Homework 04

STAT 430, Fall 2017

Due: Friday, October 6, 11:59 PM

Please see the homework instructions document for detailed instructions and some grading notes. Failure to follow instructions will result in point reductions.

Exercise 1 (Comparing Classifiers)

[**8 points**] This exercise will use data in `hw04-trn-data.csv` and `hw04-tst-data.csv` which are train and test datasets respectively. Both datasets contain multiple predictors and a categorical response `y`.

The possible values of `y` are `"dodgerblue"` and `"darkorange"` which we will denote mathematically as B (for blue) and O (for orange).

Consider four classifiers.

$$\hat{C}_1(x) = \begin{cases} B & x_1 > 0 \\ O & x_1 \leq 0 \end{cases}$$

$$\hat{C}_2(x) = \begin{cases} B & x_2 > x_1 + 1 \\ O & x_2 \leq x_1 + 1 \end{cases}$$

$$\hat{C}_3(x) = \begin{cases} B & x_2 > x_1 + 1 \\ B & x_2 < x_1 - 1 \\ O & \text{otherwise} \end{cases}$$

$$\hat{C}_4(x) = \begin{cases} B & x_2 > (x_1 + 1)^2 \\ B & x_2 < -(x_1 - 1)^2 \\ O & \text{otherwise} \end{cases}$$

Obtain train and test error rates for these classifiers. Summarize these results using a single well-formatted table.

- Hint: Write a function for each classifier.
- Hint: The `ifelse()` function may be extremely useful.

```
train_data = read.csv("hw04-trn-data.csv")
test_data = read.csv("hw04-tst-data.csv")
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```r
classified = function(x, boundary1, boundary2, nums,above = "dodgerblue", below = "darkorange") {
  if (nums == 1){
    ifelse(x > boundary1, above, below)
  }
  else{
    ifelse(x > boundary1 | x < boundary2, above, below)
  }
}
error = function(x,b1, b2, nums, above,below, actual)
{
  pred = classified(x, b1, b2, nums, above, below)
  tab = table(predicted = pred, actual = actual)
  con_mat  = confusionMatrix(tab, positive = "dodgerblue")
  1 - con_mat$overall["Accuracy"]
}
```

```r
trn_err1 = error(train_data$x1, 0, 0, 1, "dodgerblue", "darkorange", train_data$y)

tst_err1 = error(test_data$x1, 0, 0, 1, "dodgerblue", "darkorange", test_data$y)


trn_err2 = error(train_data$x2, train_data$x1 + 1, 0, 1,
                 "dodgerblue", "darkorange", train_data$y)

tst_err2 = error(test_data$x2, test_data$x1 + 1, 0, 1, "dodgerblue", "darkorange", test_data$y)


trn_err3 = error(train_data$x2, train_data$x1 + 1,
                 train_data$x1 - 1, 2, "dodgerblue", "darkorange", train_data$y)

tst_err3 = error(test_data$x2, test_data$x1 + 1, test_data$x1 - 1,
                 2, "dodgerblue", "darkorange", test_data$y)

trn_err4 = error(train_data$x2, (train_data$x1 + 1)^2,
                 -(train_data$x1 - 1)^2, 2, "dodgerblue", "darkorange", train_data$y)

tst_err4 = error(test_data$x2, (test_data$x1 + 1)^2, -(test_data$x1 - 1)^2,
                 2, "dodgerblue", "darkorange", test_data$y)
```

```r
df = data.frame(
  classifer = c("1", "2", "3", "4"),
  train_error = c(trn_err1, trn_err2, trn_err3, trn_err4),
  test_error = c(tst_err1, tst_err2, tst_err3, tst_err4)
  )
knitr::kable(df)
```

classifer	train_error	test_error
1	0.468	0.5160
2	0.216	0.2240
3	0.096	0.1270
4	0.050	0.0665

Exercise 2 (Creating Classifiers with Logistic Regression)

[**8 points**] We'll again use data in hw04-trn-data.csv and hw04-tst-data.csv which are train and test datasets respectively. Both datasets contain multiple predictors and a categorical response `y`.

The possible values of `y` are `"dodgerblue"` and `"darkorange"` which we will denote mathematically as B (for blue) and O (for orange).

Consider classifiers of the form

$$\hat{C}(x) = \begin{cases} B & \hat{p}(x) > 0.5 \\ O & \hat{p}(x) \le 0.5 \end{cases}$$

Create (four) classifiers based on estimated probabilities from four logistic regressions. Here we'll define $p(x) = P(Y = B \mid X = x)$.

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0$$

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2$$

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

Note that, internally in `glm()`, `R` considers a binary factor variable as `0` and `1` since logistic regression seeks to model $p(x) = P(Y = 1 \mid X = x)$. But here we have `"dodgerblue"` and `"darkorange"`. Which is `0` and which is `1`? Hint: Alphabetically.

