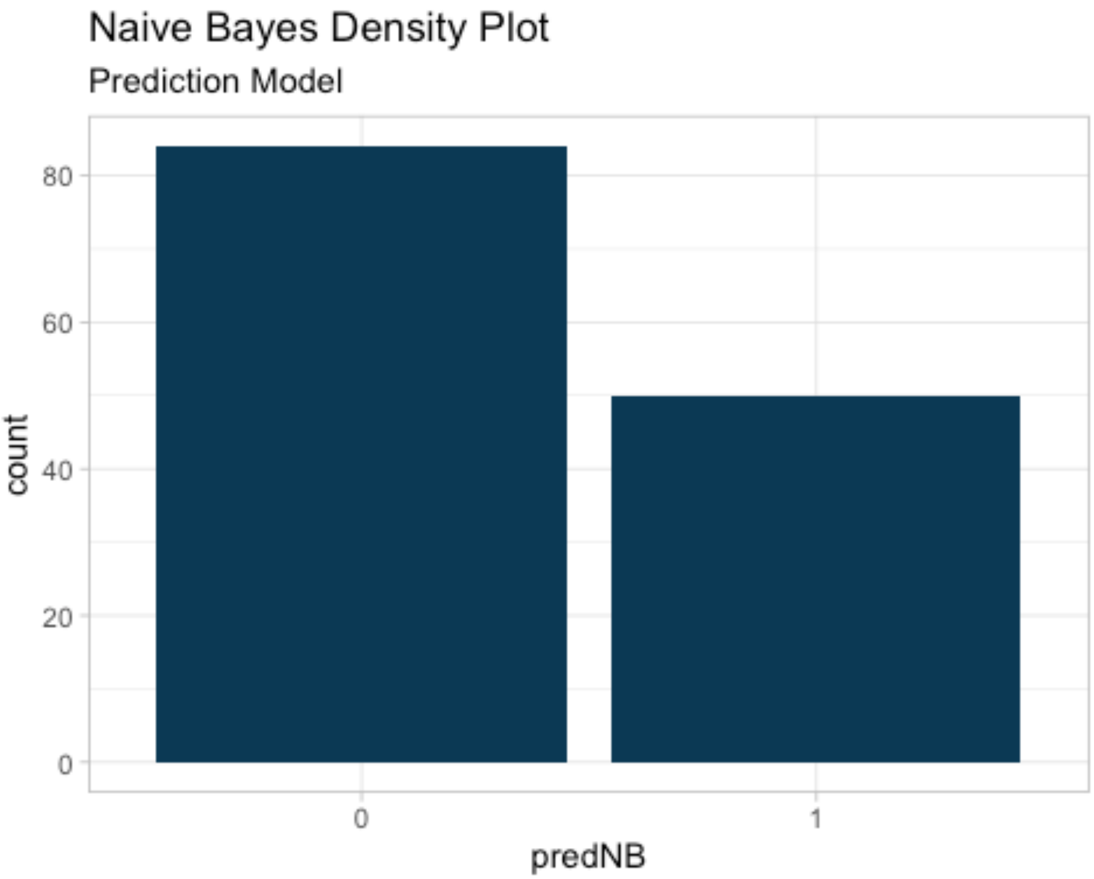
```
# Create 2 data frames with one including playoff results and one excluding p
layoff results
nflTestPlayoff <- nflTest$playoffs # Keeps just the result column
nflTestNoPlayoff <- nflTest[,-66] # Removes result column
```

