Homework 4

Yuning Li

2024-03-05

```
train_data <- read.table("/Users/yuningli/Desktop/zip.train.gz")
test_data<- read.table("/Users/yuningli/Desktop/zip.test.gz")
library(class)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
############################question 1: knn model#################################
# Extract labels and features
train_labels <- train_data[, 1]
train_features <- train_data[, -1]
test_labels <- test_data[, 1]
test_features <- test_data[, -1]

# Function to normalize features (if needed)
normalize <- function(data) {
  (data - min(data)) / (max(data) - min(data))
}

# Normalize features to [0, 1] (assuming grayscale values)
train_features_normalized <- apply(train_features, 2, normalize)
test_features_normalized <- apply(test_features, 2, normalize)

# Train k-NN model
k <- 5  # You can adjust the value of k
knn_model <- knn(train_features_normalized, test_features_normalized, train_labels, k = k)

# Evaluate the model
error_rate <- mean(knn_model != test_labels)
cat('Error rate:', error_rate, '\n')
```

```
## Error rate: 0.05480817
```

```
########################Q2#################################
# Extract labels and features
labels <- train_data[, 1]
features <- train_data[, -1]

# Convert labels to factor
labels <- as.factor(labels)

# Set up control parameters for 5-fold cross-validation
```

```r
folds <- createFolds(labels, k = 5, list = TRUE)
k_values <- 1:20

# Perform k-fold cross-validation
cv_errors <- numeric(length(k_values))

for (i in seq_along(k_values)) {
  k <- k_values[i]
  current_errors <- numeric(length(folds))

  for (j in seq_along(folds)) {
    train_indices <- unlist(folds[-j])
    test_indices <- folds[[j]]

    knn_model <- knn(features[train_indices, ], features[test_indices, ], labels[train_indices], k = k)

    current_errors[j] <- mean(knn_model != labels[test_indices])
  }

  cv_errors[i] <- mean(current_errors)
