STA 314H1S: Problem Set 4

Except for question 4, the questions on this assignment are practice for the quiz on Friday, and are not to be handed in.
For question 4, please bring a printout of your output together with the source code. You will be asked to submit this printout at the end of the quiz.

1. (*Textbook Exercises*) Answer questions 1,2,4,5,6 from section 8.4, and questions 1,2,3 from section 9.7.

2. (*Support Vector Machine*) Recall the SVM classifier and the kernel trick that was used to construct it.

   (a) Show that if $\mathbf{A} \in \mathbb{R}_{p\times p}$ is any PD matrix, then $K(\mathbf{z_1}, \mathbf{z_2}) = \mathbf{z_1}^T\mathbf{A}\mathbf{z_2}$ is a valid inner-product for $\mathbf{z_1}, \mathbf{z_2} \in \mathbb{R}^p$.

   (b) Show that if $\mathbf{x} \in \mathbb{R}^p$ is a random vector with zero mean and covariance Σ_x, then $K(\mathbf{z_1}, \mathbf{z_2}) = \text{Cov}(\mathbf{z_1}^T\mathbf{x}, \mathbf{z_2}^T\mathbf{x})$ is a valid inner-product for $\mathbf{z_1}, \mathbf{z_2} \in \mathbb{R}^p$.

   (c) For any $\mathbf{z} \in \mathbb{R}^p$, suppose $\mathbf{\Phi}(\mathbf{z}) = (\phi_1(\mathbf{z}), ..., \phi_q(\mathbf{z}))^T \in \mathbb{R}^q$, can we conclude that $K(\mathbf{z_1}, \mathbf{z_2}) = \mathbf{\Phi}(\mathbf{z_1})^T\mathbf{\Phi}(\mathbf{z_2})$ is a valid inner-product?

   (d) Show that the kernel function $K(\mathbf{z_1}, \mathbf{z_2}) = (\mathbf{z_1}^T\mathbf{z_2})^2$ that we learned in the lecture, is not a inner product.

   (e) Describe why the Mercer's theorem implies the SVM is more useful in practice than the SVC with feature mapping?

   (f) Describe the difference between the SVC and the maximal marginal classifier.

   (g) Besides the budget parameter C, the choice of kernel in the SVM also controls the bias-variance tradeoff. Briefly explain why that is the case.

3. (*Decision Tree*) Suppose you collected data on the survival status of 6 passengers on the Titanic. The data includes their gender, age, weight, and height. The dataset is summarized in the table below.
   Using this data, you built a decision tree to predict survival status.

Passenger	Survived	Gender	Age	Weight (kg)	Height (cm)
1	Yes	Male	22	70	170
2	No	Female	30	55	160
3	No	Male	45	78	175
4	Yes	Female	25	60	166
5	No	Male	55	80	172
6	No	Female	35	58	158

Table 1: Dataset of Titanic passengers

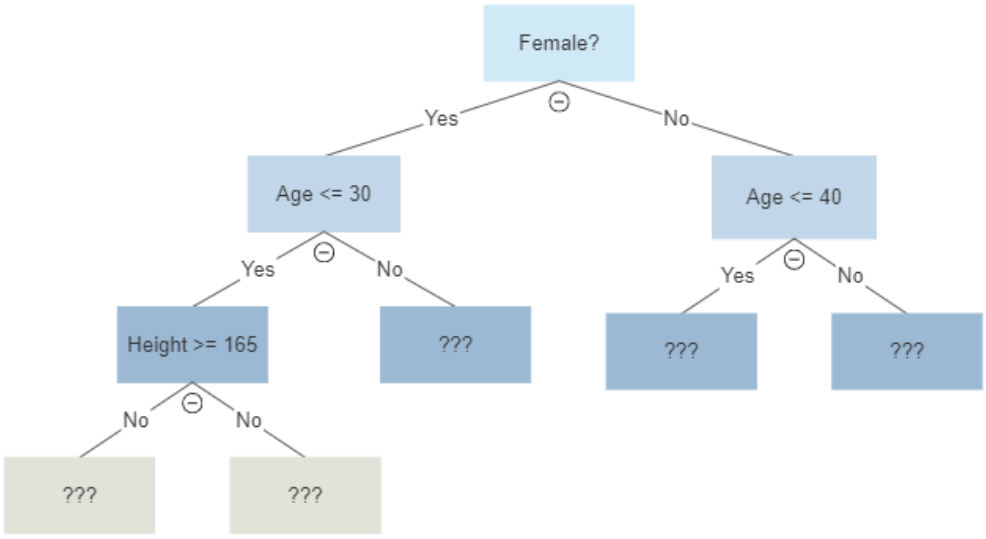

Figure 1: Decision tree for predicting survival on the Titanic.

(a) How many leaves does your decision tree have? How many observations are in each leaf?

(b) Suppose your decision tree is bulit by minizing the overall error rate in each leaf, fill in the missing classification results in each terminal leaf.

(c) Predict the survival status of the following passengers based on your decision tree and explain your reasoning:

- A 50-year-old male who weighs 76kg and is 172cm tall.
- A 28-year-old female who weighs 58kg and is 160cm tall.

(d) Critically discuss whether the decision tree might be prone to overfitting or underfitting based on the provided dataset and explain your reasoning.

4. (*Computation*) In this question, we will consider the `lung` data from the `survival` package in R. We are aiming to build a model to predict the survival time `time` using three features: `age`, `sex` (Male=1 Female=2) and `wt.loss` (the loss of weight in pounds, in the last six months). You have just received a new patient with feature $\mathbf{x}_{new} = (\texttt{age} = 21, \texttt{sex} = 2, \texttt{wt.loss} = 5)$.

