IST687 Project

Jon Kaplan

Jeffrey Bertucci

27 March 2019

library(tidyverse)  
library(lubridate)  
library(plotly)  
library(kernlab)  
library(randomForest)

**Importing the Data:**

#LOADING IN THE DATA:  
  
#1) all the GAME data  
OutdoorStadium <- c("Open", "Outdoor", "Outdoors", "Outside", "Retr. Roof - Open", "outdoor", "Oudoor","Ourdoor", "Outddors", "Outdor", "outdoor", "Outdor")  
  
IndoorStadium <- c("Dome", "Indoors", "Retr. Roof - Closed", "Indoor", "Closed Dome", "Dome, closed", "Domed, closed", "Indoor, non-retractable roof", "Indoors Domed")  
  
  
gamedata <- read\_csv("IST687 Final Project/game\_data.csv")

## Parsed with column specification:  
## cols(  
## GameKey = col\_double(),  
## Season\_Year = col\_double(),  
## Season\_Type = col\_character(),  
## Week = col\_double(),  
## Game\_Date = col\_datetime(format = ""),  
## Game\_Day = col\_character(),  
## Game\_Site = col\_character(),  
## Start\_Time = col\_time(format = ""),  
## Home\_Team = col\_character(),  
## HomeTeamCode = col\_character(),  
## Visit\_Team = col\_character(),  
## VisitTeamCode = col\_character(),  
## Stadium = col\_character(),  
## StadiumType = col\_character(),  
## Turf = col\_character(),  
## GameWeather = col\_character(),  
## Temperature = col\_double(),  
## OutdoorWeather = col\_character()  
## )

gamedata <- gamedata %>%   
 mutate(StadiumType2=case\_when(StadiumType %in% OutdoorStadium~"Outdoor",  
 StadiumType %in% IndoorStadium~"Indoor",  
 TRUE~"Unknown"))   
  
  
  
#2) CONCUSSION data  
videoreview <- read\_csv("IST687 Final Project/video\_review.csv")

## Parsed with column specification:  
## cols(  
## Season\_Year = col\_double(),  
## GameKey = col\_double(),  
## PlayID = col\_double(),  
## GSISID = col\_double(),  
## Player\_Activity\_Derived = col\_character(),  
## Turnover\_Related = col\_character(),  
## Primary\_Impact\_Type = col\_character(),  
## Primary\_Partner\_GSISID = col\_character(),  
## Primary\_Partner\_Activity\_Derived = col\_character(),  
## Friendly\_Fire = col\_character()  
## )

videoreview <- videoreview %>%   
 mutate(Concussion= 1) #add concussion field with 1 value  
  
  
  
#videoreview$Concussion <- as.numeric(videoreview$Concussion)  
  
#3) PLAY level data (score and play description), converted game clock to minutes and seconds since was hms format before  
playinformation <- read\_csv("IST687 Final Project/play\_information.csv", col\_types=cols(Game\_Date=col\_skip(), Game\_Clock=col\_time(format = "%M:%S")))  
  
#4) read in all player roles in punt plays  
playplayerrole <- read\_csv("IST687 Final Project/play\_player\_role\_data.csv")

## Parsed with column specification:  
## cols(  
## Season\_Year = col\_double(),  
## GameKey = col\_double(),  
## PlayID = col\_double(),  
## GSISID = col\_double(),  
## Role = col\_character()  
## )

#read in just punters and punt returners  
playplayerrolepunts <- read\_csv("IST687 Final Project/play\_player\_role\_data.csv") %>%   
 dplyr::filter(Role=="P" | Role=="PR")

## Parsed with column specification:  
## cols(  
## Season\_Year = col\_double(),  
## GameKey = col\_double(),  
## PlayID = col\_double(),  
## GSISID = col\_double(),  
## Role = col\_character()  
## )

#JOINING the data into one master file  
puntmaster <- gamedata %>% inner\_join(playinformation, by=c("GameKey", "Season\_Year", "Season\_Type", "Week")) %>%  
 inner\_join(playplayerrole, by=c("GameKey", "Season\_Year", "PlayID")) %>%   
 left\_join(videoreview) %>%   
 mutate(Quarter=as.factor(Quarter)) %>%   
 mutate(Concussion=if\_else(is.na(Concussion), 0, 1))

## Joining, by = c("GameKey", "Season\_Year", "PlayID", "GSISID")

#combining gamedata and videoreview for more game day information about concussion games  
  
  
gamevideo <- inner\_join(gamedata, videoreview) %>%   
 select(-Game\_Date)

## Joining, by = c("GameKey", "Season\_Year")

#OPTIONAL STUFF/notes from importing  
  
#%>%   
 #mutate(FullGameClock=((Quarter-1)\*15+Game\_Clock))  
  
#sums the total concussions in concussions column of puntmaster  
#sum(puntmaster$Concussion, na.rm = TRUE)  
  
  
#taking NA in concussions field and replacing with 0  
  
#puntmaster[is.na(puntmaster$Concussion),] <- 0  
#puntmaster %>% filter(is.na(Concussion))  
  
  
#OPTIONAL STUFF:  
  
