# Tracking Data Project Long Form

Sean Hellingman

2023-07-31

## Discription of Problem

*Events describe passes from open play in the same general area of the field. The player locations data is a record for the locations of the players and the ball for each of these passing events. These events can be joined together in both locations using the GameEventID field. The passing player is denoted by the IsEPlayer in the player locations data, and EPlayerID in the event data. Whether or not these individual possessions resulted in a goal is given by the IsPossGoal field. Note that the orientation of the coordinates is always going from left to right (negative to positive towards the opposition goal).*

*We would like you to demonstrate your skillset by using the player locations data to make insights about the probability of individual passes resulting in goals (denoted by the IsPossGoal field). We are looking for constructed aggregate measures (players in front of the passer, opponents in proximity to ball, etc..) of the player locations data that might be useful in predicting the previously stated IsPossGoal response. You do not necessarily have to make predictions on the IsPossGoal field, rather we are looking for constructed measurements of the on-field situation that might be used for such, as well as their general specific signifigance.*

## Packages

The following R packages were used for this analysis.

```r
set.seed(14)
#rm(list = ls())
if(!require(readr)) install.packages("readr")
library(readr)
if(!require(tidyverse)) install.packages("tidyverse")
library(tidyverse)
if(!require(ggplot2)) install.packages("ggplot2")
library(ggplot2)
#if(!require(fitdistrplus)) install.packages("fitdistrplus")
#library(fitdistrplus)
#if(!require(EnvStats)) install.packages("EnvStats")
#library(EnvStats)
if(!require(caret)) install.packages("caret")
library(caret)
#if(!require(AER)) install.packages("AER")
#library(AER)
#if(!require(iccCounts)) install.packages("iccCounts")
#library(iccCounts)
if(!require(lme4)) install.packages("lme4")
library(lme4)
```

```
if(!require(knitr)) install.packages("knitr")
library(knitr)
if(!require(kableExtra)) install.packages("kableExtra")
library(kableExtra)
if(!require(broom)) install.packages("broom")
library(broom)
if(!require(corrplot)) install.packages("corrplot")
library(corrplot)
#if(!require(ROSE)) install.packages("ROSE")
#library(ROSE)
#if(!require(sjPlot)) install.packages("sjPlot")
#library(sjPlot)
#if(!require(ICCbin)) install.packages("ICCbin")
#library(ICCbin)
#if(!require(stargazer)) install.packages("stargazer")
#library(stargazer)
```

# Introduction

In order to complete this task, information from the player locations dataset was aggregated and then combined with the events dataset to determine the outcomes of the passes as they relate to immediately scoring a goal or not. Logistic regression was chosen as a modelling technique due to computational simplicity and ease of interpretation.

# Data

```
Events <- read_csv("Events.csv")
PlayerLocations <- read_csv("PlayerLocations.csv")
```

To account for any potential timing errors the location of the player passing the ball at the time of the pass was considered as the reference point and not the location of the ball. The passes included in this dataset originate in roughly the same area of the pitch. They occur centrally and in the offensive half of the pitch. Therefore, very little was done to characterize the passing locations. A variable indicating the absolute distance from center on the $y$ axis was constructed.

The first of the aggregated variables was constructed as the number of defensive players (including the keeper) between the passer and the byline they are attacking. This variable was then refined to only include the number of players centrally between the passer and the byline they are attacking. In other words, how many defensive players are actually between the passer and the goal.

Next, the same was done for attacking players ahead of the passer. Both the number of attacking players ahead of the ball and the number of attacking players centrally ahead of the ball were recorded.

