Assignment 4.1

In this assignment, you will apply your knowledge and understanding of the topics presented in the module readings and materials.

Instructions:

1. Answer the assignment questions on the following pages.
2. Create a single document that combines your solutions to all question prompts.

Final Format for Submission:

Submit one assignment file using these guidelines:

- The file must be either a **Microsoft Word** or a **PDF** document file.
- **Do not** combine and submit files into a zipped compressed folder.
- Please use the following naming conventions for your **Word** or **PDF** assignment file:
  File Naming: **`LastName_FirstName_Assignment<Number>.pdf`**
  Example: **`Smith_James_Assignment4.1.pdf`**
- Answer all parts of a question in one place and answer questions in the order they appear in the assignment.
- For programming answers using **R**, it is recommended that the answers are written in R Markdown and 'knitted' to a Word/PDF file.
  - Do not print data frames in your submission, if you want to make a point about data you can use head(df) to print the first few rows.
  - Submit the code used to answer the questions in the assignment with your name on it, answers without code and appropriate results will not get full credit.
  - It is not a professional practice, but in case of difficulty, you can take screenshots of code and outputs and submit them in a Word/PDF file.
- Maximum number of pages should be 15. Any submissions that exceed 15 pages will **not be graded.**

Grading and Scoring:

- Use common sense to gauge the expectations of the answer to the number of points assigned to the question. See the Scoring Rubric in Blackboard for details.

University of San Diego
SHILEY-MARCOS SCHOOL OF ENGINEERING

Assignment 4.1 – Questions

1. (20 points) Simulate a single predictor and a nonlinear relationship, such as a sin wave shown in Fig. 7.7 of textbook, and investigate the relationship between the cost, ε, and kernel parameters for a support vector machine model:

   ```
   set.seed(100)
   x <- runif(1000, min = 2, max = 10)
   y <- sin(x) + rnorm(length(x)) * .25
   sinData <- data.frame(x = x, y = y)
   plot(x, y)
   ## Create a grid of x values to use for prediction
    dataGrid <- data.frame(x = seq(2, 10, length = 1000)).
   ```

   a) Fit different models using a radial basis function and different values of the cost (the C parameter) and ε assigning sigma a constant value of 1. To study the impact of each parameter, keep all other parameters constant and vary only one parameter at a time. Plot the fitted curves and discuss what happens when cost and epsilon parameters are varied in terms of overfitting/underfitting.
   For example:

   ```
   library(kernlab)
   rbfSVM <- ksvm(x = x, y = y, data = sinData, kernel ="rbfdot", kpar = list(sigma = 1),
   C = 1, epsilon = 0.1)
   modelPrediction <- predict(rbfSVM, newdata = dataGrid)
   ## This is a matrix with one column. We can plot the
   ## model predictions by adding points to the previous plot
   points(x = dataGrid$x, y = modelPrediction[,1], type = "l", col = "blue") (12 points)
   ```

   b) The σ parameter can be adjusted using the kpar argument, such as kpar = list(sigma = 1). Try different values of σ to understand how this parameter changes the model fit. How do the cost, ε, and σ values affect the model? (8 points)

   For this problem, you will perform your analysis in R.

2. (30 points) Start R and use these commands to load the data:

   ```
   library(AppliedPredictiveModeling)
   data(chemicalManufacturingProcess)
   ```

   The matrix processPredictors contains the 57 predictors (12 describing the input biological material and 45 describing the process predictors) for the 176 manufacturing runs. yield contains the percent yield for each run. This describes the data for a chemical

manufacturing process. Split the data into a training (80%) and a test set (20%). A small percentage of cells in the predictor set contain missing values. Use an imputation function to fill in these missing values in both training and test data sets, also perform centering and scaling. Build and tune nonlinear regression models Neural network, MARS, SVM and KNN.

a) Which nonlinear regression model gives the optimal resampling and test set performance (use RMSE metric to make this determination)? (20 points)
b) What are the ten most important predictors in the optimal nonlinear regression model? Do either the biological or process variables dominate the list? (5 points)
c) How do the important predictors compare to the important predictors from the optimal linear model you built in module 3 assignment? (5 points)

For this problem, you will perform your analysis in R.

3. (20 points) Recreate the simulated data as shown below:

```
library(mlbench)
set.seed(200)
simulated <- mlbench.friedman1(200, sd = 1)
simulated <- cbind(simulated$x, simulated$y)
simulated <- as.data.frame(simulated)
colnames(simulated)[ncol(simulated)] <- "y"
```

In this data set V6 to V10 are columns that are random noise and have no relation to the response.

a) Fit a random forest model to all of the predictors, then estimate the variable importance scores:

```
library(randomForest)
library(caret)
model1 <- randomForest(y ~ ., data = simulated, importance = TRUE,
ntree = 1000)
varImp(model1)
```

Did the random forest model significantly use the uninformative predictors (V6 – V10)? What do you think the impact of having uninformative predictors will be on random forest model performance? (10 points)

b) Now add an additional predictor that is highly correlated with one of the informative predictors, V1.