Results

Naive Bayes Model

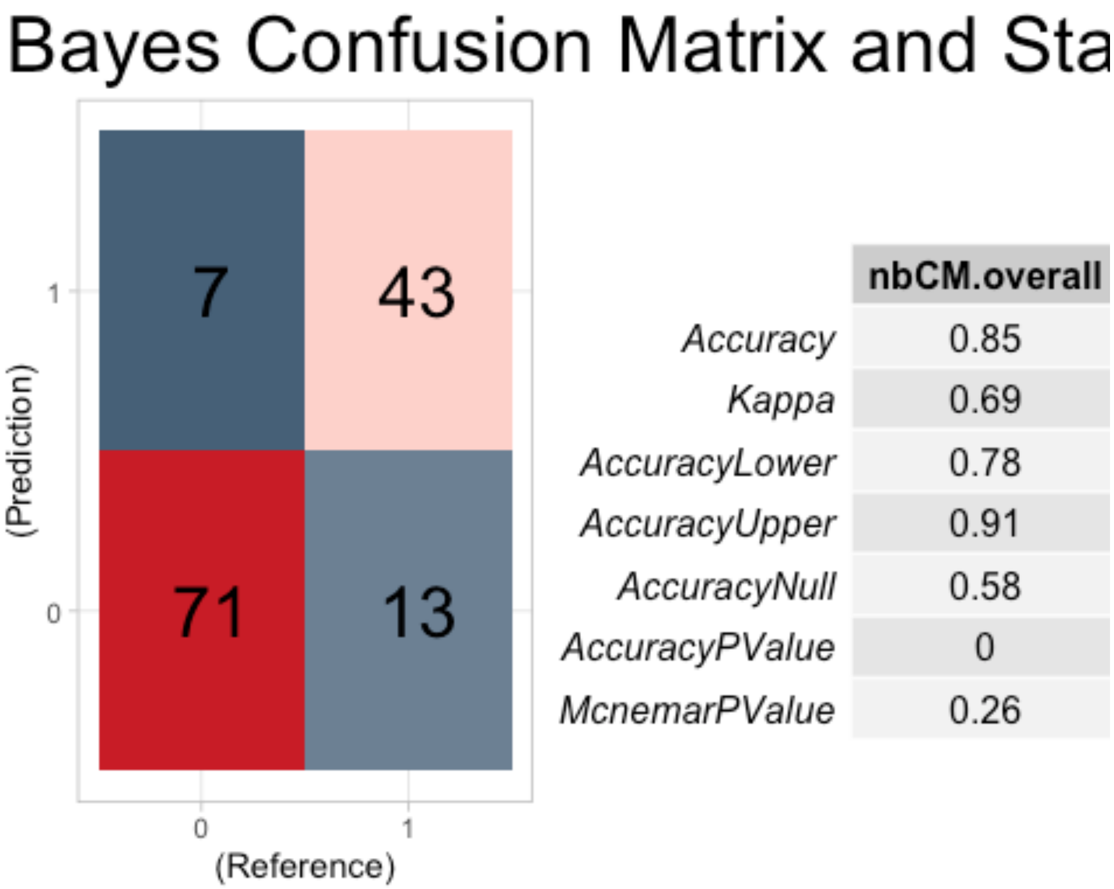
Naive Bayes is a technique that assigns class labels to vectors of feature values. The Naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. There are some drawbacks, as Naive Bayes can rely on oversimplified assumptions which can affect the accuracy of the model.

```
# Naive Bayes Model
set.seed(4321)
nflNB <- naiveBayes(playoffs~., data = nflTrain, na.action = na.pass)
predNB <- predict(nflNB, nflTestNoPlayoff)
predNB2 <- as.data.frame(predNB)
ggplot(data = predNB2, aes(x = predNB)) + geom_bar(fill = "#0B3954") + theme_
light() +
  ggtitle("Naive Bayes Density Plot", subtitle = "Prediction Model")
```



Naive Bayes Confusion Matrix

```
# Naive Bayes Confusion Matrix
nbCM <- confusionMatrix(predNB, nflTestPlayoff)
nbCM_table <- as.data.frame(nbCM$table)
nbCM_stats <- data.frame(nbCM$overall)
nbCM_stats$nbCM.overall <- round(nbCM_stats$nbCM.overall, 2)
nbCM_d_p <- ggplot(data = nbCM_table, aes(x=(Reference), y=(Prediction), fill
= Freq)) + geom_tile() +
  geom_text(aes(label = paste("",Freq)), color = 'black', size = 8) +
  theme_light() +
  guides(fill=FALSE) + scale_fill_gradient2(low="#0B3954", mid = "white", hig
h = "#c81d25", midpoint = 35)
nbCM_stats_p <- tableGrob(nbCM_stats)
grid.arrange(nbCM_d_p, nbCM_stats_p,nrow = 1, ncol = 2,
             top=textGrob("Naive Bayes Confusion Matrix and Statistics",gp=gp
ar(fontsize=25,font=1)))
```