A player was considered to be in a central position if they were in the middle third of the $y$ axis defined in the data. This was done because all of the passes originated in a central position and goals in soccer are commonly scored from a central position.

```
#Use passer location instead of ball location due to potential lags
PlayerLocations$PasserX <- PlayerLocations$IsePlayer *PlayerLocations$PlayerX

PlayerLocations <- PlayerLocations %>%
          group_by(GameEventID) %>%
          mutate(
            PasserX = PasserX[which.max(abs(PasserX))]
          )


PlayerLocations$PasserY <- PlayerLocations$IsePlayer *PlayerLocations$PlayerY

PlayerLocations <- PlayerLocations %>%
          group_by(GameEventID) %>%
          mutate(
            PasserY = PasserY[which.max(abs(PasserY))]
          )

#Number of defensive players behind the ball
PlayerLocations$DefBehBall <- ifelse(PlayerLocations$IsTM == 0
                      & PlayerLocations$PlayerX > PlayerLocations$PasserX,1,0)

#Number of attacking players ahead of the ball
PlayerLocations$AttAhead <- ifelse(PlayerLocations$IsTM == 1
                      & PlayerLocations$PlayerX > PlayerLocations$PasserX,1,0)
```

```
#Defensive Players in a Central position
PlayerLocations$Cent <- ifelse(PlayerLocations$PlayerY + 40 < 80/3, 0,
                                  ifelse(PlayerLocations$PasserY + 40 > 80-80/3, 0,1))

#Between Ball and Goal
PlayerLocations$GoalDef <- PlayerLocations$Cent*PlayerLocations$DefBehBall

#Attacking players in a central position ahead of the ball
PlayerLocations$AttCent <- PlayerLocations$Cent*PlayerLocations$AttAhead

#Remove Unneeded
PlayerLocations$Cent <- NULL
```

Another potentially influential variable in determining if a pass results in a goal is the presence of defenders pressuring the passer. A function to determine the Cartesian distance of defenders to the passer was used to determine pressure. Any defenders within 2 units of distance to the passer were recorded as pressuring the pass. This variable was then simplified to a binary variable indicating if the passer was under pressure or not.

```
#Cartesian distance formula
Distance <- function(x1,y1,x2,y2){
  sqrt((x2-x1)^2 + (y2-y1)^2)
}

#Distance of every player from the passer
PlayerLocations$Dist <- Distance(PlayerLocations$PlayerX,PlayerLocations$PlayerY
                                 ,PlayerLocations$PasserX,PlayerLocations$PasserY)

#Binay under pressure or not
PlayerLocations$Pressure <- as.numeric(PlayerLocations$IsTM == 0)*
                                  ifelse(PlayerLocations$Dist < 2,1,0)
```

```
Data <- PlayerLocations %>%
  group_by(GameEventID) %>%
  mutate(DefBehBall = cumsum(DefBehBall),AttAhead = cumsum(AttAhead),GoalDef =
           cumsum(GoalDef),AttCent = cumsum(AttCent),Pressure = cumsum(Pressure))

Data <- Data %>%
  group_by(GameEventID) %>%
  slice_tail()

#Remove duplicate variables
Data <- Data %>%
  dplyr::select(!c(BallX,BallY))
```

After the aggregated variables were constructed the data from the events dataset were matched by event IDs to create the final dataset used in the analysis. The constructed variable names and descriptions can be found in Table~1.

```
Data <- merge(Data,Events,by = "GameEventID")

#Binary factor for pressure
table(Data$Pressure)
```

```
##
##    0    1    2    3    4
## 2166 1275  105    1    1
```

```
Data$PressureFAC <- ifelse(Data$Pressure > 0,1,0)
Data$PressureFAC <- factor(Data$PressureFAC)

Data$ABSPasserY <- abs(Data$PasserY)
```

Table 1: Constructed Variables

| Label: | Description: |
|---|---|
| DefBehBall | Defensive players between the passer and goal |
| AttAhead | Attacking players ahead of the passer |
| GoalDef | Defensive players centrally between the passer and goal |
| AttCent | Attacking players centrally ahead of the passer |
| Pressure | Number of defensive players within two distance units of the passer |
| PressureFAC | (Binary) At least one defensive player within two distance units of the passer |
| ABSPasserY | Absolute lateral distance of the passer from the middle of the pitch |

