Homework 2

Alec Rospierski

September 23th, 2022

1. Write an R function to predict sales using TV budgets by implementing the k nearest neighbor averaging method. Note that you are not asked to apply an existing R function or package (such as the knn() function) to perform the task. Instead, write your own function. More instructions and helper code is in HW2code_knn.R. The dataset is Advertising.csv, shared on Canvas. Upload your R code, and report the prediction results (i.e., four numbers) in your answer sheet.

```
mk <- read.csv(file="~/development/dsa6000/hw_2/Advertising.csv", header=T)
mk[c(1,3,4)] <- NULL  # Remove the unwanted columns, keep only TV and sales column
head(mk)  # mk should now be a data frame with two columns
```

```
##      TV sales
## 1 230.1  22.1
## 2  44.5  10.4
## 3  17.2   9.3
## 4 151.5  18.5
## 5 180.8  12.9
## 6   8.7   7.2
```

```
# Write an R function to predict the sales given a TV advertising budget value,
# by implementing the nearest neighbor averaging method.
# Specifically, the function should return a numeric value, i.e., the predicted sales given the newTV v
# The predicted sales should be the average of the k sales quantities in the training data whose corres
# TV budget values are closest to the newTV value
# Complete the function body
newsales <- function(newTV, train = mk, k = 3){
  # Your code here:
  train$dist <- abs(newTV - train$TV)
  return(mean(train[order(train$dist), ]$sales[1:k]))
}

# Test code
newsales(200)  # The result should be 13.43
```

```
## [1] 13.43333
```

```
newsales(200, k=5)  # The result should be 15.14
```

```
## [1] 15.14
```

```
# Report your prediction results for the following cases (keep two digits after the decimal point)
sprintf('New Price: 180 k: 1 :: Prediction: %.2f', newsales(newTV = 180, k = 1))
```

```
## [1] "New Price: 180 k: 1 :: Prediction: 12.90"
```

```
sprintf('New Price: 180 k: 3 :: Prediction: %.2f', newsales(newTV = 180, k = 3))
```

```
## [1] "New Price: 180 k: 3 :: Prediction: 17.07"
```

```r
sprintf('New Price: 180 k: 5 :: Prediction: %.2f', newsales(newTV = 180, k = 5))
```

```
## [1] "New Price: 180 k: 5 :: Prediction: 15.62"
```

```r
sprintf('New Price: 180 k: 7 :: Prediction: %.2f', newsales(newTV = 180, k = 7))
```

```
## [1] "New Price: 180 k: 7 :: Prediction: 15.50"
```

2. Run the R code provided in bias_variance_code.pdf, do the Exercises 1 – 3 stated in Lines 74 – 76, and report your findings. (The script below Line 80 provides a solution to Exercise 4, so you do not need to do Exercise 4 separately). While doing the exercise, it is helpful to think about why these questions are asked (or how the exercise serve to enhance understanding of the bias- variance tradeoff). Your answers to the Exercise questions should briefly reflect your learning and understanding of the theory the simulation serves to demonstrate.

Exercise 1: Change the sampsize from 100 to 300, re-run, observe the difference in the output. Explain.

After changing the sample size from 100 to 300, there was a clear difference in the models. For the flexible model, there was a decrease in variance, which we know would also increase the bias. The increase is bias is more noticeable in the simple and moderate models where the simple model became flatter (i.e. the slope decreased). Whereas the moderate model lost some extra curves.

Exercise 2: Increase the flexibility of the model, e.g., using y~ns(x, ...), try df=8, 20 and , observe the difference in the output. Explain.

Increasing the degrees of freedom in each of the models allows the model to more strongly fit the training data. It creates more curves and variance.

Exercise 3: Does picking a different x0 affect the observations?

Picking a different x_0 greatly affects the model. After changing the x_0 value to 25 (down from 50), all models changed their estimates and seem to have more curves and therefore more variance.

3. ISLR 2.4 Question 4 (a) (b) (c). Describe real-life applications in which classification, regression and clustering might be useful. One (not three) case for each method. (Interesting / uncommon / extraordinary applications will be granted a small amount of extra credit at the discretion of the professor).

Classification example: Detecting cancer in patients is one of the most important and useful emerging classification examples in machine learning. Using a variety of different imaging techniques, doctors can send images to a machine learning model to predict whether or not a patient has a form of cancer. Usually this is done with deep learning. One of the challenges of this is that there far fewer people who have cancer as opposed to no cancer. This creates a training set that is not balanced for learning.

Regression example: My favorite use of regression in machine learning is in predicting the outcome of sports games. Using baseball as an example, one could predict the projected win / loss percentage using a variety of predictors. Another case could be predicting the chance of a team winning any given game. Such techniques are useful for fantasy sports and are largely used in the sports betting field.

Clustering example: Clustering can be used to identify previously unknown "classes" of people in a set of unknown people. Consider an ad agency has collected millions of rows of data on millions of different people. A clustering algorithm or model could parse through this data and find relationships between different people with similar rows. This would allow the agency to know what kind of people they should be displaying ads to.