}

# Print the estimated average test error for each 'k'
print(data.frame(k = k_values, avg_error = cv_errors))
```

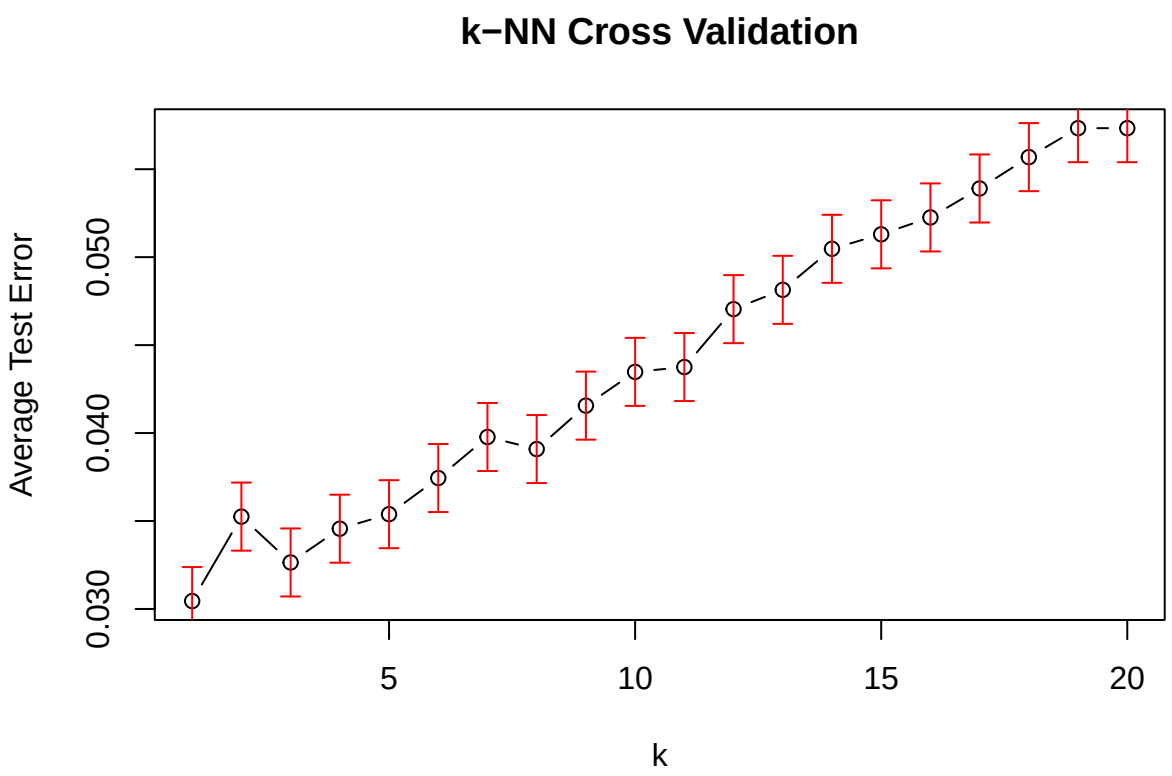
```
##     k  avg_error
## 1   1 0.03044854
## 2   2 0.03524917
## 3   3 0.03264304
## 4   4 0.03456320
## 5   5 0.03538653
## 6   6 0.03744367
## 7   7 0.03977563
## 8   8 0.03908986
## 9   9 0.04155880
## 10 10 0.04347868
## 11 11 0.04375303
## 12 12 0.04704474
## 13 13 0.04814232
## 14 14 0.05047447
## 15 15 0.05129742
## 16 16 0.05225764
## 17 17 0.05390354
## 18 18 0.05568662
## 19 19 0.05733233
## 20 20 0.05733233
```

```r
#########################Q3#########################

# Calculate standard errors
se <- sd(cv_errors) / sqrt(length(cv_errors))

# Plot the estimated average test error as a function of 'k' with error bars
plot(k_values, cv_errors, type = 'b', xlab = 'k', ylab = 'Average Test Error', main = 'k-NN Cross Valida
```

```r
# Add error bars
arrows(k_values, cv_errors - se, k_values, cv_errors + se, angle = 90, code = 3, length = 0.05, col = "
```




```r
##############################Q4##########################################
# Find the index of the minimum cross-validated error
min_error_index <- which.min(cv_errors)

# Identify the minimum error and its associated 'k' value
min_error <- cv_errors[min_error_index]
best_k <- k_values[min_error_index]

# Calculate the standard error
se <- sd(cv_errors) / sqrt(length(cv_errors))

# Identify the largest 'k' within one standard error of the minimum
selected_k <- k_values[cv_errors <= min_error + se][1]

cat('Minimum Cross-validated Error:', min_error, '\n')
```

```
## Minimum Cross-validated Error: 0.03044854
```

```r
cat('Best k Value:', best_k, '\n')
```

```
## Best k Value: 1
```

```r
cat('Selected k using One-Standard Error Rule:', selected_k, '\n')
```

```
## Selected k using One-Standard Error Rule: 1
```

```r
