What Makes a Winning NFL Team?

Julia Lee

2022-05-28

1 Introduction

The rise in legalized sports betting and fantasy football have made people interested in figuring out how to predict a team's chance of winning.[1] Although there is information available about different facets of every team in the form of statistics, the question is whether such information can provide insight into the outcome of games. The question is whether there are factors that can be used to make a judgment about whether or not a team will win. What information about a team's track-record can help a gambler?

It is not only fans that want to know these things, but the teams themselves are interested in how data can help them win[2]. With the rise of sports analytics, teams in different sports want to know whether there might be previously not examined information that might provide important clues to a winning formula.

By examining certain data, we aim to answer (1) whether we can predict whether or not an NFL team will win a game using classification methods and (2) what factors most strongly impact a team's chances of winning.

We will compare and contrast various methods to explore which method or methods are best for analyzing the data. The methods that will be used to explore this problem are decision trees, Random Forests, Linear discriminant analysis, quadratic discriminant analysis, and neural networks.

Lastly, we will also study how the teams are similar to each other by clustering on the data. We use hierarchical clustering, K-means, and self organizing maps.

2 Data:

The Data is game data for all 32 NFL teams from 2016 - 2021 (3454 observations). The data being used for training the classification methods and clustering in the game data from 2019 - 2021 (1924 observations) and the test data is game data for teams from 2016 - 2018 (1530 observations). The data was collected by merging aggregate play by play data by teams[3] and publicly available game data[4]

The data we will use is basic offensive game data, such as the number and kinds of plays by a team in a game, and the number of certain successful outcomes, such as yards, touchdowns and first downs. Some of the data pertains to failures, such as the number of fumbles and interceptions.

Using this data, we will examine what variables or combination of variables made it more likely that a team would be successful in winning a game. We want to determine whether there is a clear connection between certain data and a team's likelihood of winning a game, and identify which factors are the most prevalent for a winning team.

We will also study how the teams are similar to each other by clustering on the data. To do this we grouped the training data by team and are using the averages of the numeric features (a data set of 32 x 13 variables)

[1] https://www.espn.com/chalk/story/_/id/19309213/chalk-why-fantasy-sports-sports-betting-collision-course-us
[2] https://www.cxotalk.com/episode/data-analytics-nfl
[3] http://nflsavant.com/about.php
[4] http://www.habitatring.com/standings.csv

and expect total number of wins.

2.1 Variables:

DidWin: Did the team win? [1: Win. 0: lost or tied] Team: one of the 32 NFL teams

Rushes: Number of rushes the team made in the game.

Passes: Number of passes the team made in the game.

FirstDown: Number of first-Downs the team made in the game.

Sacks: Number of sacks made against the team.

Interception: The number of interceptions the team threw.

Fumbles: Number of fumbles that the team made.

Incomplete: Number of incomplete passes for the team in the game.

TDs: Number of touchdowns the team scored in the game.

Twopoint: Number of successful two point conversions.

Yards: Number of total yards the team gained in the game.

N: Number of total offensive plays.

IsHome: Is the team the home Team? (0 or 1)

Score: The number of points the individual team scored.

Favorby: The number of points a team was favored by before the game. A positive number means the team was favored by that many points, a negative number means their opponent was favored by that many points.

Roof: What was the status of the stadium's roof? values: closed, open ,dome, outdoors.

Surface : Grass vs artifical turf.

RatioPasstoRush: the number of passes the team made to the number of rushes the team made in the game.

3 Classification Methods

3.1 Decision Tree:

There are many advantages of using trees for our problem. Trees are easy to understand and intuitive. Some argue that the decision trees more closely mirror human decision-making than than other classification approaches. Trees can be displayed graphically, and are easily interpreted (even better when small). Using the cross-validation error rate we can find the optimal number of nodes for the decision Tree.

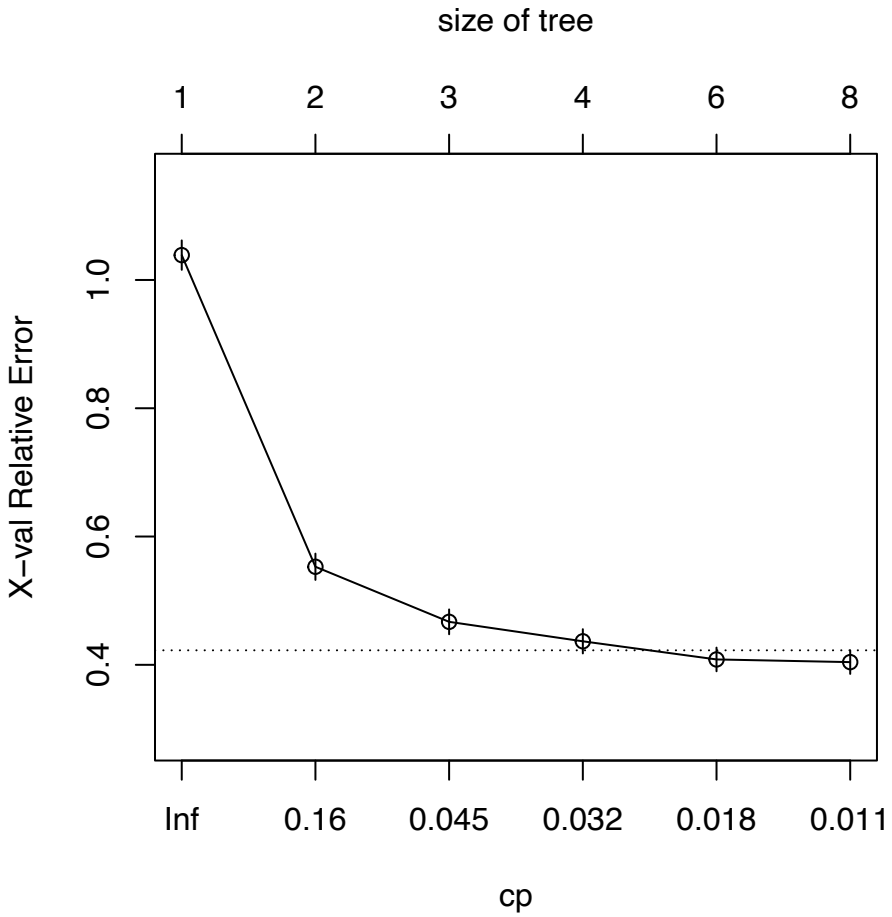


Figure 1: CV plot to find number of nodes

We found that the the tree with the lowest cross-validation error rate is the tree with 8 terminal nodes.

Table 1: confusion Table of Classification Tree

	0	1
0	639	163
1	132	596

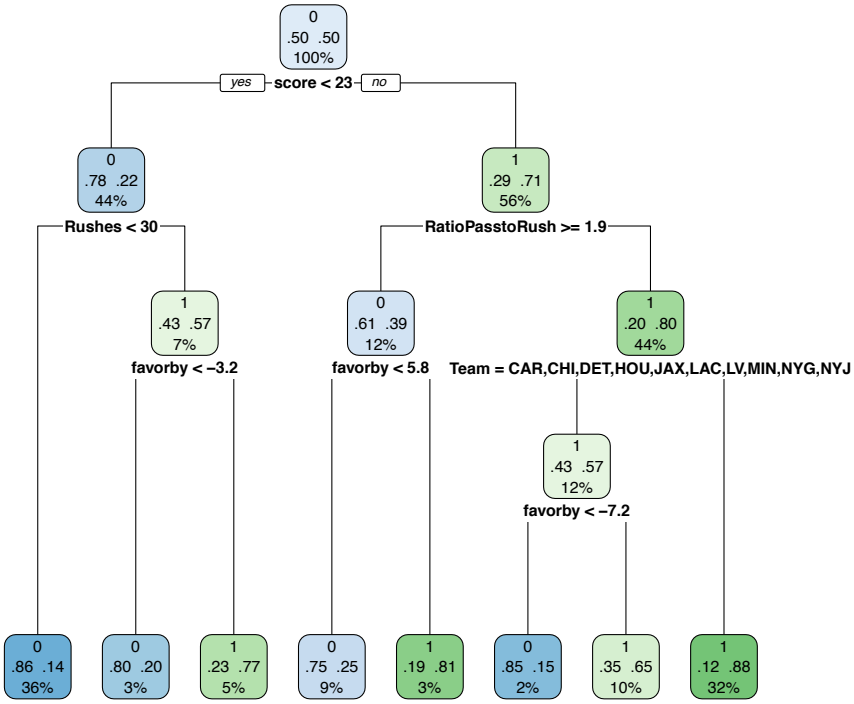


Figure 2: Decision Tree

We can see that most important indicator of current mental health status appears to be whether the team scores more or less than 23 points.

