PSTAT 131 - Homework 2

Niveditha Lakshminarayanan, Celeste Herrera

10/30/2020

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Linear regression

In this problem, we will make use of the **Auto** data set, which is part of the *ISLR* package and can be directly accessed by the name Auto once the *ISLR* package is loaded. The dataset contains 9 variables of 392 observations of automobiles. The qualitative variable **origin** takes three values: 1, 2, and 3, where 1 stands for American car, 2 stands for European car, and 3 stands for Japanese car.

1. Fit a linear model to the data, in order to predict "mpg" using all of the other predictors except for "name". Present the estimated coefficients. For each predictor, comment on whether you can reject the null hypothesis that there is no linear associaiton between that predictor and "mpg", conditional on the other predictors in the model.

```
data("Auto")
#glm.fit <- glm(mpg ~ . - name - origin + as.factor(origin), data = Auto)

mpg <- Auto$mpg
cylinders<- Auto$cylinders
displacement<- Auto$displacement
horsepower<- Auto$horsepower
weight<- Auto$weight
acceleration<- Auto$acceleration
year<- Auto$year
origin = as.factor(Auto$origin)

glm.fit <- glm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + origin)

summary(glm.fit)
```

```
##
## Call:
## glm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.009  -2.078  -0.098   1.986  13.361
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.80e+01   4.68e+00   -3.84  0.00014 ***
## cylinders    -4.90e-01   3.21e-01   -1.52  0.12821
## displacement  2.40e-02   7.65e-03    3.13  0.00186 **
## horsepower   -1.82e-02   1.37e-02   -1.33  0.18549
## weight       -6.71e-03   6.55e-04  -10.24  < 2e-16 ***
## acceleration  7.91e-02   9.82e-02    0.81  0.42110
## year          7.77e-01   5.18e-02   15.01  < 2e-16 ***
## origin2       2.63e+00   5.66e-01    4.64  4.7e-06 ***
## origin3       2.85e+00   5.53e-01    5.16  3.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 10.93)
##
##     Null deviance: 23819.0  on 391  degrees of freedom
## Residual deviance:  4187.4  on 383  degrees of freedom
## AIC: 2061
##
## Number of Fisher Scoring iterations: 2
```

For the cylinder, horsepower, and acceleration predictors we fail to reject the null hypothesis because there is a p-value which is larger than .05, so there will be no linear association between cylinders and mpg, horsepower and mpg, & acceleration and mpg. For the displacement, weight, year, and origin predictors we reject the null hypothesis because there is a p-value which is smaller than .05, so there will be linear association between displacement and mpg, weight and mpg, year and mpg, & origin and mpg.

2. What is the training mean squared error of this model?

```
mean(glm.fit$residuals^2)
```

```
## [1] 10.68
```

3. What gas mileage do you predict for a Japanese car with 3 cylinders, displacement 111, horsepower of 95, weight of 2900, acceleration of 22, built in the year 1981? (Be sure to check how "year" is coded in the dataset).

```
new_Japanese_car <- data.frame(cylinders = 3, displacement = 111, horsepower = 95, weight = 2900, accele
year = 81, origin = factor(3))

predict(glm.fit, newdata = new_Japanese_car)
```

```
##     1
## 29.58
```

4. On average, holding all other covariates fixed, what is the difference between the mpg of a Japanese car and the mpg of an American car? What is the difference between the mpg of a European car and the mpg of an American car?

From our linear regression, we can see that on average, American cars have the lowest mpg compared to Japanese and European cars. The average difference between Japanese and American cars is 2.85 miles per gallon. The average difference between European and American cars is 2.63 miles per gallon.

5. On average, holding all other covariates fixed, what is the change in mpg associated with a 10-unit change in horsepower?

```
-1.80e+01 + (-1.82e-02*10)
```

```
## [1] -18.18
```

We can see that there is a 18.18 mpg decrease associated with a 10-unit change in horsepower.