##############################Q5 ##########################################
```

```r
# Extract labels and features
train_labels <- train_data[, 1]
train_features <- train_data[, -1]

# Convert labels to factor
train_labels <- as.factor(train_labels)

# Fit the final k-NN model using the full training dataset and 'k=1'
final_k <- 1  # Use the selected 'k' from the one-standard error rule
final_knn_model <- knn(train_features, train_features, train_labels, k = final_k)

# Print the final k-NN model details
cat('Final k-NN Model (k =', final_k, '):\n')
```

```
## Final k-NN Model (k = 1 ):
```

```r
print(final_knn_model)
```

```
##    [1] 6 5 4 7 3 6 3 1 0 1 7 0 1 1 7 7 4 8 0 1 4 8 7 4 8 7 3 7 4 1 3 6 7 4 1 3 7
##   [38] 7 4 5 4 2 7 4 1 3 7 7 4 0 6 3 2 0 8 6 6 2 0 8 7 8 2 0 9 0 2 2 0 8 1 2 0 8
##   [75] 3 3 2 8 2 2 0 8 1 4 4 8 9 8 9 6 7 6 1 9 7 0 8 0 4 6 8 0 0 3 0 8 0 9 0 3 8
##  [112] 0 1 2 2 9 0 6 6 5 9 2 0 9 1 4 1 2 7 1 0 9 0 8 0 7 9 1 3 0 4 4 3 5 1 6 8 5
##  [149] 4 4 6 8 4 4 8 6 4 0 2 3 9 8 6 8 9 3 5 6 8 0 2 2 6 8 4 1 0 2 7 1 0 2 2 7 1
##  [186] 0 9 2 7 0 4 8 0 8 7 2 7 1 3 2 7 3 2 2 2 7 1 0 2 2 8 5 4 2 2 7 8 7 0 2 7 1
##  [223] 0 2 6 0 2 7 6 0 8 2 7 1 1 0 1 7 7 6 4 4 6 2 9 1 1 9 0 3 1 1 8 1 0 3 1 6 1
##  [260] 1 7 5 5 3 6 5 0 6 8 8 0 0 0 1 4 9 5 7 0 0 0 1 6 0 1 9 8 6 0 1 9 8 6 8 0 0
##  [297] 9 6 8 5 4 4 9 4 0 8 6 4 8 1 2 1 2 1 2 6 8 0 2 1 3 0 2 3 2 3 5 8 0 2 7 9 0
##  [334] 0 0 0 6 8 9 7 7 9 4 9 1 2 9 4 5 3 8 5 7 1 8 0 1 4 6 5 5 3 2 0 1 0 6 6 6 6
##  [371] 0 9 0 2 6 0 5 5 9 6 1 7 7 7 6 1 8 2 0 6 0 5 2 1 6 1 8 6 6 6 0 9 0 2 6 0 2
##  [408] 5 6 0 9 0 2 6 0 5 2 1 6 0 5 4 6 6 4 7 3 6 1 8 2 1 6 0 9 0 2 6 0 4 3 5 6 0
##  [445] 5 6 1 8 2 0 6 0 9 0 2 6 1 8 0 1 6 0 5 2 2 6 1 7 6 1 6 0 4 7 3 1 8 2 6 0 9
##  [482] 0 2 6 1 7 0 1 6 0 5 2 2 0 9 2 7 6 0 9 0 2 6 1 8 2 1 6 1 9 1 3 6 0 9 0 2 6
##  [519] 1 9 0 0 4 6 2 6 1 8 0 1 4 6 9 9 2 9 7 1 0 2 8 0 7 0 2 9 7 1 0 2 8 2 3 4 2
##  [556] 8 2 1 2 2 