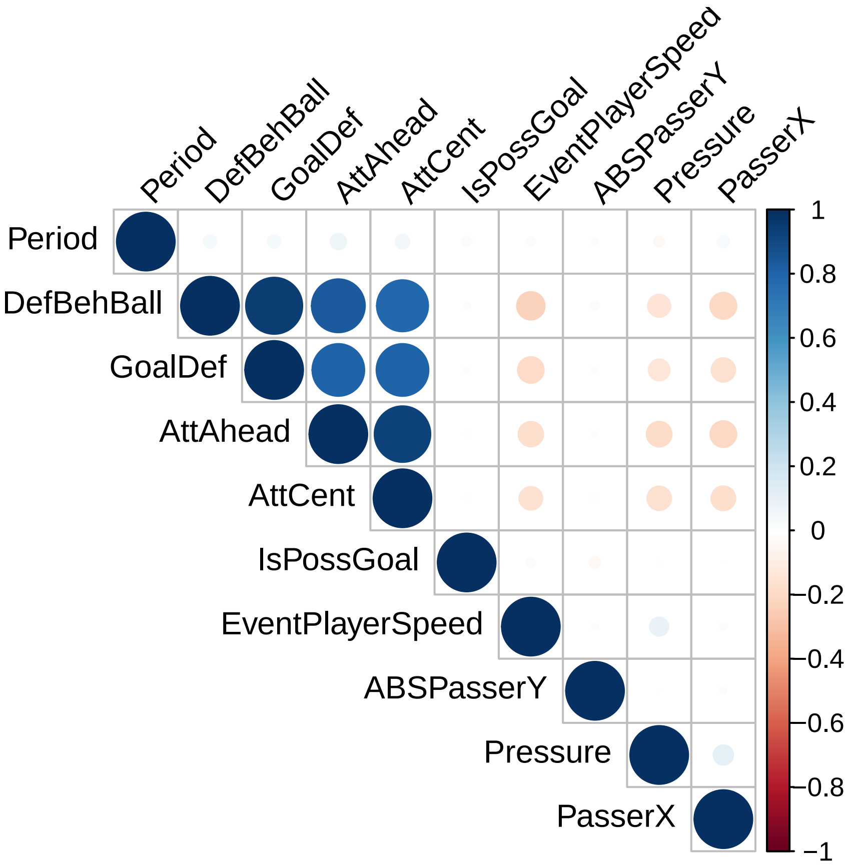
## Logistic Regression

Logistic regression was chosen for this task due to a binary outcome variable, the ease of interpretation of model estimates, and the overall computational simplicity. Probit regression could also be used for the same reasons. Alternative methods such as binary classifiers, random forests, or neural networks could be used for this task but they do not offer the same transparent interpretations of the impacts of specific variables on the outcome variable.

The linear correlations of the variables considered for the logistic regression was checked. There is some correlation between the number of attacking players ahead of the ball and the number of defenders behind the ball. This makes sense as defenders will try to mark and/or track the runs of attacking players.

```
CorDat <- Data[,c(24,10,11,12,13,15,17,23,8,27)]
res <- cor(CorDat)

corrplot(res, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



```
rm(CorDat) #Not needed
```

An initial logistic regression model including the number of defensive players centrally between the passer and goal (GoalDef), the number of attacking players centrally ahead of the passer (AttCent), if the passer was under pressure or not (PressureFAC), which half the game was in (Period), The speed of the passing player (EventPlayerSpeed), the passer's $x$ coordinate (PasserX), and the absolute lateral distance on the $y$ axis of the passer from the middle of the pitch (ABSPasserY). There was very little significance in this model suggesting that something was was missing.

Next, a logistic regression model including all of the variables from the first model and all of their interactions was estimated. This proved to be more informative as controlling for interactions in the variables yeilded significant results.

Finally, a logistic regression model including only the significant terms from the second model was estimated. Conclusions about the impacts of certain variables on the probability of scoring were drawn from this model.

### Initial Logistic Regression Estimates

