HW7

2024-02-29

code : https://github.com/megurukiss/ds241/blob/main/HW7.Rmd

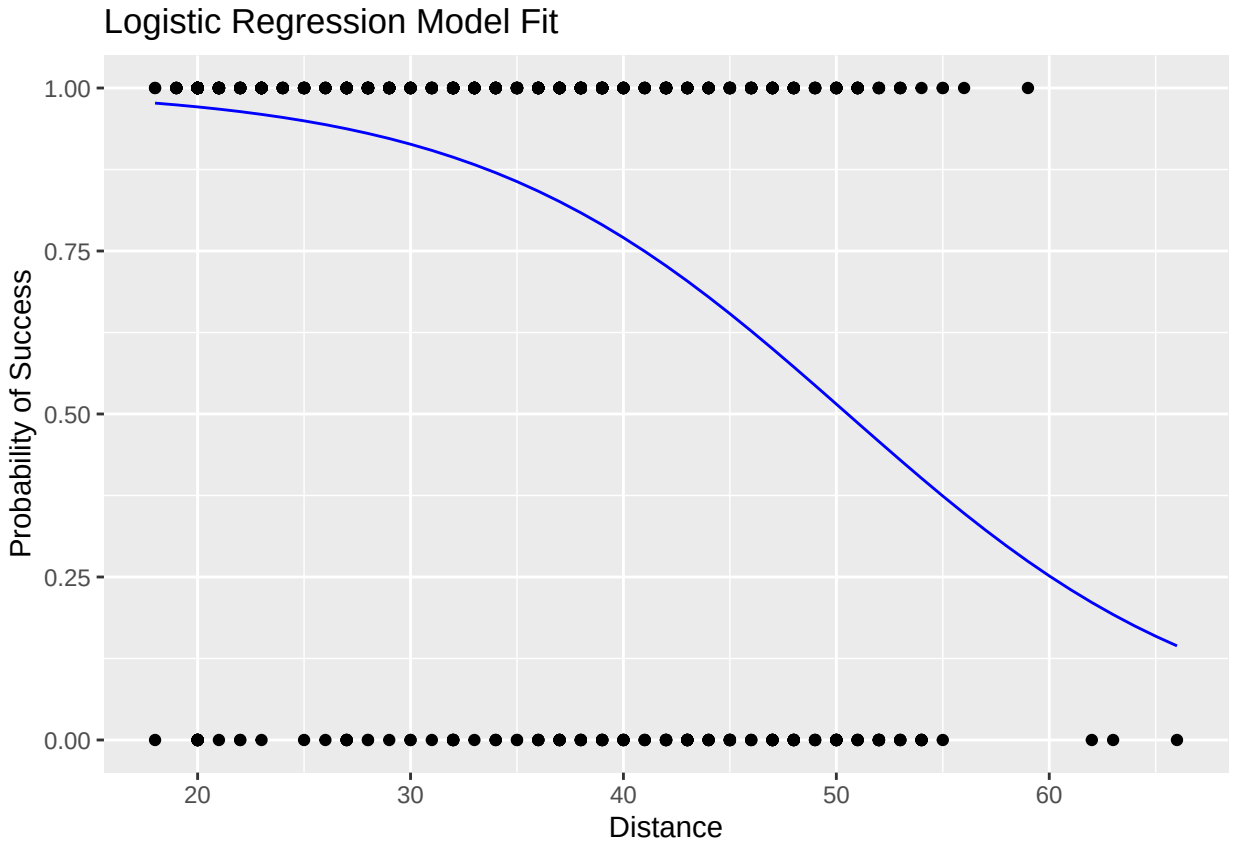
1

a

```
library(ggplot2)

dataset <- read.csv("./Placekick.csv")

model <- glm(good ~ distance, data=dataset, family=binomial(link="logit"))
summary(model)
```

```
##
## Call:
## glm(formula = good ~ distance, family = binomial(link = "logit"),
##     data = dataset)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.812080   0.326277   17.81   <2e-16 ***
## distance    -0.115027   0.008339  -13.79   <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: 1013.43  on 1424  degrees of freedom
## Residual deviance:  775.75  on 1423  degrees of freedom
## AIC: 779.75
##
## Number of Fisher Scoring iterations: 6
```

```
prediction_data <- data.frame(distance=seq(min(dataset$distance), max(dataset$distance), by=1))
prediction_data$predicted_good <- predict(model, newdata=prediction_data, type="response")

# Plot
ggplot(dataset, aes(x=distance, y=good)) +
  geom_point() +
  geom_line(data=prediction_data, aes(x=distance, y=predicted_good), color='blue') +
  labs(title="Logistic Regression Model Fit", x="Distance", y="Probability of Success")
```