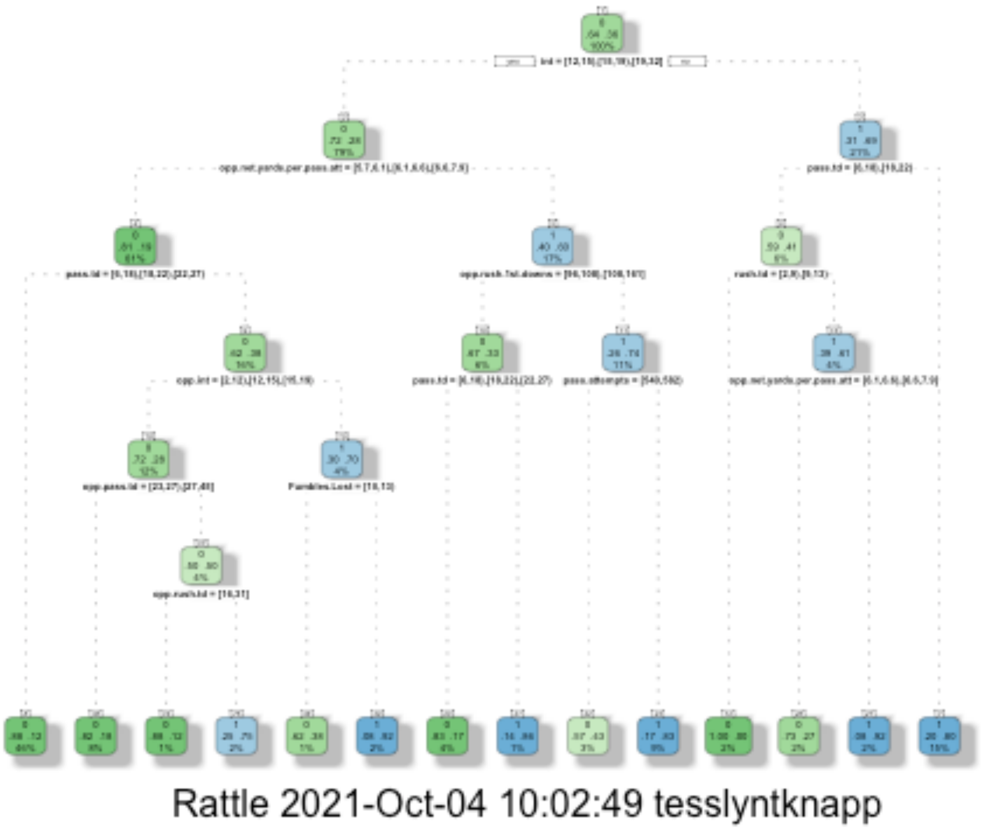
	nbCM.overall
Accuracy	0.85
Kappa	0.69
AccuracyLower	0.78
AccuracyUpper	0.91
AccuracyNull	0.58
AccuracyPValue	0
McnemarPValue	0.26

The Naive Bayes model achieved an accuracy of 85.07% in correctly predicting NFL playoff eligibility.

Decision Tree Model

A Decision Tree is a flowchart-like structure in which each internal node represents a “test” on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules. Decision Trees tend to have issues with overfitting, but can still be helpful for analyses.

```r
# Decision Tree Model
set.seed(4321)
nflDT <- rpart(playoffs~., data = nflTrain, method = "class")
fancyRpartPlot(nflDT)
```


yes
int = [12,15),[15,18),[18,32]
no
opp.net.yards.per.pass.att = [5.7,6.1),[6.1,6.6),[6.6,7.9]
pass.td = [6,18),[18,22]
pass.td = [6,18),[18,22),[22,27]
opp.rush.1st.downs = [96,108),[108,161]
rush.td = [2,9),[9,13]
opp.int = [2,12),[12,15),[15,18]
pass.td = [6,18),[18,22),[22,27]
pass.attempts = [540,592]
opp.net.yards.per.pass.att = [6.1,6.6),[6.6,7.9]
opp.pass.td = [23,27),[27,40]
Fumbles.Lost = [10,13]
opp.rush.td = [16,31]
Rattle 2021-Oct-04 10:02:49 tesslyntknapp

Interceptions were the biggest indicator of a team's playoff chances.

