STATS 790
Assignment 1

Roxanne Li
Student id: 400181170

1/20/23

Q1.

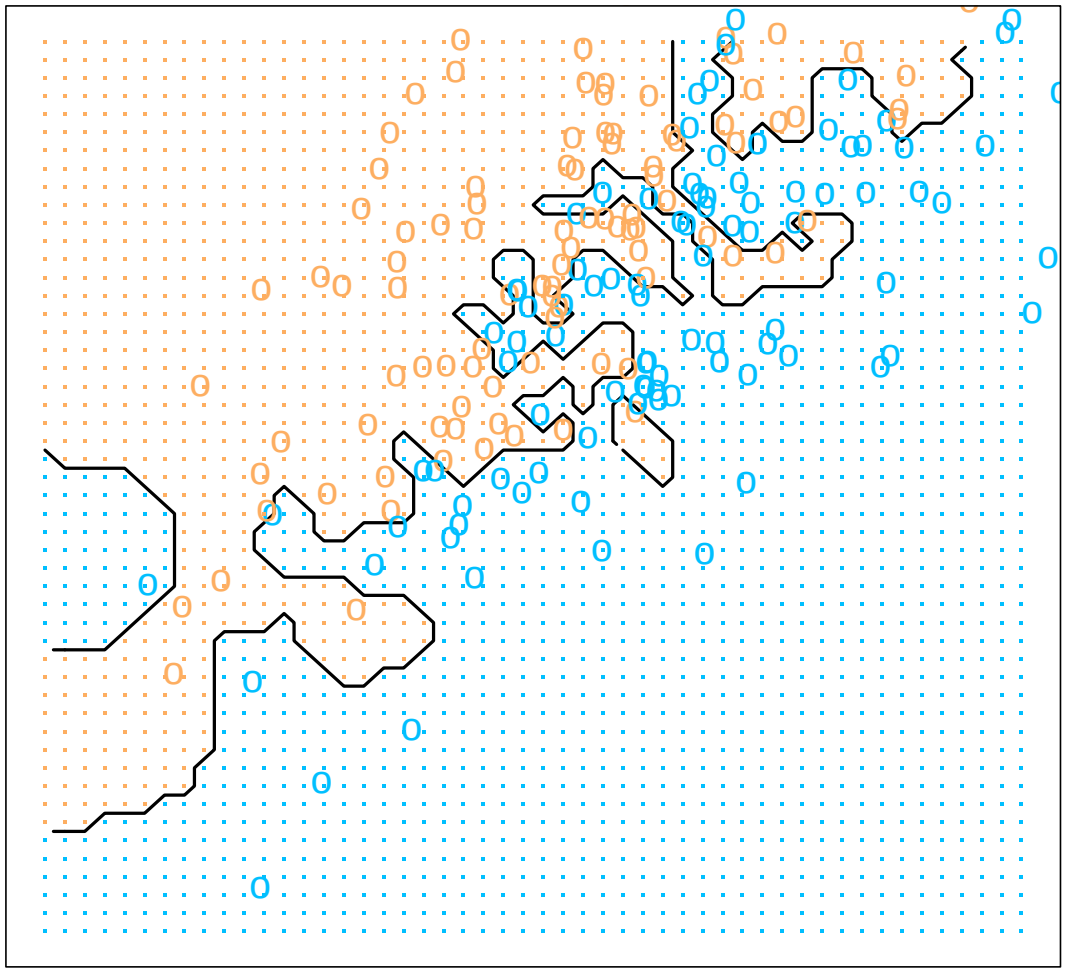
I agree with the final remarks of Breiman's Two Culture of Statistical Modelling article that statisticians should work with the data and check theory against data. In many of our Statistics projects, one seemingly integral step when we got the data was to check for outliers and remove some of them, after which we would try to fit the statistical models we have learned in class to the data and examine the fit. After reading the article, I think what that conventional step has been doing could be seen as somewhat imposing the data models on the data. Removing the oddities inherent in the data of the real world or the nature doesn't solve the problem in general, we should instead aim to seek a solution (such as algorithmic) that is more stable and generalizable to all cases.

Q2.

R code for producing **Figure2.3** in ESL:

```
library(MASS)
library(caret)
set.seed(1)

# generating 2-dim Gaussian mixture train data
sigma <- matrix(c(4,3,3,4), 2, 2)
```

```r
orange_X <- mvrnorm(100, c(5,7.5), sigma)
orange_y <- rep(1, 100)
orange <- data.frame(orange_X, orange_y)
colnames(orange) = c("X1", "X2", "y")

blue_X <- mvrnorm(100, c(6.5,7), sigma)
blue_y <- rep(0, 100)
blue <- data.frame(blue_X, blue_y)
colnames(blue) = c("X1", "X2", "y")

data <- rbind(orange, blue)
train <- data[sample(1:nrow(data)), ]
rownames(train) <- NULL

# generating test data (the grid)
x1 <- seq(0, 10, length.out=50)
x2 <- seq(0, 10, length.out=50)
test <- expand.grid(X1=x1, X2=x2)

# KNN (k=15) prediction of test data
test$y_pred <- knn3Train(train=train[,c(1,2)], test=test, cl=train$y, k=1)
matrix_pred <- matrix(test$y_pred, length(x1), length(x2))

# color the grid and draw the contour
contour(x1, x2, matrix_pred, levels=0.5, labels="", xlab="", ylab="",
        main="1-Nearest Neighbor Classifier", lwd=2, axes=FALSE)
points(train$X1, train$X2, pch="o", cex=1.5, col = ifelse(train$y==1, "#FDAE61", "#00BFFF"))
points(test$X1, test$X2, pch=".", cex=2, col=ifelse(test$y==1, "#FDAE61", "#00BFFF"))
box()
```



