Random Simulations and Data Modelling

Manbir Singh Panesar

10/26/2021

3. For this question you have to simulate a dataset.

(a) Perform the following simulations.

Generate 500 random values from X1 ~ Uniform[0, 1], X2 ~ Uniform[0, 1], X3 ~ Uniform[0, 1], X4 ~ Uniform[0, 1], X5 ~ Uniform[0, 1]

Generate, f(X) = 4[sin(pix1x2)+8(x3-0.5)^3+1.5x4-x5-0.77]. Here, pi = 3.14…

Generate Y ~ Bernoulli (p(X) = exp(f(X)) / 1 + exp(f(X)))

```
### Question 3 ###
## 3a) Random Simulations ##
set.seed(1000660251)
n <- 500
X1 <- runif(n, 0, 1)
X2 <- runif(n, 0, 1)
X3 <- runif(n, 0, 1)
X4 <- runif(n, 0, 1)
X5 <- runif(n, 0, 1)
fX <- 4*(sin(pi*X1*X2) + 8*(X3 - 0.5)^3 + 1.5*X4 - X5 - 0.77)
pX <- (exp(fX))/(1 + exp(fX))
Y <- rbinom(n, 1, pX)
```

(b) Fit a logistic regression where Y is the outcome and X1,X2, …, X5 are the predictors. Show the coefficients table. Produce the ROC curve. State the AUC value and interpret.

```
## 3b) Logistic Regression ##
library(nnet)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
df <- c(list(X1, X2, X3, X4, X5))
#df
multi.mod <- multinom(Y ~ X1 + X2 + X3 + X4 + X5, data = df)
```

```
## # weights:  7 (6 variable)
## initial  value 346.573590
## iter  10 value 208.899879
## final  value 208.839465
## converged
```

```
summary(multi.mod)
```

```
## Call:
## multinom(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = df)
##
## Coefficients:
##                  Values Std. Err.
## (Intercept) -5.257012 0.6177702
## X1           2.155847 0.4400079
## X2           2.625741 0.4496319
## X3           3.534753 0.4688883
## X4           5.467585 0.5802606
## X5          -3.390456 0.4882242
##
## Residual Deviance: 417.6789
## AIC: 429.6789
```

```
logit.mod <- glm(Y ~ X1 + X2 + X3 + X4 + X5, family = binomial(link = logit), data = df)
summary(logit.mod)
```

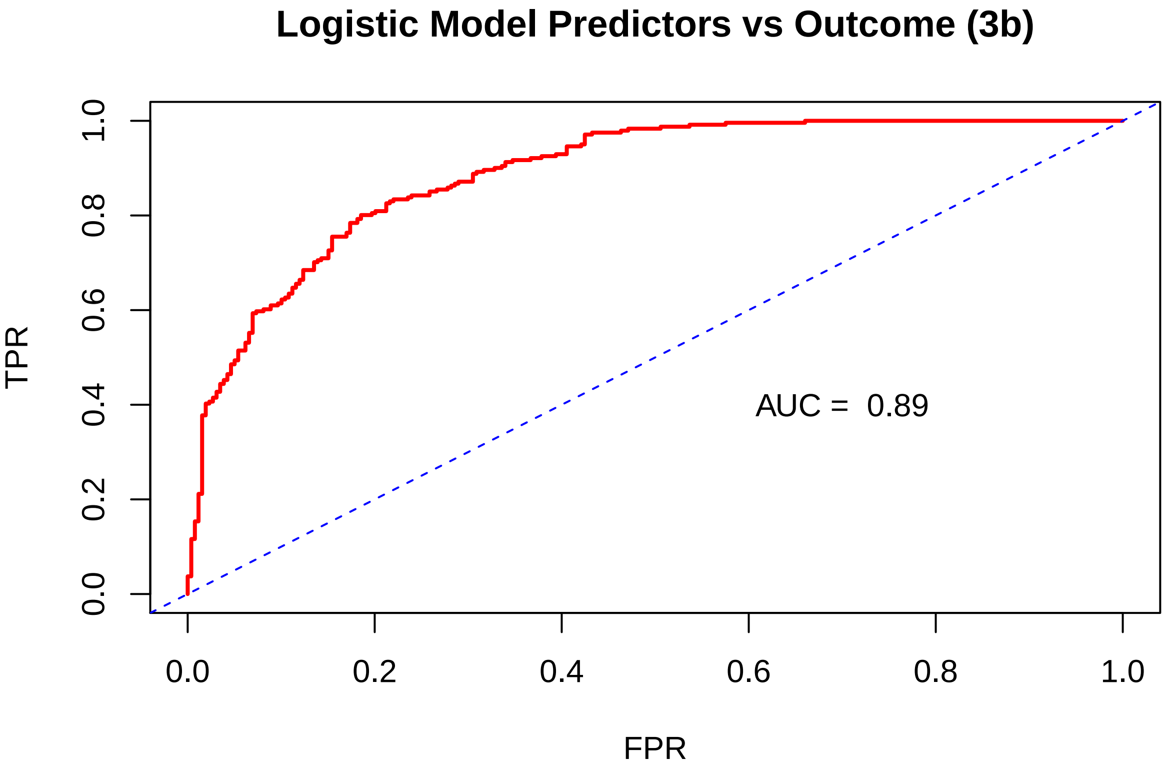
```
##
## Call:
## glm(formula = Y ~ X1 + X2 + X3 + X4 + X5, family = binomial(link = logit),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0276  -0.6113  -0.1468   0.6685   2.2369
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.2570     0.6178  -8.510  < 2e-16 ***
## X1            2.1558     0.4400   4.900 9.60e-07 ***
## X2            2.6258     0.4496   5.840 5.22e-09 ***
## X3            3.5348     0.4689   7.539 4.74e-14 ***
## X4            5.4676     0.5802   9.423  < 2e-16 ***
## X5           -3.3904     0.4882  -6.945 3.79e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 692.50  on 499  degrees of freedom
## Residual deviance: 417.68  on 494  degrees of freedom
## AIC: 429.68
##
## Number of Fisher Scoring iterations: 5
```

```
# ROC Curve #
library(pROC)
pred <- predict(logit.mod, type = "response")
roc_logit <- roc(Y ~ pred)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities
pdf("ROC_LOG_REG.pdf")
```

```
plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red', main = "Logistic
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```