  
#trying to figure out how to make gameclock for full 60 minutes intead of just quarters  
#playgamevideo <- left\_join(playinformation, gamevideo) %>%   
 #mutate(FullGameClock=hms((Quarter-1)\*15+Game\_Clock))

Importing the NGS Data:

# since NGS data is so massive, tried to create a way to import and also remove previous dataset that is imported

# 2016 import

NGS2016Post <- read\_csv(“NGS-2016-post.csv”, col\_types = cols(Event=col\_character()))

NGS2016Pre <- read\_csv(“NGS-2016-pre.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGS2016Post, NGS2016Pre) rm(NGS2016Post, NGS2016Pre)

NGS2016RegPart1 <- read\_csv(“NGS-2016-reg-wk1-6.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2016RegPart1) rm(NGS2016RegPart1)

NGS2016RegPart2 <- read\_csv(“NGS-2016-reg-wk7-12.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2016RegPart2) rm(NGS2016RegPart2)

NGS2016RegPart3 <- read\_csv(“NGS-2016-reg-wk13-17.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2016RegPart3) rm(NGS2016RegPart3)

# 2017 import

NGS2017Post <- read\_csv(“NGS-2017-post.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2017Post) rm(NGS2017Post)

NGS2017Pre <- read\_csv(“NGS-2017-pre.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2017Pre) rm(NGS2017Pre)

NGS2017Part1 <- read\_csv(“NGS-2017-reg-wk1-6.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2017Part1) rm(NGS2017Part1)

NGS2017Part2 <- read\_csv(“NGS-2017-reg-wk7-12.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2017Part2) rm(NGS2017Part2)

NGS2017Part3 <- read\_csv(“NGS-2017-reg-wk13-17.csv”, col\_types = cols(Event=col\_character()))

NGSData <- dplyr::union(NGSData, NGS2017Part3) rm(NGS2017Part3)

# join player roles with NGS data

NGSPlayerRole <- inner\_join(NGS2016Post, playplayerrole)

# join player roles with NGS data, only Punt and Punt Returner roles

NGSPlayerRolePRP <- inner\_join(NGSData, playplayerrolepunts)

# used union function

# NGSPlayerRole <- inner\_join(playplayerrole, NGS2016Post) %>%

# union(inner\_join(playplayerrole, NGS2016Pre))

# NGSPlayerRole <- inner\_join(playplayerrole, NGS2016Post) %>%

# dplyr::union(inner\_join(playplayerrole, NGS2016Pre))%>% #dplyr::union(inner\_join(playplayerrole, NGS2016RegPart1))%>% #dplyr::union(inner\_join(playplayerrole, NGS2016RegPart2))%>% #dplyr::union(inner\_join(playplayerrole, NGS2016RegPart3))%>% #dplyr::union(inner\_join(playplayerrole, NGS2017Post))%>% #dplyr::union(inner\_join(playplayerrole, NGS2017Pre))%>% #dplyr::union(inner\_join(playplayerrole, NGS2017Part1))%>% #dplyr::union(inner\_join(playplayerrole, NGS2017Part2))%>% #dplyr::union(inner\_join(playplayerrole, NGS2017Part3))

# 66,492,490 observations in NGS data

# OPTIONAL STUFF:

# remove all unnecessary data after joined

# rm(NGS2016Pre, NGS2016Post, NGS2016RegPart1, NGS2016RegPart2, NGS2016RegPart3, NGS2017Post, NGS2017Pre, NGS2017Part1, NGS2017Part2, NGS2017Part3)

# joining concussion occurences from gamevideo and adding velocities and speed data from NGSPlayerRole

# ConcussionNGS <- inner\_join(gamevideo, NGSPlayerRolePRP)

# events

# ConcussionNGS[!is.na(ConcussionNGS$Event),]

**Average Temperature for Concussions**

averagetempconcussions <- mean(gamevideo$Temperature)  
averagetempconcussions

## [1] 60.72973

#60.73 degrees

Season Type Count

#analysis of where concussions occurred during season  
seasontype <- gamevideo$SeasonType

## Warning: Unknown or uninitialised column: 'SeasonType'.

gamevideo %>% group\_by(Season\_Type) %>% count()

## # A tibble: 2 x 2  
## # Groups: Season\_Type [2]  
## Season\_Type n  
## <chr> <int>  
## 1 Pre 12  
## 2 Reg 25

**Stadium type count**

#analysis of outdoor vs indoor stadiums  
  
  
   
stadiumtypecounts <- gamedata %>% group\_by(StadiumType) %>% count() %>% arrange(desc(n))  
view(stadiumtypecounts)  
  
stadiumcount <- gamedata %>% group\_by(StadiumType2) %>% count()