Q6.

Preparing training and test data:

```
library(dplyr)
library(class)
library(reshape2)
```

```
zip_train <- read.table("zip.train", quote="\"", comment.char="")
zip_test <- read.table("zip.test", quote="\"", comment.char="")

# prepare data
train <- zip_train %>% filter(V1 == "2" | V1 == "3")
test <- zip_test %>% filter(V1 == "2" | V1 == "3")
```

Using Linear Regression model for classification and examine the training and test error:

```
# Linear Regression
LR <- lm(V1~., data = train)
train_pred <- round(predict(LR, train))
test_pred <- round(predict(LR, test))
train_error <- 1 - length(which(train_pred==train$V1)==TRUE)/length(train_pred)
test_error <- 1 - length(which(test_pred==test$V1)==TRUE)/length(test_pred)
sprintf("Using linear regression, train error is %f, test error is %f", train_error, test_er
```

```
[1] "Using linear regression, train error is 0.005760, test error is 0.041209"
```

Using KNNs with different K values for classification and examine the training and test error:

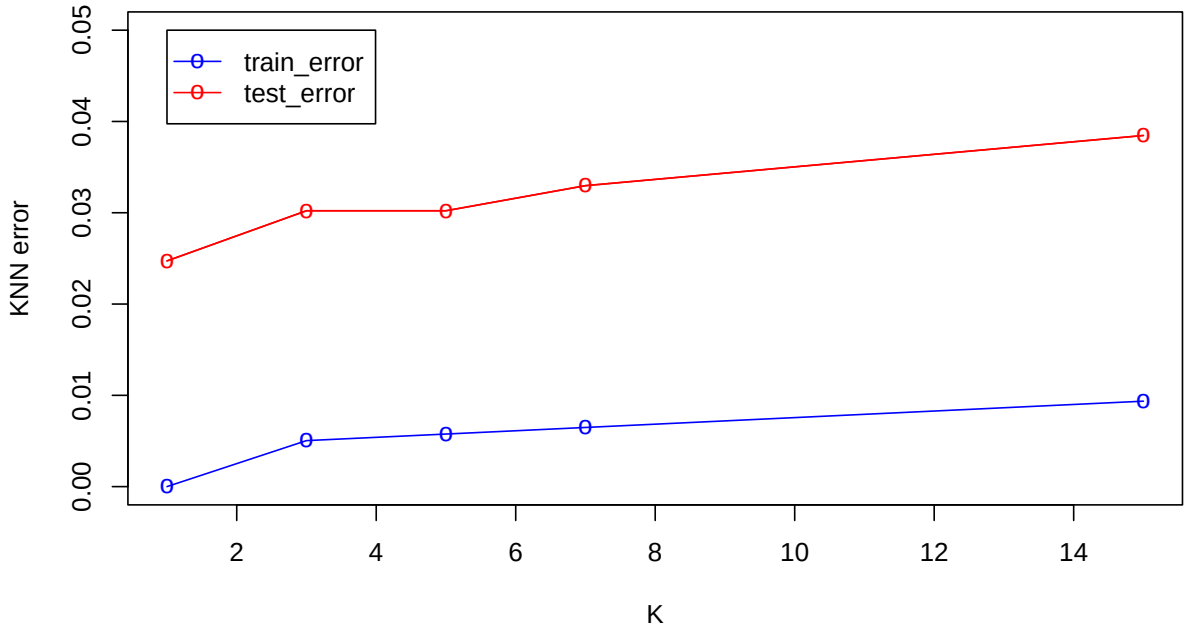
```
# KNN
train_error_knn <- list()
test_error_knn <- list()
K <- list(1,3,5,7,15)
for (k in K) {
  train_pred <- knn(train[,-1], train[,-1], train$V1, k)
  test_pred <- knn(train[,-1], test[,-1], train$V1, k)
  train_error_knn <- append(train_error_knn, 1 - length(which(train_pred==train$V1)==TRUE)/l
  test_error_knn <- append(test_error_knn, 1 - length(which(test_pred==test$V1)==TRUE)/lengt
}

error_knn = data.frame(unlist(K), unlist(train_error_knn), unlist(test_error_knn))
```

```
colnames(error_knn) = c("K","train_error", "test_error")
print(error_knn)
```

```
   K train_error test_error
1  1 0.000000000 0.02472527
2  3 0.005039597 0.03021978
3  5 0.005759539 0.03021978
4  7 0.006479482 0.03296703
5 15 0.009359251 0.03846154
```