```
dev.off()
```

```
## pdf
##   3
```

```
auc(roc_logit)
```

```
## Area under the curve: 0.8888
```

We see that the AUC is 0.8888 indicating the model can discriminate between true positive rate and a false positive rate 88% of the time. From the coefficients we see that for a one point increase in X1, X2, X3, X4 there is an increase in the difference of the log odds. While there is a decrease in the difference in the log odds of 3.3904 for an increase in X5.

(c) Now instead of using the original X1,X2, . . . , X5 as predictors, transform the variables in such a way that they resembles the individual terms in f(X). That, is create new variables from X1,X2, . . . , X5 in such a way that f(X) is transformed to a linear predictor. Now run a logistic regression using the new variables. Show the coefficients table. Produce the ROC curve. State the AUC value. (Hint: You have to create 4 new variables from X1,X2, . . . , X5)

```r
## 3c) Linear Transformation Predictor ##
new_X1 <- 4*(sin(pi*X1*X2))
new_X2 <- 32*(X3 - 0.5)^3
new_X3 <- 4*(1.5*X4)
new_X4 <- 4*(-X5 - 0.77)

transformed.logit.mod <- glm(Y ~ new_X1 + new_X2 + new_X3 + new_X4,
                             family = binomial(link = logit), data = df)
summary(transformed.logit.mod)
```

