NCAA Division Three Football Predictors for Winning

Ben Garski |Information Studies 691| 05-05-2019

Outline

Introduction

The NFL and NCAA Division One Football have gained considerable attention to data enthusiasts. The MLB and NBA have also been carefully analyzed, so much so that it is the norm to see beautiful up-to-date statistic representations on live sporting events and websites. Indeed, there are lots of bets done on these professional events, including the ever-popular fantasy football leagues. Unfortunately, there is less attraction to sports in lower divisions, especially Division Three. The purpose of this report is to analyze an ignored area of study, in an expanding area of talent, which is NCAA Division Three Football.

The common stats like third down percentage, yards per play, and turnovers, I presume to be big predictors of winning, but I wanted to find out how much so, along with what other under-looked variables would be predictors for a team winning. With time I hope more coaches realize the power of data science and what it can do to for their team in combination with successful coaching.

Background Information

All the data used in this report originated from the NCAA website (link shown later). This data is from the 2017-2018 NCAA Division Three Football Season. I will refer to this as the 2017 season, since all the games occur in 2017 anyway. Further, if one follows D3 Football, they would know that there are a select handful of teams that have been good for years and continue to win. For example, Mount Union, UW-Whitewater, Mary Hardin-Baylor, St. Thomas, and UW-Oshkosh. To emphasize, the above teams have been the only teams who were involved in the National Championship game since 2005. That is 14 years of the same five teams making Nationals! Therefore, there are trends to be found in this division because of the select dominance by such few teams.

Many of the stats I will use are obvious to football enthusiasts, buy maybe slightly less so to someone who does not follow football. With that being said, I will briefly explain some of the statistics which will be used in this report below:

- 3rd Down Percentage = $\frac{3rd\ Down\ Conversions}{3rd\ Down\ Attempts}$
- Penalty Yards Per Game = $\frac{Total\ Penalty\ Yards}{Total\ Games}$
- Average Time of Possession = $\frac{Total\ Time\ of\ Possession}{Total\ Games}$
- Turnover Margin = $Turnovers\ Gained\ (Fumbles + Interceptions) - Turnovers\ Lost\ (Fumbles + Interceptions)$

- Winning Percentage = $\frac{Total\ Wins}{Total\ Games}$
- Net Punting Yards = $\frac{Total\ Punting\ Yards - Opp.\ Punt\ Return\ Yards - Touchback\ Yards}{Total\ Punts}$
- Kickoff Return Average Yards = $\frac{Total\ Kickoff\ Return\ Yards}{Total\ Kickoff\ Returns}$
- Average Attendance = $\frac{Total\ Attendance}{Total\ Games\ Attendace\ Taken}$

Data Collection

All the data from this report were collected from: http://stats.ncaa.org/rankings/change_sport_year_div for the year 2017 (termed 2017-2018 on website). I first chose the exact stats I was interested in from the drop-down after going to the "Team" tab. Then, I selected the maximum number of entries to be shown in the drop-down to see all the teams. I then sorted by team name and copied the text, including the headers, down to the bottom (excluding the teams that were reclassifying). Finally, I pasted that into Notepad ++ and made sure there were no empty rows. The files were all saved into my working directory for RStudio as text files (.txt), which were technically tab-separated values (evident by how I imported them into RStudio with read.csv)

Also, I did go to the "Misc. Reports" tab of the website, instead of the "Team" one, to pull the attendance data. This data I was more unsure of its reliability. It appeared that a lot of games did not have attendance data, which sort of makes sense for D3 Football. Therefore, I had eight total files in my working directory when all was said in done.

Data Wrangling

First, I imported the files and removed unnecessary columns. For the punting data, I split the record into two columns of "Wins" and "Losses" so I could then calculate a win percentage and add that column. The time of possession data was a little more work. To make one column with the average time of possession, I needed to condense the "minute: second" format into something the computer could interpret. So, I split up minutes and seconds and turned minutes into seconds and added it to the other seconds column to end up with one column of average time of possession in seconds. To make sure all the individual files matched up with one another, in other words, the team names were all sorted properly and had the same teams in them, I combined the columns with the team name from each into one data frame.