We can also see that how many points that the where favored by going into the game is also an important predicting whether the the team wins the game. This indicates that perhaps the spread line used by experts in accurate and helpful into predicting the outcome of the game. We can also see that the team has an impact on whether the outcome of the game would be a win or not. This perhaps indicates that the teams are different and not evenly matched.

We should also keep in mind the the influence of Team would probably diminish over using a training data set that spanned over decades where teams overall records are similar.

We can also see that the number of Rush plays and the ratio of plays that were passes to rushes were also contributing factors to a classifer.

The decision tree gives us a misclassification rate of 19.28105% on our test data (game data for each team from 2016-2018).

3.2 Random Forest:

We will use random forest to see if we can improve upon our single decision tree.

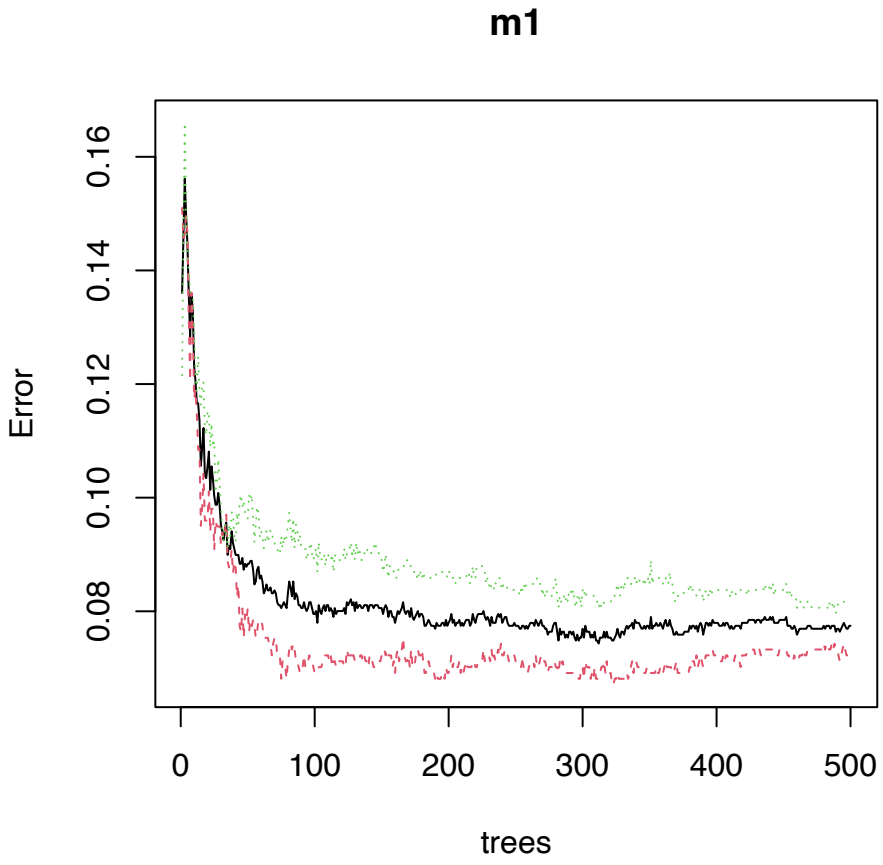


Figure 3: OOB error of random forest

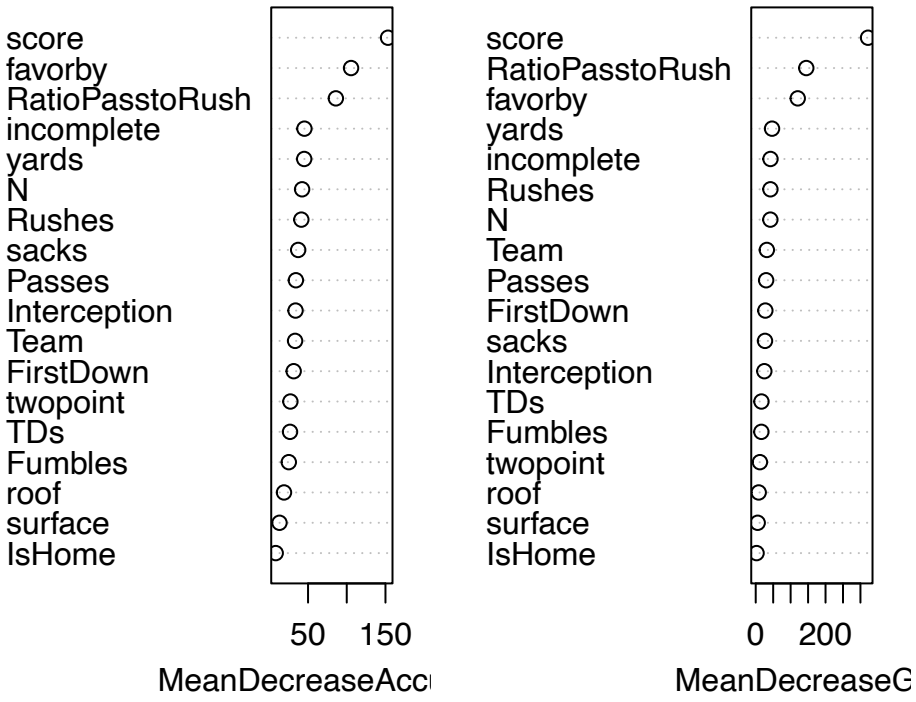


Figure 4: Dotchart of variable importance as measured by a Random Forest

We can also use random forest to measure variable importance using node purity.

We can see that score, ratio of pass to rush, points favored by, yards achieved and number of rushes are the five most important variables in classifying whether or not the team won. These variables account for the largest decrease in node impurity.

Random forest gives us a misclassification rate of 16.99346% on our test data. This lower than our previous pruned single decision tree.

3.3 Discriminant analysis

After some data exploration it looks like that most of features to classify wins appear to be distributed roughly normal. This is the assumption we need to perform Linear Discriminant Analysis and Quadratic Analysis.

