HW7

Sandra Villamar and Shobhit Dronamraju

2023-03-07

Problem 1A

Fit a logistic regression model with good as response and distance as predictor. Interpret the fitted model coefficients and visualize the model fit.

```
dat = read.csv("../data/Placekick.csv")
fit = glm(good ~ distance, family = binomial(link = "logit"), data = dat)
fit$coefficients
```

```
## (Intercept)    distance
##   5.8120798  -0.1150267
```

```
exp(fit$coefficients)
```

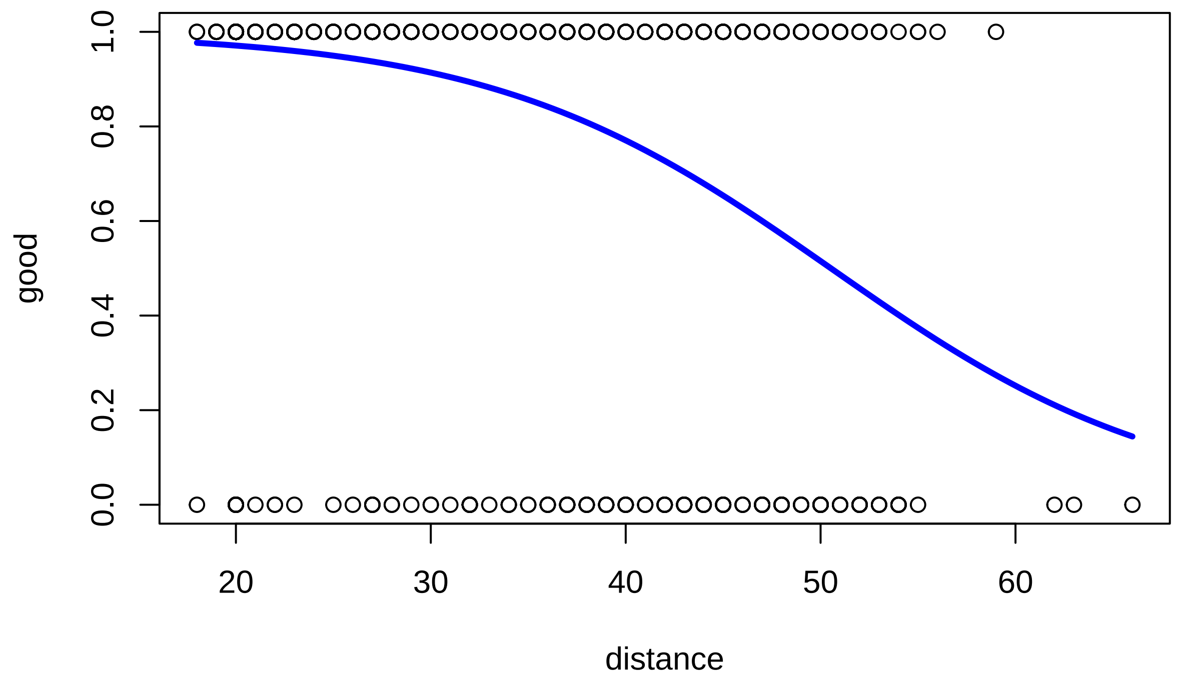
```
## (Intercept)    distance
## 334.3136943   0.8913424
```

In logistic regression, the coefficients represent the change in the log odds of the response variable (i.e., the dependent variable) for a one-unit increase in the corresponding predictor variable (i.e., the independent variable), holding all other predictor variables constant.

In this case, for a one unit increase in distance, the odds increases by a factor of $e^{-0.1150267} \approx 0.891$. This is about an 11% **decrease** in the odds of y i.e. the success of a placekick.

The intercept in this model corresponds to the probability of the success of a placekick when distance is zero. This obviously doesn't make quite sense as distance zero means you are already in the goal.

```
# visualize the model fit
plot(dat$distance, dat$good, xlab = "distance", ylab = "good")
curve(expr = predict(object = fit,
                     newdata = data.frame(distance = x),
                     type = "response"),
      col = "blue", add= TRUE, lwd=3)
```



From the plot we see that at around 50 yards, the success of a placekick tapers off below 50%. This is intuitive as the farther away one is from the goal, the harder it is to make the goal.

