Lesson 25 NFL Field Goals (continued) (6.3)

LTC J.K. Starling

```
# Load packages
library(tidyverse)

fg <- read.csv("https://raw.githubusercontent.com/jkstarling/MA376/main/fg_2021-2.csv",
               header=T, stringsAsFactors = TRUE)
fg <- fg %>%  mutate(field_goal_result = fct_rev(as.factor(field_goal_result)))

# setwd("C:/Users/james.starling/OneDrive - West Point/Teaching/MA376/JimsLessons_AY23-1/Block III/LSN2
# field_goal <- read.csv("fg_2021-2.csv", header = TRUE, stringsAsFactors = FALSE)
```

Background

(0) Last class we looked at the 2021 NFL Season and field goal kicks. In this lesson we will build our previous models.

We created a logistic regression model on the surface and distance.

```
contrasts(fg$surface)
```

```
##       turf
## grass    0
## turf     1
```

```
surface.glm <- fg %>% glm(field_goal_result ~ surface, data = ., family = 'binomial')
distance.glm <- glm(field_goal_result ~  kick_distance, data = fg, family = 'binomial')
summary(surface.glm)
```

```
##
## Call:
## glm(formula = field_goal_result ~ surface, family = "binomial",
##     data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0120   0.5323   0.5323   0.5844   0.5844
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.6808     0.1089  15.433   <2e-16 ***
## surfaceturf   0.2015     0.1781   1.131    0.258
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 897.79  on 1075  degrees of freedom
## Residual deviance: 896.49  on 1074  degrees of freedom
## AIC: 900.49
##
## Number of Fisher Scoring iterations: 4
```

```
summary(distance.glm)
```

```
##
## Call:
## glm(formula = field_goal_result ~ kick_distance, family = "binomial",
##     data = fg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8578   0.2061   0.3583   0.6103   1.5728
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     6.5466     0.5142   12.73   <2e-16 ***
## kick_distance  -0.1127     0.0111  -10.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 897.79  on 1075  degrees of freedom
## Residual deviance: 760.84  on 1074  degrees of freedom
## AIC: 764.84
##
## Number of Fisher Scoring iterations: 5
```

(1) We then took a look at the coefficients for each model and found the confidence intervals on those coefficients. How did we interpret these coefficients? What conclusions can we draw based on the confidence intervals?

```
coef(surface.glm)
```

```
## (Intercept) surfaceturf
##   1.6808279   0.2015285
```

```
confint(surface.glm)
```

```
##                 2.5 %    97.5 %
## (Intercept)  1.472349 1.8997523
## surfaceturf -0.144254 0.5551773
```

```
coef(distance.glm)
```

```
##   (Intercept) kick_distance
##     6.5465796    -0.1127346
```

```
confint(distance.glm)
```

```
##                    2.5 %      97.5 %
## (Intercept)    5.5771665  7.59618940
## kick_distance -0.1351956 -0.09162642
```

(2) We can also take the exponential of the coefficients and their CIs. Why would we do this? What is the interpretation of these results?

```
exp(coef(surface.glm))
```

```
## (Intercept) surfaceturf
##    5.370000    1.223271
```

```
exp(confint(surface.glm))
```

```
##                 2.5 %   97.5 %
## (Intercept) 4.3594623 6.684238
## surfaceturf 0.8656678 1.742250
```

```
exp(coef(distance.glm))
```

```
##   (Intercept) kick_distance
##   696.8565647     0.8933877
```

```
exp(confint(distance.glm))
```

```
##                     2.5 %      97.5 %
## (Intercept)   264.3215934 1990.596069
## kick_distance   0.8735451    0.912446
```

(3) Create a new logistic regression model that predicts the field goal result, taking into account the surface type and the kick distance. Discuss the results of this model.

```
surf.dist.glm <- glm(field_goal_result ~ surface + kick_distance, data=fg, family='binomial')
summary(surf.dist.glm)
```

```
##
## Call:
## glm(formula = field_goal_result ~ surface + kick_distance, family = "binomial",
##     data = fg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8377   0.2089   0.3628   0.5945   1.5316
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.48549    0.52038  12.463   <2e-16 ***
## surfaceturf    0.14362    0.19121   0.751    0.453
## kick_distance -0.11260    0.01111 -10.134   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 897.79  on 1075  degrees of freedom
## Residual deviance: 760.27  on 1073  degrees of freedom
## AIC: 766.27
##
## Number of Fisher Scoring iterations: 5
```

```
coef(surf.dist.glm)
```

```
##   (Intercept)   surfaceturf kick_distance
##     6.4854868     0.1436217    -0.1126032
```

```
confint(surf.dist.glm)
```

```
##                    2.5 %      97.5 %
## (Intercept)    5.5033623  7.54640190
## surfaceturf   -0.2283777  0.52238602
## kick_distance -0.1350950 -0.09146743
```

