Homework 3

In this homework, you should expect the code for some problems to take a couple of minutes to complete. Re-knitting your file can take a long time so you should consider using `cache=TRUE` option in R chunks that involve code which takes a while to run. Please load the following packages for this homework:

```
library(tidyverse)
library(ISLR)
library(glmnet)
library(tree)
library(maptree)
library(randomForest)
library(gbm)
library(ROCR)
```

Predicting carseats sales using regularized regression methods

We will use the `Carseats` dataset in the `ISLR` package. The dataset contains 400 observations on 11 variables. In this question, we will apply regularized regression methods in order to predict `Sales` using all other predictors.

The following code randomly split `Carseats` into a training set consisting of only 30 observations (in order to make a small n case) and a test set consisting of the remaining observations. We use `set.seed(123)` in the beginning of the R code chunk to ensure reproducibility, and use `model.matrix` to construct the design matrix for appropriate dummy encoding for categorical predictors.

```
set.seed(123)

dat <- model.matrix(Sales~., Carseats)
train = sample(nrow(dat), 30)
x.train = dat[train, ]
y.train = Carseats[train, ]$Sales

# The rest as test data
x.test = dat[-train, ]
y.test = Carseats[-train, ]$Sales
```

(a). (2 pts) Fit a ridge regression model to the training set to predict `Sales` using all other variables as predictors. Use the built-in cross-validation in `cv.glmnet` to choose the optimal value of tuning parameter λ from the following list of λ values using a 5-fold CV. (2 pts) Report the ridge coefficient estimates corresponding to the selected value of λ.

```
lambda.list.ridge = 1000 * exp(seq(0, log(1e-5), length = 100))
```

(b). (2 pts) What is the training MSE for the model corresponding to the optimal value of λ selected by the cross-validation above? (2 pts) What is the test MSE for that same model? (1 pts) Comment on your findings.

(c). (2 pts) Fit a lasso model to the training set to predict `Sales` using all other variables as predictors. Use the built-in cross-validation in `cv.glmnet` to choose the optimal value of tuning parameter λ from the following list of λ

values using a 10-fold CV. (2 pts) Report the lasso coefficient estimates corresponding to the selected value of λ. (2 pts) Are there any coefficients set to zero in the model selected by cross-validation? Comment on your findings.

```
lambda.list.lasso = 2 * exp(seq(0, log(1e-4), length = 100))
```

(d). (2 pts) What is the training MSE for the lasso model corresponding to the optimal value of λ selected by cross-validation? (2 pts) What is the test MSE for that same model? (1 pts) Comment on your findings.

(e). (2 pts) Comment on the comparison between ridge and lasso estimates in this application.

Analyzing drug use

In this homework, we will apply several classification methods that we have covered in this course in analyzing the drug use data (`drug.csv` file attached in the Homework Assignment 3 on GauchoSpace). The data set includes a total of 1885 observations on 32 variables. A detailed description of the data set can be found here. Each row of the data contains observations of the following predictors:

- ID: number of record in original database. Used for reference only.
- Age: Age of the participant
- Gender: Gender of the participant (M/F)
- Education: Level of education of the participant
- Country: Country of current residence of the participant
- Ethnicity: Ethnicity of the participant
- Nscore: NEO-FFI-R Neuroticism (Ranging from 12 to 60)
- Escore: NEO-FFI-R Extraversion (Ranging from 16 to 59)
- Oscore: NEO-FFI-R Openness (Ranging from 24 to 60)
- Ascore: NEO-FFI-R Agreeableness (Ranging from 12 to 60)
- Cscore: NEO-FFI-R Conscientiousness (Ranging from 17 to 59)
- Impulsive: Impulsiveness measured by BIS-11
- SS: Sensation Seeking measured by ImpSS

In addition to these predictors, participants were also questioned concerning their use of 18 legal and illegal drugs (alcohol, amphetamines, amylnitrite, benzodiazepine, cannabis, chocolate, cocaine, caffeine, crack, ecstasy, heroin, ketamine, legal highs, LSD, methadone, mushrooms, nicotine and volatile substance abuse) and one fictitious drug (Semeron) which was introduced to identify over-claimers. All of the drugs use the class system of CL0-CL6:

- CL0 = "Never Used"
- CL1 = "Used over a decade ago"
- CL2 = "Used in last decade"
- CL3 = "Used in last year"
- CL4 = "Used in last month"
- CL5 = "Used in last week"
- CL6 = "Used in last day".