**Turf count**

#figuring out the different turfs of fields  
  
turfcount <- gamevideo %>% group\_by(Turf) %>% count()  
turfcount

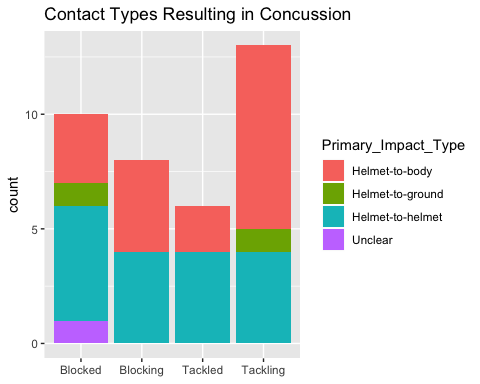
## # A tibble: 9 x 2  
## # Groups: Turf [9]  
## Turf n  
## <chr> <int>  
## 1 A-Turf Titan 1  
## 2 Artificial 4  
## 3 Field Turf 2  
## 4 FieldTurf 4  
## 5 FieldTurf 360 1  
## 6 Grass 14  
## 7 Natural grass 1  
## 8 Natural Grass 8  
## 9 UBU Speed Series-S5-M 2

turfcount2 <- gamedata %>% group\_by(Turf) %>% count()  
turfcount2

## # A tibble: 23 x 2  
## # Groups: Turf [23]  
## Turf n  
## <chr> <int>  
## 1 <NA> 1  
## 2 A-Turf Titan 20  
## 3 Artifical 3  
## 4 Artificial 61  
## 5 AstroTurf GameDay Grass 3D 1  
## 6 DD GrassMaster 21  
## 7 Field turf 1  
## 8 Field Turf 65  
## 9 FieldTurf 53  
## 10 FieldTurf 360 8  
## # … with 13 more rows

**Types of Impacts for Concussions:**

# histogram of types of concussion impacts  
ggplot(videoreview, aes(x=Player\_Activity\_Derived, fill = Primary\_Impact\_Type))+  
 geom\_bar(stat = 'Count')+  
 ggtitle("Contact Types Resulting in Concussion")+  
 xlab(NULL)

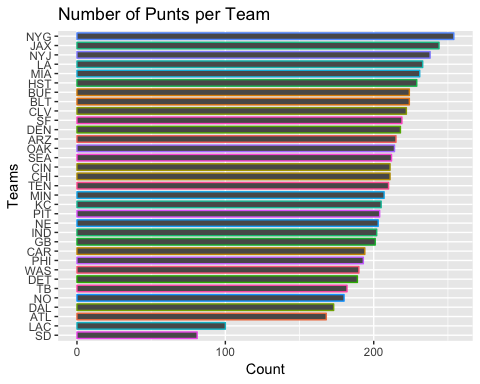


videoreview %>% group\_by(Primary\_Impact\_Type) %>% count()

## # A tibble: 4 x 2  
## # Groups: Primary\_Impact\_Type [4]  
## Primary\_Impact\_Type n  
## <chr> <int>  
## 1 Helmet-to-body 17  
## 2 Helmet-to-ground 2  
## 3 Helmet-to-helmet 17  
## 4 Unclear 1

**Punts per Team**

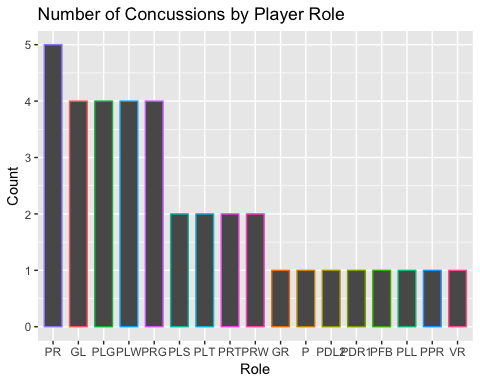
#looking at amount of punts by team to get general overview  
  
# punts per team, orientation changed  
  
playinformationsummary <- playinformation %>%   
 group\_by(Poss\_Team) %>%   
 summarize(NumPunts=n())  
  
#can change color to fill if want different color  
ggplot(playinformationsummary, aes(x=reorder(Poss\_Team, NumPunts), y=NumPunts, color=Poss\_Team))+  
 geom\_col(width=.7)+coord\_flip()+  
 ggtitle("Number of Punts per Team")+  
 xlab("Teams")+  
 ylab("Count")+  
 theme(legend.position="none")



#37/6681  
#0.55% chance of a concussion on each punt

**Player Roles and Concussion**

#finding out what player role types most affected by punt concussions  
  
concussedroles <- puntmaster %>%   
 filter(Concussion == 1) %>%   
 group\_by(Role) %>%   
 summarize(NumConcussions=n())  
  
  
ggplot(concussedroles, aes(x=reorder(Role, -NumConcussions), y=NumConcussions, color=Role))+  
 geom\_col(width=.7)+  
 ggtitle("Number of Concussions by Player Role")+  
 xlab("Role")+  
 ylab("Count")+  
 theme(legend.position="none")



**Distance between Punt and Punt Returner**

# using the NGS data to find out what distance is between Punt and punt returner on punt plays, seeing if distance is a factor in punt concussions

# use NGSData with filter for only P and PR roles

# find way to figure out which field to filter by, decide on “punt”

unique(NGSPlayerRolePRP$Event)