```r
#BallY is an absolute distance from center
#

f1 <- glm(IsPossGoal~GoalDef+AttCent+PressureFAC+Period+EventPlayerSpeed+
            PasserX+ABSPasserY, family = binomial, data = Data)
#summary(f1) #Not Good...
f1%>%
  tidy() %>%
  kable(digits = 4)
```

| term | estimate | std.error | statistic | p.value |
|---|---:|---:|---:|---:|
| (Intercept) | -2.6545 | 0.7258 | -3.6573 | 0.0003 |
| GoalDef | -0.0224 | 0.0556 | -0.4023 | 0.6875 |
| AttCent | 0.0110 | 0.0665 | 0.1661 | 0.8680 |
| PressureFAC1 | -0.0178 | 0.1387 | -0.1283 | 0.8979 |
| Period | 0.2274 | 0.1352 | 1.6814 | 0.0927 |
| EventPlayerSpeed | 0.0480 | 0.0392 | 1.2234 | 0.2212 |
| PasserX | -0.0039 | 0.0237 | -0.1658 | 0.8683 |
| ABSPasserY | -0.0924 | 0.0456 | -2.0239 | 0.0430 |

### Logistic Regression Estimates with Interactions

```
f2 <- glm(IsPossGoal~(GoalDef+AttCent+PressureFAC+Period+EventPlayerSpeed+
                        PasserX+ABSPasserY)^2, family = binomial, data = Data)
f2%>%
  tidy() %>%
  kable(digits = 4)
```

| term | estimate | std.error | statistic | p.value |
|:---|---:|---:|---:|---:|
| (Intercept) | 5.0223 | 3.7059 | 1.3552 | 0.1753 |
| GoalDef | -1.4992 | 0.5347 | -2.8040 | 0.0050 |
| AttCent | 0.9935 | 0.6548 | 1.5172 | 0.1292 |
| PressureFAC1 | 0.1011 | 1.4232 | 0.0710 | 0.9434 |
| Period | -1.5829 | 1.4215 | -1.1135 | 0.2655 |
| EventPlayerSpeed | 0.1170 | 0.3855 | 0.3035 | 0.7615 |
| PasserX | -0.3047 | 0.1391 | -2.1908 | 0.0285 |
| ABSPasserY | 0.5829 | 0.4588 | 1.2705 | 0.2039 |
| GoalDef:AttCent | 0.0155 | 0.0177 | 0.8757 | 0.3812 |
| GoalDef:PressureFAC1 | -0.0890 | 0.1162 | -0.7660 | 0.4437 |
| GoalDef:Period | 0.0136 | 0.1166 | 0.1167 | 0.9071 |
| GoalDef:EventPlayerSpeed | -0.0304 | 0.0325 | -0.9347 | 0.3500 |
| GoalDef:PasserX | 0.0634 | 0.0195 | 3.2482 | 0.0012 |
| GoalDef:ABSPasserY | -0.0075 | 0.0389 | -0.1928 | 0.8471 |
| AttCent:PressureFAC1 | 0.1922 | 0.1407 | 1.3657 | 0.1720 |
| AttCent:Period | -0.0298 | 0.1395 | -0.2139 | 0.8306 |
| AttCent:EventPlayerSpeed | 0.0215 | 0.0413 | 0.5192 | 0.6036 |
| AttCent:PasserX | -0.0498 | 0.0234 | -2.1304 | 0.0331 |
| AttCent:ABSPasserY | 0.0143 | 0.0464 | 0.3088 | 0.7575 |
| PressureFAC1:Period | 0.3124 | 0.2865 | 1.0902 | 0.2756 |
| PressureFAC1:EventPlayerSpeed | -0.0425 | 0.0815 | -0.5208 | 0.6025 |
| PressureFAC1:PasserX | -0.0254 | 0.0496 | -0.5114 | 0.6091 |
| PressureFAC1:ABSPasserY | 0.0676 | 0.0951 | 0.7106 | 0.4774 |
| Period:EventPlayerSpeed | 0.1178 | 0.0821 | 1.4349 | 0.1513 |
| Period:PasserX | 0.0615 | 0.0499 | 1.2318 | 0.2180 |
| Period:ABSPasserY | -0.0766 | 0.0952 | -0.8050 | 0.4208 |
| EventPlayerSpeed:PasserX | -0.0041 | 0.0142 | -0.2899 | 0.7719 |
| EventPlayerSpeed:ABSPasserY | -0.0048 | 0.0275 | -0.1756 | 0.8606 |
| PasserX:ABSPasserY | -0.0227 | 0.0165 | -1.3780 | 0.1682 |