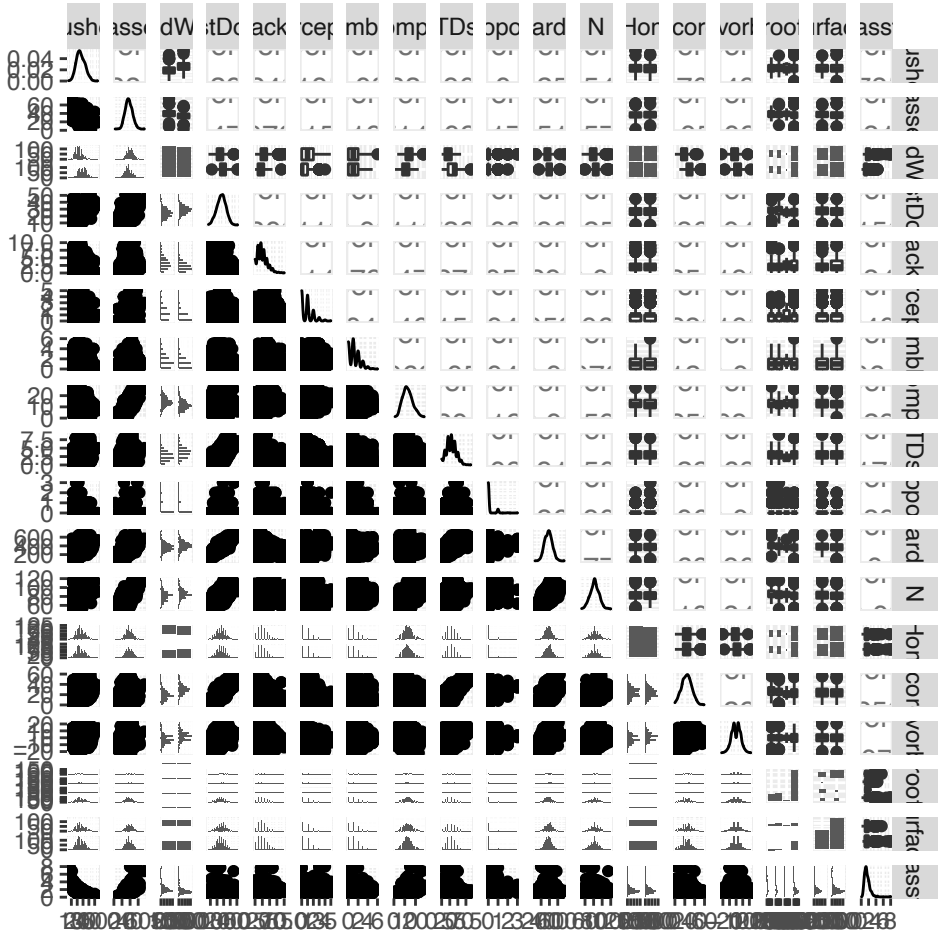


Figure 5: data exploration

3.3.1 LDA:

The LDA output indicated prior probabilities of $\pi_0 = 0.5046948, \pi_1 = 0.4953052$; in other words, 50.46948 of the training observations correspond to instances where a team did not win.

The LDA predictions are accurate almost 85.1633% or wrongly classifies the data at about a rate of 14.836601307% of the time.

3.3.2 QDA:

The QDA predictions are accurate almost 63.85620915% or misclassifies about 36.143790849% of the time.

We can see that it appears that LDA outperforms QDA. So we can conclude that QDA is too flexible for the data we have.

Table 2: coefficients of linear discriminants output

	x
Rushes	-0.0552910
Passes	-0.1239715
FirstDown	-0.0347343
sacks	-0.2282908
Interception	-0.1313423
Fumbles	-0.0711646
incomplete	-0.0989723
TDs	-0.1210499
twopoint	-0.2949621
yards	-0.0008927
N	0.1303852
IsHome1	-0.0482523
score	0.0644218
favorby	0.0595460
roofdome	-0.0191112
roofopen	0.1943729
roofoutdoors	0.0650480
surfacegrass	-0.0076837
RatioPasstoRush	0.1012001

Table 3: LDA prediction Table

class	DidWin	n
0	0	662
0	1	118
1	0	109
1	1	641

Table 4: QDA prediction Table

class	DidWin	n
0	0	508
0	1	290
1	0	263
1	1	469

3.4 Neural Network

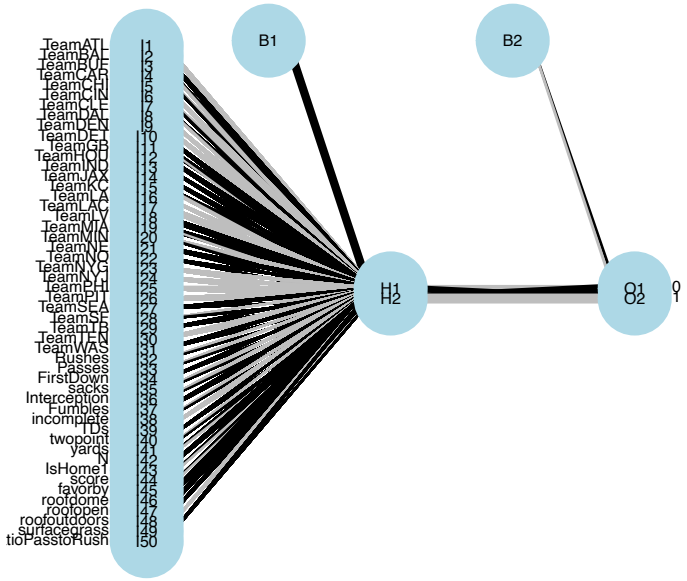


Figure 6: Neural Network Plot, 50-2-2

We will try to us a neural network with a 50-2-2 topology with 100 starting values.

Using a neural network with topology 50-2-2 and using 100 starting values, we misclassify at about a rate of 16.47%.

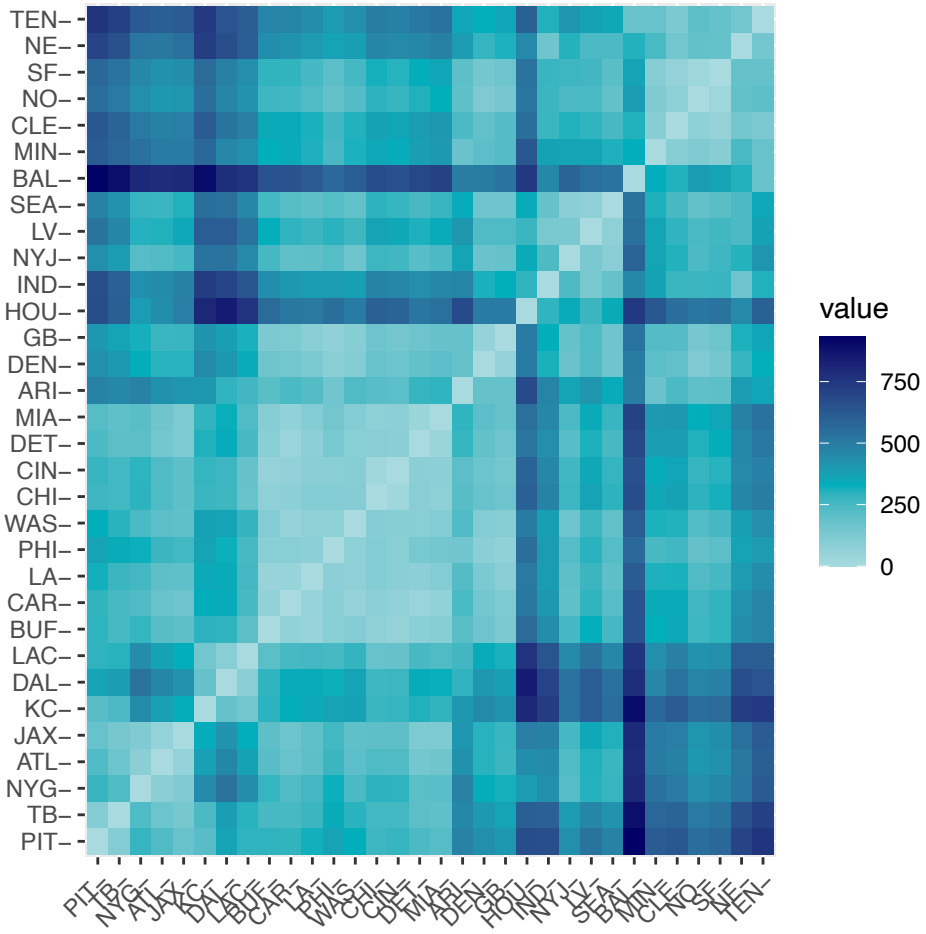
3.5 Results/Conclusion for Classification Methods

The method that performed the best on our data was linear discriminant analysis with a misclassification rate of 14.836% compared to the other methods.

we were able to conclude that there is a possible correlation between the score, ratio of pass to rush, points favored by, yards achieved and number of rushes are the five most important variables in classifying whether or not the team won.

We also know that some of our features are correlated to each other. For example, the number of offensive plays is correlated with the number of first downs which is also correlated to the number of yards. We also know that score is correlated with number of Touchdowns. To better account for this in the future, we should find a interaction terms that describe these relationships. This would improve the results for our classification methods.

4 Clustering



When we look at the distance plot we can see that some teams are very similar to each other and others are not. We can already see some clusters.

4.1 Hierarchical Clustering:

We will first use hierarchical Clustering complete linkage. We then use silhouette diagrams to figure out the optimal number of clusters and we find that the optimal number of clusters is 7.

