STAT 4360 (Introduction to Statistical Learning, Fall 2025)

Mini Project 1

Name: Van Nguyen

Question 1

(a) I used KNN with $K = 1, 6, 11, \ldots, 200$. For each K, I calculated the training error rate and the test error rate.

(b) The plot of training and test error vs K is shown in Figure 1.

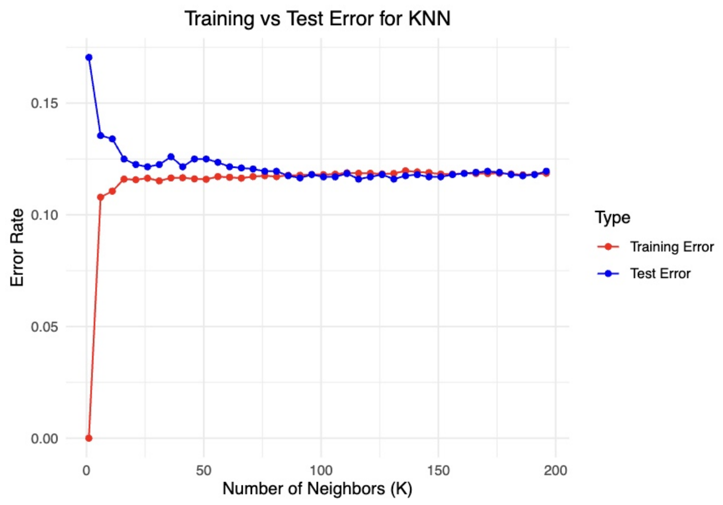

Figure 1: Training vs Test Error for K.

(c) From the test error, the best K is 116 with training error 0.1186 and test error 0.1160.

(d) The decision boundary plot for $K = 116$ is shown in Figure 2.

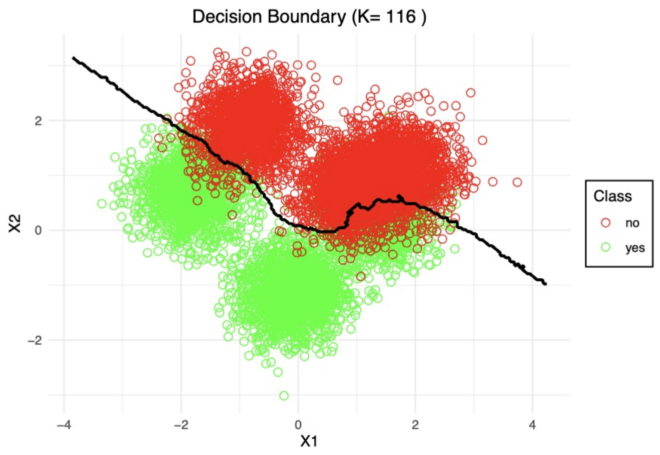

Figure 2: Decision Boundary for $K = 116$.

(e) Looking at the decision boundary, points far away from the boundary are classified clearly, and most of them are in the same class. The points near the boundary are more mixed and can change if K is different.

Question 2

(a) The Mean Squared Error can be decomposed as follows:

$$MSE(\hat{f}(x_0)) = E\big[(\hat{f}(x_0) - f(x_0))^2\big]$$

$$= E\Big[(\hat{f}(x_0) - E[\hat{f}(x_0)] + E[\hat{f}(x_0)] - f(x_0))^2\Big]$$

$$= E\big[(\hat{f}(x_0) - E[\hat{f}(x_0)])^2\big] + (E[\hat{f}(x_0)] - f(x_0))^2 + 2E\big[(\hat{f}(x_0) - E[\hat{f}(x_0)])(E[\hat{f}(x_0)] - f(x_0))\big]$$

$$= Var(\hat{f}(x_0)) + Bias(\hat{f}(x_0))^2$$

The cross-term is zero because $E[\hat{f}(x_0)] - f(x_0)$ is a constant, and

$$E[\hat{f}(x_0) - E[\hat{f}(x_0)]] = E[\hat{f}(x_0)] - E[\hat{f}(x_0)] = 0.$$

(b) The expected prediction error is

$$E[(\hat{Y}_0 - Y_0)^2], \quad \text{where } \hat{Y}_0 = \hat{f}(x_0),\ Y_0 = f(x_0) + \epsilon_0.$$

$$E[(\hat{Y}_0 - Y_0)^2] = E[(\hat{f}(x_0) - f(x_0) - \epsilon_0)^2]$$

$$= E[(\hat{f}(x_0) - f(x_0))^2] - 2E[(\hat{f}(x_0) - f(x_0))\epsilon_0] + E[\epsilon_0^2]$$

$$= MSE(\hat{f}(x_0)) + \sigma^2$$

$$= Var(\hat{f}(x_0)) + Bias(\hat{f}(x_0))^2 + \sigma^2$$

The cross-term vanishes because $\hat{f}(x_0)$ is independent of the noise ϵ_0, and $E[\epsilon_0] = 0$.

