Homework 8

Name: Put your name here

I worked with:

Click the "Knit" button in RStudio to knit this file to a pdf.

Problem 1: Random guessing

Suppose you have n observations and you would like to construct a classifier to predict positive outcomes. We know that n_N are actually negative cases and n_P are actually positive cases.

Suppose we are a lazy data scientist and don't actually build a model with predictors to classify our cases. Instead, we use a fair coin to classify negative and positive cases! This "coin classifier" will result in an even split between *expected* positive and negative predictions:

$$\hat{n}_N = \hat{n}_P = \frac{n}{2}$$

Using the same logic, half of the actual failures will be successes and failures:

$$TN = FP = \frac{n_N}{2}$$

and half of the actual successes will be successes and failures:

$$FN = TP = \frac{n_P}{2}$$

confusion matrix	truth negative	truth positive	total
predict negative	$TN = \frac{n_N}{2}$	$FN = \frac{n_P}{2}$	$\hat{n}_N = \frac{n}{2}$
predict positive	$FP = \frac{n_N}{2}$	$TP = \frac{n_P}{2}$	$\hat{n}_P = \frac{n}{2}$
total	n_N	n_P	n

a.

Answer: Your answer here

```
# Your code here
```

b.

Answer: Your answer here

```
# Your code here
```

c.

Answer: Your answer here

```
# Your code here
```

d.

Answer: Your answer here

```
# Your code here
```

Problem 2: Spam using k-nn

This example looks at a data set of about 4600 emails that are classified as spam or not spam, along with over 50 variables measuring different characteristic of the email. Details about these variables are found on the Spambase example on the machine learning data archive. The dataset linked to below is a slightly cleaned up version of this data. The only extra column in the data is `rgroup` which is a randomly assigned grouping variable (groups 0 through 99) which we will eliminate from the data.

Read the data in using the commands below to create a response `class` variable that contains the factor levels `spam` and `nonspam` with `spam` the first level.

```
# tsv = tab separated values!
spam <- read_delim("http://math.carleton.edu/kstclair/data/spamD.txt",
        delim="\t")

# some clean up
spam <- spam %>%
  mutate(class = fct_recode(
    spam,
    spam = "spam" ,
    nonspam = "non-spam"), # rename levels because caret doesn't like "non-spam"
    class = fct_relevel(class, "spam") # make "spam" the first level (our "positive")
    ) %>%
  select(-rgroup, -spam) # don't need random group variable and  spam variable
levels(spam$class)  # verify "spam" is level 1
## [1] "spam"    "nonspam"
```

a.

Answer: Your answer here

```
# Your code here
```

b.

```
set.seed(757302859)  # set a seed
```

Answer: Your answer here

```
# Your code here
```

c.

Make a recipe for fitting k nearest-neighbor algorithm to the training data by inputting the formula and the preprocessing steps.

Answer: Your answer here

```
# Your code here
```

d.

Answer: Your answer here

```
# Your code here
```

e.

Answer: Your answer here

```
# Your code here
```

f.

Answer: Your answer here

```
# Your code here
```

g.

Answer: Your answer here

```
# Your code here
```

h.

Use the `tidymodels` package to do 10-fold cross validation as follows:

- use the 80% training data split from part b.
- tune your knn spam classifier based on accuracy
- consider neighborhood sizes ranging from size 1 to 31

Use the `results` to get the training set cross-validated estimates of the accuracy, precision, sensitivity and specificity of your final ("best") classifier.

And use the following seed before running your `train` command:

```
set.seed(30498492)
```

Answer: Your answer here

```
# Your code here
```

i.

Answer: Your answer here

```
# Your code here
```

Problem 3: Incoming student characteristic

We will look at a "classic" college data set of a random sample of colleges and universities. To simplify our look at this data, we will filter to only look at MN, MA, and CA schools

```
colleges <- read_csv("http://math.carleton.edu/kstclair/data/Colleges.csv")
names(colleges)
##  [1] "State"       "College"     "SATM"        "SATV"        "AppsReceive"
##  [6] "AppsAccept"  "HStop10"     "HStop25"     "FullTime"    "Tuition"
## [11] "RoomBoard"   "Books"       "Ratio"       "Donate"      "Expend"
## [16] "GradRate"    "Type"        "AvgSalary"   "NumFaculty"
colleges2 <- colleges %>%
  filter(State %in% c("MN","MA","CA"))
colleges2 %>% count(State)
## # A tibble: 3 x 2
##   State     n
##   <chr> <int>
## 1 CA       21
## 2 MA       19
## 3 MN       11
```

We will also just focus on student body characteristics (incoming class averages) for SAT and the HS variables (which are are the proportion of the incoming class that is in the top 10% or 25% of their HS class). Here we select just these characteristics and college name and state.

```
colleges2 <- colleges2 %>% select(1,2,3,4,7,8)
colleges2
## # A tibble: 51 x 6
##    State College                          SATM  SATV HStop10 HStop25
##    <chr> <chr>                           <dbl> <dbl>   <dbl>   <dbl>
##  1 CA    California Institute of Technolo   750   660      98     100
##  2 CA    California Lutheran University     495   436      23      52
```

```
##  3 CA    California Polytechnic-San Luis    547   455      47      73
##  4 CA    Chapman University                 501   456      23      48
##  5 CA    Claremont McKenna College          670   600      71      93
##  6 CA    Harvey Mudd College                740   630      95     100
##  7 CA    Pitzer College                     590   560      37      73
##  8 CA    Pomona College                     700   640      80      98
##  9 CA    Scripps College                    590   560      60      83
## 10 CA    Occidental College                 570   510      52      81
## # ... with 41 more rows
```

Let's cluster schools by their incoming class characteristics.

(a)

Answer: Your answer here

```
# Your code here
```

(b)

Answer: Your answer here

```
# Your code here
```

(c)

Answer: Your answer here

```
# Your code here
```

(d)

Answer: Your answer here

```
# Your code here
```

(e)

Answer: Your answer here

```
# Your code here
```

(f)

Answer: Your answer here

```
# Your code here
```

(g)

```
library(GGally)   # install if needed
colleges2 %>%
  ggpairs(aes(color = cluster_km3),
          columns=c("SATM", "SATV", "HStop10", "HStop25"))
```

Answer: Your answer here

```
# Your code here
```