The logistic regression model of good over distance is not so representative, it doesn't fit well when distance is less than 50. The groud truth shows that when distance is less than 50, the good can both be 0 or 1, with a similar number, while the logistic regression only generate higher probility for good to be 1.

b

```
library(MASS)
initial_model <- glm(good ~ 1, data = dataset, family = binomial)
full_model <- glm(good ~ ., data = dataset, family = binomial)
forward_selected_model <- step(initial_model, scope = list(lower = initial_model, upper = full_model), d
```

```
## 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(forward_selected_model)
```

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

The model chosen is good ~ distance + PAT + change + wind.

c

```
model <- glm(good ~ distance + PAT + change + wind, data = dataset, family = binomial)

summary(model)
```

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

```
coefficients <- coef(model)
print(coefficients)
```

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

The decision boundary when probability of success is 0.5 is $4.75 - 0.087 * distance + 1.22 * Pat - 0.33 * change - 0.52 * wind = 0$

d

The decision boundary when probability of success is 0.9 is

$4.75 - 0.087 * distance + 1.22 * Pat - 0.33 * change - 0.52 * wind = log(9)$

The probability of success 0.5 represents a scenario where the odds of success and failure are equal. The decision boundary at this threshold is where the model is indifferent between predicting success or failure. The output is classified to be success which is good 1 in this case when the output probability is larger than 0.5, and be good 0 when is smaller than 0.5.

The probability of success 0.9 indicates a much higher certainty in predicting success. The decision boundary for this threshold would be more conservative, which means the boundary is lower than 0.5 case so that the chance be classified good 1 is higher than be classified good 0.

2

a

```
bootGLM <- function(x, y, B = 1000){
  if (!is.data.frame(x)) {
    x <- as.data.frame(x)
  }

  data <- cbind(x, y)
  colnames(data)[ncol(data)] <- "y"

  coef_matrix <- matrix(NA, nrow = B, ncol = ncol(x))
  colnames(coef_matrix) <- colnames(x)

  for (i in 1:B) {
    sample_indices <- sample(1:nrow(data), replace = TRUE)
    resampled_data <- data[sample_indices, ]

    model <- glm(y ~ ., data = resampled_data, family = binomial)

    coef_matrix[i, ] <- coef(model)[-1]
  }

  standard_errors <- apply(coef_matrix, 2, sd)

  return(standard_errors)
}
```

b

```
x <- dataset[, c("distance", "PAT", "change", "wind")]
y <- dataset$good

bootstrap_se <- bootGLM(x, y, B = 1000)
print("Bootstrap Standard Errors:")
```

```
## [1] "Bootstrap Standard Errors:"
```

```
print(bootstrap_se)
```

```
##   distance        PAT     change       wind
```

```
## 0.01108679 0.38529878 0.19184160 0.29195527
```

```
model <- glm(good ~ distance + PAT + change + wind, data = dataset, family = binomial)
model_summary <- summary(model)

print("Model Summary Standard Errors:")
```

```
## [1] "Model Summary Standard Errors:"
```

```
print(sqrt(diag(vcov(model))))
```

```
## (Intercept)    distance         PAT      change        wind
##  0.47199923  0.01111711  0.38474229  0.19334821  0.31315327
```

The standard error for both distance and change are almost same in Bootstarp and Model summary. However, The SDE for PAT in Bootstrap is a bit higher than PAT in Model summary. Also the SDE for wind in Bootstrap is a bit lower than in the Model summary.

The higher SDE for PAT in Bootstrap method can attribute to the a missing important predicator. This means in the model selection step, maybe We missed an important predicator that we should have included into our model, thus caused the SDE to be higher. To solve it, I think maybe we should use other variable selection methods like backward selection or LASSO to see if there is any additional predicator selected. Also the higher SDE may attribute to the violations to model assumption, but since distance and change have a similar SDE, so I think the posibility should be decluded.

The lower SDE for wind in Bootstrap, may caused by the model overfitting, or because the distribution of wind variable is skewed. To solve this, again, maybe its a good idea to combine other variable selection methods. Also to dive into the distribution of wind variable to check it.