STA 545 Statistical Data Mining I, Fall 2020

Homework 2

Stella Liao

September 15, 2020

1. (40 points) The table below provides a training data set containing six observations, three predictors, and one qualitative response variable. Suppose we wish to use this data set to make a prediction for Y when X 1 = X 2 = X 3 = 0 using k-nearest neighbors.

```
##   Obs. X1 X2 X3     Y
## 1    1  0  3  0   Red
## 2    2  2  0  0   Red
## 3    3  0  1  3   Red
## 4    4  0  1  2 Green
## 5    5 -1  0  1 Green
## 6    6  1  1  1   Red
```

(a) Compute the Euclidean distance between each observation and the test point, X1 = X2 = X3 = 0.

The answer is shown in the output of the following chunk.

```
data1_eucDist <- data1%>%
  mutate(eucDist = sqrt(X1**2+X2**2+X3**2))

data1_eucDist
```

```
##   Obs. X1 X2 X3     Y  eucDist
## 1    1  0  3  0   Red 3.000000
## 2    2  2  0  0   Red 2.000000
## 3    3  0  1  3   Red 3.162278
## 4    4  0  1  2 Green 2.236068
## 5    5 -1  0  1 Green 1.414214
## 6    6  1  1  1   Red 1.732051
```

(b) What is our prediction with k = 1?

At this point, we should choose the nearest point, which is Obs.5. Therefore, our prediction should be 'Green'.

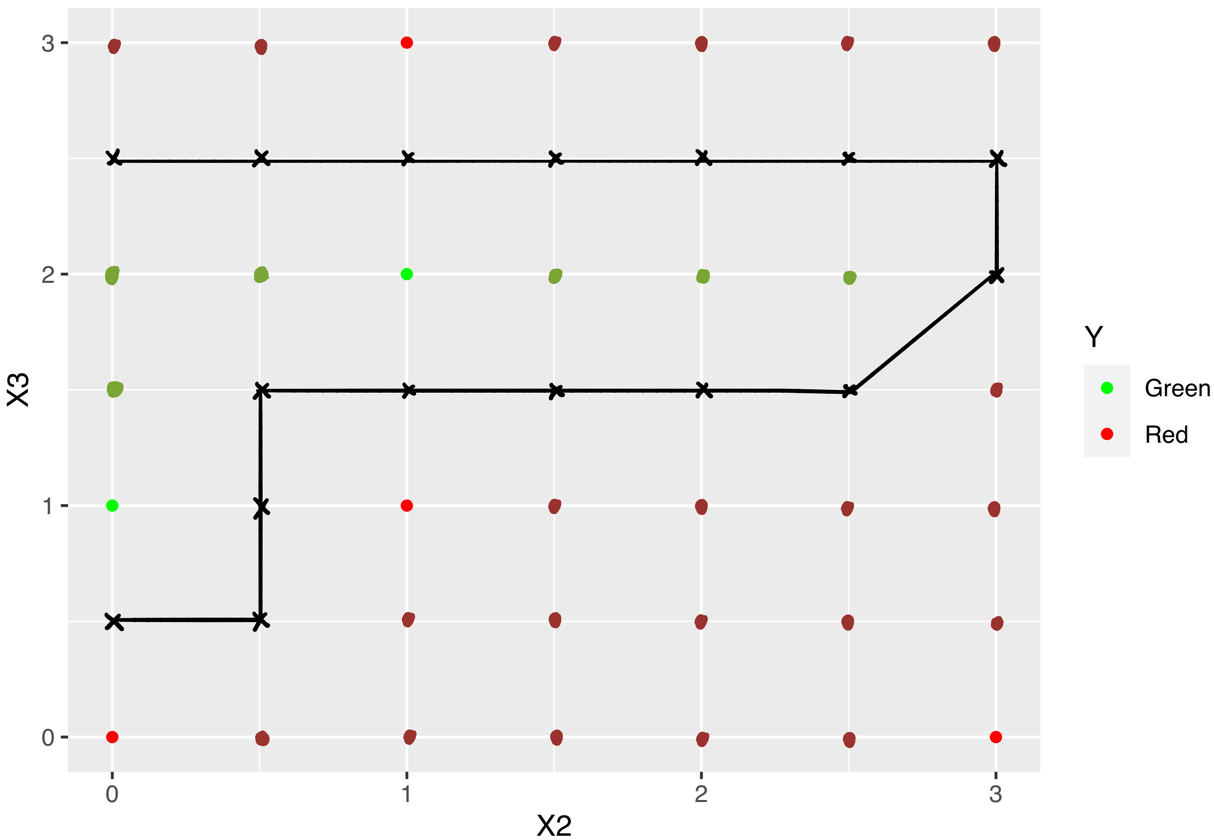
(c) What is our prediction with k = 3?

