k Nearest Neighbors

Consider the training and test data.

a) KNN is fitted for K = 1, 6, . . . 200.

```
q1.train<-read.csv("1-training_data.csv")
q1.test<-read.csv("1-test_data.csv")
head(q1.train)
```

```
##           x.1        x.2   y
## 1  1.18621488  0.2222768 yes
## 2 -0.94728411 -1.0978430 yes
## 3 -0.34557360 -1.1217680 yes
## 4  0.25232375 -0.4216226 yes
## 5 -1.33450560  0.3180441 yes
## 6 -0.01441298 -1.5883013 yes
```

```
str(q1.train)
```

```
## 'data.frame':    10000 obs. of  3 variables:
##  $ x.1: num  1.186 -0.947 -0.346 0.252 -1.335 ...
##  $ x.2: num  0.222 -1.098 -1.122 -0.422 0.318 ...
##  $ y  : chr  "yes" "yes" "yes" "yes" ...
```

```
head(q1.test)
```

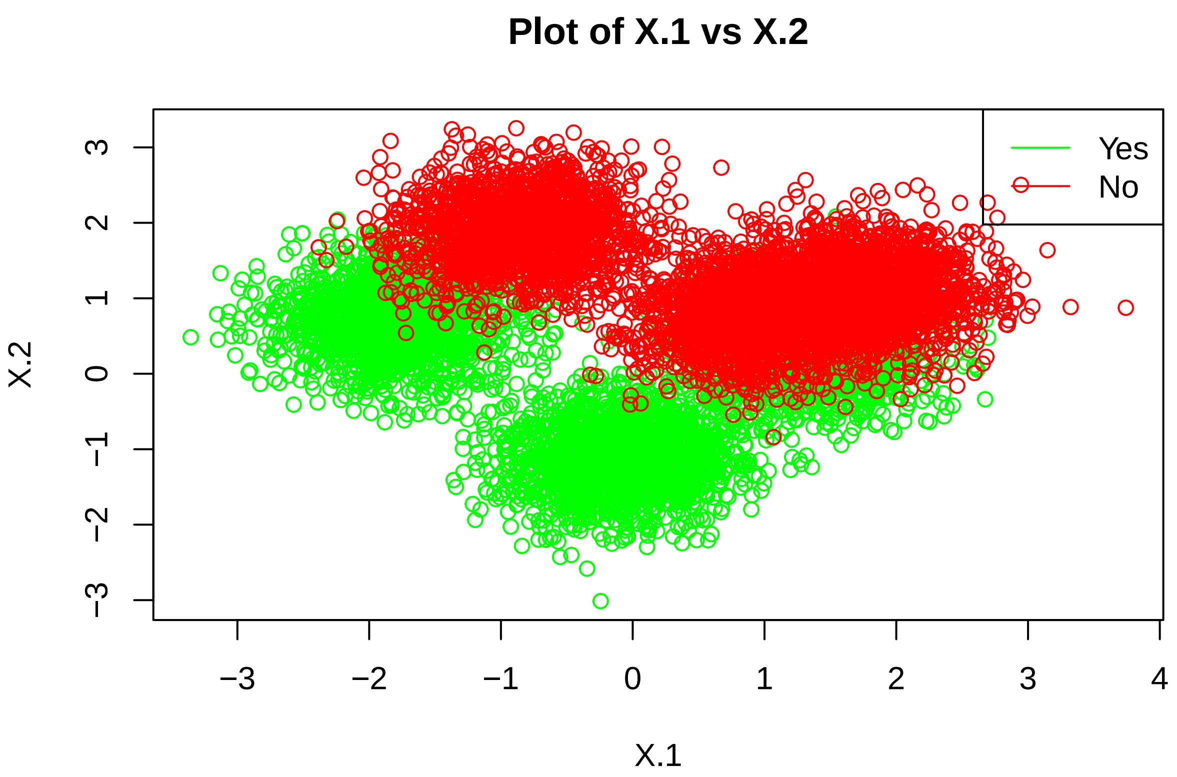
```
##           x.1         x.2   y
## 1  1.09403735  0.07560646 yes
## 2  0.09707596 -1.68666666 yes
## 3 -2.37512241  1.44326696 yes
## 4  0.25188685 -0.62926630 yes
## 5 -0.66783080 -0.87785728 yes
## 6  0.60319185 -1.03247879 yes
```

```
str(q1.test)
```

```
## 'data.frame':    2000 obs. of  3 variables:
##  $ x.1: num  1.094 0.0971 -2.3751 0.2519 -0.6678 ...
##  $ x.2: num  0.0756 -1.6867 1.4433 -0.6293 -0.8779 ...
##  $ y  : chr  "yes" "yes" "yes" "yes" ...
```

```
#subsetting data
train.X<-q1.train[,-3]
train.Y<-q1.train[,3]
test.X<-q1.test[,-3]
test.Y<-q1.test[,3]

#Visualizing data
plot(train.X, xlab = "X.1", ylab = "X.2", col = ifelse(train.Y == "yes", "green", "red"))
title("Plot of X.1 vs X.2")
legend("topright", lty = 1, col = c("green", "red"), legend = c("Yes", "No"))
```