# rename x and y for P and PR specific

ngspunt <- NGSPlayerRolePRP %>% filter(Event==“punt”, Role==“P”) %>% rename(XP=x, YP=y)

ngspr <- NGSPlayerRolePRP %>% filter(Event==“punt”, Role==“PR”) %>% rename(XPR=x, YPR=y)

# use distance formula to calculate using X and Y

prpdist <- inner\_join(ngspunt, ngspr, by=c(“GameKey”, “Season\_Year”, “Time”, “PlayID”)) %>% mutate(Distance=sqrt((XP-XPR)2+(YP-YPR)2))

# new data with concussions joined

prpdistconc <- inner\_join(prpdist, videoreview, by=c(“GameKey”, “PlayID”, “Season\_Year”))

# linear regression

# glm()

# distmodel <- glm(Concussion~Distance, data=prpdistconc, family=binomial())

# summary(distmodel)

range(prpdist$Distance) #24.15795 - 74.91791

range(prpdistconc$Distance) #41.44668 - 69.59000

mean(prpdistconc$Distance) #57.50733

mean(prpdist$Distance) #56.62935

mean(gamedata$Temperature)

# boxplot

ggplot(prpdistconc, aes(x=Concussion, y=Distance)) + geom\_boxplot()

puntmasterdistance <- inner\_join(puntmaster, prpdist)

# boxplot

ggplot(puntmasterdistance, aes(x=Concussion==1, y=Distance)) + geom\_boxplot()

**Regression Comparing Game\_Day, Temperature, Turf, and Quarter**

#glm()  
#?glm()  
  
model1 <- glm(Concussion~Temperature+Game\_Day+Turf, data=puntmaster, family=binomial())  
summary(model1)

##   
## Call:  
## glm(formula = Concussion ~ Temperature + Game\_Day + Turf, family = binomial(),   
## data = puntmaster)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.0724 -0.0263 -0.0212 -0.0180 4.2374   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.224e+00 1.553e+00 -4.652 3.29e-06 \*\*\*  
## Temperature -1.770e-02 8.437e-03 -2.098 0.0359 \*   
## Game\_DayMonday -7.498e-02 1.250e+00 -0.060 0.9522   
## Game\_DaySaturday 1.146e-01 1.120e+00 0.102 0.9185   
## Game\_DaySunday -4.794e-01 1.051e+00 -0.456 0.6484   
## Game\_DayThursday 6.978e-01 1.058e+00 0.659 0.5097   
## Game\_DayWednesday -1.532e+01 3.322e+03 -0.005 0.9963   
## TurfArtifical -1.459e+01 3.320e+03 -0.004 0.9965   
## TurfArtificial 5.696e-01 1.119e+00 0.509 0.6109   
## TurfDD GrassMaster -1.547e+01 1.176e+03 -0.013 0.9895   
## TurfField turf -1.466e+01 4.891e+03 -0.003 0.9976   
## TurfField Turf -2.090e-01 1.227e+00 -0.170 0.8648   
## TurfFieldTurf 5.122e-01 1.120e+00 0.457 0.6474   
## TurfFieldTurf 360 1.822e+00 1.427e+00 1.277 0.2016   
## Turfgrass -1.495e+01 3.255e+03 -0.005 0.9963   
## TurfGrass 3.653e-01 1.038e+00 0.352 0.7249   
## TurfNatrual Grass -1.448e+01 3.697e+03 -0.004 0.9969   
## TurfNatural -1.471e+01 2.827e+03 -0.005 0.9958   
## TurfNatural grass 6.494e-01 1.430e+00 0.454 0.6498   
## TurfNatural Grass 3.489e-01 1.067e+00 0.327 0.7436   
## TurfNaturall Grass -1.485e+01 6.403e+03 -0.002 0.9981   
## TurfSynthetic -1.490e+01 4.900e+03 -0.003 0.9976   
## TurfUBU Speed Series S5-M -1.486e+01 3.993e+03 -0.004 0.9970   
## TurfUBU Speed Series-S5-M 1.564e-01 1.233e+00 0.127 0.8991   
## TurfUBU Sports Speed S5-M -1.524e+01 1.289e+03 -0.012 0.9906   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 680.11 on 133474 degrees of freedom  
## Residual deviance: 660.00 on 133450 degrees of freedom  
## (13054 observations deleted due to missingness)  
## AIC: 710  
##   
## Number of Fisher Scoring iterations: 22

model2 <- glm(Concussion~Temperature, data=puntmaster, family=binomial())  
summary(model2)

##   
## Call:  
## glm(formula = Concussion ~ Temperature, family = binomial(),   
## data = puntmaster)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.0322 -0.0246 -0.0229 -0.0217 4.1159   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.552792 0.521483 -14.483 <2e-16 \*\*\*  
## Temperature -0.010192 0.008149 -1.251 0.211   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 680.22 on 133672 degrees of freedom  
## Residual deviance: 678.72 on 133671 degrees of freedom  
## (12856 observations deleted due to missingness)  
## AIC: 682.72  
##   
## Number of Fisher Scoring iterations: 11

model3 <- glm(Concussion~Game\_Day, data=puntmaster, family=binomial())  
summary(model3)