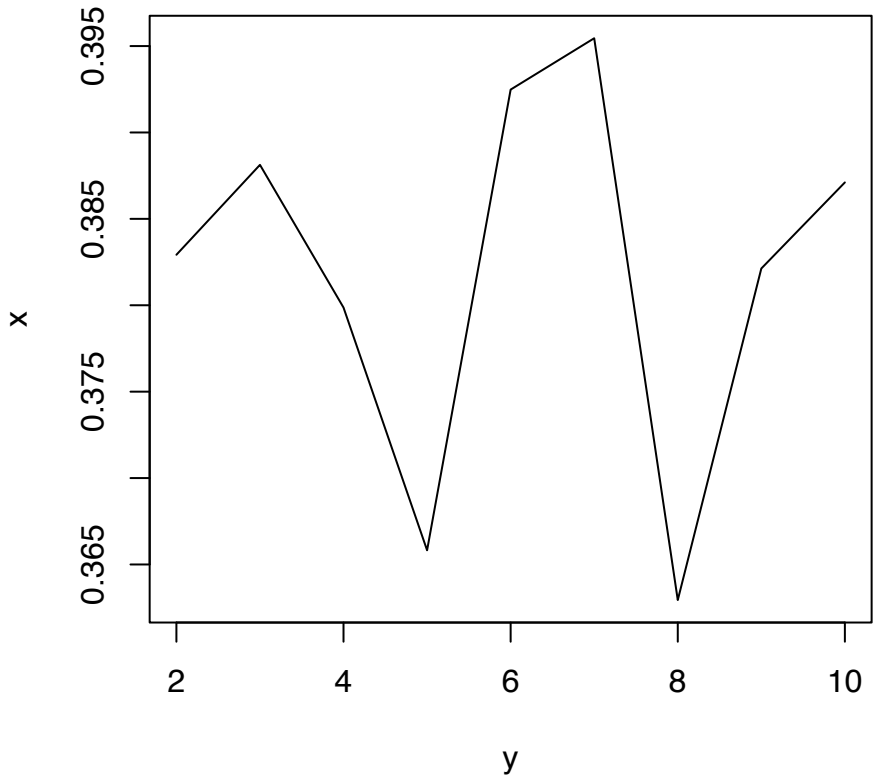


Figure 7: Finding the opitmal number of clusters for dendrogram using avg width

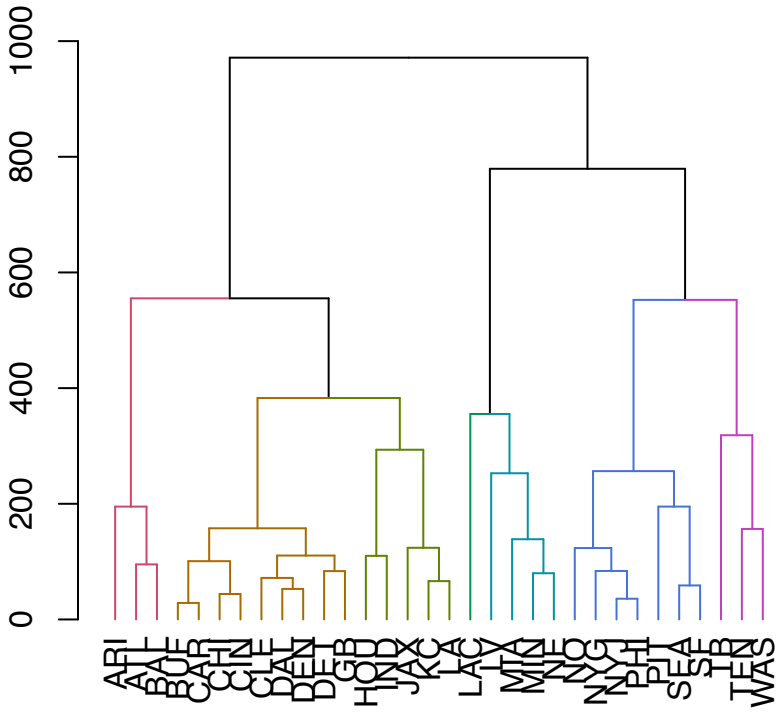


Figure 8: NFL Team Dendrogram, K = 7

We will also look at a scaled heat map of the variables. Looking at the scaled heat-map we can see that some teams are quite similar on some metrics than others. This gives us a more in-depth look at how these teams compare.

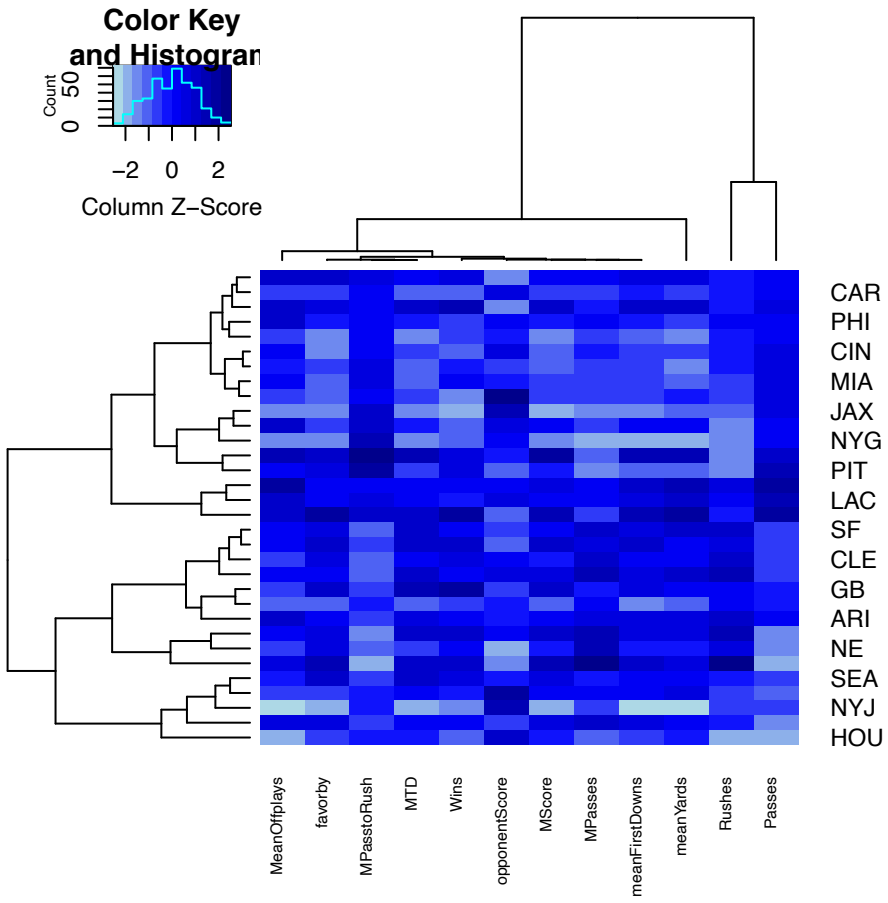


Figure 9: NFL Team Heat Map

4.2 K-means:

We then use k-means clustering and we find that the optimal number of clusters is 6.