Of the teams with fewer than 12 interceptions in the year, 80% made the playoff if they threw more than 22 passing touchdowns.

Of the teams that missed the playoffs, 46% threw between 6 and 27 passing touchdowns and allowed between 5.7 and 7.9 yards per pass attempt on defense in addition to throwing between 12 and 39 interceptions.

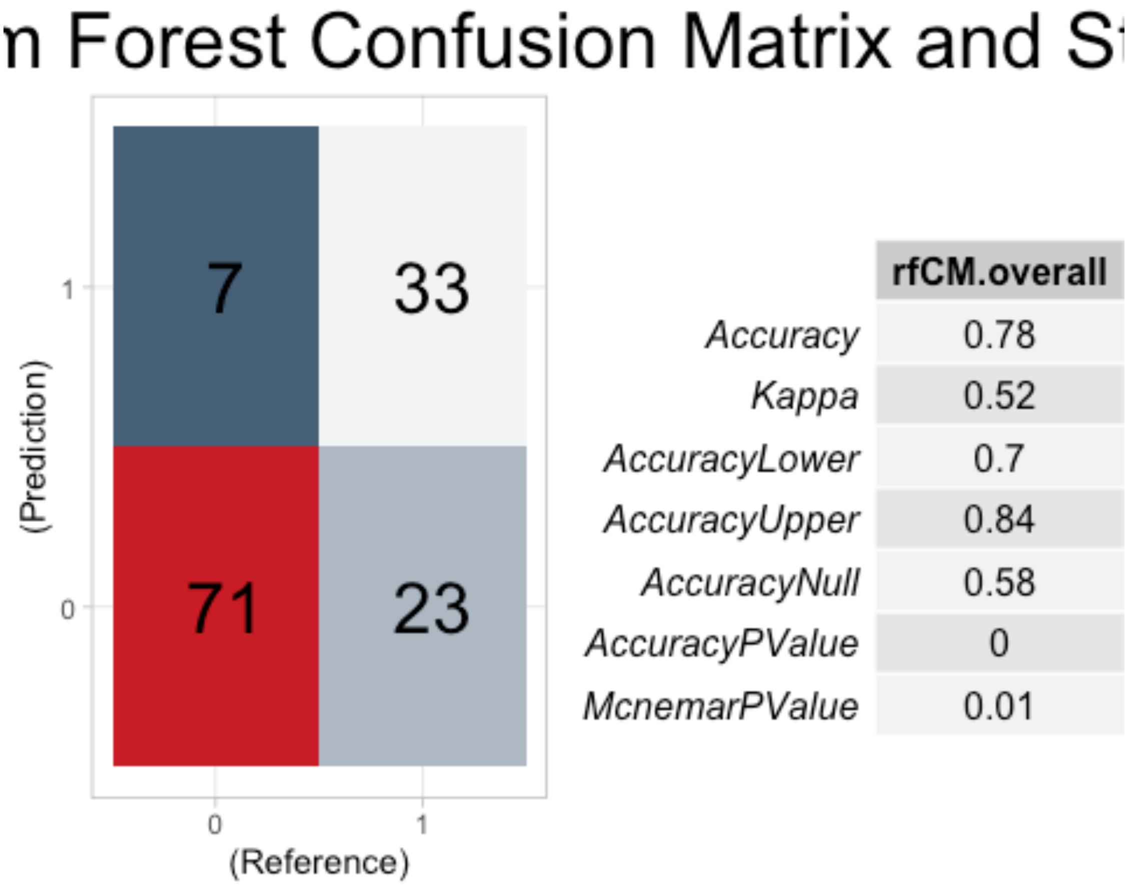
Random Forest Model

Random Forests are an ensemble learning method for classification, regression, and other tasks by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. Random Forests are advantageous to regular Decision Trees because they correct for the overfitting errors commonly found in Decision Tree models. Therefore, they often outperform Decision Trees.

```r
# Random Forest
set.seed(4321)
nflRF <- randomForest(playoffs~., data = nflTrain)
predRF <- predict(nflRF, nflTestNoPlayoff)
```