Further, the attendance data caused me the most work. I did mention before that I was hesitant on its reliability. In short, I found out that there were three teams included in the

attendance data that was not included in my other data. So, I removed these three teams and resorted the file. I then could add a column with the appropriate average attendance for each team. Lastly, I create a data frame called “Data” that housed all the team names and their proper stats that I would be analyzing. The complete R code for data importing/cleaning is shown below.

```
# Source: http://stats.ncaa.org/rankings/change_sport_year_div #
# *Note* Copy and Pasted From Header Down (ommiting the Reclassifying Teams at Bottom) into Notepad ++
# Already Sorted by Team Name From Website

library(tidyr)
library(dplyr)
library(corrplot)
library(NbClust)
library(ROCR)

# Import Punting Data, Seperate Record into Wins/Losses, Calculate Win %
Punt2017 <- read.csv("2017 - Punt.txt", sep = "\t", stringsAsFactors = FALSE)
Punt2017 <- separate(Punt2017, W.L, c("Wins", "Losses"), sep = "-")
Punt2017$Wins <- as.integer(Punt2017$Wins)
Punt2017$Losses <- as.integer(Punt2017$Losses)
Punt2017$Win_Percentage <- Punt2017$Wins/(Punt2017$Wins+Punt2017$Losses)
Punt2017 <- Punt2017[,c(2,10,11)]

# Import Kick Return Data
KR2017 <- read.csv("2017 - KR.txt", sep = "\t", stringsAsFactors = FALSE)
KR2017 <- KR2017[,c(2,8)]

# Import Penalty Data
Pen2017 <- read.csv("2017 - Penalties.txt", sep = "\t", stringsAsFactors = FALSE)
Pen2017 <- Pen2017[,c(2,7)]

# Import 3rd Down Data
Third2017 <- read.csv("2017 - 3D.txt", sep = "\t", stringsAsFactors = FALSE)
Third2017 <- Third2017[,c(2,7)]

# Import Time of Possession Data, Split TOP into Minute/Second, Combine into Seconds
TOP2017 <- read.csv("2017 - TOP.txt", sep = "\t", stringsAsFactors = FALSE)
TOP2017 <- TOP2017[,c(2,6)]
TOP2017 <- separate(TOP2017, AvgTOP, c("Minute", "Second"), sep = ":")
TOP2017$Minute <- as.numeric(TOP2017$Minute)
TOP2017$Second <- as.numeric(TOP2017$Second)
TOP2017$AvgTOP <- floor((TOP2017$Minute * 60) + TOP2017$Second)
TOP2017 <- TOP2017[,c(1,4)]
```

```
# Import Turnover Margin Data
TOM2017 <- read.csv("2017 - TOM.txt", sep = "\t", stringsAsFactors = FALSE)
TOM2017 <- TOM2017[,c(2,11)]

# Import Total Offense Data
TO2017 <- read.csv("2017 - TO.txt", sep = "\t", stringsAsFactors = FALSE)
TO2017 <- TO2017[,c(2,7)]

# Import Game Attendance Data (Came From Reports Tab)
Att2017 <- read.csv("2017 - Attendance.txt", sep = "\t")

# Check for Name/Row Matching (2018 Data Had Issues)
All <- data.frame("Punt2017" = Punt2017$Team,"KR2017" = KR2017$Team, "Pen2017" = Pen2017$Team,
                  "Third2017" = Third2017$Team, "TO2017" = TO2017$Team,"TOM2017" = TOM2017$Team,
                  "TOP2017" = TOP2017$Team) # All Match

# Attendance - Check for Team Name Discrepancies
All <- separate(All, Punt2017, c("Team"), sep = "\\),\\(|\\)|\\(", remove = FALSE)
test <- data.frame("Team" = All$Team, "Attendance_Team" = Att2017$Institution[1:242])
# Belhaven (Row 18),Brevard (Row 27),Finlandia (Row 71),McMurry (Row 134)
Att2017 <- Att2017[-c(18,27,71,134),c(1,4)]
rownames(Att2017) <- seq(length = nrow(Att2017))
Att2017$Avg.Attendance <- as.numeric(gsub(",","",Att2017$Avg.Attendance))

# Make Data Frame with All Data
Data <- data.frame("Team" = KR2017$Team,"Win_Percentage" = Punt2017$Win_Percentage,
                   "Average_Attendance" = Att2017$Avg.Attendance,"KR_Avg_Yards" = KR2017$Avg,
                   "Avg_Penalty_Yards_Per_Game" = Pen2017$YPG, "Net_Punting_Yards" = Punt2017$Net.Yds,
                   "Third_Down_Percentage" = Third2017$Pct,"Average_Yards_Per_Play" = TO2017$Yds.Play,
                   "Turnover_Margin" = TOM2017$Margin, "Average_Time_of_Possesion" = TOP2017$AvgTOP)

Data$Team <- as.character(Data$Team)
```