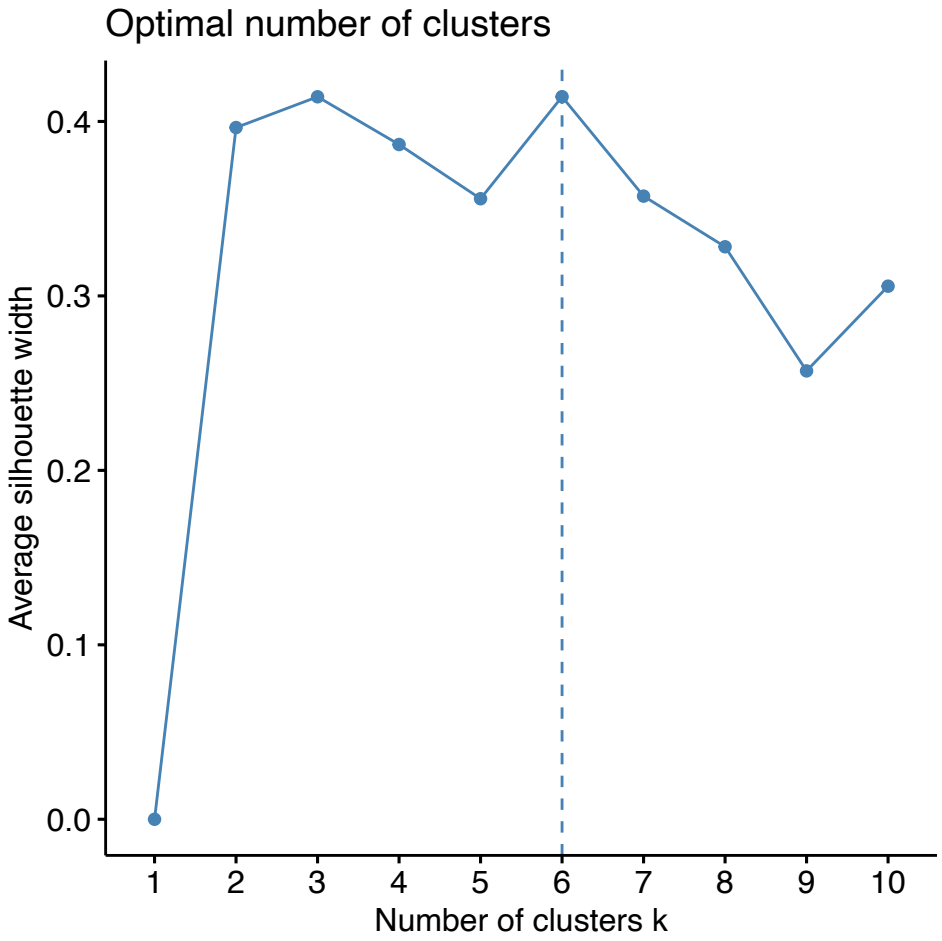


Figure 10: using Shilouette width to find opitmal number of clusters for k means

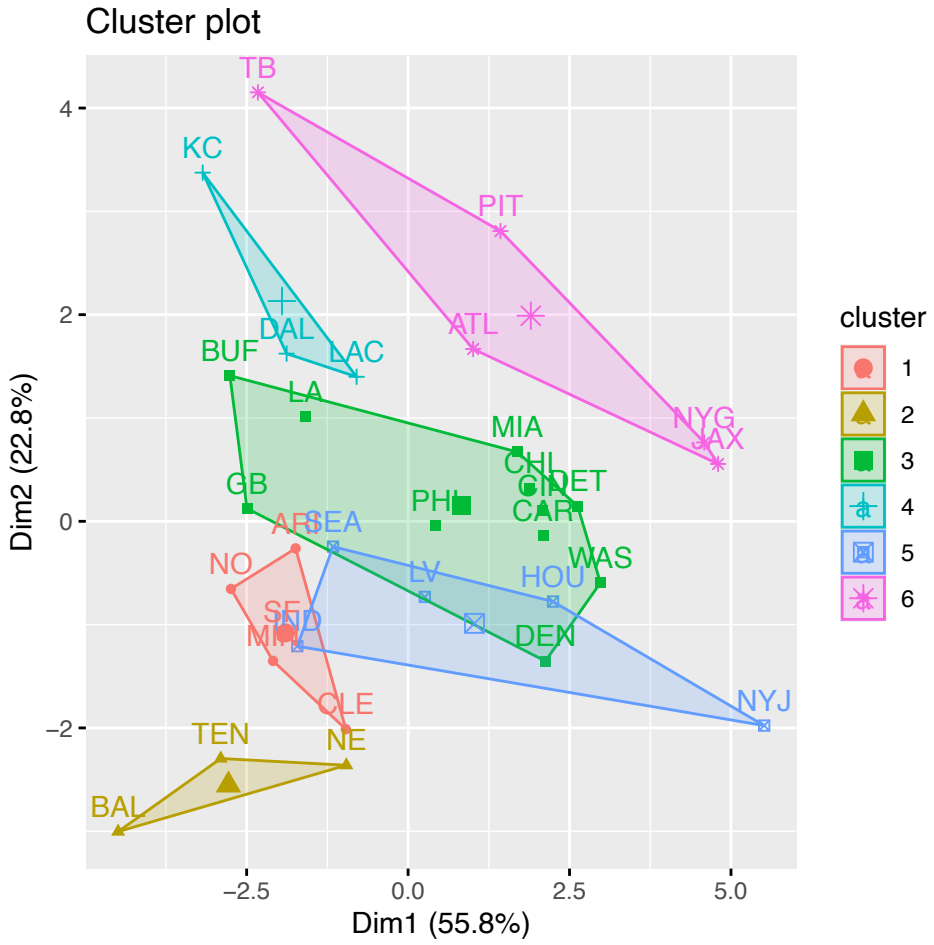


Figure 11: K means, k = 6

We see that some of the clusters overlap and not completely separate from each other.

4.3 Self-organizing Maps:

Lastly, we built a self organizing map with a hexagonal topology using 6 clusters. We can see that some clusters are different in size and that teams we would not to expect to be similar are.

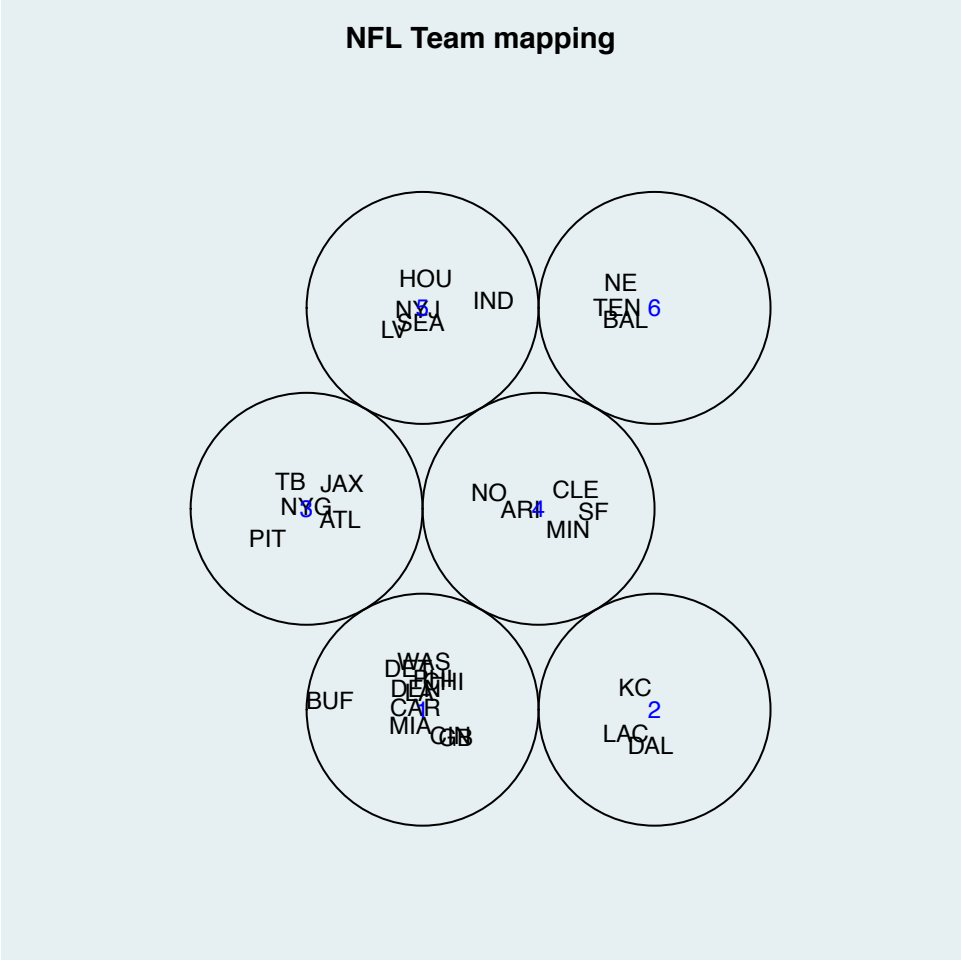


Figure 12: Self-organizing Map

4.4 Results/Conclusion for Clustering Methods

After examining these clusters; it would be interesting to go more in depth and see what exactly makes teams similar to each other. Perhaps the teams have similar styles of play or similar coaching lineages, similar players.

5 Final Thoughts

In conclusion, we can see that there are many factors that make a winning NFL team and that some teams are similar to each other based on these factors.

