Homework #2

Madhav Viswesvaran

February 10, 2025

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Linear Regression

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. With a 0.01 threshold, comment on whether you can reject the null hypothesis that there is no linear association between mpg with any of the predictors.

```
#fitting linear model to predict mpg, except for name
#View(Auto)
Auto_new <- Auto %>% select(-name)
#setting origin as a factor
Auto_new$origin = as.factor(Auto_new$origin)
#View(Auto_new)
mpg_lm <- lm(mpg ~., data = Auto_new)
summary(mpg_lm)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = Auto_new)
##
## 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
##
## Residual standard error: 3.31 on 383 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.821
## F-statistic:  224 on 8 and 383 DF,  p-value: <2e-16
```

At the 0.01 threshold we can reject the null hypothesis that there is no linear association between mpg and any of the predictors. The displacement, weight, year and origin are all significant predictors of mpg.

2. Take the whole dataset as training set. What is the training mean squared error of this model? Can you calculate the test mean squared error?

```
#calculating MSE
mse <- mean(mpg_lm$residuals^2)
```

The mse on the whole dataset is about 10.85, and we cannot caluclate the test mse, since we used the whole dataset for fitting the model (training).

3. What gas mileage do you predict for an European car with 3 cylinders, displacement 132, horsepower of 115, weight of 3050, acceleration of 32, built in the year 1995? (Be sure to check how year is coded in the dataset).

```
new_data <- data.frame(origin = factor(2, levels = c(1,2,3)),  cylinders = 3, displacement = 132, horsep
predict(mpg_lm, newdata = new_data,
        level = 0.95, interval = 'predict')
```

```
##     fit   lwr   upr
## 1 40.16 32.48 47.85
```

The predicted mpg for a car with the given specifications is 40.16.

4. On average, holding all other features 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?

```
euro_vs_usa <- coef(mpg_lm)["origin2"]
jpn_vs_usa <- coef(mpg_lm)["origin3"]
print("Japanese vs American")
```

```
## [1] "Japanese vs American"
```

```
jpn_vs_usa
```

```
## origin3
##   2.853
```

```
print("European vs American")
```

```
## [1] "European vs American"
```

```
euro_vs_usa
```

```
## origin2
##    2.63
```

The difference between the mpg of a Japanese car and the mpg of an American car is 2.853. The difference between the mpg of a European car and the mpg of an American car is 2.63, holding all other features fixed.

5. On average, holding all other predictor variables fixed, what is the change in mpg associated with a 30-unit increase in displacement?

```
coef(mpg_lm)["displacement"] * 30
```

```
## displacement
##       0.7194
```

The change in mpg associated with a 30-unit increase in displacement is 0.7194.

Logistic Regression

```
#reading data
algae <- read_table2("algaeBloom.txt", col_names=
c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4',
'oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
na="XXXXXXX")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mnO2 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double(),
##   a2 = col_double(),
##   a3 = col_double(),
##   a4 = col_double(),
##   a5 = col_double(),
##   a6 = col_double(),
##   a7 = col_double()
## )
```

```
#transforming data
algae.transformed <- algae %>% mutate_at(vars(4:11), funs(log(.)))
algae.transformed <- algae.transformed %>%
mutate_at(vars(4:11),funs(ifelse(is.na(.),median(.,na.rm=TRUE),.)))
# a1 == 0 means low
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))
```

```
algae.transformed
```

```
## # A tibble: 200 x 18
##    season size  speed   mxPH  mnO2    Cl    NO3   NH4  oPO4   PO4   Chla a1
##    <chr>  <chr> <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl> <fct>
##  1 winter small medium  2.08  2.28  4.11  1.83   6.36  4.65  5.14  3.91  0
##  2 spring small medium  2.12  2.08  4.06  0.253  5.91  6.06  6.33  0.262 0
##  3 autumn small medium  2.09  2.43  3.69  1.67   5.85  4.83  5.23  2.75  0
```

```
##  4 spring small medium  2.09  1.57  4.35  0.834  4.59  4.11  4.93  0.336 0
##  5 autumn small medium  2.09  2.20  4.01  2.34   5.45  4.06  4.58  2.35  1
##  6 winter small high    2.11  2.57  4.19  2.22   6.06  2.90  4.04  3.35  1
##  7 summer small high    2.10  2.33  4.29  0.429  4.70  4.11  4.72  1.16  0
##  8 autumn small high    2.09  2.36  4.08  1.61   5.33  3.80  4.35  1.93  1
##  9 winter small medium  2.16  1.22  3.09 -0.121  4.63  3.59  4.26  1.71  1
## 10 winter small high    2.07  2.29  2.08  0.329  1.76  3.31  3.84 -0.223 1
## # i 190 more rows
## # i 6 more variables: a2 <dbl>, a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>,
## #   a7 <dbl>
```

```
#classfication error rate function
calc_error_rate <- function(predicted.value, true.value){
return(mean(true.value != predicted.value))
}
#traning/test seeds
set.seed(1)
test.indices = sample(1:nrow(algae.transformed), 50)
algae.train=algae.transformed[-test.indices,]
algae.test=algae.transformed[test.indices,]
```

