Assigned: Due:
January 29, 2023 Homework 2 February 12, 2023

Please complete the assigned problems to the best of your abilities. Ensure that the work you do is entirely your own, external resources are only used as permitted by the instructor, and all allowed sources are given proper credit for non-original content.

1 Recitation Exercises

These exercises are to be found in: **Introduction to Statistical Learning, 2nd Edition (Online Edition)** by *Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani.*

1.1 Chapter 4

Exercises: 4,6,7,9

1.2 Chapter 5

Exercises: 2,3

2 Practicum Problems

These problems will primarily reference the *lecture materials and the examples given in class* using **R** and **CRAN**. It is suggested that a *RStudio* session be used for the programmatic components.

2.1 Problem 1

Load the *abalone* sample dataset from the UCI Machine Learning Repository (**abalone.data**) into **R** using a dataframe. Remove all observations in the Infant category, keeping the Male/Female classes. Using the **caret** package, use *createDataPartition* to perform an 80/20 test-train split (80% training and 20% testing). Fit a logistic regression using all feature variables via **glm**, and observe which predictors are relevant. Do the confidence intervals for the predictors contain 0 within the range? How does this relate to the null hypothesis? Use the *confusionMatrix* function in **caret** to observe testing results (use a 50% cutoff to tag Male/Female) - how does the accuracy compare to a random classifier ROC curve? Use the **corrplot** package to plot correlations between the predictors. How does this help explain the classifier performance?

2.2 Problem 2

Load the *mushroom* sample dataset from the UCI Machine Learning Repository (**agaricus-lepiota.data**) into **R** using a dataframe (**Note**: There are missing values with a *?* character, you will have to explain your handling of these).

Create a Naive Bayes classifier using the **e1071** package, using the *sample* function to split the data between 80% for training and 20% for testing. With the target class of interest being *edible* mushrooms, calculate the accuracy of the classifier both in-training and in-test. Use the **table** function to create a confusion matrix of predicted vs. actual classes - how many false positives did the model produce?

2.3 Problem 3

Load the *Yacht Hydrodynamics* sample dataset from the UCI Machine Learning Repository (**yacht_hydrodynamics.data**) into **R** using a dataframe (**Note**: The feature labels need to be manually specified). Use the **caret** package to perform a 80/20 test-train split (via the **createDataPartition** function), and obtain a training fit for a linear model. (**Hint**: The model fit should use all available features with the *residuary resistance* as the target.). What are the training MSE/RMSE and R^2 results? Next, use the **caret** package to perform a bootstrap from the full sample dataset with N=1000 samples for fitting a linear model (via the **trainControl** method), resulting in a training MSE/RMSE and R^2 for each resample. Plot a histogram of the RMSE values, and provide a mean RMSE and R^2 for the fit. How do these values compare to the basic model? How does the performance on the test set for the original and boostrap model compare?

2.4 Problem 4

Load the *German Credit Data* sample dataset from the UCI Machine Learning Repository (**german.data-numeric**) into **R** using a dataframe (**Note**: The final column is the class variable coded as 1 or 2). Use the **caret** package to perform a 80/20 test-train split (via the **createDataPartition** function), and obtain a training fit for a logistic model via the **glm** package. (**Hint**: You may select a subset of the predictors based on exploratory analysis, or use all predictors for simplicity.). What are the training Precision/Recall and F_1 results? Next, use the **trainControl** and **train** functions to perform a k=10 fold cross-validation fit of the same model, and obtain cross-validated training Precision/Recall and F_1 values. How do these values compare to the original fit? How does the performance on the test set for the original and cross-validated model compare?