Attribution:

All Code and data used for this paper will be available at this github repository: https://github.com/JuliaClaireLee/NFLDataProject208

Code appendix

Julia Lee

```
library(readr)
library(float)
library(devtools)
source_url
('https://gist.githubusercontent.com/fawda123/7471137/raw/466c14
  74d0a505ff044412703516c34f1a4684a5/nnet_plot_update.r')

library(kableExtra)
library(clusterGeneration)
library(tictoc)
library(dplyr)
library(MASS)
library(dplyr)
library(DT)
library(readr)
library(nloptr)
library(e1071)
library(ISLR)
library(GGally)
library(caret)
library(nnet)
library(rpart)
library(MASS)
library(dplyr)
library(ISLR)
library(cluster)
library(flashClust)
library(factoextra)
library(ape)
library(ggdendro)
library(dendextend)
library(ggplot2)
#install.packages("klaR")
library(klaR)
library(gplots)
library(kohonen)
library(circlize)
library(rpart.plot)
library(rattle)
library(tree)
library(class)
library(randomForest)
library(readr)
library(readxl)
```

```
library(ggplot2)
library(tm)
library(e1071)
library(gridExtra)
library(class)
library(ISLR)
library(dplyr)
library(nnet)
```

```
train1 <- read_csv("train.csv")
test1 <- read_csv("test.csv")
```

Data:

```
head(test1)
```

```
## # A tibble: 6 x 19
##   Team  Rushes Passes DidWin FirstDown sacks Interception Fumbles incomplete
##   <chr>  <dbl>  <dbl>  <dbl>     <dbl> <dbl>        <dbl>   <dbl>      <dbl>
## 1 ATL       20     44      0        17     6            1       1         22
## 2 PHI       28     40      1        19     2            1       2         17
## 3 ARI       14     35      0        16     2            1       1         14
## 4 BAL       33     41      1        38     2            0       3         14
## 5 BUF       23     35      0        10     6            2       1         21
## 6 CAR       30     30      1        25     3            0       2         11
## # ... with 10 more variables: TDs <dbl>, twopoint <dbl>, yards <dbl>, N <dbl>,
## #   IsHome <fct>, score <dbl>, favorby <dbl>, roof <chr>, surface <chr>,
## #   RatioPasstoRush <dbl>
```

```
head(train1)
```

```
## # A tibble: 6 x 19
##   Team  Rushes Passes DidWin FirstDown sacks Interception Fumbles incomplete
##   <chr>  <dbl>  <dbl> <fct>      <dbl> <dbl>        <dbl>   <dbl>      <dbl>
## 1 CHI       16     49 0             15     5            1       1         20
## 2 GB        21     32 1             14     6            0       2         14
## 3 ARI       26     56 0             28     5            1       0         25
## 4 ATL       17     50 0             24     5            2       1         13
## 5 BAL       43     29 1             41     1            0       0          5
## 6 BUF       23     40 1             25     1            3       2         13
## # ... with 10 more variables: TDs <dbl>, twopoint <dbl>, yards <dbl>, N <dbl>,
## #   IsHome <fct>, score <dbl>, favorby <dbl>, roof <chr>, surface <chr>,
## #   RatioPasstoRush <dbl>
```

Decision Tree:

```
set.seed(10064)
mod1<-tree(DidWin~.,data=train1, method = "class")
cv_mod1 = cv.tree(mod1)
#plot(cv_mod1$size, cv_mod1$dev, type = 'b')
fittree2<-rpart(DidWin~.,train1,method="class")
plotcp(fittree2)
```

```
fittree2<-rpart(DidWin~.,train1,method="class")
rpart.plot(fittree2,extra=104)
```

```
tree_pred = predict(fittree2, test1, type="class")
table(tree_pred, test1$DidWin) %>%
  kbl(caption = "confusion Table of Classification Tree") %>%
  kable_styling()
```

Random Forest:

```
m1 <- randomForest(
  formula = DidWin~.,
  data    = train1,
   mtry = 20, importance = TRUE)

random_forest_estimate = predict(m1,
                                 newdata = test1)
```

```
plot(m1)
```

```
importance(m1)
```

```
varImpPlot(m1, main = "Random Forest")
```

```
table(random_forest_estimate, test1$DidWin)%>%
kbl(caption = "random_forest_estimate") %>%
  kable_styling()
```

Discriminant analysis

```
ggpairs(train1[,-1])
```

LDA:

```
model_LDA = lda(DidWin~., data = train1)
predictions_LDA = data.frame(predict(model_LDA, test1))
GMeans<-model_LDA$means
GMeans<-t(GMeans)
GMeans%>%
  kbl(caption = "Group Means of Training Data") %>%
  kable_styling()
```

```
model_LDA = lda(DidWin~., data = train1)
predictions_LDA = data.frame(predict(model_LDA, test1))
coeff<-model_LDA$scaling

coeff<-coeff[32:50,]

coeff%>%
  kbl(caption = "coefficients of linear discriminants output") %>%
  kable_styling()
```

```
predictions_LDA = cbind(test, predictions_LDA)

LDA<-predictions_LDA %>%
  count(class, DidWin) %>%
  kbl(caption = "LDA prediction Table") %>%
  kable_styling()
LDA
```

QDA:

```
model_QDA = qda(DidWin~., data = train1)
predictions_QDA = data.frame(predict(model_QDA, test1))

predictions_QDA = cbind(test, predictions_QDA)

Q<-predictions_QDA %>%
  count(class, DidWin) %>%
  kbl(caption = "QDA prediction Table") %>%
  kable_styling()
Q
```

```
set.seed(109)
minval<-1e10
lam<-0.2
for (k in 1:100) {
  init<-runif(12,-0.7,0.7)
fitnn<-nnet(class.ind(DidWin)~.,train1,size=2,decay=lam,entropy=TRUE,maxit=5000,wts=init)
 if (fitnn$value<minval) {
minval<-fitnn$value
fitnn.save<-fitnn
}
}
```

Neural Network

```
plot.nnet(fitnn.save, cex.val = 0.5 )
```

```
k<-test1[,-4]
class.predict<-predict(fitnn.save,k,type="class")
```

```
correct<-(test1$DidWin==class.predict)
n<-length(correct[correct== TRUE])
(1-(n/1530))*100
```

Clustering

```
clust <- read_csv("clust.csv")
C<-clust[,2:13]
c<-data.matrix(C)
rownames(c) <- clust$Team
distance <- get_dist(c)
fviz_dist(distance, gradient = list(low = "#FFFFFF", mid = "#00AFBB", high = "#000066"))
```

Hierarchical Clustering:

```
dist.map <- dist(clust)
hclustavg <- hclust(dist.map,method= "complete")
labels(hclustavg) <- clust$Team


x = c()
y = c(2:10)
N <- 10
for (k in 2:N) {
 p<-cutree(hclustavg,k=k)
 s<-silhouette(p,dist.map)
 S<-mean(s[,3])
 x<-append(x,S)
}
d<-cbind(y,x)
plot(d, type = "line")
```

```
dend<-as.dendrogram(hclustavg)%>%
  color_branches(k=7)
plot(dend)
```

```
set.seed(100)
mycol <- colorRampPalette(c("lightblue","blue","darkblue"))(12)
C<-clust[,2:13]
c<-data.matrix(C)
rownames(c) <- clust$Team

heatmap.2(c, col=mycol, trace="none",scale="column",cexCol =0.65 ,margins=c(7,5))
```

K-means:

```
fviz_nbclust(c, kmeans, method='silhouette')
```

```
set.seed(100)
k<-kmeans(c, centers = 6,nstart = 100)
fviz_cluster(k, data = c)
```

Self-organizing Maps:

```
SOM<-som(c,grid=somgrid(2,3,topo="hexagonal"),rlen=1000)
coords<-SOM$grid$pts
par(bg = rgb(0.9,0.94,0.95), font = 1, pch = 16, pch = 16, cex = 0.75)
plot(SOM,type="mapping",labels = rownames(c), cex.lab = 0.65,
     main = "NFL Team mapping", cex.main = 5.4)
text(coords,labels=seq(1,6), col = "blue")
```

