Report Code

Maysen Pagan

December 12, 2023

```r
#libraries
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(nnet))
suppressPackageStartupMessages(library(rstanarm))
suppressPackageStartupMessages(library(lme4))
suppressPackageStartupMessages(library(kableExtra))
suppressPackageStartupMessages(library(gridExtra))
suppressPackageStartupMessages(library(reshape2))
suppressPackageStartupMessages(library(flexmix))

#load data
pitch <- read.csv("~/Desktop/MSSP/Fall/MA 678/Final/pitch.csv")
#move pitcher and batter columns up one
pitch <- pitch %>% mutate_at(c("Pitcher"), tibble::lst("Pitcher"=lead), n = 1)
pitch <- pitch %>% mutate_at(c("Batter"), tibble::lst("Batter"=lead), n = 1)

pitch <- pitch[,-c(2,4)]
#remove NA rows
pitch <- pitch %>% na.omit()

#replace arrows of vertical and horizontal break with up, down, left, right
pitch$X.2 <- ifelse(pitch$X.2=="↓", "down", "up")
pitch$X.3 <- ifelse(pitch$X.3=="←", "left",
                    ifelse(pitch$X.3=="→", "right",""))
pitch <- pitch %>% select(-1)

#rename columns
data <- pitch
colnames(data) <- c("pitcher", "batter", "game.pitch", "pitcher.pitch",
                    "plate.app", "inning", "pitcher.result", "pitch.type",
                    "velo", "spin", "vbreak", "vbreak.direc", "hbreak",
                    "hbreak.direc")

#rename row names
row.names(data) <- seq(1:nrow(data))

head(data) %>% select(1:7) %>%
  kable(format = "latex",
        booktabs = TRUE,
        caption = "First 6 rows of dataset") %>%
  kable_styling(latex_options="scale_down") %>%
  kable_classic(html_font = "Cambria")
```

```r
head(data) %>% select(8:14) %>%
  kable(format = "latex", booktabs = TRUE) %>%
  kable_styling(font_size = 11) %>%
  kable_classic(html_font = "Cambria")

#add team name of pitcher to pitch data
sox <- c("Kenley Jansen", "Chris Martin", "Nick Pivetta", "Garrett Whitlock",
         "Josh Winckowski", "Brennan Bernardino",
         "Tanner Houck", "Brayan Bello", "Corey Kluber", "Joe Jacques",
         "Chris Murphy", "Kaleb Ort", "Mauricio Llovera",
         "John Schreiber", "Kutter Crawford", "Nick Robertson", "Zack Weiss",
         "Brandon Walter")
yanks <- c("Jimmy Cordero", "Nick Ramirez", "Albert Abreu", "Gerrit Cole",
           "Clay Holmes", "Tommy Kahnle",
           "Wandy Peralta", "Domingo German", "Ron Marinaccio", "Michael King",
           "Clarke Schmidt", "Isiah Kiner-Falefa",
           "Matt Krook", "Greg Weissert", "Luis Severino", "Keynan Middleton",
           "Ian Hamilton", "Jhony Brito",
           "Jonathan Loaisiga", "Randy Vasquez", "Matt Bowman", "Anthony Misiewicz",
           "Zach McAllister", "Carlos Rodon")

data$team <- ifelse(data$pitcher%in%sox, "Red Sox", "Yankees")
data <- data %>% select('team', everything())

#team the batter is on
#if pitcher is on red sox, batter is on yankees
#exit$Team <- ifelse(exit$Pitcher%in%sox, "Yankees", "Red Sox")

#SWINGING STRIKE AND CALLED STRIKE WILL BE THE STRIKES, EVERYTHING
#ELSE IS NOT A STRIKE
#create new binary column `result` that is 1 if strike and 0 if not
data <- data %>% mutate(result = ifelse(data$pitcher.result == "Swinging Strike" |
                                          data$pitcher.result == "Called Strike" |
                                          data$pitcher.result == "Foul" |
                                          data$pitcher.result == "Foul Tip" |
                                          data$pitcher.result == "Foul Bunt" |
                                          data$pitcher.result == "Missed Bunt",
                                        1, 0)) %>%
  select("team", "pitcher", "batter", "result", everything())

#data only for red sox pitchers
sox <- data %>% filter(team=="Red Sox")
#add in red sox pitchers' WAR
pwar <- read.csv("~/Desktop/MSSP/Fall/MA 678/Final/pitcherWAR.csv", header = TRUE)
