Mini Project 1

of

Stat 4360

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

# read training and test data
setwd("/Users/vannguyen/Downloads")
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)
```

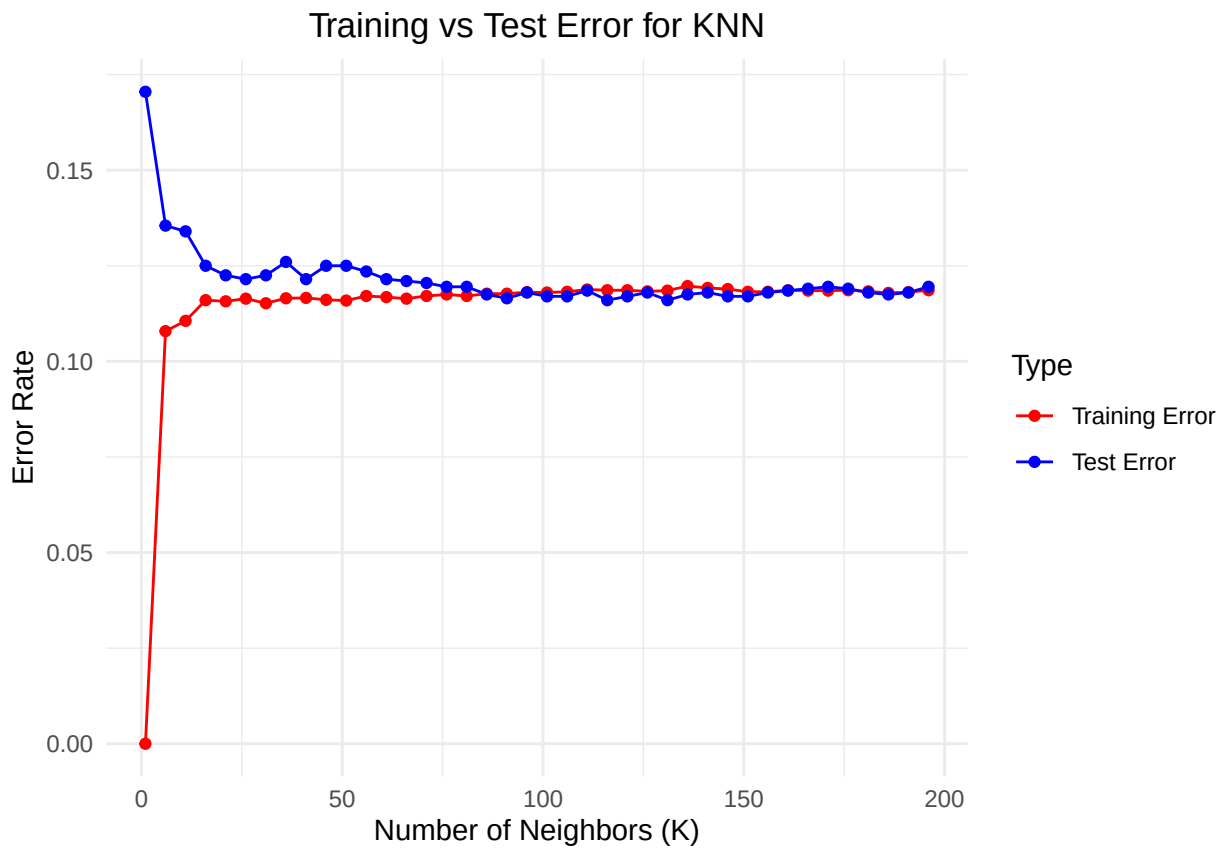
```
##    K TrainError TestError
## 1  1     0.0000    0.1705
## 2  6     0.1079    0.1355
## 3 11     0.1106    0.1340
## 4 16     0.1160    0.1250
## 5 21     0.1157    0.1225
```

```
## 6 26     0.1164    0.1215
```

```r
## (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))
```



```r
## (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")
```

```
## Best K = 116
```

```r
cat("Train error =", round(best.train.error, 4), "\n")
```

```
## Train error = 0.1186
```

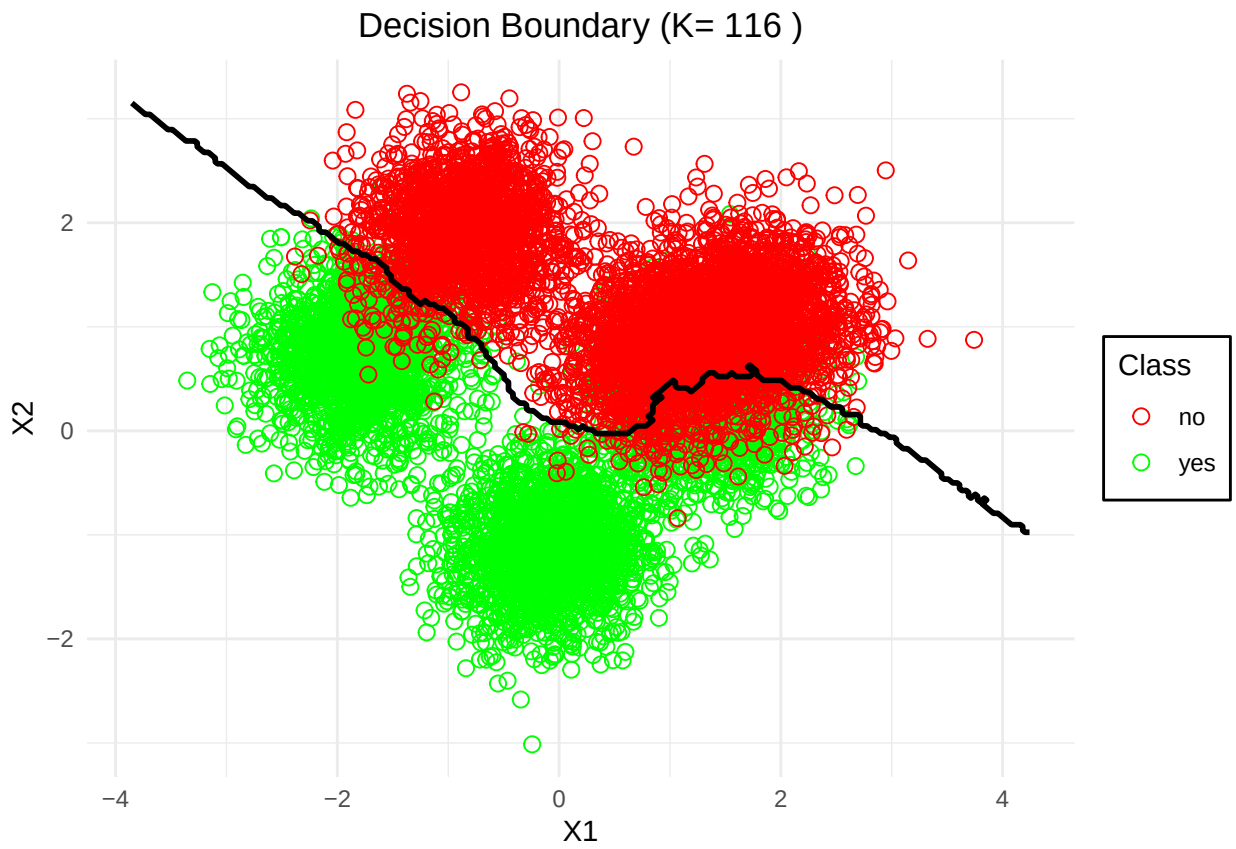
```r
cat("Test error =", round(best.test.error, 4), "\n")
```

```
## Test error = 0.116
```

```r
## (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
```