##   
## Call:  
## glm(formula = Concussion ~ Game\_Day, family = binomial(), data = puntmaster)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.0317 -0.0235 -0.0192 -0.0192 4.1464   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.4591 1.0001 -8.458 <2e-16 \*\*\*  
## Game\_DayMonday 0.2630 1.2249 0.215 0.830   
## Game\_DaySaturday 0.4713 1.0956 0.430 0.667   
## Game\_DaySunday -0.1369 1.0275 -0.133 0.894   
## Game\_DayThursday 0.8657 1.0446 0.829 0.407   
## Game\_DayWednesday -11.1069 449.6490 -0.025 0.980   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 687.01 on 146528 degrees of freedom  
## Residual deviance: 680.11 on 146523 degrees of freedom  
## AIC: 692.11  
##   
## Number of Fisher Scoring iterations: 18

model4 <- glm(Concussion~Turf, data=puntmaster, family=binomial())  
summary(model4)

##   
## Call:  
## glm(formula = Concussion ~ Turf, family = binomial(), data = puntmaster)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.0377 -0.0241 -0.0234 -0.0217 4.1950   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.35702 1.00012 -8.356 <2e-16 \*\*\*  
## TurfArtifical -14.20904 1691.35332 -0.008 0.993   
## TurfArtificial 0.27655 1.11817 0.247 0.805   
## TurfDD GrassMaster -14.20904 725.10322 -0.020 0.984   
## TurfField turf -14.20904 2966.26768 -0.005 0.996   
## TurfField Turf -0.44166 1.22487 -0.361 0.718   
## TurfFieldTurf 0.33865 1.11818 0.303 0.762   
## TurfFieldTurf 360 1.10923 1.41455 0.784 0.433   
## TurfFieldTurf360 -14.20904 2421.94749 -0.006 0.995   
## Turfgrass -14.20904 1979.17860 -0.007 0.994   
## TurfGrass 0.15695 1.03522 0.152 0.879   
## TurfNatrual Grass -14.20904 2242.28769 -0.006 0.995   
## TurfNatural -14.20904 1714.74208 -0.008 0.993   
## TurfNatural grass 0.55932 1.41444 0.395 0.693   
## TurfNatural Grass 0.21249 1.06079 0.200 0.841   
## TurfNaturall Grass -14.20904 3883.75595 -0.004 0.997   
## TurfSynthetic -14.20904 2971.90162 -0.005 0.996   
## TurfUBU Speed Series S5-M -14.20904 2421.94749 -0.006 0.995   
## TurfUBU Speed Series-S5-M -0.06333 1.22489 -0.052 0.959   
## TurfUBU Sports Speed S5-M -14.20904 790.52634 -0.018 0.986   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 686.91 on 146330 degrees of freedom  
## Residual deviance: 678.40 on 146311 degrees of freedom  
## (198 observations deleted due to missingness)  
## AIC: 718.4  
##   
## Number of Fisher Scoring iterations: 21

#(intercept) is quarter 1 in summary because converted to factors  
model5 <- glm(Concussion~Quarter, data=puntmaster, family=binomial())  
summary(model5)

##   
## Call:  
## glm(formula = Concussion ~ Quarter, family = binomial(), data = puntmaster)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.0284 -0.0235 -0.0235 -0.0199 4.2129   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.8740 0.4472 -19.842 <2e-16 \*\*\*  
## Quarter2 0.6788 0.5394 1.258 0.2082   
## Quarter3 1.0554 0.5210 2.026 0.0428 \*   
## Quarter4 0.3430 0.5856 0.586 0.5580   
## Quarter5 -10.6920 432.9405 -0.025 0.9803   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 687.01 on 146528 degrees of freedom  
## Residual deviance: 681.40 on 146524 degrees of freedom  
## AIC: 691.4  
##   
## Number of Fisher Scoring iterations: 18

#Quarter 3 is significant, 0.0428

**Animated plot**

# decided to create an animated plot for player positions during concussion plays

# used tidyverse, plotly, lubridate packages

# create vectors for receiving and punting coverage, 4 variables for plot

rec <- c(“PDR1”, “PDR2”, “PDR3”, “PDL3”, “PDL2”, “PDL1”, “PLR”, “PLM”, “PLL”, “VL”, “VRi”, “VRo”, “PFB”)

punt <- c(“GL”, “PLW”, “PLT”, “PLG”, “PLS”, “PRG”, “PRT”, “PRW”, “GR”, “PC”, “PPR”, “PPL”)

p <- c(“P”)

pr <- c(“PR”)

# EXAMPLE 1

# going to animate an instance of PR being concussed, helmet to helmet impact

playplayerrolepunter <- read\_csv(“IST687 Final Project/play\_player\_role\_data.csv”) %>% dplyr::filter(Role==“PR”) #filter just PR

justpunter <- inner\_join(gamevideo, playplayerrolepunter) #join concussion data with just PR

gamevideoroles <- inner\_join(gamevideo, playplayerrole) #get all roles

# load data

NGS2016RegPart2 <- read\_csv(“NGS-2016-reg-wk7-12.csv”, col\_types = cols(Event=col\_character()))

input <- readr::read\_csv(“NGS-2016-reg-wk7-12.csv”, col\_types = cols(Event=col\_character())) %>% dplyr::filter(PlayID==3509)

input2 <- inner\_join(input, playplayerrole) %>% mutate(Side=case\_when(Role %in% rec~“Punt Return”, Role %in% punt~“Punt Coverage”, Role == p~“Punter”, Role == pr~“Punt Returner”)) %>% mutate(display\_time=as.numeric(Time-min(Time)))