8 6 5 8 2 8 7 3 9 2 8 2 1 7 2 8 2 7 2 0 0 9 2 2 8 3 6 2 8 3 0 3
##  [593] 2 8 6 0 1 2 8 1 0 2 8 2 6 5 2 8 2 6 2 8 4 3 2 8 2 3 2 8 1 1 2 3 2 2 6 2 1
##  [630] 2 2 2 1 1 6 2 7 0 2 2 4 0 1 2 3 4 5 2 3 0 6 0 9 3 0 0 3 0 0 8 8 3 0 0 8 3
##  [667] 3 0 2 4 0 3 0 8 1 5 3 0 2 7 4 3 0 8 0 9 3 0 0 3 0 2 3 0 2 4 7 8 1 1 1 1 2
##  [704] 1 3 7 2 1 0 3 8 0 3 7 1 3 0 8 5 1 6 8 5 7 0 8 9 1 1 9 8 0 0 0 5 2 0 0 1 1
##  [741] 2 0 0 3 7 5 9 4 0 2 9 7 4 1 0 2 1 3 0 0 2 2 1 5 0 2 1 0 9 0 2 1 4 6 0 1 7
##  [778] 2 0 0 1 9 0 6 0 2 1 3 8 0 1 9 7 0 0 1 8 4 1 0 1 3 6 4 0 2 6 0 1 0 1 8 3 0
##  [815] 9 1 0 4 9 8 0 7 2 0 0 6 5 0 9 0 0 6 3 0 0 0 9 0 7 0 0 9 3 4 4 0 6 9 0 0 7
##  [852] 0 8 0 6 6 0 0 9 1 8 6 5 4 0 0 6 0 4 0 0 7 5 8 0 0 2 3 0 0 6 6 0 0 9 0 7 0
##  [889] 0 9 2 3 0 0 9 1 2 0 0 6 9 7 0 7 5 4 2 3 5 5 7 1 8 8 6 0 0 1 8 0 2 0 2 0 4
##  [926] 8 2 8 0 6 0 2 1 3 2 9 7 1 0 4 0 1 8 5 0 2 3 8 5 3 4 5 8 9 1 0 7 3 8 1 5 2
##  [963] 4 6 2 0 6 0 1 0 7 4 6 9 1 2 2 3 2 0 4 8 8 3 8 6 6 6 1 4 9 7 0 0 6 9 7 4 0
## [1000] 5 9 8 0 0 2 9 0 9 8 4 0 8 4 9 5 5 0 4 8 2 3 2 2 6 9 8 4 8 0 8 4 0 7 2 0 8
## [1037] 5 0 0 6 5 3 7 0 6 9 2 5 7 6 5 7 1 9 6 5 4 6 8 5 4 4 6 8 0 0 9 6 8 8 0 3 6
## [1074] 8 1 1 4 8 5 0 2 7 8 5 7 1 8 8 5 2 2 7 1 1 8 9 3 1 9 0 1 1 9 9 7 7 1 9 7 1
## [1111] 4 1 9 1 1 9 7 1 4 1 9 7 2 6 0 0 0 6 6 2 2 2 4 6 3 1 6 4 6 3 1 4 6 6 5 4 7
## [1148] 3 9 0 0 1 9 0 0 4 5 7 0 0 6 2 7 0 8 0 9 7 0 0 9 4 0 2 9 6 4 0 5 0 5 4 1 0
## [1185] 1 7 2 4 0 7 9 4 1 2 2 8 4 2 5 4 1 4 1 7 0 1 4 1 5 3 1 4 0 9 1 4 4 3 0 1 4
## [1222] 2 1 0 1 4 3 1 1 4 1 4 7 2 4 1 9 4 2 5 0 1 4 2 5 0 1 4 2 5 0 1 4 1 1 3 9 1
## [1259] 4 7 0 3 2 1 2 3 2 1 2 0 2 2 1 2 6 8 2 1 2 0 2 1 2 6 2 1 4 0 4 2 1 7 0 1 1
## [1296] 2 1 2 2 8 3 0 3 6 6 3 0 3 2 6 3 0 3 5 6 0 3 4 9 3 0 3 0 2 6 3 0 3 4 4 3 0
```

```