Data Clean Appendix

Julia Lee

Training Data Set Prep

```
pbp_2021 <- read_csv("pbp-2021.csv")
names(pbp_2021)
as.numeric(pbp_2021$Minute)
as.numeric(pbp_2021$Second)
pbp<-pbp_2021%>%
mutate(m= ((Minute*60)+Second)/60) %>%
mutate(TimeLeftInQ=m) %>%
  filter(IsPenalty == 0) %>%
mutate(TimeinGame = ifelse(Quarter == 1, 15-m,ifelse(Quarter == 2, 30-m,ifelse(Quarter == 3, 45-m, 60-m
```

```
pbp_2021$Rush<-as.numeric(pbp_2021$IsRush)
pbp_2021$GOff<-paste(pbp_2021$GameDate, pbp_2021$OffenseTeam)
game<-pbp_2021 %>%
  group_by(GameDate,OffenseTeam,DefenseTeam, GameId) %>%
  summarise(Rushes=sum(Rush),Passes=sum(IsPass), win=sum(TeamWin), FirstDown=sum(SeriesFirstDown), sack

mean(game$FirstDown)
```

```
games <- read_csv("http://www.habitatring.com/games.csv")
g2021<-games %>%
  filter(season == 2021)
 g2021$GameId <- g2021$old_game_id
game21 <- game %>%
  full_join(g2021, by = "GameId")


game21<-game21 %>%
  mutate(S=ifelse(OffenseTeam == home_team, 1, 0)) %>%
 mutate(DidWin = ifelse(S == 1 & result > 0 , 1, ifelse(S == 0 & result < 0, 1, 0) ) )
```

```
game21$IsHome<-game21$S

game21 <- game21 %>%
  mutate(score = ifelse(IsHome == 1, home_score , away_score)) %>%
    mutate(QB = ifelse(IsHome == 1, home_qb_name,  away_qb_name)) %>%
  mutate(coach = ifelse(IsHome == 1, home_coach,  away_coach))%>%
   mutate(Team = ifelse(IsHome == 1, home_team,  away_team)) %>%
  mutate(favorby = ifelse(IsHome == 1, spread_line,   spread_line*-1)) %>%
  mutate(opponent = ifelse(IsHome == 1, away_team,   home_team)) %>%
    mutate(opponentScore = ifelse(IsHome == 1, away_score,   home_score)) %>%
  mutate(RatioPasstoRush = Passes/Rushes)
```

```
game_small<- game21 %>%
  group_by(GameDate, Team, opponent) %>%
  summarise(Team, Rushes,Passes, DidWin, FirstDown, sacks,Interception, Fumbles,incomplete, TDs,twopoin
```

```
game_clean<-game_small %>%
   na.omit()
```

```
pbp_2020 <- read_csv("pbp-2020.csv")
names(pbp_2020)
as.numeric(pbp_2020$Minute)
as.numeric(pbp_2020$Second)
pbp<-pbp_2020%>%
mutate(m= ((Minute*60)+Second)/60) %>%
mutate(TimeLeftInQ=m) %>%
  filter(IsPenalty == 0) %>%
mutate(TimeinGame = ifelse(Quarter == 1, 15-m,ifelse(Quarter == 2, 30-m,ifelse(Quarter == 3, 45-m, 60-m
```

```
pbp_2020$Rush<-as.numeric(pbp_2020$IsRush)
pbp_2020$GOff<-paste(pbp_2020$GameDate, pbp_2020$OffenseTeam)
game<-pbp_2020 %>%
  group_by(GameDate,OffenseTeam,DefenseTeam, GameId) %>%
  summarise(Rushes=sum(Rush),Passes=sum(IsPass), win=sum(TeamWin), FirstDown=sum(SeriesFirstDown), sacks

mean(game$FirstDown)
```

```
games <- read_csv("http://www.habitatring.com/games.csv")
g2020<-games %>%
  filter(season == 2020)
 g2020$GameId <- g2020$old_game_id
game20 <- game %>%
  full_join(g2020, by = "GameId")


game20<-game20 %>%
  mutate(S=ifelse(OffenseTeam == home_team, 1, 0)) %>%
 mutate(DidWin = ifelse(S == 1 & result > 0 , 1, ifelse(S == 0 & result < 0, 1, 0) ) )
```

```
game20$IsHome<-game20$S

game20 <- game20 %>%
  mutate(score = ifelse(IsHome == 1, home_score , away_score)) %>%
    mutate(QB = ifelse(IsHome == 1, home_qb_name,  away_qb_name)) %>%
  mutate(coach = ifelse(IsHome == 1, home_coach,  away_coach))%>%
   mutate(Team = ifelse(IsHome == 1, home_team,  away_team)) %>%
  mutate(favorby = ifelse(IsHome == 1, spread_line,   spread_line*-1)) %>%
  mutate(opponent = ifelse(IsHome == 1, away_team,   home_team)) %>%
    mutate(opponentScore = ifelse(IsHome == 1, away_score,   home_score)) %>%
  mutate(RatioPasstoRush = Passes/Rushes)
```

```
game_small2<- game20 %>%
  group_by(GameDate, Team, opponent) %>%
  summarise(Team, Rushes,Passes, DidWin, FirstDown, sacks,Interception, Fumbles,incomplete, TDs,twopoin
```

```
game_clean_2<-game_small2 %>%
   na.omit()
```

```
x<-as.data.frame(rbind(game_clean_2, game_clean))
```

```
pbp_2019 <- read_csv("pbp-2019.csv")

as.numeric(pbp_2019$Minute)
as.numeric(pbp_2019$Second)
pbp<-pbp_2019%>%
mutate(m= ((Minute*60)+Second)/60) %>%
mutate(TimeLeftInQ=m) %>%
  filter(IsPenalty == 0) %>%
mutate(TimeinGame = ifelse(Quarter == 1, 15-m,ifelse(Quarter == 2, 30-m,ifelse(Quarter == 3, 45-m, 60-m)
```

```
pbp_2019$Rush<-as.numeric(pbp_2019$IsRush)
pbp_2019$GOff<-paste(pbp_2019$GameDate, pbp_2019$OffenseTeam)
game<-pbp_2019 %>%
  group_by(GameDate,OffenseTeam,DefenseTeam, GameId) %>%
  summarise(Rushes=sum(Rush),Passes=sum(IsPass), win=sum(TeamWin), FirstDown=sum(SeriesFirstDown), sacks

mean(game$FirstDown)
```

```
games <- read_csv("http://www.habitatring.com/games.csv")
g2019<-games %>%
  filter(season == 2019)
 g2019$GameId <- g2019$old_game_id
game19 <- game %>%
  full_join(g2019, by = "GameId")


game19<-game19 %>%
  mutate(S=ifelse(OffenseTeam == home_team, 1, 0)) %>%
 mutate(DidWin = ifelse(S == 1 & result > 0 , 1, ifelse(S == 0 & result < 0, 1, 0) ) )
```

```
game19$IsHome<-game19$S
game19 <- game19 %>%
  mutate(score = ifelse(IsHome == 1, home_score , away_score)) %>%
    mutate(QB = ifelse(IsHome == 1, home_qb_name,  away_qb_name)) %>%
  mutate(coach = ifelse(IsHome == 1, home_coach,  away_coach))%>%
   mutate(Team = ifelse(IsHome == 1, home_team,  away_team)) %>%
  mutate(favorby = ifelse(IsHome == 1, spread_line,   spread_line*-1)) %>%
  mutate(opponent = ifelse(IsHome == 1, away_team,   home_team)) %>%
    mutate(opponentScore = ifelse(IsHome == 1, away_score,   home_score)) %>%
  mutate(RatioPasstoRush = Passes/Rushes)
```

```
game_small9<- game19 %>%
  group_by(GameDate, Team, opponent) %>%
  summarise(Team, Rushes,Passes, DidWin, FirstDown, sacks,Interception, Fumbles,incomplete, TDs,twopoint
game_clean_2<-game_small9 %>%
   na.omit()
x<-as.data.frame(rbind(game_clean_2,x))
```

```
x<-x%>%
  mutate(Team=ifelse(Team == "SD", "LAC" ,Team)) %>%
  mutate(Team=ifelse(Team == "OAK", "LV" ,Team))%>%
  mutate(surface=ifelse(surface == "grass","grass", "artificial"))
write.csv(x, "Train.csv")
```

