Do Penalty Shot Expected Goals Models Mean Anything?

—

2023-07-21

Do penalty shot expected goals models mean anything…at all? We all know that expected goals models can be pretty accurate - you input some data about the shot distance, angle, off the rush, and more, and can spit out a probability of a goal using

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

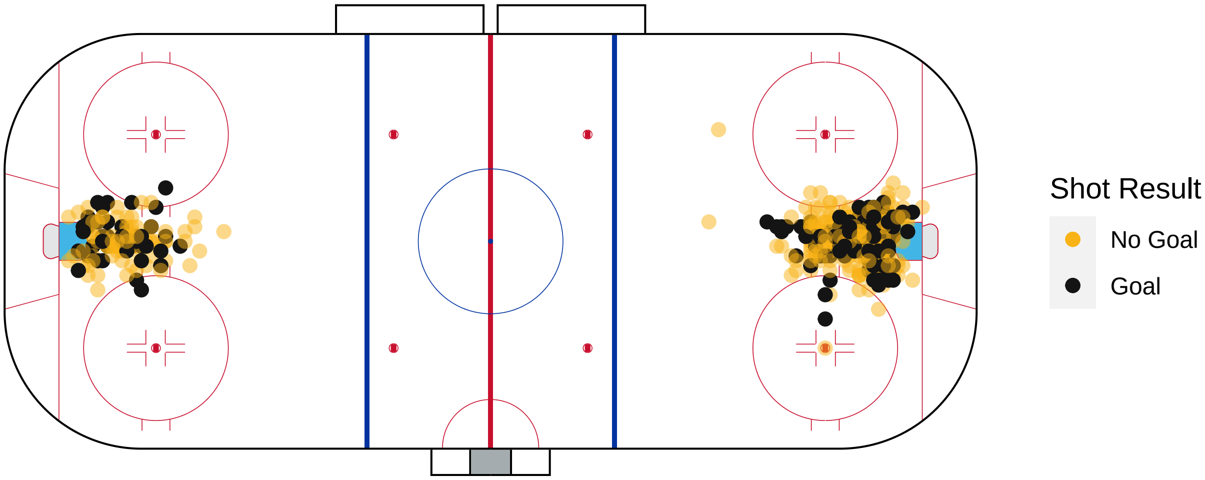
```
library(sportyR)
library(ggplot2)
library(hockeyR)
library(hexbin)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
LogLoss<-function(actual, predicted)
{
  result=-1/length(actual)*(sum((actual*log(predicted)+(1-actual)*log(1-predicted))))
  return(result)
}

library(ggimage)

#list.files("C:/Users/matth/OneDrive/Desktop/XG Model/",pattern="png", full.names=TRUE)

logos_colors <- team_logos_colors %>% mutate(filename = paste0(substr(team_logo_espn, 43, 45), ".png",

for (i in 1:nrow(team_logos_colors)){
  url <- team_logos_colors[i, ]$team_logo_espn
  name <- paste0(substr(url, 43, 45), ".png", '')
  if (!file.exists(name)){
    download.file(url, name, mode = "wb")
  }
}


#shots <- fread("shots_2007-2021.csv")
ps <- read.csv(file = "penalty_shots_10_23.csv") %>%
  filter(abs(x) >= 5) %>%
  mutate(goal = ifelse(event == 'Goal', 1, 0),
         event_player_2_name = ifelse(is.na(event_player_2_name),
                                      ifelse(event_team_type == 'home', away_goalie, home_goalie)
                                      ,event_player_2_name),
         strength_code = as.factor(strength_code))

ps[(ps$event_player_1_name == 'Connor.McDavid') & is.na(ps$event_player_2_name),]$event_player_2_name <
ps <- ps %>% left_join(logos_colors, by = c("event_team_abbr" = "team_abbr"))
```