Obtain train and test error rates for these classifiers. Summarize these results using a single well-formatted table.

```
dodgerblue is 1 while darkorange is 0
```

```r
calc_class_err = function(actual, predicted) {
  mean(actual != predicted)
}
class_err = function(model,act, data){
  model_pred = ifelse(predict(model, newdata = data, type = "response") > 0.5,
                      "dodgerblue", "darkorange")
  calc_class_err(act, model_pred)
}
```

```r
model1 = glm(y ~ 1, data = train_data, family = "binomial")
train_err1 = class_err(model1, train_data$y, data = train_data)
test_err1 = class_err(model1, test_data$y, data = test_data)
```

```r
model2 = glm(y ~ x1 + x2, data = train_data, family = "binomial")
train_err2 = class_err(model2, train_data$y, data = train_data)
test_err2 = class_err(model2, test_data$y, data = test_data)
```

```
model3 = glm(y ~ x1 + x2 + I(x1^2) + I(x2^2), data = train_data, family = "binomial")
train_err3 = class_err(model3, train_data$y, data = train_data)
test_err3 = class_err(model3, test_data$y, data = test_data)
```

```
model4 = glm(y ~ I(x1^2) + I(x2^2) + x1*x2, data = train_data, family = "binomial")
train_err4 = class_err(model4, train_data$y, data = train_data)
test_err4 = class_err(model4, test_data$y, data = test_data)
```

```
md1 = "y ~ 1"
md2 = "y ~ x1 + x2"
md3 = "y ~ x1 + x2 + x1^2 + x2^2"
md4 = "y ~ x1 + x2 + x1^2 + x2^2 + x1*x2"
df1 = data.frame(
  model = c(md1,md2,md3,md4),
  train_error = c(train_err1, train_err2, train_err3, train_err4),
  test_error = c(test_err1, test_err2, test_err3, test_err4)
  )
knitr::kable(df1)
```

model	train_error	test_error
y ~ 1	0.334	0.3305
y ~ x1 + x2	0.334	0.3305
y ~ x1 + x2 + x1^2 + x2^2	0.320	0.3500
y ~ x1 + x2 + x1^2 + x2^2 + x1*x2	0.080	0.1110

Exercise 3 (Bias-Variance Tradeoff, Logistic Regression)

[**8 points**] Run a simulation study to estimate the bias, variance, and mean squared error of estimating $p(x)$ using logistic regression. Recall that $p(x) = P(Y = 1 \mid X = x)$.

Consider the (true) logistic regression model

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = 1 + 2x_1 - x_2$$

To specify the full data generating process, consider the following R function.

```
make_sim_data = function(n_obs = 25) {
  x1 = runif(n = n_obs, min = 0, max = 2)
  x2 = runif(n = n_obs, min = 0, max = 4)
  prob = exp(1 + 2 * x1 - 1 * x2) / (1 + exp(1 + 2 * x1 - 1 * x2))
  y = rbinom(n = n_obs, size = 1, prob = prob)
  data.frame(y, x1, x2)
}
```

So, the following generates one simulated dataset according to the data generating process defined above.

```
sim_data = make_sim_data()
```

Evaluate estimates of $p(x_1 = 1, x_2 = 1)$ from fitting three models:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

Use 1000 simulations of datasets with a sample size of 25 to estimate squared bias, variance, and the mean squared error of estimating $p(x_1 = 1, x_2 = 1)$ using $\hat{p}(x_1 = 1, x_2 = 1)$ for each model. Report your results using a well formatted table.

At the beginning of your simulation study, run the following code, but with your nine-digit Illinois UIN.

```
set.seed(650178134)
```

```
get_bias_square = function(estimate, truth) {
  (mean(estimate) - truth)^2
}

get_var = function(estimate) {
mean((estimate - mean(estimate)) ^ 2)
}
get_mse = function(truth, estimate) {
mean((estimate - truth) ^ 2)
}
```

```
data = data.frame(x1 = 1, x2 = 1)
```

```
n_sims = 1000
n_models = 3
predictions = matrix(0, nrow = n_sims, ncol = n_models)
for(sim in 1:n_sims) {

  # simulate new, random, training data
  # this is the only random portion of the bias, var, and mse calculations
  # this allows us to calculate the expectation over D
  sim_data = make_sim_data()

  # fit models
  m1 = glm(y ~ 1, data = sim_data, family = "binomial")
  m2 = glm(y ~ x1 + x2, data = sim_data, family = "binomial")
  m3 = glm(y ~ I(x1^2) + I(x2^2) + x1*x2, data = sim_data, family = "binomial")

  # get predictions
  predictions[sim, 1] = predict(m1,newdata = data, type = "response")
  predictions[sim, 2] = predict(m2,newdata = data, type = "response")
  predictions[sim, 3] = predict(m3,newdata = data, type = "response")
}
```

```
f = function(x1, x2){
  exp(1 + 2 * x1 - x2) / (1 + exp(1 + 2*x1 - x2))
}
real = f(data$x1, data$x2)
```

```r
bias = apply(predictions, 2, get_bias_square, truth = real)
Variance = apply(predictions, 2, get_var)
mse = apply(predictions, 2, get_mse, truth = real)
df2 = data.frame(
model = c(md1,md2,md4),
bias_sqaured = round(bias , 5),
Variance = round(Variance, 5),
MSE = round(mse, 5)
)
knitr::kable(df2)
```

model	bias_sqaured	Variance	MSE
y ~ 1	0.04989	0.00943	0.05932
y ~ x1 + x2	0.00001	0.01017	0.01018
y ~ x1 + x2 + x1^2 + x2^2 + x1*x2	0.00031	0.02579	0.02609

Exercise 4 (Concept Checks)

[**1 point each**] Answer the following questions based on your results from the three exercises.

(a) Based on your results in Exercise 1, do you believe that the true decision boundaries are linear or non-linear?

non-linear since the test error decreases when it is non-linear boundaries, and classifer 4 with non-linear boundary has the smallest value of error

(b) Based on your results in Exercise 2, which of these models performs best?

The last model with maximum number of parameter, since the error is smallest

(c) Based on your results in Exercise 2, which of these models are underfitting?

Model 1 2 3 are underfitting

(d) Based on your results in Exercise 2, which of these models are overfitting??

There is no model overfitting among the three models

(e) Based on your results in Exercise 3, which models are performing unbiased estimation?

The second and the third one, since they have small value of bias squared

(f) Based on your results in Exercise 3, which of these models performs best?

the second one. since the mse is smallest among the three models.