Spam detection with spambase dataset

```
#read in spambase.tab dataset
spam <- read_table2("spambase.tab", guess_max = 2000)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```
spam <- spam %>%
mutate(y = factor(y, levels=c(0,1), labels=c("good", "spam"))) %>% # label as factors
mutate_at(.vars=vars(-y), .funs=scale) # scale others

#misclassification error rate
calc_error_rate <- function(predicted.value, true.value) {
  return(mean(true.value!=predicted.value))
}

#training/test sets
set.seed(1)
test.indices = sample(1:nrow(spam),1000)
spam.train=spam[-test.indices,]
spam.test=spam[test.indices,]
```

Logistic regression

6. In a binary classification problem, let p represent the probability of class label "1", which implies that 1 represents probability of class label "0". The *logistic function* (also called the "inverse logit") is the cumulative distribution function of logistic distribution, which maps a real number z to the open interval (0,1):

$$p(z) = \frac{e^z}{1 + e^z}$$

(a) Show that indeed the inverse of a logistic function is the *logit* function:

$$z(p) = ln\left(\frac{p}{1-p}\right)$$

To show that that the inverse of the logistic function is the *logit* function, we must show that p(z(p)) = p and z(p(z)) = z.

First, let's prove that p(z(p)) = p.

$$\begin{aligned} p(z(p)) &= \frac{e^{ln\left(\frac{p}{1-p}\right)}}{1 + e^{ln\left(\frac{p}{1-p}\right)}} \\ &= \frac{\frac{p}{1-p}}{1 + \frac{p}{1-p}} \\ &= \frac{\frac{p}{1-p}}{\frac{1}{1-p}} \\ &= \frac{p}{1-p} * \frac{1-p}{1} \\ &= p \end{aligned}$$

Now, we must show that z(p(z)) = z.

$$\begin{aligned} z(p(z)) &= ln\left(\frac{\frac{e^z}{1+e^z}}{1 - \frac{e^z}{1+e^z}}\right) \\ &= ln\left(\frac{\frac{e^z}{1+e^z}}{\frac{1}{1+e^z}}\right) \\ &= ln\left(\frac{e^z}{1+e^z} * \frac{1+e^z}{1}\right) \\ &= ln(e^z) \\ &= z \end{aligned}$$

So, since we can see that p(z(p)) = p and z(p(z)) = z, where z(p) is the *logit* function, and p(z) is the logistic function. So, this proves that the inverse of a logistic function is indeed the *logit* function.

(b) The logit function is a commonly used *link function* for a generalized linear model of binary data. One reason for this is that implies interpretable coefficients. Assume that $z = \beta_0 + \beta_1 x_1$, and $p = \text{logistic}(z)$. How does the odds of the outcome change if you increase x_1 by two? Assume β_1 is negative: what value does p approach $x_1 \to \infty$? What value does p approach as $x_1 \to -\infty$?

If we increase x_1 by two, the odds of the outcome changes by $2e^{\beta_1}$. If β_1 is negative and $x_1 \to \infty$, then $p \to -\infty$. If β_1 is negative and $x_1 \to -\infty$, then $p \to \infty$.

7. Use logistic regression to perform classification. Logistic regression specifically estimates the probability that an observation as a particular class label. We can define a probability threshold for assigning class labels based on the probabilities returned by the glm fit.

In this problem, we will simply use the "majority rule". If the probability is larger than 50% class as spam. Fit a logistic regression to predict spam given all other features in the dataset using the glm function. Estimate the class labels using the majority rule and calculate the training and test errors using the **calc_error_rate** defined earlier.

```
#logistic regression
glm.fits = glm(y ~ ., data = spam.train, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#training error rate
probs.train = predict(glm.fits, newdata = spam.train, type="response")

calc_error_rate(ifelse(probs.train <= 0.5, "good", "spam"), spam.train$y)
```

```
## [1] 0.06804
```