For example:
simulated$duplicate1 <- simulated$V1 + rnorm(200) * .1
cor(simulated$duplicate1, simulated$V1)
Fit another random forest model to these data. Did the importance score for V1 change? What happens if you add one more predictor to the data set that is also highly correlated with V1 (after this you should have 12 predictors, 10 original and two added correlated predictors) and fit another random forest model again to this data. Did the importance score for V1 change? (10 points)

For this problem, you will perform your analysis in R.

4. (20 points) In stochastic gradient boosting the bagging fraction and learning rate will govern the construction of the trees as they are guided by the gradient. Although the optimal values of these parameters should be obtained through the tuning process, it is helpful to understand how the magnitudes of these parameters affect magnitudes of variable importance. Figure 8.24 provides the variable importance plots for boosting using two extreme values for the bagging fraction (0.1 and 0.9) and the learning rate (0.1 and 0.9) for the solubility data. The left-hand plot has both parameters set to 0.1, and the right-hand plot has both set to 0.9:
    a) Why does the model on the right focus its importance on just the first few of predictors, whereas the model on the left spreads importance across more predictors? (7 points)
    b) Which model do you think would be more predictive of other samples? (7 points)
    c) How would increasing interaction depth affect the slope of predictor importance for either model in Figure 8.24 below? (6 points)

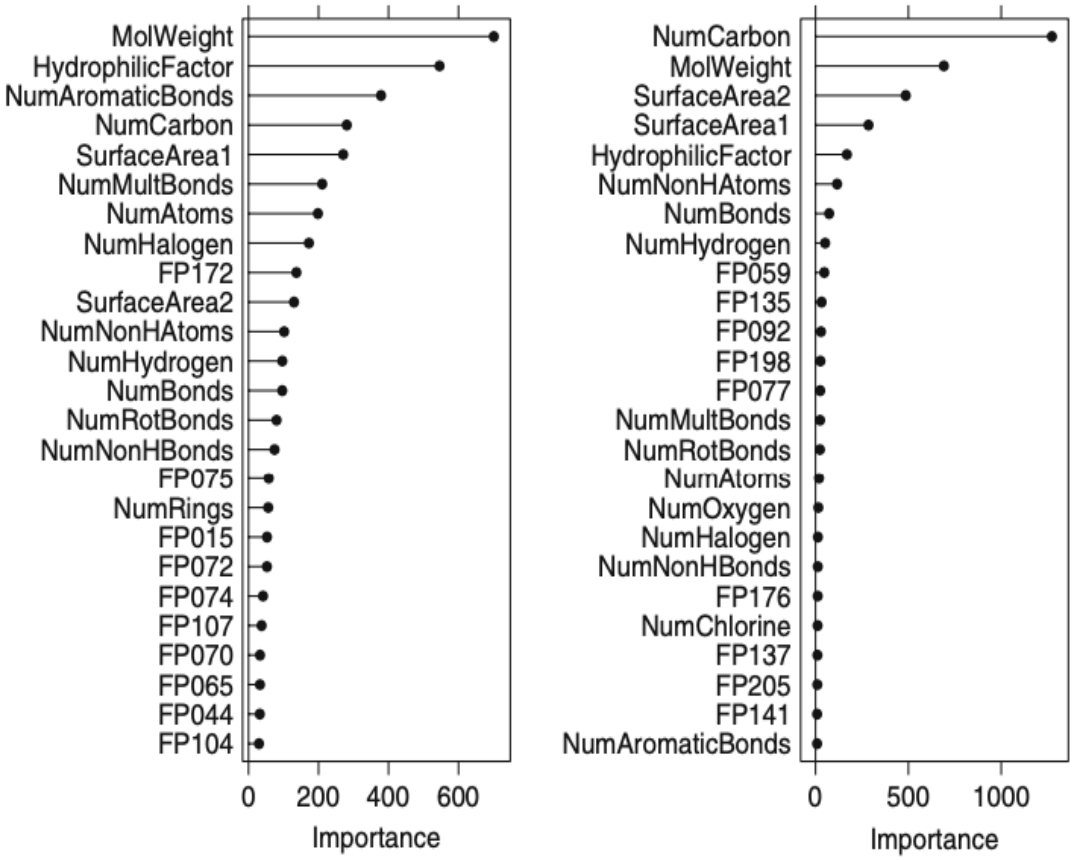

Fig. 8.24: A comparison of variable importance magnitudes for differing values of the bagging fraction and shrinkage parameters. Both tuning parameters are set to 0.1 in the *left* figure. Both are set to 0.9 in the *right* figure