## [1333] 3 2 7 1 9 9 6 6 1 9 8 0 3 1 9 7 1 3 1 9 5 1 9 0 6 1 9 8 0 4 1 9 7 1 3 6 7
## [1370] 2 1 7 7 7 4 8 6 7 0 6 2 7 2 4 1 7 2 5 0 3 7 2 5 1 2 7 2 2 0 3 7 2 0 3 2 3
## [1407] 7 4 0 5 3 8 1 5 1 3 8 1 3 4 3 7 0 7 5 3 8 1 3 8 3 8 1 1 7 3 7 2 0 2 9 6 0
## [1444] 7 3 7 0 8 3 8 0 5 0 1 9 8 9 9 1 9 9 6 8 1 9 7 2 0 1 9 8 8 0 4 0 3 1 9 7 1
## [1481] 1 1 9 7 1 1 1 7 7 1 7 4 0 4 1 7 6 0 1 1 7 0 1 3 1 7 5 7 3 1 7 0 1 1 1 7 5
## [1518] 7 3 1 7 3 6 0 1 7 8 3 7 1 7 5 4 7 1 7 0 6 6 1 7 6 0 4 3 0 0 1 7 7 6 3 7 5
## [1555] 7 3 1 6 9 1 5 7 3 4 0 1 1 7 0 5 5 1 7 3 1 1 7 5 7 3 1 7 3 0 6 1 7 0 2 1 7
## [1592] 2 6 8 1 7 4 0 7 1 7 3 3 3 1 7 0 1 1 1 7 0 5 5 1 9 7 2 0 1 9 7 2 6 0 0 1 1
## [1629] 9 8 0 4 1 9 7 0 3 1 9 7 1 1 9 8 5 0 1 9 7 0 7 9 7 0 0 1 9 8 0 3 9 8 5 1 9
## [1666] 8 9 1 0 1 9 6 1 9 0 6 7 0 2 2 7 7 1 1 0 0 1 9 7 2 6 0 0 1 0 1 9 8 0 5 1 9
## [1703] 8 9 7 9 8 9 9 1 4 2 0 1 8 0 2 1 7 1 1 0 1 9 7 2 6 0 0 1 0 0 5 4 0 1 0 5 3
## [1740] 4 5 0 5 4 0 1 0 5 4 0 1 0 5 6 4 1 0 8 8 7 0 2 8 8 2 0 3 7 7 0 0 6 2 2 6 0
## [1777] 6 0 3 8 0 6 7 9 0 0 8 7 0 6 0 9 1 3 5 0 6 1 0 3 0 6 0 8 7 0 6 0 8 7 0 4 0
## [1814] 9 6 0 4 9 6 3 4 9 2 0 0 4 7 5 1 5 3 6 3 1 4 0 1 4 1 0 6 0 4 9 0 4 9 3 0 1
## [1851] 4 2 0 0 9 7 2 0 1 9 7 2 0 9 9 7 1 0 2 9 7 2 1 0 1 8 6 4 2 1 9 3 2 0 1 9 3
## [1888] 3 5 1 8 0 4 9 1 8 1 0 1 1 5 9 1 8 9 0 1 1 8 7 0 4 1 9 1 0 1 1 9 1 3 2 1 9
## [1925] 1 0 1 1 9 1 0 1 1 9 2 5 5 1 9 1 0 4 1 9 1 2 4 1 9 1 4 1 1 9 0 1 3 1 9 0 0
## [1962] 3 1 9 1 3 0 1 9 1 0 1 1 9 0 3 1 1 0 1 1 0 3 1 9 0 1 0 1 0 7 1 9 0 1 5 1 9
## [1999] 0 1 0 1 9 0 2 5 1 9 8 6 1 9 1 2 0 7 0 1 1 5 7 0 6 0 5 7 0 1 6 0 7 0 7 0 8
## [2036] 1 7 0 1 0 7 0 0 1 7 1 1 1 7 1 1 1 1 1 4 4 6 9 6 8 1 7 6 7 1 6 1 4 7 4 6 6
## [2073] 7 2 3 2 3 6 6 7 2 7 4 5 7 0 2 0 2 5 9 5 8 5 0 6 4 9 7 0 7 5 9 7 1 4 9 9 7
## [2110] 4 5 1 4 9 7 9 1 3 9 7 3 0 9 0 9 0 9 7 3 0 9 0 9 3 0 2 5 7 0 1 3 9 0 4 2 4
## [2147] 5 3 9 0 1 3 9 0 4 2 2 0 0 1 3 2 0 9 2 0 4 9 2 0 5 9 2 0 0 0 1 2 0 0 0 6 2
## [2184] 0 2 0 3 1 0 0 2 0 0 6 9 9 9 1 2 0 2 3 2 0 0 0 7 2 0 0 2 0 0 1 3 2 0 0 0 6
## [2221] 1 9 9 5 6 1 9 7 1 4 1 9 7 1 