Random Forest Model Confusion Matrix

```
# Random Forest Confusion Matrix
rfCM <- confusionMatrix(predRF, nflTestPlayoff)
rfCM_table <- as.data.frame(rfCM$table)
rfCM_stats <- data.frame(rfCM$overall)
rfCM_stats$rfCM.overall <- round(rfCM_stats$rfCM.overall, 2)
rfCM_d_p <- ggplot(data = rfCM_table, aes(x=(Reference), y=(Prediction), fill
= Freq)) +
  geom_tile() +geom_text(aes(label = paste("",Freq)), color = 'black', size =
8) +
  theme_light() + guides(fill=FALSE) + scale_fill_gradient2(low="#0B3954", mi
d = "white", high = "#c81d25", midpoint = 35)
rfCM_stats_p <- tableGrob(rfCM_stats)
grid.arrange(rfCM_d_p, rfCM_stats_p,nrow = 1, ncol = 2,
             top=textGrob("Random Forest Confusion Matrix and Statistics",gp=
gpar(fontsize=25,font=1)))
```


n Forest Confusion Matrix and St

	rfCM.overall
Accuracy	0.78
Kappa	0.52
AccuracyLower	0.7
AccuracyUpper	0.84
AccuracyNull	0.58
AccuracyPValue	0
McnemarPValue	0.01

This initial Random Forest Model achieved an accuracy of 77.62% in correctly predicting NFL playoff eligibility.