```
#summary(f2) #Better
```

### Final Logistic Regression Model Estimates

```
#Significant Terms from f2
f3 <- glm(IsPossGoal~(GoalDef+AttCent+PasserX+GoalDef:PasserX+AttCent:PasserX),
          family = binomial, data = Data)
#summary(f3) #This is the model!

f3%>%
  tidy() %>%
  kable(digits = 4)
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 4.9304 | 2.1815 | 2.2601 | 0.0238 |
| GoalDef | -1.6466 | 0.4699 | -3.5037 | 0.0005 |
| AttCent | 1.2163 | 0.5640 | 2.1565 | 0.0310 |
| PasserX | -0.2975 | 0.0882 | -3.3730 | 0.0007 |
| GoalDef:PasserX | 0.0652 | 0.0189 | 3.4535 | 0.0006 |
| AttCent:PasserX | -0.0484 | 0.0227 | -2.1267 | 0.0334 |

These estimates are presented in the *Log Odds* format, meaning that a positive coefficient indicates an increase in probability of a goal being scored and negative values indicate a decrease in probability of a goal being scored per unit change of the variables.

When controlling for interactions, an increase in the number of defenders centrally between the passer and the goal decreases the probability of the pass leading to a goal. Furthermore, an increase in attacking players centrally ahead of the ball increases the probability of the pass leading to a goal. Both of these results are intuitive from a soccer point of view.

As the passer gets closer to the goal the probability of the pass leading to a goal being scored seems to diminish. This could be because as the passer gets closer to the goal, the spaces between defenders and attackers is smaller, thus making it more difficult to score.

## Stepwise Estimation

A stepwise algorithm was used to estimate the best model based on the AIC. This methodology did not appear to be very informative.

```
#Let's try AIC with interactions and see how they compare

#Best models based on AIC

#null model
f0 = glm(formula = IsPossGoal ~ 1, family = binomial, data = Data)
#summary(f0)


#this adds and takes away lower is min scope and upper is max scope (included variables)
#AIC
f_step_aic <- step(f0, scope = list(upper = ~DefBehBall+AttAhead+GoalDef+
                                      AttCent+Pressure+Period+EventPlayerSpeed+
GoalDef:AttCent  +
GoalDef:PressureFAC   +
GoalDef:Period    +
GoalDef:EventPlayerSpeed   +
GoalDef:PasserX   +
GoalDef:ABSPasserY  +
AttCent:PressureFAC   +
AttCent:Period     +
AttCent:EventPlayerSpeed +
AttCent:PasserX +
AttCent:ABSPasserY  +
PressureFAC:Period +
PressureFAC:EventPlayerSpeed   +
PressureFAC:PasserX  +
PressureFAC:ABSPasserY +
Period:EventPlayerSpeed   +
Period:PasserX +
Period:ABSPasserY +
EventPlayerSpeed:PasserX  +
EventPlayerSpeed:ABSPasserY +
PasserX:ABSPasserY,lower = ~1),
                   trace = FALSE, #print all the steps
                   direction = "both")
#summary(f_step_aic) #This is not very insightful
```

## Random Slopes

A random slopes model was explored as there is a potential second source of variability from the player actually passing the ball. Although, there were too many passers with only a single observation to determine if the second source of variability was actually significant, the model was able to identify which player has the largest individual intercept. In other words, based on the random slopes model, player 315169 making the pass has the highest probability of a goal occurring.

```
#Significant Terms from f2

Data$EventPlayerIDFAC <- factor(Data$EventPlayerID)


RS1 <- glmer(IsPossGoal~GoalDef+AttCent+PasserX+GoalDef:PasserX+AttCent:PasserX +
               (1| EventPlayerIDFAC),
                family = binomial,data=Data, nAGQ=0)
#summary(RS1)

#plot_model(RS1,type = "re",show.values = F)
RandomSlopes <- coef(RS1)$EventPlayerIDFAC

RandomSlopes %>% slice_max(`(Intercept)`)
```

```
##        (Intercept)   GoalDef  AttCent    PasserX GoalDef:PasserX
## 315169    5.394758 -1.652713 1.232228 -0.2965355       0.0654981
##        AttCent:PasserX
## 315169     -0.04897277
```