(c) When the model is not flexible, bias is high and test MSE is large. As flexibility increases, bias decreases, so test MSE decreases. If the model is too flexible, variance grows and test MSE increases again. This explains the U-shape.

R Code

```
## Question 1 ##
## (a) Fit KNN with K = 1, 6, 11, ..., 200
library(class)
library(ggplot2)
library(tidyr)

# read training and test data
train <- read.csv("1-training_data.csv")
test  <- read.csv("1-test_data.csv")

# first 2 columns = predictors, 3rd = class label
train.X <- as.matrix(train[, 1:2])
train.Y <- as.factor(train[, 3])
test.X  <- as.matrix(test[, 1:2])
test.Y  <- as.factor(test[, 3])

# K values to try (1, 6, 11, ... , 196)
K.values <- seq(1, 200, by = 5)
train.error <- rep(0, length(K.values))
test.error  <- rep(0, length(K.values))

# loop through K
set.seed(1) # reproducibility
for (i in 1:length(K.values)) {
  k <- K.values[i]

  # predict on training data and compute error
  pred.train <- knn(train.X, train.X, train.Y, k = k)
  train.error[i] <- mean(pred.train != train.Y)

  # predict on test data and compute error
  pred.test <- knn(train.X, test.X, train.Y, k = k)
  test.error[i] <- mean(pred.test != test.Y)
}

# results data
results <- data.frame(K = K.values,
                      TrainError = train.error,
                      TestError = test.error)
head(results)
```

```
## (b) Plot training and test error
# reshape into long format for plotting
results_long <- results %>%
  pivot_longer(cols = c(TrainError, TestError),
               names_to = "Type",
               values_to = "Error")

results_long$Type <- factor(results_long$Type,
                            levels = c("TrainError", "TestError"),
                            labels = c("Training Error", "Test Error"))

# plot train vs test error
ggplot(results_long, aes(x = K, y = Error, color = Type)) +
  geom_line() +
  geom_point() +
  labs(title = "Training vs Test Error for KNN",
       x = "Number of Neighbors (K)",
       y = "Error Rate") +
  scale_color_manual(values = c("Training Error" = "red",
                                "Test Error" = "blue")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

## (c) Find best K
# index of minimum test error
best.index <- which.min(test.error)

# store the best K and its errors
best.K <- K.values[best.index]
best.train.error <- train.error[best.index]
best.test.error  <- test.error[best.index]

# print the best K and error rates
cat("Best K =", best.K, "\n")
cat("Train error =", round(best.train.error, 4), "\n")
cat("Test error =", round(best.test.error, 4), "\n")

## (d) Decision boundary plot
```

```
# build a grid of X1 and X2 values
x1.range <- seq(min(c(train.X[,1], test.X[,1]))-0.5,
                max(c(train.X[,1], test.X[,1]))+0.5,length.out=200)
x2.range <- seq(min(c(train.X[,2], test.X[,2]))-0.5,
                max(c(train.X[,2], test.X[,2]))+0.5,length.out=200)
grid <- expand.grid(X1 = x1.range, X2 = x2.range)

# predict class for each grid point
grid.pred <- knn(train.X, grid, train.Y, k = best.K)
grid$Class <- factor(grid.pred, levels = levels(train.Y))
grid$ClassNum <- as.numeric(grid$Class)

# plot training points and boundary
ggplot() +
  geom_point(data = train,
             aes(x=train[,1],y=train[,2],color=train.Y),
             shape=1,size=2.5) +
  stat_contour(data=grid,
               aes(x=X1,y=X2,z=ClassNum),
               breaks=1.5,color="black",linewidth=1) +
  labs(title=paste("Decision Boundary (K=",best.K,")"),
       x="X1",y="X2",color="Class") +
  scale_color_manual(values=c("red","green")) +
  theme_minimal() +
  theme(plot.title=element_text(hjust=0.5),
        legend.background=element_rect(fill="white",color="black"))

## (e) Observations about decision boundary is written in Section 1
```

Output: Best K = 116, Train error = 0.1186, Test error = 0.1160