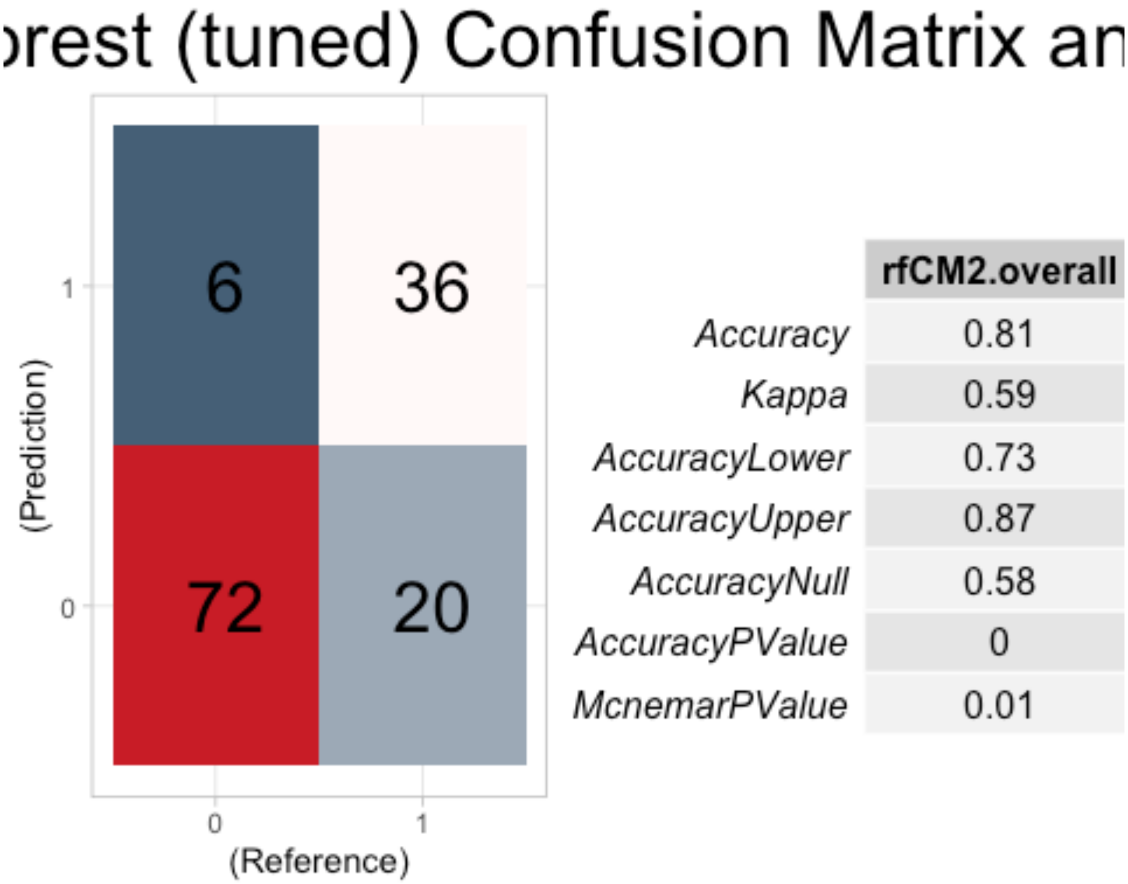
Tuned Random Forest Model

Next, the Random Forest Model was tuned in attempts to increase the model's accuracy. It was tuned by setting a number of trees at 500 and the number of variables available for splitting at each tree node (mtry) to 6.

```
# Tune RF Model
set.seed(4321)
nflRF2 <- randomForest(playoffs~., data = nflTrain, ntree = 500, mtry = 6)
predRF2 <- predict(nflRF2, nflTestNoPlayoff)
rfCM2 <- confusionMatrix(predRF2, nflTestPlayoff)
```

Tuned Random Forest Model Confusion Matrix

```
# Tuned RF Confusion Matrix
rfCM2 <- confusionMatrix(predRF2, nflTestPlayoff)
rfCM2_table <- as.data.frame(rfCM2$table)
rfCM2_stats <- data.frame(rfCM2$overall)
rfCM2_stats$rfCM2.overall <- round(rfCM2_stats$rfCM2.overall, 2)
rfCM2_d_p <- ggplot(data = rfCM2_table, aes(x=(Reference), y=(Prediction), fi
ll = Freq)) +
  geom_tile() +geom_text(aes(label = paste("",Freq)), color = 'black', size =
8) +
  theme_light() + guides(fill=FALSE) + scale_fill_gradient2(low="#0B3954", mi
d = "white", high = "#c81d25", midpoint = 35)
rfCM2_stats_p <- tableGrob(rfCM2_stats)
grid.arrange(rfCM2_d_p, rfCM2_stats_p,nrow = 1, ncol = 2,
             top=textGrob("Random Forest (tuned) Confusion Matrix and Statist
ics",gp=gpar(fontsize=25,font=1)))
```



	rfCM2.overall
Accuracy	0.81
Kappa	0.59
AccuracyLower	0.73
AccuracyUpper	0.87
AccuracyNull	0.58
AccuracyPValue	0
McnemarPValue	0.01

The tuned Random Forest Model achieved an accuracy of 80.60% in correctly predicting NFL playoff eligibility.

Association Rule Mining

Apriori Algorithm

Since the data was cleaned and all of the data types were transformed into factors, the Apriori algorithm could be used in order to generate association rules. These rules are created based off of parameters of support, confidence, and lift and are a good tool to observe associations between variables.

Support: gives an idea of how frequent an item-set is in all the transactions

Confidence: an indication of how often the rule is found to be true

Lift: the ratio of the observed support to the expected

General Rules