1. Prove that indeed the inverse of a logistic function is the logit function.

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

2. Assume that $z = \beta_0 + \beta_1 x_1$, and p = 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 as $x_1 \to \infty$? What value does p approach as $x_1 \to -\infty$?

The odds of the outcome changes to odds * $e^{2\beta_1}$ if you increase x_1 by two. If β_1 is negative, p approaches 0 as x_1 goes to infinity, and p approaches 1 as x_1 goes to negative infinity.

3. Use logistic regression to perform classification in the data application above. 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 label "1". Fit a logistic regression to predict a1 given all other features (excluding a2 to a7) 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.

```
#logisitic regression
glm.fit = glm(a1 ~ season+size+speed+mxPH+mnO2+Cl+NO3+NH4+oPO4+PO4+Chla,
              data=algae.train, family=binomial)
# Summarize the logistic regression model
summary(glm.fit)
```

```
##
```

```
## Call:
## glm(formula = a1 ~ season + size + speed + mxPH + mnO2 + Cl +
##     NO3 + NH4 + oPO4 + PO4 + Chla, family = binomial, data = algae.train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.3708    11.3012    0.21   0.8338
## seasonspring  -0.6582     0.7830   -0.84   0.4006
## seasonsummer   0.8865     0.8420    1.05   0.2924
## seasonwinter   0.6159     0.6773    0.91   0.3631
## sizemedium     0.6043     0.7567    0.80   0.4245
## sizesmall      1.9198     0.8672    2.21   0.0268 *
## speedlow       1.4404     0.8468    1.70   0.0889 .
## speedmedium    0.0774     0.6157    0.13   0.8999
## mxPH          -0.2468     5.4101   -0.05   0.9636
## mnO2           1.1671     0.9187    1.27   0.2039
## Cl            -0.3636     0.3765   -0.97   0.3342
## NO3           -0.1568     0.3718   -0.42   0.6732
## NH4            0.3828     0.2629    1.46   0.1453
## oPO4          -0.9784     0.4817   -2.03   0.0422 *
## PO4           -0.1558     0.5856   -0.27   0.7902
## Chla          -0.8376     0.2892   -2.90   0.0038 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 202.69  on 149  degrees of freedom
## Residual deviance: 113.73  on 134  degrees of freedom
## AIC: 145.7
##
## Number of Fisher Scoring iterations: 6
```

```
#training and test errors using the calc_error_rate
pred.train = predict(glm.fit, newdata = algae.train, type = "response")
pred.train = ifelse(pred.train > .5, 1,0)
pred.test = predict(glm.fit, newdata = algae.test, type = "response")
pred.test = ifelse(pred.test > .5, 1,0)
test_error <- calc_error_rate(pred.test, algae.test$a1)
train_error <- calc_error_rate(pred.train,algae.train$a1)
cat("Training error:", train_error, "\n")
```

```
## Training error: 0.2
```

```
cat("Test error:", test_error, "\n")
```

```
## Test error: 0.3
```

```
cat("Estimated Class Values:","\n")
```

```
## Estimated Class Values:
```