pwar <- pwar %>% select(2,7)
colnames(pwar) <- c("pitcher", "pwar")

sox <- merge(sox, pwar, by = "pitcher") %>%
  select("team", "pitcher", "pwar", everything())

#add in yankees batters' WAR
bwar <- read.csv("~/Desktop/MSSP/Fall/MA 678/Final/bwar.csv", header = TRUE)
bwar <- bwar %>% select(2,6)
```

```r
colnames(bwar) <- c("batter", "bwar")

sox <- merge(sox, bwar, by = "batter") %>%
  select("team", "pitcher", "pwar", "batter", "bwar", everything())

#collapse sox data to have unique pitcher, batter combos
new_data <- sox %>% group_by(pitcher, batter) %>%
  summarize(count = n(),
            pwar = mean(pwar),
            bwar = mean(bwar),
            prop = sum(result/n()),
            velo = mean(velo),
            spin = mean(spin),
            vbreak = mean(vbreak),
            hbreak = mean(hbreak)) %>%
  select(-count)

#distribution of proportion of pitches thrown that were strikes,
#for each pitcher batter combo
# hist(new_data$prop, xlab = "Proportion of Strikes Thrown")
#distribution is approximately symmetric so make threshold 50%

#if proportion of pitches thrown is above 0.5, then pitcher has a high
#proportion of pitches thrown
new_data$high <- ifelse(new_data$prop>=0.50, 1, 0)


cormat <- round(cor(new_data[,c(3:4, 6:9)]),2)
melted_cormat <- melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  labs(x = NULL, y = NULL) +
  theme(axis.text = element_text(size = 14),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 14))

new_data %>% ggplot(aes(factor(high), bwar, fill = factor(high))) +
  geom_violin(alpha = 0.5) +
  scale_fill_manual(values = c("darkgoldenrod2", "deeppink3")) +
  geom_jitter() +
  labs(x = "High-Strike", y = "Batter's WAR") +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14),
        legend.position = "none")

new_data %>% ggplot(aes(factor(high), bwar, fill = factor(high))) +
  facet_wrap(~pitcher) +
  geom_violin(alpha = 0.5) +
  scale_fill_manual(values = c("darkgoldenrod2", "deeppink3")) +
  labs(x = "High-Strike", y = "Batter's WAR") +
```

```
  theme_bw() +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14),
        legend.position = "none")

#proportion of times pitchers had high strikes thrown for each pitcher
new_data %>% group_by(pitcher) %>%
  summarize(high_prop = mean(high)) %>%
  ggplot(aes(x = reorder(pitcher, -high_prop), high_prop)) +
  geom_point(aes(color = pitcher), size = 4) +
  theme_bw() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(x = "Pitcher", y = "Proportion of High-Strikes") +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14),
        legend.position = "none")

set.seed(100)
sample <- sample(c(TRUE, FALSE), nrow(new_data), replace=TRUE, prob=c(0.8,0.2))
training  <- new_data[sample, ]
testing   <- new_data[!sample, ]

set.seed(100)
mod1 <- stan_glm(high~1, family = binomial(link = "logit"),
                 data = training,
                 refresh = 0,
                 iter = 1000)
print(mod1, digits = 4)

set.seed(100)
mod2 <- stan_glm(high ~ bwar + velo + spin + hbreak,
                 family = binomial(link = "logit"),
                 data = training,
                 refresh = 0,
                 iter = 1000)
print(mod2, digits = 4)

#partial pooling, varying intercepts
mod3 <- glmer(high ~ bwar + velo + spin + hbreak + (1+bwar|pitcher) +
                (1+velo|pitcher) + (1+spin|pitcher) + (1+hbreak|pitcher),
              data = training, family = binomial,
              control=glmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e4)))

#partial pooling plus group level predictor pwar
mod4 <- glmer(high ~ bwar + velo + spin + hbreak + pwar + (1+bwar|pitcher) +
                (1+velo|pitcher) + (1+spin|pitcher) + (1+hbreak|pitcher) +
                (1+pwar|pitcher),
