**STAT 4360 (Introduction to Statistical Learning, Spring 2023)**

**Mini Project 1  
Name: Ann Biju**

**Section 1**

1. (b)

A graph of training error rate

Description automatically generated

Observations: As K increases from 1 to 200, we can see the training and test error rates having an overlap for the bigger K values. However, when K is small, the test error seems to be very high. Though, with increased accuracy (high K values), we notice test error coming close to training error.

1. (c) - Optimal value of K is 86. Training error associated with the Optimal K is 0.1182. Testing error associated with the Optimal K is 0.116.

A red green and black dots

Description automatically generated

1. (d)
2. (e)

The decision boundary I have does not run through all the points, so it is not the most sensible. However, in a more ideal case, points farther away from the decision boundary will distinctly belong to one particular class, with a certain set of features. Points closer to the decision boundary have a higher tendency to be assigned to either class.

1. (a)

A white sheet with black text

Description automatically generated

(b)

A math equations on a grid

Description automatically generated

(c) As model flexibility increases, the variance of the test MSE increases, while the bias decreases. Additionally, the squared bias and variance change at differing rates. This inverse relationship (as one value increases, the other decreases – and vice versa) is seen in any good test set performance and is the reason behind the U-shape of the test MSE.

**Section 2 - Python Code (or R Code)**

# Question 1

install.packages("class")

library(class)

install.packages("ggplot2")

library(ggplot2)

install.packages("treemisc")

library(treemisc)

# reading the training and test files and storing them in appropriate variables

train <- read.csv("C:/ann/fall 2023/stat 4360/project 1/1-training\_data(1).csv")

test <- read.csv("C:/ann/fall 2023/stat 4360/project 1/1-test\_data(1).csv")

# part 1a

# we want to separate the X values from the Y values in the train and test datasets

x.train <- train[, -ncol(train)]

y.train <- train[, ncol(train)]

x.test <- test[, -ncol(test)]

y.test <- test[, ncol(test)]

# now fit KNN

all.K.vals <- seq(1, 200, by = 5)

for(i in all.K.vals){

predicted.values <- knn(train = x.train, test = x.test, cl = y.train, k = i)

accuracy <- mean(predicted.values == y.test)

cat("K = ", i, "accuracy = ", accuracy, "\n")

}

# part 1b

# store error rates for the training and test sets

train.errors <- numeric(length(all.K.vals))

test.errors <- numeric(length(all.K.vals))

# find the error rate for each K value

for(i in 1:length(all.K.vals)){

K <- all.K.vals[i]

#Fit the KNN classifier

pred.train <- knn(train = x.train, test = x.train, cl = y.train, k = K)

pred.test <- knn(train = x.train, test = x.test, cl = y.train, k = K)

# calculate the error rates (1 - accuracy)

train.errors[i] <- 1 - mean(pred.train == y.train)

test.errors[i] <- 1 - mean(pred.test == y.test)

}

# plot training and test errors vs. K

plot(all.K.vals, train.errors, type = "l", col = "blue", xlab = "K", ylab = "Error Rate",

main = "Training and Test Error Rates vs. K")

lines(all.K.vals, test.errors, type = "l", col = "red")

legend("topright", legend = c("Training Error", "Test Error"), col = c("blue", "red"), lty = 1)

# OBSERVATIONS:

# As K increases from 1 to 200, we can see the training and test error rates

# having an overlap for the bigger K values. However, when K is small, the test

# error seems to be very high. Though, with increased accuracy (high K values),

# we notice test error coming close to training error.

# part 1c

# find index of smallest test error

optimal.K.ind <- which.min(test.errors)

# optimal K

optimal.K <- all.K.vals[optimal.K.ind]

cat("Optimal K value = ", optimal.K)

# find training & test errors associated with optimal K

optimal.K.training <- train.errors[optimal.K.ind]

cat("Training error rate associated with optimal K =", optimal.K.training)

optimal.K.test <- test.errors[optimal.K.ind]

cat("Test error rate associated with optimal K =", optimal.K.test)

# part 1d

n.grid <- 50

x1.grid <- seq(f = min(x.train[, 1]), t = max(x.train[, 1]), l = n.grid)

x2.grid <- seq(f = min(x.train[, 2]), t = max(x.train[, 2]), l = n.grid)

grid <- expand.grid(x1.grid, x2.grid)

k.opt <- 70

set.seed(1)

mod.opt <- knn(x.train, grid, y.train, k = optimal.K, prob = T)

prob <- attr(mod.opt, "prob") # prob is voting fraction for winning class

prob <- ifelse(mod.opt == "Up", prob, 1 - prob) # now it is voting fraction for Direction == "Up"

prob <- matrix(prob, n.grid, n.grid)

plot(x.train, col = ifelse(y.train == "yes", "green", "red"))

contour(x1.grid, x2.grid, prob, levels = 0.5, add = TRUE, color = "black", lwd = 15)