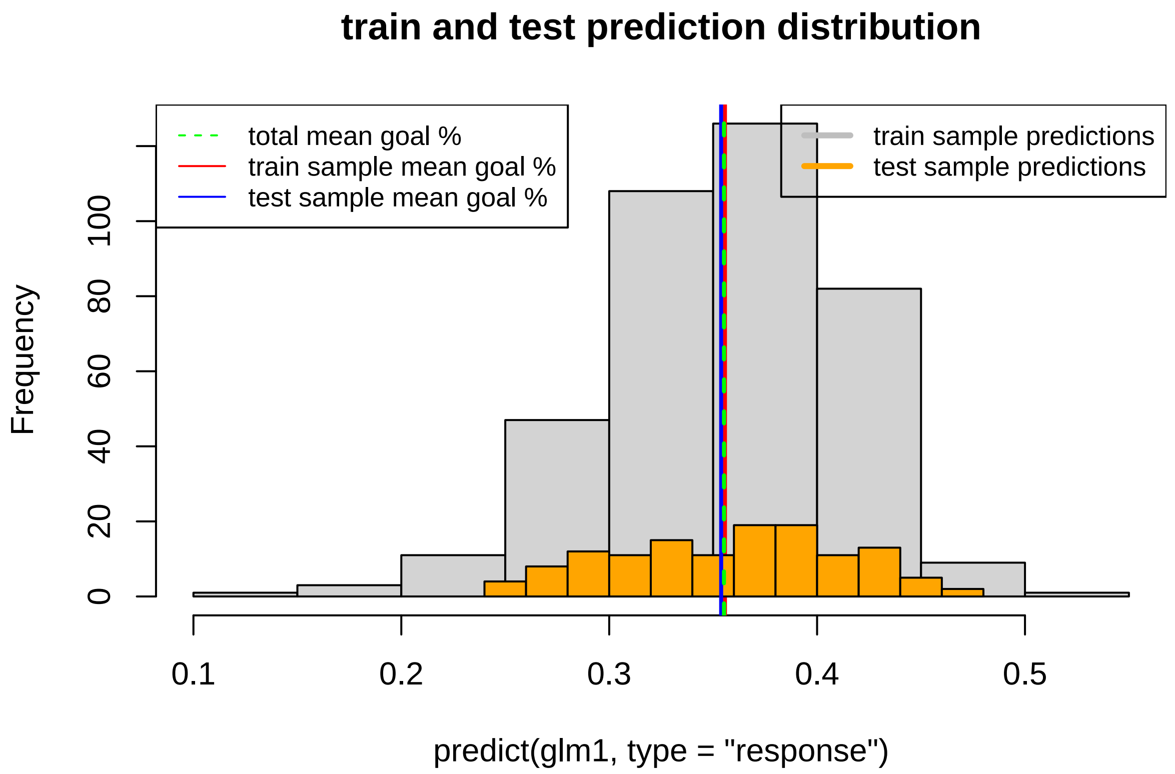
```
set.seed(37)
train_rows <- sample(nrow(ps), 0.75*nrow(ps))
train <- ps[train_rows, ] %>% select(shot_distance, shot_angle, goal)
train_viz <- ps[train_rows, ] %>% select(x, y, goal)

test <- ps[-train_rows, ] %>% select(shot_distance, shot_angle, goal)
test_viz <- ps[-train_rows, ] %>% select(x, y, goal)

glm1 <- glm(goal~., train, family = binomial)


geom_hockey("nhl") +
  geom_point(data = train_viz,
             aes(abs(x), y, color = as.factor(goal)),
             size = 2,
             alpha = ifelse(train_viz$goal == 1, 1, 0.5)) +
  geom_point(data = test_viz,
             aes(-abs(x), y, color = as.factor(goal)),
             size = 2,
             alpha = ifelse(test_viz$goal == 1, 1, 0.5)) +
  scale_color_manual(values = c("#f9b316", "#141414"),name = "Shot Result", breaks = c(0,1), labels = c
```



```
options(scipen=999)

create_viz_data_glm <- function(dta = grid){
  dta$shot_angle <- (atan(abs(dta$y)/(dta$x)) * 180) / pi
  dta$shot_distance <- sqrt((89-dta$x)^2 + abs(dta$y)^2)
  mat <- dta %>% select(shot_distance, shot_angle)
  return(mat)
}


x_viz <- seq(25, 89, .1)
y_viz <- seq(-40, 40, .1)
grid <- expand.grid(list(x = x_viz, y = y_viz))


summary(glm1)
```

```
##
## Call:
## glm(formula = goal ~ ., family = binomial, data = train)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.186988   0.359517   0.520   0.6030
## shot_distance -0.037407   0.020179  -1.854   0.0638 .
## shot_angle    -0.015650   0.007524  -2.080   0.0375 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 505.09  on 387  degrees of freedom
## Residual deviance: 498.98  on 385  degrees of freedom
## AIC: 504.98
##
## Number of Fisher Scoring iterations: 4
```

```
hist(predict(glm1, type='response'), main = 'train and test prediction distribution')
abline(v = mean(train$goal), col = 'red', lwd = 2)
abline(v = mean(ps$goal), col = 'green', lwd = 2, lty='dashed')
hist(predict(glm1, test, type='response'), main = 'test prediction distribution', add = T, col= 'orange
abline(v = mean(test$goal), col = 'blue', lwd = 2)
abline(v = mean(ps$goal), col = 'green', lwd = 2, lty='dashed')
legend("topleft", legend=c("total mean goal %", "train sample mean goal %", 'test sample mean goal %'),
       col=c("green", "red", "blue"), lty=c(2, 1, 1), cex=0.8)
legend("topright", legend=c("train sample predictions", "test sample predictions"),
       col=c("grey", "orange"), lty=1, cex=0.8, lwd = 3)
```