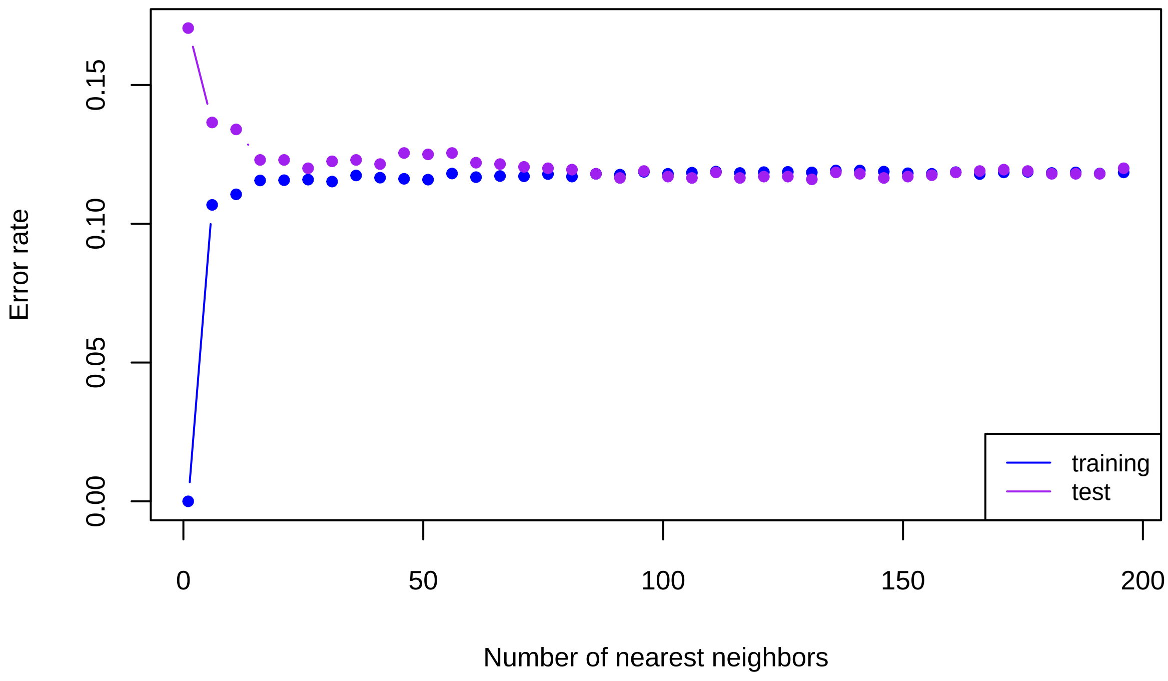
```
library(class)


#Implementing KNN algorithm for different values of K
ks <- seq(1,200, by = 5)
nks <- length(ks)
err.rate.train <- numeric(length = nks)
err.rate.test <- numeric(length = nks)
names(err.rate.train) <- names(err.rate.test) <- ks

for (i in seq(along = ks)) {
  set.seed(1)
  mod.train <- knn(train.X, train.X, train.Y, k = ks[i])
  set.seed(1)
  mod.test <- knn(train.X, test.X, train.Y, k = ks[i])
  err.rate.train[i] <- mean(mod.train != train.Y)
  err.rate.test[i] <- mean(mod.test != test.Y)
}
```

b) **Figure** shows the estimated training and test error rates for KNN with various values of K. Ignoring the fluctuations, as K increases the model flexibility decreases and we expected to see training error rate to increases. Moreover,as K increases we expected to see U shape for the test error rate due to the bias variance trade off. The results shown in **Figure** coincides with our expectation. Training error rate is 0 when K=1 and the test rate is minimized at K = 131 with 0.116 as the minimum value. The fluctuations of the curves is due to the small size of the training data set.

```
##  estimated training and test error rates for KNN with various values of K
plot(ks, err.rate.train, xlab = "Number of nearest neighbors", ylab = "Error rate",
     type = "b",ylim = range(c(err.rate.train, err.rate.test)), col = "blue",
     pch = 20,cex.lab=0.8,xaxt="n",yaxt="n")
lines(ks, err.rate.test, type="b", col="purple", pch = 20,main="",sub="thf")
axis(2,cex.axis=0.8)
axis(1,cex.axis=0.8)
legend("bottomright", lty = 1, col = c("blue", "purple"), legend = c("training", "test"),cex = 0.75)
```



c)
- Optimal value of K : 131
- Training error rate associated with the optimal K : 0.116
- Test error rate associated with the optimal K : 0.1185

```
 ## Error rates
result <- data.frame(ks, err.rate.train, err.rate.test)
result[err.rate.test == min(result$err.rate.test), ]
```

```
##      ks err.rate.train err.rate.test
## 131 131         0.1185         0.116
```

d) **Figure** represent the plot of the training data and the black line is the decision boundary for the optimal K=131. Green represents observations belong to yes class and red represent observations belong to no. According to the graph, although there are some red points lies below the decision boundary, most(large proportion) of the red points lies above the decision boundary. Similarly, although there are some green points lies above the decision boundary, most(large proportion) of the green points lies below the decision boundary. Therefore the decision boundary seem sensible.