test data

```
pbp_2021 <- read_csv("pbp-2016.csv")
names(pbp_2021)
as.numeric(pbp_2021$Minute)
as.numeric(pbp_2021$Second)
pbp<-pbp_2021%>%
mutate(m= ((Minute*60)+Second)/60) %>%
mutate(TimeLeftInQ=m) %>%
  filter(IsPenalty == 0) %>%
mutate(TimeinGame = ifelse(Quarter == 1, 15-m,ifelse(Quarter == 2, 30-m,ifelse(Quarter == 3, 45-m, 60-m
```

```
pbp_2021$Rush<-as.numeric(pbp_2021$IsRush)
pbp_2021$GOff<-paste(pbp_2021$GameDate, pbp_2021$OffenseTeam)
game<-pbp_2021 %>%
  group_by(GameDate,OffenseTeam,DefenseTeam, GameId) %>%
  summarise(Rushes=sum(Rush),Passes=sum(IsPass), win=sum(TeamWin), FirstDown=sum(SeriesFirstDown), sack

mean(game$FirstDown)
```

```
games <- read_csv("http://www.habitatring.com/games.csv")
g2021<-games %>%
  filter(season == 2016)
 g2021$GameId <- g2021$old_game_id
game21 <- game %>%
  full_join(g2021, by = "GameId")


game21<-game21 %>%
  mutate(S=ifelse(OffenseTeam == home_team, 1, 0)) %>%
 mutate(DidWin = ifelse(S == 1 & result > 0 , 1, ifelse(S == 0 & result < 0, 1, 0) ) )
```

```
game21$IsHome<-game21$S

game21 <- game21 %>%
  mutate(score = ifelse(IsHome == 1, home_score , away_score)) %>%
    mutate(QB = ifelse(IsHome == 1, home_qb_name,  away_qb_name)) %>%
  mutate(coach = ifelse(IsHome == 1, home_coach,  away_coach))%>%
   mutate(Team = ifelse(IsHome == 1, home_team,  away_team)) %>%
  mutate(favorby = ifelse(IsHome == 1, spread_line,   spread_line*-1)) %>%
  mutate(opponent = ifelse(IsHome == 1, away_team,   home_team)) %>%
    mutate(opponentScore = ifelse(IsHome == 1, away_score,   home_score)) %>%
  mutate(RatioPasstoRush = Passes/Rushes)
```

```
game_small<- game21 %>%
  group_by(GameDate, Team, opponent) %>%
  summarise(Team, Rushes,Passes, DidWin, FirstDown, sacks,Interception, Fumbles,incomplete, TDs,twopoin
```

```
game_clean<-game_small %>%
   na.omit()
```

```
pbp_2020 <- read_csv("pbp-2017.csv")
names(pbp_2020)
as.numeric(pbp_2020$Minute)
as.numeric(pbp_2020$Second)
```

```
pbp<-pbp_2020%>%
mutate(m= ((Minute*60)+Second)/60) %>%
mutate(TimeLeftInQ=m) %>%
  filter(IsPenalty == 0) %>%
mutate(TimeinGame = ifelse(Quarter == 1, 15-m,ifelse(Quarter == 2, 30-m,ifelse(Quarter == 3, 45-m, 60-m)
```

```
pbp_2020$Rush<-as.numeric(pbp_2020$IsRush)
pbp_2020$GOff<-paste(pbp_2020$GameDate, pbp_2020$OffenseTeam)
game<-pbp_2020 %>%
  group_by(GameDate,OffenseTeam,DefenseTeam, GameId) %>%
  summarise(Rushes=sum(Rush),Passes=sum(IsPass), win=sum(TeamWin), FirstDown=sum(SeriesFirstDown), sacks

mean(game$FirstDown)
```

```
games <- read_csv("http://www.habitatring.com/games.csv")
g2020<-games %>%
  filter(season == 2017)
 g2020$GameId <- g2020$old_game_id
game20 <- game %>%
  full_join(g2020, by = "GameId")


game20<-game20 %>%
  mutate(S=ifelse(OffenseTeam == home_team, 1, 0)) %>%
 mutate(DidWin = ifelse(S == 1 & result > 0 , 1, ifelse(S == 0 & result < 0, 1, 0) ) )
```

```
game20$IsHome<-game20$S

game20 <- game20 %>%
  mutate(score = ifelse(IsHome == 1, home_score , away_score)) %>%
    mutate(QB = ifelse(IsHome == 1, home_qb_name,  away_qb_name)) %>%
  mutate(coach = ifelse(IsHome == 1, home_coach,  away_coach))%>%
   mutate(Team = ifelse(IsHome == 1, home_team,  away_team)) %>%
  mutate(favorby = ifelse(IsHome == 1, spread_line,   spread_line*-1)) %>%
  mutate(opponent = ifelse(IsHome == 1, away_team,   home_team)) %>%
    mutate(opponentScore = ifelse(IsHome == 1, away_score,   home_score)) %>%
  mutate(RatioPasstoRush = Passes/Rushes)
```

```
game_small2<- game20 %>%
  group_by(GameDate, Team, opponent) %>%
  summarise(Team, Rushes,Passes, DidWin, FirstDown, sacks,Interception, Fumbles,incomplete, TDs,twopoint
```

```
game_clean_2<-game_small2 %>%
   na.omit()
y<-as.data.frame(rbind(game_clean_2, game_clean))
```

```
pbp_2019 <- read_csv("pbp-2018.csv")

as.numeric(pbp_2019$Minute)
as.numeric(pbp_2019$Second)
pbp<-pbp_2019%>%
mutate(m= ((Minute*60)+Second)/60) %>%
mutate(TimeLeftInQ=m) %>%
  filter(IsPenalty == 0) %>%
mutate(TimeinGame = ifelse(Quarter == 1, 15-m,ifelse(Quarter == 2, 30-m,ifelse(Quarter == 3, 45-m, 60-m)
```

```
pbp_2019$Rush<-as.numeric(pbp_2019$IsRush)
pbp_2019$GOff<-paste(pbp_2019$GameDate, pbp_2019$OffenseTeam)
game<-pbp_2019 %>%
  group_by(GameDate,OffenseTeam,DefenseTeam, GameId) %>%
  summarise(Rushes=sum(Rush),Passes=sum(IsPass), win=sum(TeamWin), FirstDown=sum(SeriesFirstDown), sacks

mean(game$FirstDown)
```

```
games <- read_csv("http://www.habitatring.com/games.csv")
g2019<-games %>%
  filter(season == 2018)
 g2019$GameId <- g2019$old_game_id
game19 <- game %>%
  full_join(g2019, by = "GameId")


game19<-game19 %>%
  mutate(S=ifelse(OffenseTeam == home_team, 1, 0)) %>%
 mutate(DidWin = ifelse(S == 1 & result > 0 , 1, ifelse(S == 0 & result < 0, 1, 0) ) )
```

```
game19$IsHome<-game19$S
game19 <- game19 %>%
  mutate(score = ifelse(IsHome == 1, home_score , away_score)) %>%
    mutate(QB = ifelse(IsHome == 1, home_qb_name,  away_qb_name)) %>%
  mutate(coach = ifelse(IsHome == 1, home_coach,  away_coach))%>%
   mutate(Team = ifelse(IsHome == 1, home_team,  away_team)) %>%
  mutate(favorby = ifelse(IsHome == 1, spread_line,   spread_line*-1)) %>%
  mutate(opponent = ifelse(IsHome == 1, away_team,   home_team)) %>%
    mutate(opponentScore = ifelse(IsHome == 1, away_score,   home_score)) %>%
  mutate(RatioPasstoRush = Passes/Rushes)
```

```
game_small9<- game19 %>%
  group_by(GameDate, Team, opponent) %>%
  summarise(Team, Rushes,Passes, DidWin, FirstDown, sacks,Interception, Fumbles,incomplete, TDs,twopoint
game_clean_2<-game_small9 %>%
   na.omit()
y<-as.data.frame(rbind(game_clean_2,y))
```

```
y<-y%>%
  mutate(Team=ifelse(Team == "SD", "LAC" ,Team)) %>%
  mutate(Team=ifelse(Team == "OAK", "LV" ,Team))%>%
  mutate(surface=ifelse(surface == "grass","grass", "artificial"))
write.csv(y, "test.csv")
```

cluster data

```
Games20and21<-rbind(x)
clustdata<- Games20and21 %>%
  group_by(Team) %>%
  summarise(MPasstoRush=mean(RatioPasstoRush), MPasses=mean(Passes), MPasses=mean(Rushes), MTD= mean(TDs

write_csv(clustdata, "clust.csv")
```