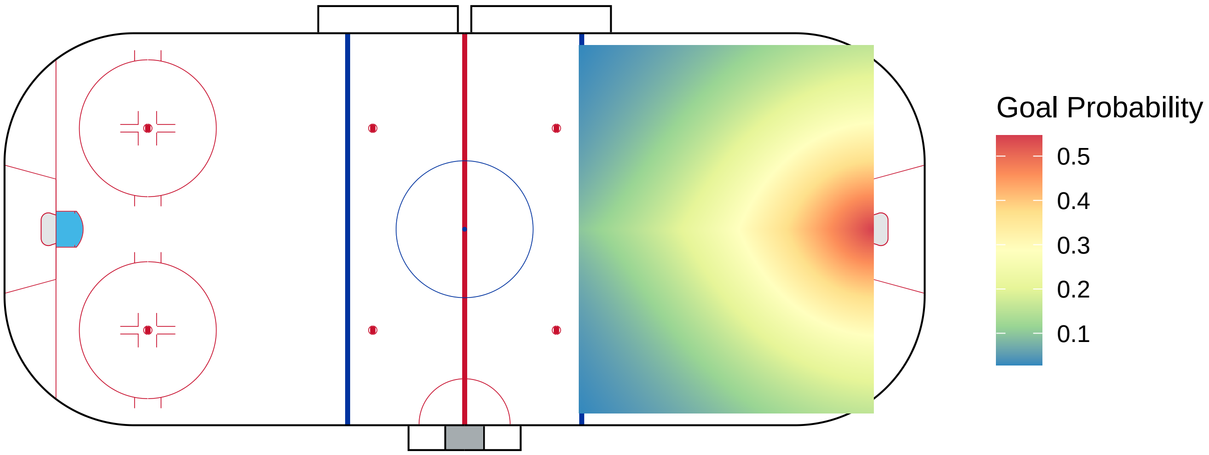
```
LogLoss(train$goal, predict(glm1, type = 'response'))
```

```
## [1] 0.6430112
```