# build the plot

p <- ggplot2::ggplot(input2, ggplot2::aes(x = x, y = y, color = Side, ids = GSISID, frame = display\_time)) + ggplot2::geom\_point(size = 3, alpha = .6) + ggplot2::geom\_text(ggplot2::aes(label = Role), size = 1.5, color = ‘black’) + ggplot2::theme\_minimal() + ggplot2::theme(legend.position = “none”)

# display the plot

p

# convert the ggplot into a plotly

animated\_p <- plotly::ggplotly(p) %>% plotly::animation\_opts(frame = 100)

# execute

animated\_p

# EXAMPLE2

# helmet to body impact concussion on position PLW

input3 <- readr::read\_csv(“NGS-2016-pre.csv”, col\_types = cols(Event=col\_character())) %>% dplyr::filter(PlayID==3129)

input4 <- inner\_join(input, playplayerrole) %>% mutate(Side=case\_when(Role %in% rec~“Punt Return”, Role %in% punt~“Punt Coverage”, Role == p~“Punter”, Role == pr~“Punt Returner”)) %>% mutate(display\_time=as.numeric(Time-min(Time)))

# finds all unique values for punt return and punt coverage

# unique(input2[input2$Side==“Punt Return”,]$Role) #unique(input2[input2$Side==“Punt Coverage”,]$Role)

# build the plot

p2 <- ggplot2::ggplot(input4, ggplot2::aes(x = x, y = y, color = Side, ids = GSISID, frame = display\_time)) + ggplot2::geom\_point(size = 3, alpha = .6) + ggplot2::geom\_text(ggplot2::aes(label = Role), size = 1.5, color = ‘black’) + ggplot2::theme\_minimal() + ggplot2::theme(legend.position = “none”)

# display the plot

p2

# convert the ggplot into a plotly

animated\_p2 <- plotly::ggplotly(p) %>% plotly::animation\_opts(frame = 100)

# execute

animated\_p2

# found this idea from here:

# <http://jimkloet.com/animated_nfl_play.html>

**Machine learning**

# used kernlab and randomForest

puntmaster <- gamedata %>% inner\_join(playinformation, by=c(“GameKey”, “Season\_Year”, “Season\_Type”, “Week”)) %>% inner\_join(playplayerrole, by=c(“GameKey”, “Season\_Year”, “PlayID”)) %>% left\_join(videoreview) %>% mutate(Quarter=as.factor(Quarter)) %>% mutate(Concussion=if\_else(is.na(Concussion), 0, 1))

puntmaster2<-puntmaster

rem<-c(1,2, 5, 7, 9:14, 18, 20, 21, 23, 25:27, 29, 30, 32:37)

right = function (string, char){ substr(string,nchar(string)-(char-1),nchar(string)) }

left = function (string,char){ substr(string,1,char) } puntmaster2Score\_Home\_Visiting) #puntmaster2<-puntmaster2[, -rem]

puntmaster2Score\_Home\_Visiting, 2)) puntmaster2Score\_Home\_Visiting, 2))

puntmaster2Score\_Home-puntmaster2$Score\_visiting

rem2<-c(11,14,15)

puntmaster2<-puntmaster2[, -rem2]

puntmaster3<-puntmaster2

blanks<-which(puntmaster3$GameWeather=="") puntmaster3$GameWeather<-as.character(puntmaster3GameWeather[blanks]<-gsub(“”, “Normal”, puntmaster3$GameWeather[blanks])

# Assuming that if no note was made regarding weather, we will describe no note as “normal conditions”

# a different description than “sunny” conditions.

Chance<-c(“30% Chance of Rain”, “Chance of Showers”, “Cloudy with Possible Stray Showers/Thundershowers”, “Cloudy, chance of rain”, “Cloudy, Humid, Chance of Rain”, “Partly Cloudy, Chance of Rain 80%”, “Rain likely, temps in low 40s.”)

puntmaster3$GameWeather<-gsub("30% Chance of Rain", "Chance of Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Chance of Showers", "Chance of Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy with Possible Stray Showers/Thundershowers", "Chance of Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy, chance of rain", "Chance of Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy, Humid, Chance of Rain", "Chance of Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Partly Cloudy, Chance of Rain 80%", "Chance of Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Rain likely, temps in low 40s.", "Chance of Rain", puntmaster3$GameWeather)

Rain<-c(“Cloudy with periods of rain, thunder possible. Winds shifting to WNW, 10-20 mph.”, “Cloudy with rain”, “Cloudy, Rain”, “Light Rain”, “Rain”, “Showers”)

puntmaster3$GameWeather<-gsub("Cloudy with periods of rain, thunder possible. Winds shifting to WNW, 10-20 mph.", "Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy with rain", "Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy, Rain", "Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Light Rain", "Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Rain", "Rain", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Showers", "Rain", puntmaster3$GameWeather)