```
##
## Call:
## glm(formula = Y ~ new_X1 + new_X2 + new_X3 + new_X4, family = binomial(link = logit),
##     data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.98659  -0.54712  -0.09419   0.55662   2.81098
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1854     0.6390  -0.290    0.772
## new_X1        0.9183     0.1124   8.168 3.14e-16 ***
## new_X2        0.8001     0.1032   7.753 8.96e-15 ***
## new_X3        1.0129     0.1069   9.476  < 2e-16 ***
## new_X4        0.9512     0.1325   7.178 7.09e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 692.50  on 499  degrees of freedom
## Residual deviance: 377.23  on 495  degrees of freedom
## AIC: 387.23
##
## Number of Fisher Scoring iterations: 6
```

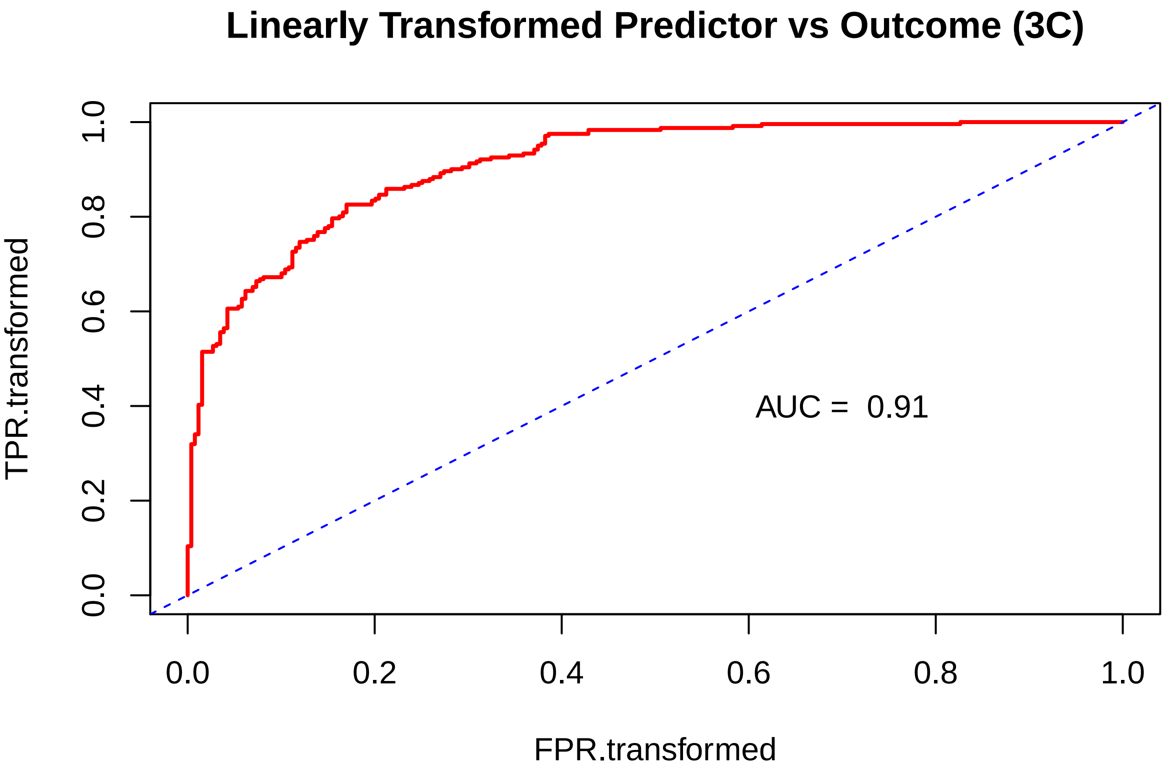
```r
# ROC Curve #
library(pROC)
transformed.pred <- predict(transformed.logit.mod, type = "response")
roc_logit.transformed <- roc(Y ~ transformed.pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
## The True Positive Rate ##
TPR.transformed <- roc_logit.transformed$sensitivities
## The False Positive Rate ##
FPR.transformed <- 1 - roc_logit.transformed$specificities
pdf("ROC_LOG_REG_TRANSFORMED.pdf")
```

```r
plot(FPR.transformed, TPR.transformed, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col =
     main = "Linearly Transformed Predictor vs Outcome (3C)")
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit.transformed),2)))
```



```
dev.off()
```

```
## pdf
##   3
```

```
auc(roc_logit.transformed)
```

```
## Area under the curve: 0.9105
```

Here we see that the AUC is 0.9105,indicating that this model can discriminate between a true positive rate from a false positive rate 91% of the time. All four newly transformed variables provide an increase by some value (different for each transformed predictor) in the difference in the log odds with a one point increase in their respective transformed predictor value.

(d) Compare your results in (b) and (c). How did your coefficients and AUC change from (b) to (c)? Explain why you think this happened.

First we see that the logit.mod (from 3b) has a higher AIC value (429.68) compared to the transformed.logit.mod (in 3c) with an AIC value of 387.23. This indicates that the transformed.logit.mod has a better goodness of fit.

Secondly, the coefficents in the transformed model(3c) are smaller absolute values than the logistic regression(3b). Additionally, there is 1 negative coefficent in the transformed model, whereas there are 2 in the logistic regression. This could suggest that the transformed model has more accurate odds given the very large and extreme odds found in the logistic regression coefficients.

Thirdly, the AUC for the transformed model is 0.9105 and the AUC for the logistic regression model is 0.8888. This suggests the transformed model is much better at discriminating TPR from FPR compared to the logistic model.

All in all, these findings indicate that the transformed model is a better model for predicting compared to the logistic model in 3b. This could be due to the fact that the new_X variables(3c) that were linearly transformed to match the fX function, which was used in the simulating Y, the outcome variable, better than the X variables in 3b. In other words the linearly transformed variables were better at simulating values that Y was also producing. This lead to better overall prediction due to lower AIC score, goodness of fit and stronger coefficients and AUC.