At this point, we should choose the average from the three nearest neighbors(Obs.5, 6 and 2). Therefore,our prediction should be 'Red'.

(d) Suppose we only use features X2 and X3. Please draw the decision boundary of the k-nearest neighbor classifier with k = 1 in a figure.

The decision boundary is drawn with black ink in the output plot of the following chunk.

```
library(ggplot2)
data1 %>%
  ggplot(aes(X2, X3)) +
  geom_point(aes(color = Y))+
  scale_color_manual(values=c("#00ff00", "#ff0000"))
```



(e) If the Bayes decision boundary in this problem is highly nonlinear, would we expect the best value for k to be large or small? Why?

With K increasing, the boundary becomes linear. Therefore, we would expect the best value for K to be small at this point.

2. (60 points) Data for this question come from the handwritten ZIP codes on envelopes from U.S. postal mail. Each image is a segment from a five digit ZIP code, isolating a single digit. The images are 16×16 eight-bit grayscale maps, with each pixel ranging in intensity from 0 to 255. The images have been normalized to have approximately the same size and orientation. The task is to predict, from the 16×16 matrix of pixel intensities, the identity of each image (0, 1, . . . , 9) quickly and accurately. The zipcode data are available from the book website www-stat.stanford.edu/ElemStatLearn. Please consider only the 2's and 3's in the data.

(a) Fit a linear regression model where we code Y = 1 if the label of the image is 2, and Y = −1 if the label of the image is 3. Show both the training misclassification error and test misclassification error for this binary classification problem.

training misclassification error = 0.09924689
test misclassification error = 0.6066614

```
train <- as.data.frame(read.table('zip.train.gz'))%>%
  filter(V1 %in% c(2,3))%>%
  mutate(Y = replace(V1, V1==2, 1))%>%
  mutate(Y = replace(Y, V1==3, -1))%>%
  subset(select = -V1)

test <- as.data.frame(read.table('zip.test.gz'))%>%
  filter(V1 %in% c(2,3))%>%
  mutate(Y = replace(V1, V1==2, 1))%>%
  mutate(Y = replace(Y, V1==3, -1))%>%
  subset(select = -V1)

lm.fit <- lm( Y ~ .,train)
lm.prediction = predict(lm.fit, newdata = test[,1:256])
training_misclassi cation_error <- mean(lm.fit$residuals^2)
test_misclassi cation_error <- mean((lm.prediction-test$Y)^2)

training_misclassi cation_error
```

```
## [1] 0.09924689
```

```
test_misclassi cation_error
```

```
## [1] 0.6066614
```

(b) Consider the k-nearest neighbor classifiers with k = 1, 3, 5, 7 and 15. Show both the training error and test error for each choice.

The answer is shown in the output of the following chunk.

```
library(class)

n.K=15

train.error=rep(NA,n.K)
test.error=rep(NA,n.K)

# Calculate the training error and test error for each K
for(i in 1:n.K){
```

```r
  knn.pred=knn(train = train[,1:256],
               test = test[,1:256],
               cl = train[,257], k = i)
  test.error[i]=mean(knn.pred!=test[,257])

  knn.train=knn(train = train[,1:256],
                test = train[,1:256],
                cl = train[,257], k = i)
  train.error[i]=mean(knn.train!=train[,257])
}

errors <- matrix(ncol = 3, nrow = 8)
for (i in 1:8){
  j = 2 * i -1
  errors[i,1] <- j
  errors[i,2] <- train.error[j]
  errors[i,3] <- test.error[j]
}
errors <- as.data.frame(errors)%>%
  setNames(c("K","Train error","Test error"))%>%
  filter( K %in% c(1,3,5,7,15))
errors
```

```
##    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
```