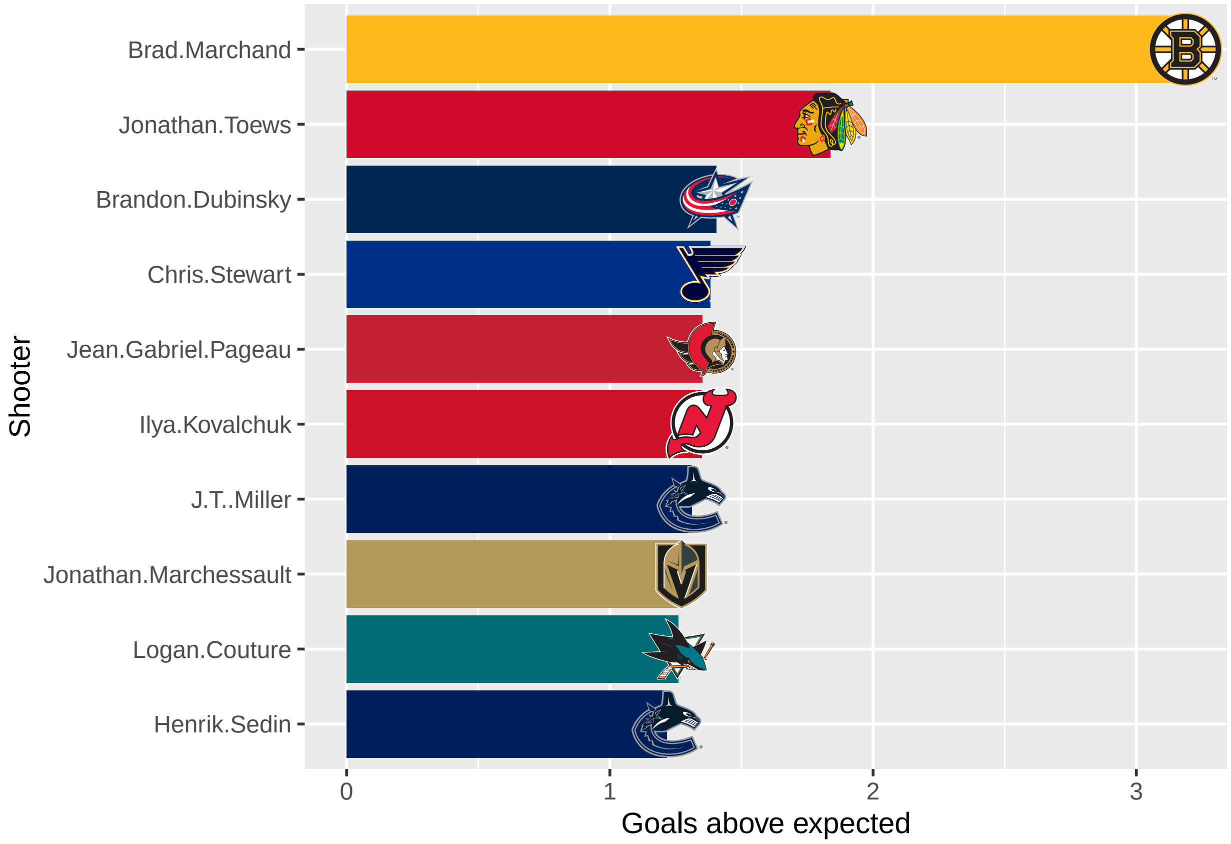
```
LogLoss(test$goal, predict(glm1, newdata = test, type = 'response'))
```

```
## [1] 0.6508571
```

```
viz_predictions <- predict(glm1, create_viz_data_glm(), type='response')

viz_data <- cbind(grid, viz_predictions)
geom_hockey("NHL") +
  geom_raster(data = viz_data, aes(x, y, fill=viz_predictions)) +
  scale_fill_distiller(palette = "Spectral", direction = -1) +
  labs(fill = "Goal Probability")
```



```
ps$xG <- predict(glm1, ps, type='response')

best <- ps %>% group_by(event_player_1_name) %>% summarize(tot_xG = sum(xG), tot_G = sum(goal), GaX = t

#worst <- ps %>% group_by(event_player_1_name) %>% summarize(tot_xG = sum(xG), tot_G = sum(goal), GaX =

best %>% ggplot(aes(x = reorder(event_player_1_name, +GaX), y = GaX, fill = team_col)) +
  geom_bar(stat = 'identity') +
  coord_flip() +
  scale_fill_identity() +
  geom_image(aes(image = logo), size = 0.1) +
  xlab("Shooter") +
  ylab("Goals above expected")
```



```
ps %>% group_by(event_player_1_name) %>% summarize(tot_xG = sum(xG), tot_G = sum(goal), GaX = tot_G - t
```

```
## # A tibble: 10 x 7
##    event_player_1_name   tot_xG tot_G     GaX   att team_col logo
##    <chr>                  <dbl> <dbl>   <dbl> <int> <chr>    <chr>
##  1 Brad.Marchand          2.81      6  3.19       8 #FFB81C  BOS.png
##  2 Andreas.Athanasiou     1.95      3  1.05       5 #CE1126  DET.png
##  3 Jonathan.Marchessault  1.73      3  1.27       5 #B4975A  VGK.png
##  4 Jonathan.Toews         1.16      3  1.84       3 #CF0A2C  CHI.png
##  5 Antoine.Roussel        1.38      2  0.619      4 #006847  DAL.png
##  6 Boone.Jenner           1.00      2  0.998      3 #002654  CBJ.png
##  7 Brandon.Dubinsky       0.595     2  1.41       2 #002654  CBJ.png
##  8 Cam.Atkinson           1.95      2  0.0481     5 #002654  CBJ.png
##  9 Chris.Stewart          0.618     2  1.38       2 #002F87  STL.png
## 10 Connor.McDavid         2.22      2 -0.219      6 #FF4C00  EDM.png
```