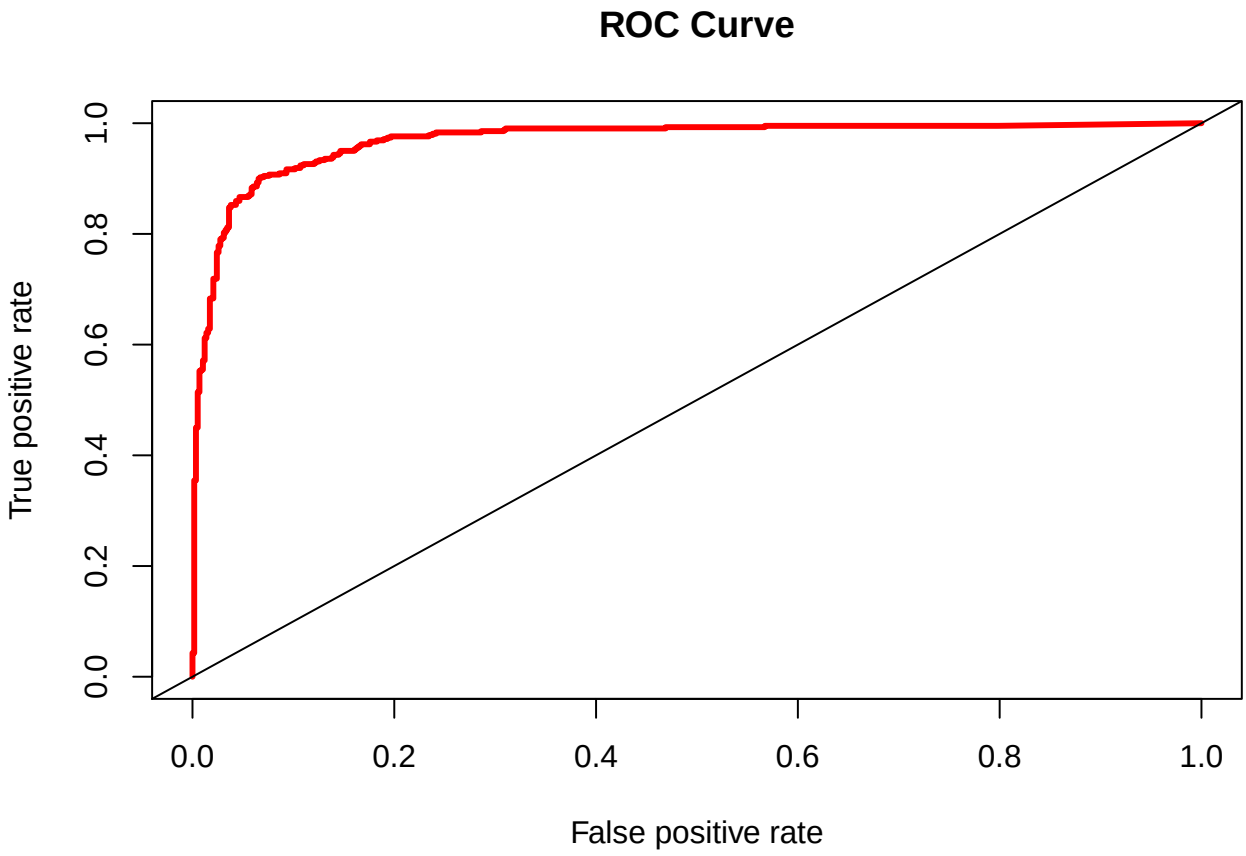
```
#test error rate
probs.test = predict(glm.fits, newdata = spam.test, type="response")

calc_error_rate(ifelse(probs.test <= 0.5, "good", "spam"), spam.test$y)
```

```
## [1] 0.086
```

8. We will construct ROC curve based on the predictions of the *test* data from the model we obtained from the logistic regression above. Plot the ROC for the test data for the logistic regression fit. Compute the area under the curve (AUC).

Hints: In order to construct the ROC curves one needs to use the vector of predicted probabilities for the test data. The usage of the function predict() may be different from model to model. For logistic regression, one needs to predict type response, see Lab 3.

```
#ROC curve
pred = prediction(probs.test, spam.test$y)
perf = performance(pred, measure = 'tpr', x.measure = 'fpr')
plot(perf, col=2, lwd = 3, main = 'ROC Curve')
abline(0,1)
```




```
#AUC
auc = performance(pred, "auc")@y.values
auc
```

```
## [[1]]
## [1] 0.9685
```

9. In the SPAM example, take "positive" to mean "spam". If you are the designer of a spam filter, are you more concerned about the potential for false positive rates that are too large or true positive rates that are too small? Argue your case.

If I was a designer I would be more concerned about the false positive rates that are too large because it would indicated that there would be a bug within the spam filter than would need to be indicated in order to fix the problem. Whether as having true positive test rate that is too small can also just mean the spam filter itself is great at filtering out the spam mail that is being recieved. So, we can conclude having a false positive rates at a large number is not good for filtering out spam compared to having true positive rates that are too small.