# For cloud coverage, basically if it is the only thing of note.

Cloudy<-c(“cloudy”, “Cloudy”, “Cloudy and cold”, “Cloudy and Cold”, “Cloudy, steady temps”, “Coudy”, “Mostly cloudy”, “Mostly Cloudy”, “Mostly CLoudy”, “Mostly Coudy”)

puntmaster3$GameWeather<-gsub("cloudy", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy and cold", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy and Cold", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy, steady temps", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Mostly cloudy", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Mostly Cloudy", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Mostly CLoudy", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Mostly Coudy", "Cloudy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Coudy", "Cloudy", puntmaster3$GameWeather)

Thunder<-c(“T-Storms”, “Scattered thunderstorms”)

puntmaster3$GameWeather<-gsub("T-Storms", "Thunder Storms", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Scattered thunderstorms", "Thunder Storms", puntmaster3$GameWeather)

# distinguishing clear/normal from sunny, mentality->clear means peak football conditions all around

Sunny<-c(“Suny”, “Sunny, highs to upper 80s”, “Sunny Skies”, “Sunny Intervals”, “Sunny intervals”, “Sunny and warm”, “Sunny and cool”, “Sunny and Clear”, “Sunny”, “Sun & clouds”, “Sunny”, “Party Cloudy”, “Partly Sunny”, “Partly sunny”, “Partly Cloudy, lows to upper 50s.”, “Partly CLoudy”, “Partly Cloudy”, “Mostly Sunny”)

puntmaster3$GameWeather<-gsub("Suny", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny, highs to upper 80s", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny Skies", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny Intervals", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny intervals", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny and warm", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny and cool", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny and Clear", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sun & clouds", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Party Cloudy", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Partly Sunny", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Partly sunny", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Partly Cloudy, lows to upper 50s.", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Partly CLoudy", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Partly Cloudy", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Mostly Sunny", "Sunny", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Sunny", "Sunny", puntmaster3$GameWeather)

Snow<-c(“Snow showers”, “Snow”, “Heavy lake effect snow”, “Snow Rain, 3 to 5 inches expected.”, “Cloudy, light snow accumulating 1-3”, “Snow”)

puntmaster3$GameWeather<-gsub("Snow", "Snow", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Snow showers", "Snow", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Heavy lake effect snow", "Snow", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Snow Rain, 3 to 5 inches expected.", "Snow", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub('Cloudy, light snow accumulating 1-3"', "Snow", puntmaster3$GameWeather)

fog<-c(“Cloudy, fog started developing in 2nd quarter”, “Cloudy with patches of fog”)

puntmaster3$GameWeather<-gsub("Cloudy, fog started developing in 2nd quarter", "Foggy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cloudy with patches of fog", "Foggy", puntmaster3$GameWeather)

# Clear is a catch-all at the end, essentially meaning “no weather to report”. Thats good football.

Clear<-c(“Clear”, “CLEAR”, “Clear and Cold”, “Clear and cool”, “Clear and warm”, “Clear skies”, “Clear Skies”, “Cold”, “Fair”, “Mostly Clear. Gusting ot 14.”, “Normal”)

puntmaster3$GameWeather<-gsub("Clear", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("CLEAR", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Clear and Cold", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Clear and cool", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Clear and warm", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Clear skies", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Clear Skies", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Cold", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Fair", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Mostly Clear. Gusting ot 14.", "Clear", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Normal", "Clear", puntmaster3$GameWeather)

Hazy<-c(“Hazy, hot and humid”, “Hazy”)

puntmaster3$GameWeather<-gsub("Hazy, hot and humid", "Hazy", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Hazy", "Hazy", puntmaster3$GameWeather)

puntmaster3$GameWeather<-gsub("Indoor", "Indoors", puntmaster3$GameWeather) puntmaster3$GameWeather<-gsub("Indoorss", "Indoors", puntmaster3$GameWeather)

puntmaster3GameWeather)

puntmaster3Week)

# Similar to Weather, Turf needs to be normalized

blanks2<-which(puntmaster3$Turf=="") puntmaster3$Turf<-as.character(puntmaster3Turf[blanks2]<-gsub(“”, “Natural Grass”, puntmaster3$Turf[blanks2])

Astroturf<-c(“A-Turf Titan”, “Artifical”, “Artificial”, “AstroTurf GameDay Grass 3D”, “DD GrassMaster”, “Synthetic”, “UBU Speed Series-S5-M”, “UBU Speed Series S5-M”, “UBU Sports Speed S5-M”, “Field turf”, “Field Turf”, “FieldTurf”, “FieldTurf360”, “Synthetic 360”, “Synthetic360”)

puntmaster3$Turf<-gsub("A-Turf Titan", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Artifical", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Artificial", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("AstroTurf GameDay Grass 3D", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("DD GrassMaster", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Synthetic", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("UBU Speed Series-S5-M", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("UBU Speed Series S5-M", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("UBU Sports Speed S5-M", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Field turf", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Field Turf", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("FieldTurf", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("FieldTurf360", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Synthetic 360", "Synthetic", puntmaster3$Turf) puntmaster3$Turf<-gsub("Synthetic360", "Synthetic", puntmaster3$Turf)