We can plot the training and test error to see their trends.

```
plot(K, train_error_knn, type="o", col="blue", pch="o", ylim=c(0, 0.05), ylab="KNN error")
points(K, test_error_knn, type="o", col="red", pch="o")
lines(K, test_error_knn, col="red")
legend(1,0.05,legend=c("train_error","test_error"), col=c("blue","red"),
       pch=c("o","o"),lty=c(1,1), ncol=1)
```



We can see that the training error and test error of KNNs in general increase as K increases, and the test error is larger than the training error at all K's. When K=3, the training and test errors are close to those of the linear regression model we fitted before.

Q3, Q4, Q5

The following contains the solutions to Q3, Q4, and Q5.

1.2. $MAE(m) = E[|Y-m|]$

$$= \int_{-\infty}^{m} (m-y)\, p(y)\, dy + \int_{m}^{\infty} (y-m)\, p(y)\, dy$$

By Leibniz rule,

$$\frac{d\,MAE(m)}{dm} = \underbrace{(m-m)p(m)\cdot m'}_{=0} - \underbrace{(m-(-\infty))p(-\infty)\cdot(-\infty)'}_{=0} + \int_{-\infty}^{m} \frac{\partial}{\partial m}(m-y)\,p(y)\,dy$$

$$+ \underbrace{(\infty-m)p(\infty)\cdot\infty'}_{=0} - \underbrace{(m-m)p(m)\cdot m'}_{=0} + \int_{m}^{\infty} \frac{\partial}{\partial m}(y-m)\,p(y)\,dy$$

$$= \int_{-\infty}^{m} \frac{\partial}{\partial m}(m-y)\,p(y)\,dy + \int_{m}^{\infty} \frac{\partial}{\partial m}(y-m)\,p(y)\,dy$$

$$= \int_{-\infty}^{m} p(y)\,dy + \int_{m}^{\infty} -p(y)\,dy$$

To minimize $MAE(m)$, we set $\frac{d\,MAE(m)}{dm} = 0$.

Thus, $\int_{-\infty}^{m} p(y)\,dy + \int_{m}^{\infty} -p(y)\,dy = 0$

$$\int_{-\infty}^{m} p(y)\,dy = \int_{\infty}^{m} p(y)\,dy$$

$$F(m) = 1 - F(m)$$

$$F(m) = \frac{1}{2}, \text{ i.e. } P(Y \le m) = P(Y > m) = \frac{1}{2}$$

$\therefore$ $m = \text{Median}(Y)$.

1.7. Since our linear smoother is global mean, we have:

Linear smoother: $\hat{\mu}(x) = \sum_i y_i \hat{w}(x_i, x)$ where $\hat{w}(x_i, x) = \frac{1}{n}$

Thus $\hat{w}$ is a $n \times n$ matrix with all entries equal to $\frac{1}{n}$.

By eq.170. $df(\hat{\mu}) \equiv tr(w) = n \times \frac{1}{n} = 1$.

1.8. By 1.55, $\hat{w}(\hat{x}_i, x) = \begin{cases} \frac{1}{k}, & x_i \text{ one of the } k \text{ nearest neighbours of } x. \\ 0, & \text{o.w.} \end{cases}$

Thus, $\hat{w}$ is a $n \times n$ matrix with diagonal entries all equal to $\frac{1}{k}$ because x_i is always one of its own nearest neighbours.

$\therefore$ By eq.170, $df(\hat{\mu}) = tr(w) = n \times \frac{1}{k} = \frac{n}{k}$.

when $k = n$. $df(\hat{\mu}) = \frac{n}{n} = 1$.

Reference

Breiman, Leo. 2001. "Statistical Modeling: The Two Cultures." *Statistical Science* 16 (3): 199–215. http://www.jstor.org/stable/2676681.

Gelman, Andrew. 2021. "Reflections on Breiman's Two Cultures of Statistical Modeling." *Observational Studies* 7 (1): 95–98. https://doi.org/10.1353/obs.2021.0025.

Hastie, Trevor, Robert Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second edition. New York: Springer.

Kuhn, Max. 2022. *Caret: Classification and Regression Training*. https://CRAN.R-project.org/package=caret.

Panik, Michael J. 2005. *Advanced Statistics from an Elementary Point of View*. Boston: Elsevier/Academic Press.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with s*. Fourth. New York: Springer. https://www.stats.ox.ac.uk/pub/MASS4/.

Wickham, Hadley. 2007. "Reshaping Data with the Reshape Package." *Journal of Statistical Software* 21 (12): 1–20. http://www.jstatsoft.org/v21/i12/.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2022. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.