## Upsampling

This technique is used to help balance the sample as goals being scored from the passes are rare. As only around 6.9% of the passes actually lead to goals, this is an example of a rare event. Sampling techniques like this are more commonly used for predictive models but may uncover some other variables of interest. By balancing the sample, more may be learned about the rare events, in this case actually scoring goals. A pseudo random sample of 80% was taken to evaluate the predictive powers of the model determined through upsampling.

```
set.seed(1234)
Data$IsPossGoal <- factor(Data$IsPossGoal)
index<-createDataPartition(Data$GameEventID,p=0.8,list=FALSE) #80% for training
```

```
train<-Data[index,]
test<-Data[-index,]


trainup<-upSample(x=train,
                  y=train$IsPossGoal)


table(trainup$IsPossGoal)
```

```
##
##    0    1
## 2649 2649
```

```
f4 <- glm(IsPossGoal~(GoalDef+AttCent+PressureFAC+Period+EventPlayerSpeed+
                        PasserX+ABSPasserY)^2, family = binomial, data = trainup)
#summary(f4) #Many more significant
```

The predictive capabilities of the model with the significant variables identified by upsampling and model with the significant variables identified before upsampling were compared. Both models for this part were trained on the upsampled training set. This was done to see if attention should be paid to the model identified through upsampling.

```
set.seed(1234)

#Old on training
#Significant Terms from f2
f5 <- glm(IsPossGoal~(GoalDef+AttCent+PasserX+GoalDef:PasserX+AttCent:PasserX),
          family = binomial, data = trainup)
#summary(f5)


#New On Upsampled
f6 <- glm(IsPossGoal~GoalDef+AttCent+PressureFAC+PasserX+ABSPasserY+
            GoalDef:PasserX+AttCent:PressureFAC+AttCent:EventPlayerSpeed+
            AttCent:PasserX+PressureFAC:Period+PressureFAC:ABSPasserY+
            Period:EventPlayerSpeed+Period:PasserX+Period:ABSPasserY+
            PasserX:ABSPasserY,family = binomial, data = trainup)
#summary(f6)




predOLD <- predict(f5,test, type="response")
predOLD <- as.integer(predOLD>0.5)
confusionMatrix(as.factor(predOLD),test$IsPossGoal)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 404  32
##          1 250  22
##
##                Accuracy : 0.6017
##                  95% CI : (0.5646, 0.638)
##     No Information Rate : 0.9237
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0088
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.61774
##             Specificity : 0.40741
##          Pos Pred Value : 0.92661
##          Neg Pred Value : 0.08088
##              Prevalence : 0.92373
##          Detection Rate : 0.57062
##    Detection Prevalence : 0.61582
##       Balanced Accuracy : 0.51257
##
##        'Positive' Class : 0
##
```

```
predNEW <- predict(f6,test, type="response")
predNEW <- as.integer(predNEW>0.5)
confusionMatrix(as.factor(predNEW),test$IsPossGoal)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 405  31
##          1 249  23
##
##                Accuracy : 0.6045
##                  95% CI : (0.5674, 0.6407)
##     No Information Rate : 0.9237
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0158
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.61927
##             Specificity : 0.42593
##          Pos Pred Value : 0.92890
##          Neg Pred Value : 0.08456
##              Prevalence : 0.92373
##          Detection Rate : 0.57203
##    Detection Prevalence : 0.61582
##       Balanced Accuracy : 0.52260
##
##        'Positive' Class : 0
##
```

## Interpretations

Directly comparing the predictive capabilities of both models based on their accuracy and Kappa values suggests that the model obtained from upsampling is slightly more accurate at predicting this test set than the model identified earlier.