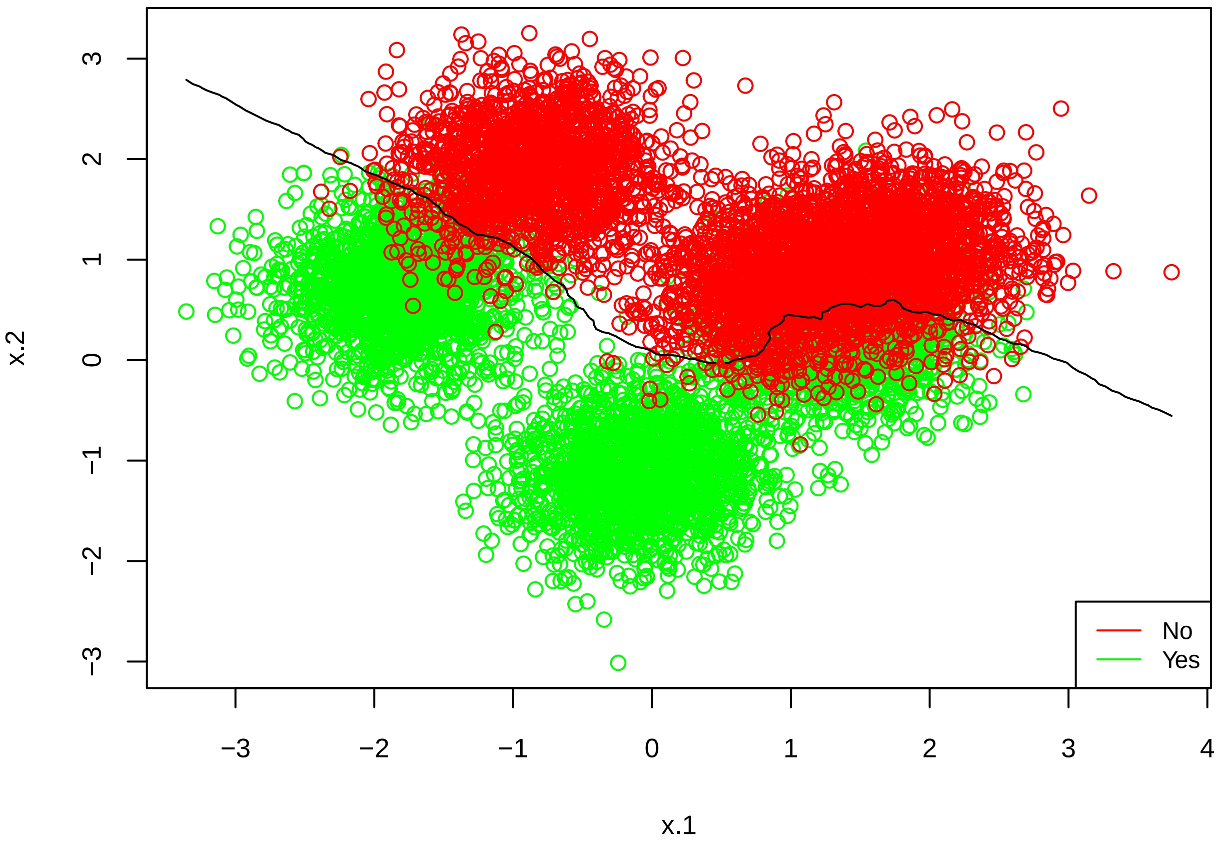
```
## Plot of the training data that shows the decision boundary for the optimal K

n.grid <- 150
x1.grid <- seq(f = min(train.X[, 1]), t = max(train.X[, 1]), l = n.grid)
x2.grid <- seq(f = min(train.X[, 2]), t = max(train.X[, 2]), l = n.grid)
grid <- expand.grid(x1.grid, x2.grid)


## KNN for optimal value of K=131
k.opt <- 131
set.seed(1)
mod.opt <- knn(train.X, grid, train.Y, k = k.opt, prob = T)
prob <- attr(mod.opt, "prob") # prob is voting fraction for winning class
prob <- ifelse(mod.opt == "yes", prob, 1 - prob) # now it is voting fraction for Direction == "yes"
prob <- matrix(prob, n.grid, n.grid)

par(mar = c(3.8, 3.8, 1,1))
```

```r
plot(train.X, col = ifelse(train.Y == "yes", "green", "red"),cex.lab=0.8,xaxt="n",yaxt="n")
axis(2,cex.axis=0.8)
axis(1,cex.axis=0.8)
contour(x1.grid, x2.grid, prob, levels = 0.49, labels = "", xlab = "", ylab = "", main = "", add = T)
legend("bottomright", lty = 1, col = c("red", "green"), legend = c("No", "Yes"),cex = 0.75)
```


x.2
x.1
No
Yes