Problem 1B

Now consider all predictors. Apply the forward selection algorithm to compute the forward selection path from 'intercept only' to 'full model' and chooses the model on that path that minimizes the AIC.

```
m0 = glm(good ~ 1, family = binomial(link = "logit"), data = dat)
m1 =  glm(good ~ ., family = binomial(link = "logit"), data = dat)
model.forward = step(m0,scope = formula(m1), direction="forward")
```

```
## Start:  AIC=1015.43
## good ~ 1
##
##            Df Deviance     AIC
## + distance  1   775.75  779.75
## + PAT       1   834.41  838.41
## + change    1   989.15  993.15
## + elap30    1  1007.71 1011.71
## + wind      1  1010.59 1014.59
## + week      1  1011.24 1015.24
## + type      1  1011.39 1015.39
## <none>         1013.43 1015.43
## + field     1  1012.98 1016.98
```

```
##
## Step:  AIC=779.75
## good ~ distance
##
##          Df Deviance    AIC
## + PAT     1   762.41 768.41
## + change  1   770.50 776.50
## + wind    1   772.53 778.53
## <none>        775.75 779.75
## + week    1   773.86 779.86
## + type    1   775.67 781.67
## + elap30  1   775.68 781.68
## + field   1   775.74 781.74
##
## Step:  AIC=768.41
## good ~ distance + PAT
##
##          Df Deviance    AIC
## + change  1   759.33 767.33
## + wind    1   759.66 767.66
## <none>        762.41 768.41
## + week    1   760.57 768.57
## + type    1   762.25 770.25
## + elap30  1   762.31 770.31
## + field   1   762.41 770.41
##
## Step:  AIC=767.33
## good ~ distance + PAT + change
##
##          Df Deviance    AIC
## + wind    1   756.69 766.69
## + week    1   757.26 767.26
## <none>        759.33 767.33
## + elap30  1   759.11 769.11
## + type    1   759.13 769.13
## + field   1   759.33 769.33
##
## Step:  AIC=766.69
## good ~ distance + PAT + change + wind
##
##          Df Deviance    AIC
## <none>        756.69 766.69
## + week    1   755.07 767.07
## + type    1   756.06 768.06
## + elap30  1   756.43 768.43
## + field   1   756.66 768.66
```

```
summary(model.forward)
```

```
##
## Call:
## glm(formula = good ~ distance + PAT + change + wind, family = binomial(link = "logit"),
##     data = dat)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9158   0.1694   0.1694   0.4729   1.3818
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.75157    0.47200  10.067  < 2e-16 ***
## distance    -0.08724    0.01112  -7.847 4.26e-15 ***
## PAT          1.22992    0.38474   3.197  0.00139 **
## change      -0.33505    0.19335  -1.733  0.08312 .
## wind        -0.52344    0.31315  -1.672  0.09462 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1013.43  on 1424  degrees of freedom
## Residual deviance:  756.69  on 1420  degrees of freedom
## AIC: 766.69
##
## Number of Fisher Scoring iterations: 6
```

After forward selection, the resulting model that minimizes AIC is $good \sim distance + PAT + change + wind$.

Problem 1C

Consider the model selected by the forward selection algorithm. Compute the decision boundary when the decision threshold for the probability of success is 0.5.

The probability $P = \frac{exp(\beta^T X)}{1+exp(\beta^T X)}$ equals 0.5 when $\beta^T X = 0$. Hence, the decision boundary is $\beta^T X = 0$ where β is the vector of coefficients and the first entry of X is 1.

```
model.forward$coefficients
```

```
## (Intercept)    distance         PAT      change        wind
##  4.75156983 -0.08723696  1.22991739 -0.33505071 -0.52343574
```

Decision Boundary:

$$0 = 4.752 - 0.087x_{distance} + 1.230x_{PAT} - 0.335x_{change} - 0.523x_{wind}$$

Problem 2A

Write a function bootGLM(x, y, B=1000) that resamples observations and returns standard errors for each of the predictor variables (when the others are present in the model) in a logistic model.

```
bootGLM <- function(x, y, B=1000){

  n = dim(x)[1]  # num of observations
  p = dim(x)[2]  # num of predictors

  coefs = matrix(NA, B, p + 1)  # store model coefs for all B trials
  for (i in 1:B) {
    indices = sample(1:n, n, replace=TRUE)
    x_boot = x[indices,]
    y_boot = y[indices]
    dat = data.frame(x_boot, y_boot)
    model = glm(y_boot ~ ., family = binomial(link = "logit"), data = dat)
    coefs[i,] = model$coefficients
  }

  colnames(coefs) = names(model$coefficients)
  se = apply(coefs, 2, sd)
  return(se)
}
```

Problem 2B

Consider the model selected by the forward selection algorithm from Problem 1B. Apply your bootGLM, and compare with the standard errors returned by the summary function.

```
# model.forward: good ~ distance + PAT + change + wind
x = dat[,names(dat) %in% c("distance", "PAT", "change", "wind")]
y = dat[,9]
se = bootGLM(x, y)
se
```

```
## (Intercept)    distance      change         PAT        wind
##  0.47761650  0.01102431  0.20000154  0.40191076  0.29900427
```

```
summary(model.forward)$coefficients
```

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)  4.75156983 0.47199923 10.066902 7.737737e-24
## distance    -0.08723696 0.01111711 -7.847089 4.258063e-15
## PAT          1.22991739 0.38474229  3.196730 1.389948e-03
## change      -0.33505071 0.19334821 -1.732888 8.311564e-02
## wind        -0.52343574 0.31315327 -1.671500 9.462293e-02
```

The standard errors given by bootGLM are very close to those returned by the summary function.