### Model Estimates from Upsampling

```r
f6 %>%
  tidy() %>%
  kable(digits = 4)
```

| term | estimate | std.error | statistic | p.value |
|---|---:|---:|---:|---:|
| (Intercept) | 8.3489 | 1.3782 | 6.0577 | 0.0000 |
| GoalDef | -2.1897 | 0.2127 | -10.2931 | 0.0000 |
| AttCent | 2.3696 | 0.2558 | 9.2626 | 0.0000 |
| PressureFAC1 | -1.3865 | 0.2546 | -5.4466 | 0.0000 |
| PasserX | -0.3613 | 0.0554 | -6.5207 | 0.0000 |
| ABSPasserY | 0.6958 | 0.1750 | 3.9770 | 0.0001 |
| GoalDef:PasserX | 0.0886 | 0.0086 | 10.3225 | 0.0000 |
| AttCent:PressureFAC1 | 0.1107 | 0.0359 | 3.0819 | 0.0021 |
| AttCent:EventPlayerSpeed | -0.0212 | 0.0086 | -2.4609 | 0.0139 |
| AttCent:PasserX | -0.0937 | 0.0103 | -9.1049 | 0.0000 |
| PressureFAC0:Period | -1.2204 | 0.5260 | -2.3202 | 0.0203 |
| PressureFAC1:Period | -0.8704 | 0.5452 | -1.5966 | 0.1103 |
| PressureFAC1:ABSPasserY | 0.1260 | 0.0407 | 3.0931 | 0.0020 |
| EventPlayerSpeed:Period | 0.0997 | 0.0178 | 5.5915 | 0.0000 |
| PasserX:Period | 0.0603 | 0.0207 | 2.9138 | 0.0036 |
| ABSPasserY:Period | -0.1835 | 0.0397 | -4.6185 | 0.0000 |
| PasserX:ABSPasserY | -0.0216 | 0.0069 | -3.1101 | 0.0019 |

The model estimated from upsampling identifies the same significant variables as the earlier model. The numbers of defenders and attackers in central positions negatively and positively impact the probabilities of scoring respectively. The closer the passer is to the goal reduces the probability that the pass results in a goal.

This model also identified other potentially significant variables and interactions. Any pressure on the passer seems to significantly decrease the probability that the pass results in a goal. Furthermore, it seems that the further the passer is from the center of the $y$ axis, the more likely that pass is to result in a goal. A possible explanation for this is that angled passes may be more difficult for defenders to intercept than straight passes. It is important to interpret these results with some caution as the number of goals in the sample was artificially increased.

## Conclusions

When controlling for interactions, an increase in the number of defenders centrally between the passer and the goal decreases the probability of the pass leading to a goal. An increase in attacking players centrally ahead of the ball increases the probability of the pass leading to a goal. As the passer gets closer to the goal the probability of the pass leading to a goal being scored decreases. Any pressure on the passer may significantly decrease the probability that the pass results in a goal. Furthermore, it seems that the further the passer is from the center of the $y$ axis, the more likely that pass is to result in a goal.

Other information may be relevant to the probability of a pass resulting in a goal. As was touched on with the mixed effects modelling, who is passing the ball may greatly impact the outcome. Also, which defensive players are behind the ball and which attacking players are ahead of the ball may impact the probabilities.

The distance between the second to last defender and the goal could be of interest, as this could indicate how much space exists for the pass to be played into. A density of defenders, or the area of a convex hull around them could also be considered. All of these variables would probably be correlated with the $x$ position of the passer and may or may not be useful.

## Resources

Logistic Regression: https://www.r-bloggers.com/2015/09/how-to-perform-a-logistic-regression-in-r/

Mixed Effects Logistic Regression: https://stats.oarc.ucla.edu/r/dae/mixed-effects-logistic-regression/

Unbalanced Samples: https://www.r-bloggers.com/2019/04/methods-for-dealing-with-imbalanced-data/