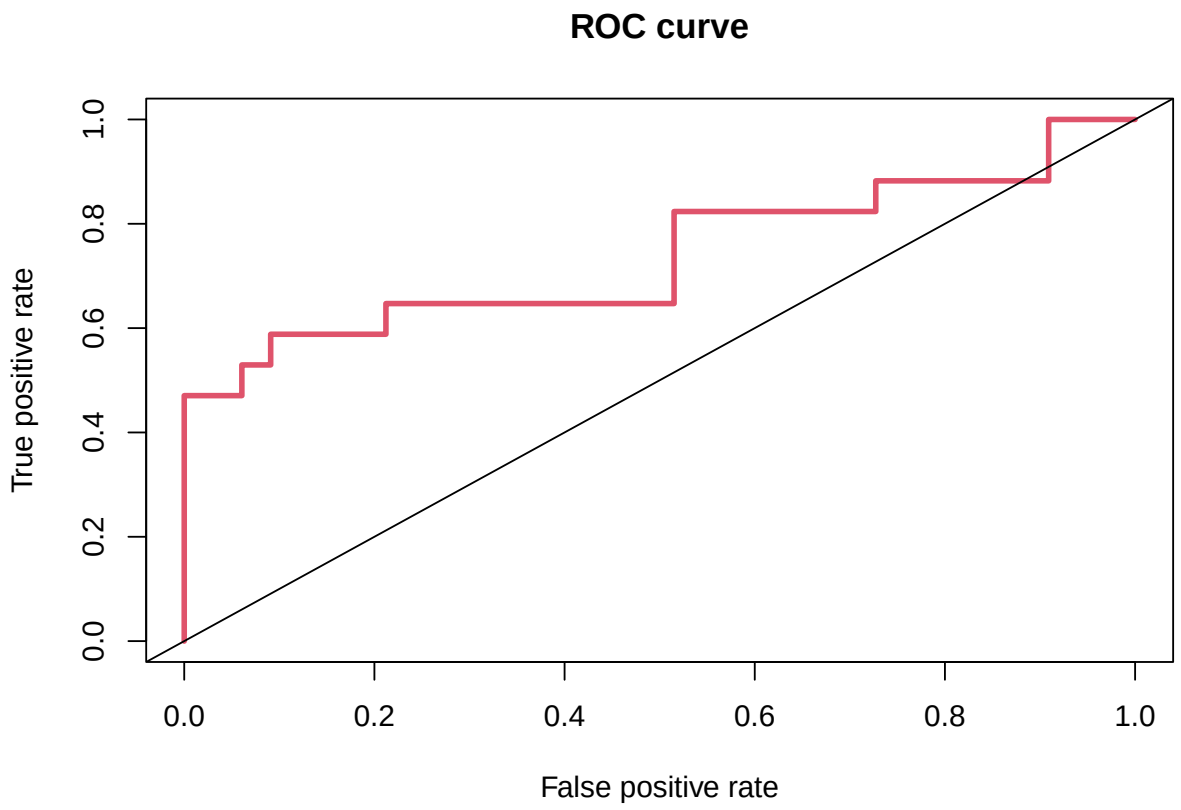
```
pred.test
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  1  0  0  0  0  1  1  1  1  0  1  1  0  1  0  0  1  0  0  1  1  0  1  0  0  1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

```
##  1  0  0  0  1  0  1  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  1  1
```

4. 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).