The target remained to obtain strong general rules, so a combination of Support = 0.14, Confidence = 0.6, and Lift >= 1 was used to generate those rules. The 10 rules with the strongest confidence of this selection are listed below.

```
# Run Association Rule Mining
# Run Apriori algorithm to generate rules with strong support and confidence.
Then, sort by lift and return top 10 rules in order of confidence.
# General Rules
nfl_rules <- apriori(discreteNFL, parameter = list(support = 0.14, confidence
= 0.6))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.14      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 93
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[94 item(s), 669 transaction(s)] done [0.00s].
## sorting and recoding items ... [94 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [76 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
nflrules_sorted <- sort(nfl_rules, by="lift")
inspect(head(sort(nflrules_sorted, by = "confidence"), 10))
```

```
##      lhs                                      rhs            support conf
idence  coverage     lift count
## [1]  {rush.td=[2,9)}                       => {playoffs=0} 0.1689088  0.8
897638 0.1898356 1.424048   113
## [2]  {opp.rush.td=[16,31],
##       opp.rush.1st.downs=[108,161]}        => {playoffs=0} 0.1434978  0.8
807339 0.1629297 1.409596    96
## [3]  {opp.net.yards.per.pass.att=[6.6,7.9]} => {playoffs=0} 0.2152466  0.8
571429 0.2511211 1.371839   144
## [4]  {opp.rush.td=[16,31]}                 => {playoffs=0} 0.2197309  0.8
448276 0.2600897 1.352128   147
## [5]  {pass.td=[6,18)}                      => {playoffs=0} 0.1928251  0.8
431373 0.2286996 1.349423   129
## [6]  {int=[19,32]}                         => {playoffs=0} 0.2257100  0.8
342541 0.2705531 1.335206   151
## [7]  {opp.rush.1st.downs=[108,161]}        => {playoffs=0} 0.2077728  0.8
176471 0.2541106 1.308627   139
## [8]  {opp.pass.td=[27,45]}                 => {playoffs=0} 0.2017937  0.7
```

```
894737 0.2556054 1.263536   135
## [9]  {opp.int=[2,12)}                           => {playoffs=0} 0.1778774  0.7
777778 0.2286996 1.244817   119
## [10] {avg.plays.per.drive=[4.56,5.37)}          => {playoffs=0} 0.1868460  0.7
763975 0.2406577 1.242608   125
```