Data Analysis

The first part of my analyses was to check for multicollinearity between the independent variables before running a regression. I excluded average yards per play in my model when I found out that third down percentage and average yards per play had a high correlation (also was an obvious predictor of success). After removing that variable, I was left with the model below which shows the correlation plot as well.

```
# Data for Tests w/o Team Name
data.simple <- Data[,-1]
data.simpler <- Data[,-c(1,8)]

# Correlation/Regression
r <- cor(data.simple)
corrplot(r)
fit <- lm(Win_Percentage ~ ., data = data.simple)
summary(fit)
varImp(fit)

# Correlation/Regression w/o Yards/Play (Average Yards/Play & 3rd Down % Had High Correlation)
r2 <- cor(data.simpler)
corrplot(r2)
fit2 <- lm(Win_Percentage ~ ., data = data.simpler)
summary(fit2)
varImp(fit2)
```

Figure 1: R Code for Multiple Regression Model

```
> summary(fit2)

Call:
lm(formula = Win_Percentage ~ ., data = data.simpler)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4780 -0.1145  0.0076  0.1141  0.3640

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)               -6.199e-01  2.063e-01  -3.005  0.00294 **
Average_Attendance         3.045e-05  1.056e-05   2.883  0.00430 **
KR_Avg_Yards               8.825e-03  4.089e-03   2.158  0.03193 *
Avg_Penalty_Yards_Per_Game 1.560e-03  7.484e-04   2.084  0.03822 *
Net_Punting_Yards          3.534e-03  3.718e-03   0.950  0.34293
Third_Down_Percentage      1.535e+00  1.773e-01   8.657 7.98e-16 ***
Turnover_Margin            1.493e-02  1.489e-03  10.030  < 2e-16 ***
Average_Time_of_Possesion  5.840e-05  8.374e-05   0.697  0.48622
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.156 on 234 degrees of freedom
Multiple R-squared:  0.6527,    Adjusted R-squared:  0.6423
F-statistic: 62.82 on 7 and 234 DF,  p-value: < 2.2e-16
```

Figure 2: Multiple Regression Model

```
> varImp(fit2)
                             Overall
Average_Attendance         2.8831723
KR_Avg_Yards               2.1582107
Avg_Penalty_Yards_Per_Game 2.0842494
Net_Punting_Yards          0.9503301
Third_Down_Percentage      8.6567651
Turnover_Margin           10.0303794
Average_Time_of_Possesion  0.6974422
```