The following code loads in the data and give proper names to each of the predictors:

```
drug <- read_csv('drug.csv',
                 col_names=c('ID','Age','Gender','Education','Country',
                             'Ethnicity','Nscore',
                             'Escore','Oscore','Ascore','Cscore',
                          'Impulsive','SS','Alcohol','Amphet','Amyl','Benzos',
                             'Caff','Cannabis', 'Choc','Coke','Crack','Ecstasy',
                             'Heroin','Ketamine','Legalh','LSD','Meth',
                             'Mushrooms','Nicotine','Semer','VSA'))
```

(a). (2 pts) Define a new factor response variable `recent_cannabis_use` which is "Yes" if a person has used cannabis within a year, and "No" otherwise. This can be done by checking if the `Cannabis` variable is *greater than or equal* to `CL3`. Hint: use `mutate` with the `ifelse` command. When creating the new factor set `levels` argument to `levels=c("No", "Yes")` (in that order).

(b). (2 pts) We will only consider a subset of all predictors in subsequent tasks. To do so, we will create a new dataset that includes a subset of the original predictors. In particular, we will focus on all variables between `age` and `SS` (inclusively) as well as the new factor `recent_cannabis_use` you obtained in part (a).

(c). (2 pts) Split the dataset you obtained in part (b) into a training data set and a test data set. The training data should include 1100 randomly sampled observation and the test data should include the remaining observations. You will need the training and the test data for subsequent analysis.

(d). (4 pts) As a benchmark method, fit a logistic regression to predict `recent_cannabis_use` using all other predictors in the training data you obtained in (c). Display the results by calling the `summary` function on the logistic regression object.

(e). (2 pts) Construct a single decision tree to predict `recent_cannabis_use` using all other predictors in the training data.

(f). (2 pts) Use 5-fold cross-validation to select the best size of a tree which minimizes the cross-validation estimate of the test error rate. Use the function `cv.tree`, and set the argument `FUN=prune.misclass`. If multiple trees have the same minimum cross validated error rate, set `best_size` to the smallest tree size with that minimum rate. (2 pts) Report the best size you obtained.

(g). (2 pts) Prune the tree to the best size selected in the previous part and plot the tree using the `draw.tree` function from the `maptree` package (see Lab 6). Set `nodeinfo=TRUE`. (2 pts) Which variable is split first in this decision tree?

(h). (2 pts) Compute and print the confusion matrix for the *test* data using the function `table`. (Hint: Recall that the `table` function takes in two arguments: the first argument is the true classes, and the second argument is the predicted classes. To generate the predicted classes for the test data, set `type="class"` in the `predict` function.) (2 pts) Calculate the true positive rate (TPR) and false positive rate (FPR) for the confusion matrix. Show how you arrived at your answer.

(i). (2 pts) Fit a boosting model to the training set with `recent_cannabis_use` as the response and the other variables as predictors. Use the `gbm` to fit a 1,000 tree boosted model and set the shrinkage value of 0.01. (2 pts) Which predictors appear to be the most important (Hint: use the `summary` function)?

(j). (2 pts) Now fit a random forest model to the same training set from the previous problem. Set `importance=TRUE` but use the default parameter values for all other inputs to the `randomForest` function. Print the random forest object returned by the random forest function. (1 pts) What is the out-of-bag estimate of error? (1 pts) How many variables were randomly considered at each split in the trees? (1 pts) How many trees were used to fit the data? Look at the variable importance. (1 pts) Is the order of important variables similar for both boosting and random forest models?

(k). (2 pts) Use both models to predict the response on the test data with a certain threshold. Predict that a person will have `recent_cannabis_use = Yes` if the predicted probability of `recent_cannabis_use` is greater than or equal to 20%. (2 pts) Print the confusion matrix for both the boosting and random forest models. (2 pts) In the random forest model, what fraction of the people predicted to use cannabis recently do in fact use cannabis recently? (Hint: use the `predict` function with `type="prob"` for random forests and `type="resonpse"` for the boosting algorithm. See Lab 7).

Problems below for 231 students only

(l). (4 pts) Plot the ROC curves for the logistic regression fit, the best pruned decision tree, the random forest, and the boosting trees that you obtained in previous parts. The ROC curves should be computed using the test data. (Hint: recall in Lab 3 we covered how to plot ROC curves for classification problems.)

(m). (4 pts) Compute the AUC for the four models and print them. Which model has larger AUC?

Using bootstrap to compute uncertainty about a parameter of interest (8 pts)

In the 2020-2021 season, Stephen Curry, an NBA basketball player, had made 337 out of 801 three point shot attempts (42.1%). Use bootstrap resampling on a sequence of 337 1's (makes) and 464 0's (misses). For each bootstrap sample compute and save the sample mean (e.g. bootstrap FG% for the player). Use 1000 bootstrap samples to plot a histogram of those values. Compute the 99% bootstrap confidence interval for Stephen Curry's "true" end-of-season FG% using the `quantile` function in R. Print the endpoints of this interval.