(a) To start with, remove the observations with missing values, and separate the data into a training set with size 200, and the rest of the data to the testing set.
Some starter code to help you get started:

```
library(tidyverse)

#### The dataset:
data <- survival::lung %>%
  select(time, age, sex, wt.loss) %>%
```

```
  drop_na()

### Separation:
set.seed(123)
training_data <- #TODO
testing_data <- #TODO
```

(b) Now, consider fitting a linear regression model with these features, and use it to predict the survival time for a female patient with `age` $= 21$ and `wt.loss` $= 5$, called $\hat{y}(\mathbf{x}_{new})$.
Some starter code to help you get started:

```
### Linear regression:
mod1 <- #TODO
new_data <- #TODO
mod1_pred <- predict(#TODO)
mod1_pred
```

(c) To compute $\text{SD}(\hat{y}(\mathbf{x}_{new})) = \sqrt{\text{Var}(\hat{y}(\mathbf{x}_{new}))}$, we can no longer do it by simulating more datasets from the true data generating process as we have done in the PS1. However, the bootstrapping method allows us to estimate $\text{SD}(\hat{y}(\mathbf{x}_{new}))$ just using the single training set. Use $B = 100$ bootstrap samples to estimate $\text{SD}(\hat{y}(\mathbf{x}_{new}))$.
Some starter code to help you get started:

```
### Bootstrapping for variance of `mod1_pred`
B <- #TODO
## The vector of yhat constructed from each bootrap sample
yhat_vector <- c()
for (i in 1:B) {
  index <- #TODO
  training_data_i <- #TODO
  boot_mod_i <- #TODO
  yhat_vector[i] <- as.numeric(predict(boot_mod_i, new_data))
}
sd(yhat_vector)
```

(d) Now, consider fitting a regression tree to this data.

```
### Try regression tree:
library(tree)
tree_mod <- tree(time ~ age + sex + wt.loss, data = training_data
    ,
                 split = "deviance",
                 control = tree.control(nobs = nrow(training_data
    ), mincut = 1, minsize = 2, mindev = #TODO))
summary(tree_mod)
mean(summary(tree_mod)$residuals^2)
```

Try to vary the value of `mindev` from 1 to 0, and see how the structure of the tree changes, as well as how the residual sum of squares changes.

(e) Now, suppose you want to use the regression tree above to obtain the prediction $\hat{y}(\mathbf{x}_{new})$. Estimate $\text{SD}(\hat{y}(\mathbf{x}_{new}))$ of your regression tree using $B = 100$ bootstrap samples. Consider both `mindev` $= 1$ and `mindev` $= 0$, what are their estimated $\text{SD}(\hat{y}(\mathbf{x}_{new}))$ respectively?
Some starter code to help you get started:

```
set.seed(123)
### Bootstrapping for variance of a tree with mindev = 0
B <- 100
## The vector of yhat constructed from each bootrap sample
yhat_vector1 <- c()
for (i in 1:B) {
  training_data_i <- #TODO
  tree_mod_i <- tree(time ~ age + sex + wt.loss, data = #TODO,
                     split = "deviance",
                     control = tree.control(nobs = nrow(training_
    data_i), mincut = 1, minsize = 2, mindev = 0))
  yhat_vector1[i] <- as.numeric(predict(tree_mod_i, new_data))
}
sd(yhat_vector1)

### Bootstrapping for variance of a tree with mindev = 1
B <- 100
## The vector of yhat constructed from each bootrap sample
yhat_vector2 <- c()
for (i in 1:B) {
  training_data_i <- #TODO
  tree_mod_i <- tree(time ~ age + sex + wt.loss, data = #TODO,
                       split = "deviance",
                       control = tree.control(nobs = nrow(training_
    data_i), mincut = 1, minsize = 2, mindev = 1))
  yhat_vector2[i] <- as.numeric(predict(tree_mod_i, new_data))
}
sd(yhat_vector2)
```

(f) Suppose you want to use Bagging to construct an ensemble prediction using multiple trees, complete the following R function:

```
### Bagging:
bagging <- function(tree_num = 100, mindev = 0, training_data,
    testing_data){
  # Initialize the prediction result yhat_vector
  yhat_vector <- matrix(nrow = nrow(testing_data), ncol = tree_
    num)
  for (i in 1:tree_num) {
    # Compute the prediction of each bootstrapped tree
    training_data_i <- # TODO
    tree_mod_i <- tree(time ~ age + sex + wt.loss, data =
    training_data_i,
                         split = "deviance",
                         control = tree.control(nobs = nrow(
    training_data_i), mincut = 1, minsize = 2, mindev = mindev))
```

```
    yhat_vector[,i] <- as.numeric(predict(tree_mod_i, testing_
    data))
  }
  ### Compute the final estimate as the sample mean
  return(apply(#TODO))
}
```

(g) Using your R function above, estimate $\text{SD}(\hat{y}(\mathbf{x}_{new}))$ of your Bagging prediction where `tree_num` $= 100$ and `mindev` $= 0$, using $B = 100$ bootstrap samples. Compare your answer to part (e).

(h) Compute the empirical testing MSE of your bagging prediction from (g) with `tree_num` $= 100$ and `mindev` $= 0$, of the single regression tree with `mindev` $= 0$ from part (e), and of the linear regression model from part (b), using the reserved testing data.