Figure 3: Multiple Regression Model's Variable Importance

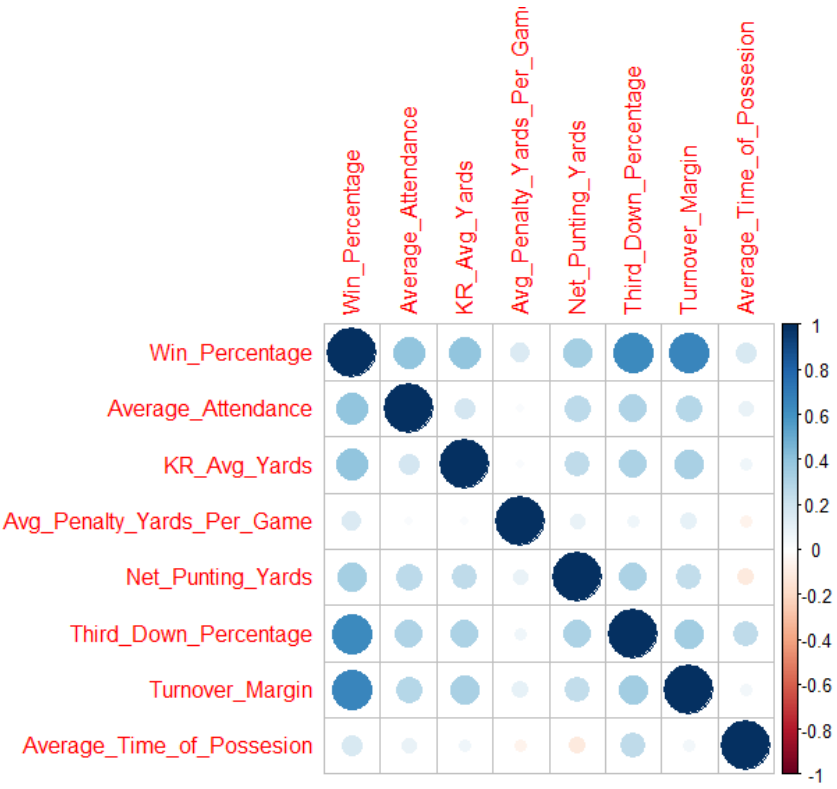

Figure 4: Correlation Plot of Variables in Multiple Regression Model

The multiple regression model shows that turnover margin has the biggest importance on winning percentage, followed by third down percentage, average attendance, kickoff return average yards, and average penalty yards per game. I would have guessed that the top two were indeed those, but I am more surprised that average kickoff return yards and average penalty yards per game were both statistically significant with 95% confidence. Attendance is also an obvious variable for success, usually better teams have more of a following compared to worse teams. Our model is not that strong though, the R-squared value is only 0.6423. This leads me to think that there are many other variables that may affect winning percentage.

Next, I perform logistic regression to see if I can predict if a team will have a winning record or not. I first create a binary predictor column from the already existing win percentage column as shown below. I run the logistic model and then use a function from https://hopstat.wordpress.com/2014/12/19/a-small-introduction-to-the-rocr-package/ that finds the optimal cutting point between the sensitivity and specificity of the model. My cut point is 0.4463935 as shown below.

```r
## Prediction via Logistic Regression ##

# Create New Binary Predictor Column
Data.Clusters$Winning_Team <- Data$Win_Percentage > 0.5
Data.Clusters$Winning_Team[Data.Clusters$Winning_Team == "False"] <- 0
Data.Clusters$Winning_Team[Data.Clusters$Winning_Team == "True"] <- 1
Data$Winning_Team <- Data.Clusters$Winning_Team

# Create New DF
Data.Log <- Data.Clusters[,-c(1,7,10)]

# Run Logistic Regression
data.glm <- glm(Winning_Team ~ ., family = "binomial", data = Data.Log)

# Get Predictions
predict <- predict(data.glm, Data.Log, type = "response")

# ROCR Plot
rocr <- prediction(predict, Data.Log$Winning_Team)
rocrp <- performance(rocr, 'tpr','fpr')
rocr.cost <- performance(rocr, "cost")
plot(rocrp)

# Function for Optimal Cut Point
# Source: https://hopstat.wordpress.com/2014/12/19/a-small-introduction-to-the-rocr-package/
opt.cut = function(perf, pred){
  cut.ind = mapply(FUN=function(x, y, p){
    d = (x - 0)^2 + (y-1)^2
    ind = which(d == min(d))
    c(sensitivity = y[[ind]], specificity = 1-x[[ind]],
      cutoff = p[[ind]])
  }, perf@x.values, perf@y.values, pred@cutoffs)
}
print(opt.cut(rocrp, rocr))

# Put Predictions in DF Based on Optimal Cut Point
Data.Clusters$Predictions <- predict
Data.Clusters$Predictions[Data.Clusters$Predictions > 0.4463935] <- 1
Data.Clusters$Predictions[Data.Clusters$Predictions <= 0.4463935] <- 0
```

Figure 5: R Code for Logistic Regression Model

```
> print(opt.cut(rocrp, rocr))
                    [,1]
sensitivity 0.8407080
specificity 0.7984496
cutoff      0.4463935
```

Figure 6: Optimal Cut Point

```
> # Confusion Matrix
> table(Data.Log$Winning_Team, Data.Clusters$Predictions) # 81% Accuracy

    0   1
0 103  26
1  19  94
```

Figure 7: Confusion Matrix from Logistic Regression Model with Cut Point

```
> summary(data.glm)

Call:
glm(formula = Winning_Team ~ ., family = "binomial", data = Data.Log)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.44741  -0.60257  -0.08577   0.55631   2.80626

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -0.28529    0.18333  -1.556   0.1197
Average_Attendance          0.43807    0.20441   2.143   0.0321 *
KR_Avg_Yards                0.05866    0.20468   0.287   0.7744
Avg_Penalty_Yards_Per_Game  0.10591    0.18039   0.587   0.5571
Net_Punting_Yards           0.07003    0.20262   0.346   0.7296
Third_Down_Percentage       1.49029    0.27016   5.516 3.46e-08 ***
Turnover_Margin             1.39337    0.26270   5.304 1.13e-07 ***
Average_Time_of_Possesion   0.23149    0.19538   1.185   0.2361
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 334.42  on 241  degrees of freedom
Residual deviance: 189.91  on 234  degrees of freedom
AIC: 205.91

Number of Fisher Scoring iterations: 6
```

Figure 8: Logistic Regression Model

After cutting the predictions at the cut point from above, my model received an 81% accuracy rate. The results were similar to the multiple regression model with similar statistically significant variables shown in Figure 8.