Natural<-c(“grass”, “Grass”, “Natrual Grass”, “Natural”, “Natural grass”, “Natural Grass”, “Natural Grass”, “Naturall Grass”, “Natrual Natural”, “Natural Natural”, “Natural Natural”, “Naturall Natural”, “Natural”)

puntmaster3$Turf<-gsub("grass", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Grass", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natrual Grass", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural grass", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural Grass", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural Grass ", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Naturall Grass", "Natural", puntmaster3$Turf)

puntmaster3$Turf<-gsub("Natrual Natural", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural Natural", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural Natural", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Naturall Natural", "Natural", puntmaster3$Turf) puntmaster3$Turf<-gsub("Natural ", "Natural", puntmaster3$Turf)

puntmaster3Turf) summary(puntmaster3$Turf)

puntmaster3Temperature)

puntmaster3StadiumType2)

puntmaster3Score\_difference)

puntmaster3Concussion)

puntmaster3Start\_Time)

# regulating all indoor weather and temperature to account for weird readings, averaging at 70 degrees

puntmaster3StadiumType2==“Indoor”)] <-“Indoor” puntmaster3StadiumType2==“Indoor”)] <-“70”

puntmaster4<-puntmaster3

# removing NAs, these are mostly temperature, none of the NAs had concussions

puntmasterna<-complete.cases(puntmaster4)

puntmaster4<-puntmaster4[puntmasterna,]

puntmaster4Score\_difference)

puntmaster4Week)

puntmaster4Quarter)

puntmaster4Role)

puntmaster4Role)

puntmaster4$Turf<-gsub("Natural", 1, puntmaster4$Turf) puntmaster4$Turf<-gsub("Synthetic", 0, puntmaster4$Turf)

puntmaster4Turf

puntmaster4<-puntmaster4[,-4]

puntmaster4NaturalTurf)

# times are very difficult to parse, removing them

puntmaster4<-puntmaster4[,-c(4, 9)]

# categorizing weather conditions into 3 groups 1, 2, 3: Optimal, Suboptimal, Bad

summary(puntmaster4$GameWeather)

Optimal<-c(“Clear”, “Cloudy”, “Indoors”, “Sunny”)

puntmaster4$GameWeather<-gsub("Clear", 1, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Cloudy", 1, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Indoors", 1, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Sunny", 1, puntmaster4$GameWeather)

SubOptimal<-c(“Chance of Rain”, “Hazy”, “Foggy”)

puntmaster4$GameWeather<-gsub("Chance of Rain", 2, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Hazy", 2, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Foggy", 2, puntmaster4$GameWeather)

Bad<-c(“Rain”, “Snow”, “Thunder Storms”)

puntmaster4$GameWeather<-gsub("Rain", 3, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Snow", 3, puntmaster4$GameWeather) puntmaster4$GameWeather<-gsub("Thunder Storms", 3, puntmaster4$GameWeather)

puntmaster4GameWeather) puntmaster4NaturalTurf)

summary(puntmaster4$NaturalTurf)

puntmasterna<-complete.cases(puntmaster4)

puntmaster4<-puntmaster4[puntmasterna,]

which(puntmasterna==FALSE)

randindex<-sample(1:dim(puntmaster4)[1]) traindata<-puntmaster4[randindex[1:8856],] testdata<-puntmaster4[randindex[8857:13284],]

rm(gamedata, playinformation, playplayerrole)

# Running a random forest based on our most recent data frame. He’s not good, but he is trying.

# The problem is trying to predict a small handful of ones in an ocean of zeros

# is incredibly difficult. We are surprised it even made an

# attempt to guess at a couple concussion plays

rf<-randomForest(Concussion ~., data=puntmaster4)

sum(rferr.rate)

sum(rferr.rate)

# no benefits from more trees

rf2<-randomForest(Concussion ~., data=puntmaster4, ntree=1000)

# KSVM actually predicts 2 concussion correctly, but gets the rest of the data wrong.

# It guessed concussion on everything, which means there is most likely more, and better factors

# that can predict these concussions, but they may not be in this dataset.

ML2<-ksvm(Concussion~NaturalTurf+Score\_difference+Quarter+GameWeather, data=traindata, kernel=“polydot”, kpar=“automatic”, C=5, cross=3, prob.model=TRUE)

prediction<-predict(ML2, testdata)

comptable<-data.frame(testdata$Concussion, prediction)

comptableprediction)

length(which(comptable$prediction==1&comptable$prediction==comptable$testdata.Concussion)) comptable

# Once again,

# what we are testing for is so small in comparison to our data, it is going to be extremely hard

# to teach a machine to rout out the main causes of these events. But, it may be possible

# with more data.

length(which(puntmaster$Concussion==1))

**Resources:**

* <http://jimkloet.com/animated_nfl_play.html>
* <https://www.kaggle.com/c/NFL-Punt-Analytics-Competition>