(4) Create a new logistic regression model that predicts the field goal result, taking into account the *interaction* between surface type and kick distance. Discuss the results of this model.

```
int.glm <- glm(field_goal_result ~ surface * kick_distance, data=fg, family='binomial')
summary(int.glm)
```

```
##
## Call:
## glm(formula = field_goal_result ~ surface * kick_distance, family = "binomial",
##     data = fg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8719   0.2105   0.3596   0.5914   1.4760
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 6.67714    0.68021   9.816  < 2e-16 ***
## surfaceturf                -0.31986    1.04167  -0.307    0.759
## kick_distance              -0.11680    0.01463  -7.982 1.43e-15 ***
## surfaceturf:kick_distance   0.01018    0.02252   0.452    0.651
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 897.79  on 1075  degrees of freedom
## Residual deviance: 760.07  on 1072  degrees of freedom
## AIC: 768.07
##
## Number of Fisher Scoring iterations: 5
```

```
coef(int.glm)                # Predicted log odds of making the field goal
```

```
##               (Intercept)                surfaceturf             kick_distance
##                 6.6771376                 -0.3198624                 -0.1167958
## surfaceturf:kick_distance
##                 0.0101827
```

```
confint(int.glm)
```

```
##                                 2.5 %      97.5 %
## (Intercept)                 5.41091047  8.08411849
## surfaceturf                -2.34793418  1.75568492
## kick_distance              -0.14676942 -0.08926685
## surfaceturf:kick_distance -0.03447005  0.05413523
```

(5) Calculate:

- the overall percentage rate for field goals made
- correct classification rate with `surface.lm`
- correct classification rate with `distance.lm`
- correct classification rate with `surf.dist.lm`
- correct classification rate with `int.lm`

(What do we need to specify in order to calculate the percentages/rates?)

```
calc.accuracy <- function(A){
  denom <- sum(A)
  TP <- 0; TN <- 0
  TP <- tryCatch(A["made","made.p"], silent=TRUE, error = function(ee) 0)
  TN <- tryCatch(A["missed","missed.p"], silent=TRUE, error = function(ee) 0)
  return ((TP + TN) / denom)
}

# calculate overall percentage
fg.attempts <- length(fg$field_goal_result) # should be 1076
fg.made <- sum(fg$field_goal_result == 'made')
overall.fg.rate <- fg.made / fg.attempts;
overall.fg.rate
```

```
## [1] 0.8531599
```

```
my.rate <- 0.5
# using surface model
fg.surf.pred <- ifelse(surface.glm$fitted.values >= my.rate, "made.p", "missed.p")
surf.tab <- table(fg$field_goal_result, fg.surf.pred)
surf.tab
```

```
##         fg.surf.pred
##          made.p
##   missed    158
##   made      918
```

```
fg.surf.rate <- calc.accuracy(surf.tab)
fg.surf.rate
```

```
## [1] 0.8531599
```

```
# using distance model
fg.dist.pred <- ifelse(distance.glm$fitted.values >= my.rate, "made.p", "missed.p")
dist.tab <- table(fg$field_goal_result, fg.dist.pred)
dist.tab
```

```
##         fg.dist.pred
##          made.p missed.p
##   missed    151        7
##   made      914        4
```

```
fg.dist.rate <- calc.accuracy(dist.tab)
fg.dist.rate
```

```
## [1] 0.855948
```

```r
# using surf + distance model
fg.sd.pred <- ifelse(surf.dist.glm$fitted.values >= my.rate, "made.p", "missed.p")
sd.tab <- table(fg$field_goal_result, fg.sd.pred)
fg.sd.rate <- calc.accuracy(sd.tab)
fg.sd.rate
```

```
## [1] 0.8550186
```

```r
# using surf * distance model
int.pred <- ifelse(int.glm$fitted.values >= my.rate, "made.p", "missed.p")
sd.tab <- table(fg$field_goal_result, int.pred)
int.rate <- calc.accuracy(sd.tab)
int.rate
```

```
## [1] 0.8531599
```

```r
c(fg.surf.rate, fg.dist.rate, fg.sd.rate, int.rate)
```

```
## [1] 0.8531599 0.8559480 0.8550186 0.8531599
```

```r
# cor(fg$field_goal_result, fg.sd.pred)
```

(6) How can we check overall model performance?

```r
my.rate <- 0.5
fgmade <- ifelse(fg$field_goal_result == 'made', 1, 0)
surf.made.p <- ifelse(unname(surface.glm$fitted.values) >= my.rate, 1, 0)
dist.made.p <- ifelse(unname(distance.glm$fitted.values) >= my.rate, 1, 0)
surfdist.made.p <- ifelse(unname(surf.dist.glm$fitted.values) >= my.rate, 1, 0)

c(sum(surf.made.p), sum(dist.made.p), sum(sum(dist.made.p)))
```

```
## [1] 1076 1065 1065
```

```r
cor(fgmade, surf.made.p)^2
```

```
## [1] NA
```

```r
cor(fgmade, dist.made.p)^2
```

```
## [1] 0.01975672
```

```
cor(fgmade, surfdist.made.p)^2
```

```
## [1] 0.01896988
```

```
\vspace{0.5in}
```

(7) In R, redo (5) and (6) above with various values for `my.rate`. What is the best choice for `my.rate` for maximizing the accuracy? for maximizing the R^2?