Next, I want to complete a K-Means Clustering on the data to group all the teams by their variables to see if we can put the teams into a good, average, bad type of category based on all their variables. I first scale the data and determine the optimal k. Ironically, it ends up being three, which works out perfectly for the good, average, and bad categories. The best plot I found that demonstrates their group membership is shown in Figure 6. I used win percentage on the x-axis and turnover margin on the y-axis. It appears that cluster two (green) are the good teams, cluster three (blue) are the bad teams, and cluster one (red) are the average teams. I checked the team and their clusters for accuracy, it was surprisingly accurate with the cluster it assigned each team to.

```r
## Cluster via K-Means ##

# Scale and Make New DF
Data.Scaled <- data.frame(scale(Data[,c(3:10)]))
Data.Scaled$Team <- Data$Team
Data.Scaled <- Data.Scaled[,c(9,1:8)]

# Find Optimal K
fviz_nbclust(Data.Scaled[,-1],kmeans,method = "wss")
nc <- NbClust(Data.Scaled[,-1],min.nc = 2,max.nc = 10,method = "kmeans") # 3 is Best Amount of Clusters

# Run K-Means
set.seed(123)
k3 <- kmeans(Data.Scaled[,-1], centers = 3)
cluster <- k3$cluster

# Create New DF with Cluster Assignment
Data.Clusters <- mutate(Data.Scaled, Cluster = cluster)

# Plot Clusters
ggplot(Data.Clusters, aes(x = Data$Win_Percentage, y = Data$Turnover_Margin, color = factor(Cluster))) +
  geom_point() + ggtitle("2017 D3 Football K-Means Clustering") + xlab("Win Percentage") +
  ylab("Turnover Margin") + labs(color = "Cluster")
```