              data = training,
              family = binomial,
              control=glmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e4)))

cat("Without group-level AIC:", extractAIC(mod3)[2])
```

```r
cat("With group-level AIC:", extractAIC(mod4)[2])
cat("Without group-level AIC:", BIC(mod3))
cat("With group-level AIC:", BIC(mod4))

#no pooling
pitchers <- unique(training$pitcher)
no_pooling_coefs <- rep(list(list()), length(pitchers))
no_pooling_mods <- rep(list(list()), length(pitchers))
no_pool <- function(x){# x is name of pitcher to filter by
  set.seed(100)
  return(stan_glm(high ~ bwar + velo + spin + hbreak,
                  family = binomial(link = "logit"),
                  data = subset(training, pitcher == x),
                  refresh = 0,
                  iter = 1000))
}
for (i in 1:length(pitchers)){
  mods <- no_pool(pitchers[i])
  coefs <- coef(no_pool(pitchers[i]))
  no_pooling_mods[[i]] <- mods
  no_pooling_coefs[[i]] <- coefs
}

models <- do.call(rbind, no_pooling_coefs)
row.names(models) <- pitchers

# #misclassification errors
misclass <- function(mod, p){
  fitted <- ifelse(posterior_epred(mod, newdata = testing, type = "response")>p, 1, 0)
  misclasses <- vector("numeric", nrow(fitted))
  for (i in 1:nrow(fitted)){
    prop <- (sum(as.vector(fitted[i,])!=testing$high))/(length(testing$high))
    misclasses[i] <- prop
  }
  return(mean(misclasses))
}

# #model1
m1error <- misclass(mod1, 0.40)
#
# #model2
m2error <- misclass(mod2, 0.40)
#
# #model3
fitted <- ifelse(predict(mod3, newdata = testing, type = "response")>0.40, 1, 0)
(m3error <- sum(testing$high!=fitted)/nrow(testing))
#
# #model5
fitted <- vector("numeric", nrow(testing))
for(i in 1:nrow(testing)){
  new_obsv <- testing[i,]
  name <- new_obsv$pitcher
  mod_index <- which(name==pitchers)
```

```r
  mod <- no_pooling_mods[[mod_index]]
  prediction <- ifelse(predict(mod, newdata = new_obsv, type = "response")>0.40, 1, 0)
  fitted[i] <- prediction
}
m5error <- sum(testing$high!=fitted)/nrow(testing)

df <- data.frame(Model = c("Null",
                           "Complete Pooling",
                           "Partial Pooling",
                           "No Pooling"),
                 `Misclassification Error` = c(round(m1error,4), round(m2error,4),
                                               round(m3error,4), round(m5error,4)))

df %>%
  kable(format = "latex", booktabs = TRUE, caption = "Misclassification Errors for 4 Models") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed")) %>%
  kable_classic(html_font = "Cambria")

#predictions
#kenley jansen
coefs <- c(-7.189097, 0.32173660, 0.05360815, 0.0010302585, 0.02228718)
new <- c(1, 1.7, 93, 2300, 5)
phat <- 1/(1+exp(-coefs%*%new)) #0.7370
#Chris Martin
coefs <- c(-8.103947, -0.04550731, 0.05328095, 0.0009696635, 0.03833361)
new <- c(1, 1.7, 93, 2300, 5)
phat <- 1/(1+exp(-coefs%*%new)) #0.3520

#appendix
hist(new_data$prop, xlab = "Proportion of Strikes Thrown", main = NULL)

coef(mod3)$pitcher %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Coefficients for partial pooling models by pitcher") %>%
  kable_styling(latex_options="scale_down",
                bootstrap_options = c("striped", "hover", "condensed")) %>%
  kable_classic(html_font = "Cambria")

coef(mod4)$pitcher %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Coefficients for partial pooling models with
        group-level predictor by pitcher") %>%
  kable_styling(latex_options="scale_down",
                bootstrap_options = c("striped", "hover", "condensed")) %>%
  kable_classic(html_font = "Cambria")

models %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Coefficients for no pooling models by pitcher") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed")) %>%
  kable_classic(html_font = "Cambria")
```