```
#plot ROC Curve
library(ROCR)
#reassigning pred.test to get probability
pred.test = predict(glm.fit, newdata = algae.test, type = "response")
pred = prediction(pred.test, algae.test$a1)
perf = performance(pred, measure="tpr", x.measure="fpr")
plot(perf, col=2, lwd=3, main="ROC curve")
abline(0,1)
```




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

```
## [[1]]
## [1] 0.738
```

Bootstrapping

1. Given a sample of size n, what is the probability that any observation j is not in a bootstrap sample? Express your answer as a function of n.

$$(1-\frac{1}{n})^n$$

2.Compute the above probability for n = 1000.

```
(1-(1/1000))^1000
```

```
## [1] 0.3677
```

3.Verify that your calculation is reasonable by resampling the numbers 1 to 1000 with replacement and printing the ratio of missing observations. Hint: use the unique and length functions to identify how many unique observations are in the sample. Note that the answer does not have to be exactly the same as what you get in b) due to randomness in sampling.

```
set.seed(123)
samp <- 1:1000
bootstrap_sample <- sample(samp, size = 1000, replace= TRUE)
1 - (length(unique(bootstrap_sample)) / 1000)
```

```
## [1] 0.362
```

Cross-validation estimate of test error

```
dat = subset(Smarket, select = -c(Year,Today))
dat$Direction = ifelse(dat$Direction == "Up", 1, 0)
```

1. Split dat into a training set of 700 observations, and a test set of the remaining observations. Fit a logistic regression model, on the training data, to predict the Direction using all other variables except for Year and Today as predictors. Calculate the error rate of this model on the test data. Use set.seed(123) in the begining of your answer.

```
#test/train split
set.seed(123)
train.indices = sample(1:nrow(dat), 700)
dat.train=dat[train.indices,]
dat.test=dat[-test.indices,]

#fitting logistic regression
dat.glm.fit = glm(Direction ~ .,
             data=dat.train, family=binomial)

#calculating test error rate
dat.pred.test = predict(dat.glm.fit, newdata = dat.test, type = "response")
dat.pred.test = ifelse(dat.pred.test > .5, 1,0)
test_error <- calc_error_rate(dat.pred.test, dat.test$Direction)
round(test_error,2)
```

```
## [1] 0.47
```

```
summary(dat.glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ ., family = binomial, data = dat.train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -0.11079    0.32154   -0.34     0.73
## Lag1        -0.02475    0.06400   -0.39     0.70
## Lag2        -0.04790    0.06880   -0.70     0.49
## Lag3         0.02605    0.06834    0.38     0.70
## Lag4        -0.00158    0.06461   -0.02     0.98
## Lag5        -0.00549    0.06532   -0.08     0.93
## Volume       0.13338    0.21172    0.63     0.53
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 969.12  on 699  degrees of freedom
## Residual deviance: 967.90  on 693  degrees of freedom
## AIC: 981.9
##
## Number of Fisher Scoring iterations: 3
```

2. Use a 10-fold cross-validation approach on the whole dat to estimate the test error rate.Report the estimated test error rate you obtain. Use set.seed(123) in the begining of your answer.

```
#CV function
do.chunk <- function(chunkid, folddef, dat, ...){
  # Get training index
  train = (folddef!=chunkid)
  # Get training set and validation set
  dat.train = dat[train, ]
  dat.val = dat[-train, ]
  # Train logistic regression model on training data
  fit.train = glm(Direction ~ ., family = binomial, data = dat.train)
  # get predicted value on the validation set
  pred.val = predict(fit.train, newdata = dat.val, type = "response")
  pred.val = ifelse(pred.val > .5, 1,0)
  data.frame(fold = chunkid,
  val.error = mean(pred.val != dat.val$Direction))
}
```

```
set.seed(123)
nfold = 10
folds = cut(1:nrow(dat), breaks=nfold, labels=FALSE) %>% sample()
error.folds = NULL
for (j in seq(10)){
    tmp = do.chunk(chunkid=j, folddef=folds, dat = dat)
    error.folds = rbind(error.folds, tmp) # combine results
}
cat("10-fold CV Test error rate:", "\n")
```

```
## 10-fold CV Test error rate:
```

```
error.folds$val.error %>% mean()
```

```
## [1] 0.4754
```