Figure 9: R Code for K-Means Clustering

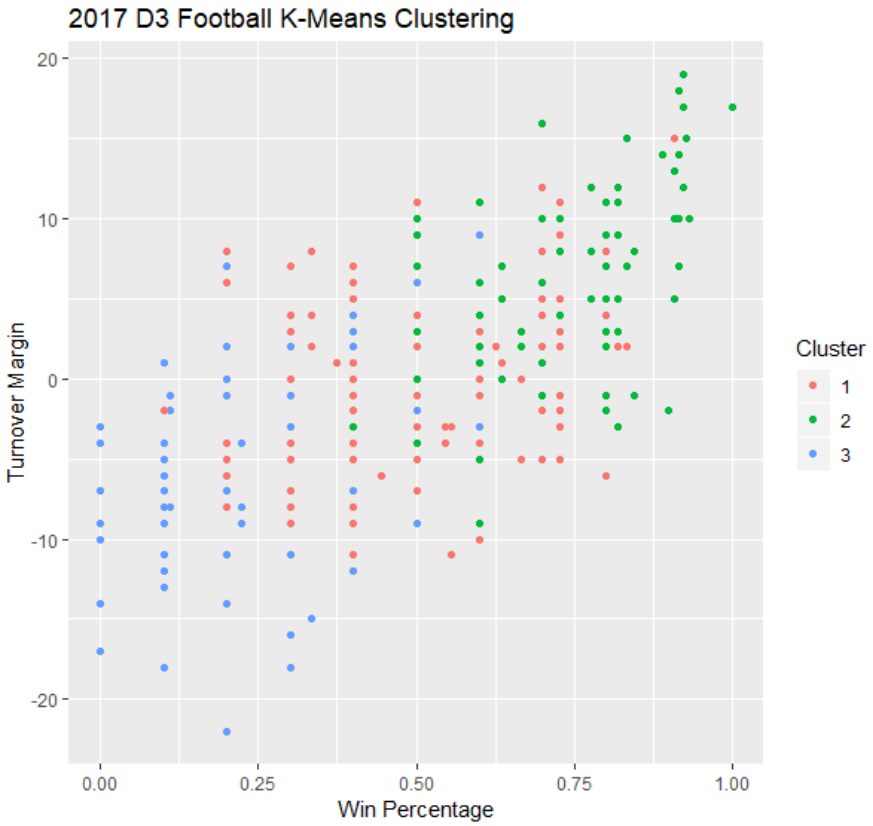

Figure 10: K-Means Clustering Plot

Lastly, I wanted to see how plotting all the variables in two-dimensions using principle component analysis (Classical MDS with Euclidean Distance) would look since I struggled to find the appropriate two axes to best plot the clusters. It appears that the better teams are on the left and the worse teams are on the right. It looks like the K-Means plot, but with the directions reversed and no correlation representation of course. I added the K-Means group assignment coloring to the plot as well, so we can easily compare the two.

```
# Principle Component Analysis/MDS
pca <- data.frame(cmdscale(dist(Data.Clusters[,2:9])))
pca$cluster <- Data.Clusters$Cluster

# Plot using PCA Dimensions Instead
ggplot(pca, aes(x = X1, y = X2, color = factor(cluster))) + geom_point() +
  ggtitle("2017 D3 Football PCA")+ xlab("Dimension 1") + ylab("Dimension 2") + labs(color = "Cluster")
```

Figure 11: R Code for PCA/MDS

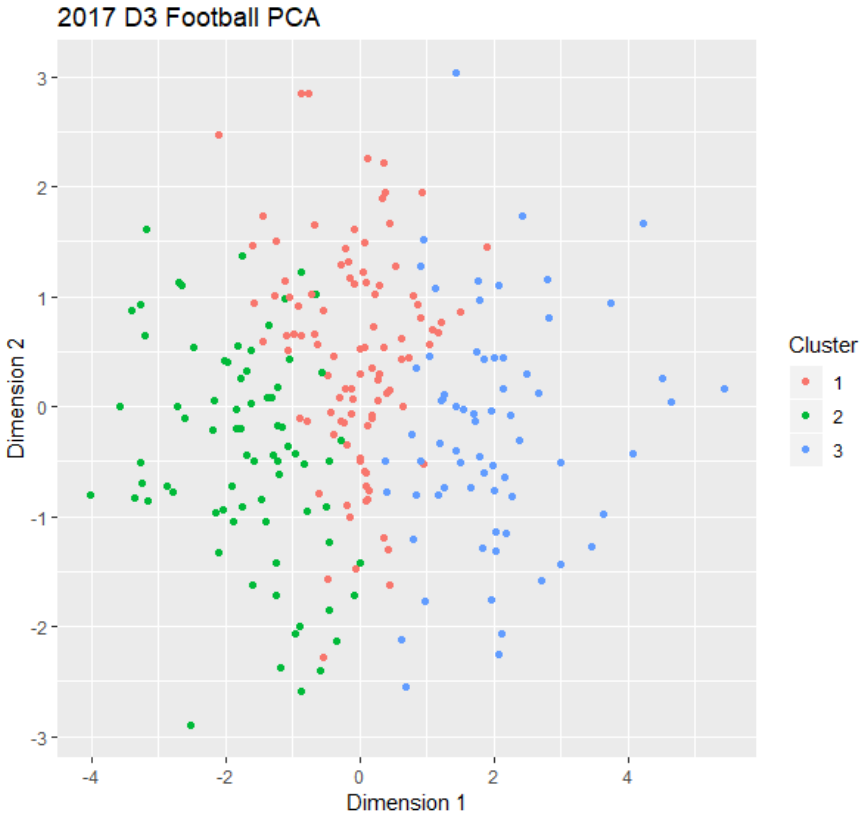

Figure 12: PCA/MDS Plot Using K-Means Group Coloring

Conclusion

In summary, it appears we can pick out some important variables that affect the winning percentage of teams. In fact, we found that third down percentage and turnovers are important to winning (not shockingly). To my surprise, we found that kickoff return yards and penalty yards may also be important. I was able to use the popular Euclidean Distance along with K-Means to put all our teams into hypothetical groupings with decent accuracy. I then attempted a multidimensional scaling on that same data to plot all the variables into two-dimensions. Further, I found we can predict with around 81% accuracy if a team will have a winning record or not from those same variables using logistic regression. Finally, there would be a benefit to doing these tests again on another year to see if the results are repeatable/comparable. Therefore, I hope my intentions of drawing interest to an overlooked area of sports will cause others to analyze Division Three and Division Two athletics for the love of exploring something different.