Teams that only totaled 2-9 rushing touchdowns per season are more likely to miss the playoffs. (Support = 0.17, Confidence = 0.89)

Additionally, teams that let opposing teams total both 16-31 rushing touchdowns and 108-161 rushing first downs are also likely to miss the playoffs. (Support = 0.14, Confidence = 0.88)

It seems to be that the team's rushing defense is the most important variable in missing the playoffs (letting the other team dictate game pace) whereas passing seems to be a slightly less-influential variable in missing the playoffs.

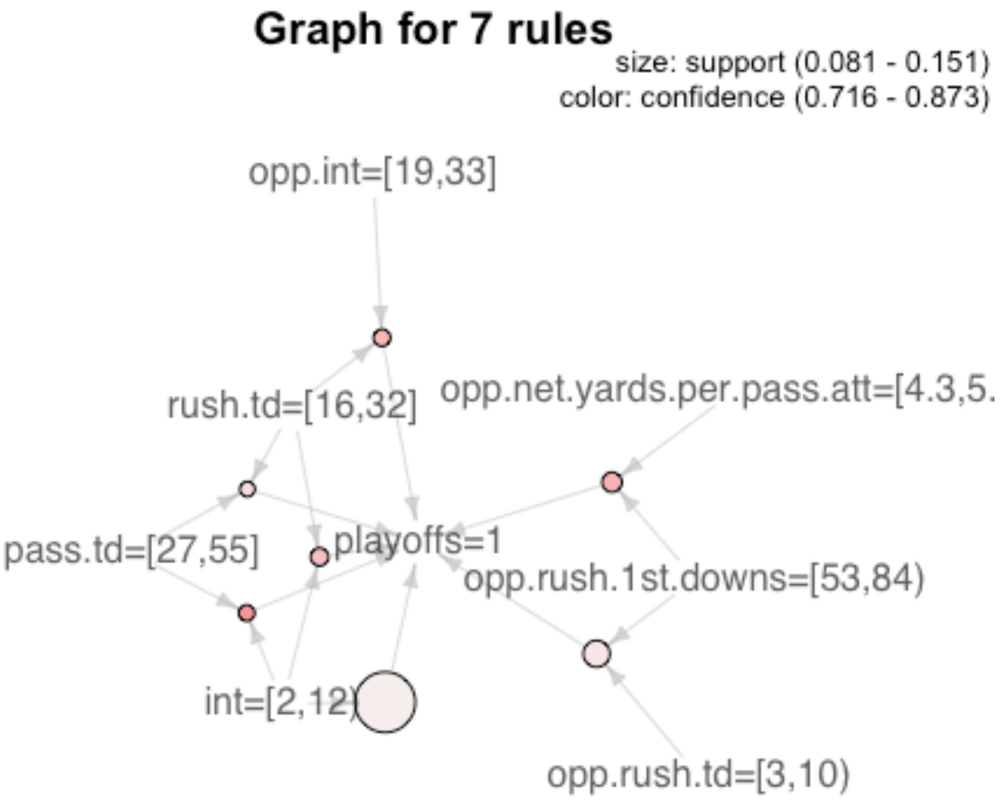
Playoff Prediction Rules

Next, the target shifted to obtain strong playoff eligibility rules, so a combination of Support = 0.08, Confidence = 0.7, and Lift >= 1 was used to generate those rules. The 10 rules with the strongest confidence of this selection are listed below.

```
# Playoff Prediction Rules
nfl_rules <- apriori(data=discreteNFL, parameter=list(supp=0.08,conf = 0.7),
appearance = list (default="lhs",rhs="playoffs=1"), control = list (verbose=F
))
nflrules_sorted <- sort(nfl_rules, by="lift")
inspect(head(sort(nflrules_sorted, by = "confidence"), 10))
```

```
##     lhs                                       rhs            support conf
idence  coverage     lift count
## [1] {pass.td=[27,55],
##      int=[2,12)}                           => {playoffs=1} 0.08221226  0.8
730159 0.0941704 2.326883    55
## [2] {opp.net.yards.per.pass.att=[4.3,5.7),
##      opp.rush.1st.downs=[53,84)}           => {playoffs=1} 0.08819133  0.8
309859 0.1061286 2.214859    59
## [3] {rush.td=[16,32],
##      opp.int=[19,33]}                      => {playoffs=1} 0.08370703  0.8
235294 0.1016442 2.194985    56
## [4] {int=[2,12),
##      rush.td=[16,32]}                      => {playoffs=1} 0.08520179  0.8
142857 0.1046338 2.170347    57
## [5] {pass.td=[27,55],
##      rush.td=[16,32]}                      => {playoffs=1} 0.08071749  0.7
500000 0.1076233 1.999004    54
## [6] {opp.rush.td=[3,10),
##      opp.rush.1st.downs=[53,84)}           => {playoffs=1} 0.09865471  0.7
333333 0.1345291 1.954582    66
```

```
## [7] {int=[2,12)}                                 => {playoffs=1} 0.15097160  0.7
163121 0.2107623 1.909214   101

# ARM Visualization
plot(nflrules_sorted, method="graph",shading="confidence")
```


Graph for 7 rules
size: support (0.081 - 0.151)
color: confidence (0.716 - 0.873)
opp.int=[19,33]
rush.td=[16,32]
opp.net.yards.per.pass.att=[4.3,5.
pass.td=[27,55]
playoffs=1
opp.rush.1st.downs=[53,84)
int=[2,12)
opp.rush.td=[3,10)

Teams that achieve between 27-55 passing touchdowns and only 2-12 interceptions throughout the season are most likely to make the playoffs. (Support = 0.08, Confidence = 0.87)

Additionally, teams that only allow opposing teams to gain between 4.3-5.7 net yards per passing attempt and 53-84 rushing first downs throughout the season are likely to make the playoffs. (Support = 0.09, Confidence = 0.83)

Conclusions

When studying American football, it is imperative to remember that it is a team game and, at the end of the season, the goal is to hold up the Lombardi trophy. In pursuit of that ultimate goal, a franchise must know the biggest pitfalls and how to avoid them given their personnel types. For many franchises, there were many years where they were on the cusp of a wildcard appearance but they could not get the crucial wins they needed to break through. There are a number of possible explanations for this; injuries to star players, poor play calling, poor player performance, and lack of talent in general. When consulting with these teams, the goal is to optimize their performance on the player side by comparing their statistics to our findings of what make a team successful.

Based on our analysis, the biggest key to success lies in the passing game. Of the different statistics that were considered, passing touchdowns, yards per pass attempt, and interceptions had the largest influence on their playoff appearances. This has become even more apparent as the past five Super Bowl winners have had quarterbacks with exceptionally high touchdown-to-interception ratios and at least 8 yards per pass attempt. For many of the teams on the edge, the missing key could be either a franchise quarterback, better protection, or a better wide receiver. It could also mean prioritizing better defensive backs to give their current offense a better chance to keep up as the next most important statistic is pass yards per attempt by an opponent.

When addressing personnel issues like those previously mentioned, it is important to consider where a particular team falls in terms of draft order, as well as their cap space going into future seasons. In the future, combining our current data with cap management patterns and draft records will allow for more in-depth analysis and more precise recommendations for each team's situation. Ultimately, time is of the essence as money and the prime years of their current stars are at stake with each passing season.