3 1 9 8 1 0 1 9 7 5 1 9 9 6 6 7 1 4 1 9 8 0 9
## [2258] 1 9 8 5 0 5 0 0 1 1 9 8 0 5 1 7 0 1 9 9 6 9 9 5 0 4 9 9 9 0 1 8 9 1 0 4 8
## [2295] 9 1 2 2 8 9 5 0 2 8 9 7 0 2 8 9 0 1 5 8 1 1 9 8 9 1 0 4 8 9 0 9 8 9 1 9 8
## [2332] 4 7 7 0 8 4 5 2 8 4 1 3 2 8 4 7 0 1 8 4 1 2 6 0 1 7 1 6 5 7 2 0 7 6 5 9 3
## [2369] 2 3 5 7 1 5 5 9 0 4 5 6 0 1 5 9 0 1 5 9 8 7 5 9 7 1 5 5 9 4 0 4 7 0 8 0 9
## [2406] 7 0 4 5 7 1 1 7 1 6 3 7 6 7 4 8 6 7 1 6 3 6 0 2 2 8 1 2 4 0 4 7 0 6 8 7 7
## [2443] 6 6 8 0 0 9 6 8 0 0 9 6 9 1 6 0 6 8 0 0 5 1 3 5 7 2 9 7 3 9 9 7 4 7 7 9 7
## [2480] 0 0 5 5 8 0 1 0 6 8 6 9 6 8 5 7 9 6 8 2 2 6 8 0 0 9 6 8 1 3 7 1 4 5 8 8 3
## [2517] 1 0 8 7 1 1 1 8 7 5 0 3 8 7 4 1 3 8 8 3 1 7 8 7 8 6 0 4 8 4 1 1 1 8 4 6 0
## [2554] 1 6 6 1 0 2 6 7 5 0 1 6 7 2 0 4 7 2 7 6 1 7 1 7 6 2 7 2 1 6 0 7 2 2 1 7 7
## [2591] 2 7 6 5 9 8 1 0 8 9 8 2 2 6 9 8 1 2 8 3 1 4 9 8 0 1 1 9 8 9 8 4 0 8 9 8 0
## [2628] 5 2 5 5 3 0 9 8 5 6 9 8 8 1 3 9 8 1 0 4 9 8 1 2 4 9 8 0 3 4 9 8 8 0 1 9 8
## [2665] 4 3 3 9 8 6 2 6 9 7 3 0 5 5 3 1 4 2 6 8 0 0 9 6 3 1 4 1 6 5 0 1 6 4 5 0 4
## [2702] 1 1 1 6 5 1 0 2 6 4 1 9 8 6 3 0 1 1 6 4 1 8 0 6 3 3 0 3 6 7 3 6 3 0 0 6 5
## [2739] 8 0 2 6 5 2 7 6 6 4 7 3 0 6 4 1 4 1 6 1 9 4 1 1 4 6 4 1 6 5 4 1 2 4 6 4 8
## [2776] 0 1 6 5 8 0 8 6 6 1 6 6 5 3 3 6 4 6 1 9 9 6 8 7 9 9 5 1 5 9 9 5 0 2 9 9 7
## [2813] 0 2 5 3 7 1 9 9 6 9 9 5 9 6 6 2 0 6 6 6 6 1 9 6 6 1 0 3 3 0 7 3 6 0 9 5 6
## [2850] 3 0 3 6 1 1 2 3 6 5 3 2 3 5 7 6 0 3 5 6 6 1 3 5 2 0 9 3 5 2 0 1 3 6 3 5 2
## [2887] 2 3 3 5 2 1 7 6 7 4 9 3 6 0 2 5 3 5 5 0 1 5 1 3 6 2 0 1 0 2 9 0 6 0 2 9 0
## [2924] 1 8 4 9 0 2 8 8 7 0 4 9 5 0 2 8 8 7 2 8 6 0 2 9 3 0 2 8 5 2 5 5 4 4 0 4 5
## [2961] 3 5 2 3 5 4 2 0 5 9 7 0 4 3 1 4 0 5 3 2 1 8 7 9 0 6 4 6 3 8 3 4 7 2 4 0 4
## [2998] 6 5 0 7 4 6 1 6 3 4 6 0 1 4 6 0 7 1 4 6 7 6 4 6 5 3 4 6 9 0 5 1 4 2 9 4 6
## [3035] 5 5 6 4 6 5 5 6 4 6 0 3 2 4 6 9 9 4 6 2 9 1 4 6 2 6 0 4 6 2 9 0 0 3 2 2 1
## [3072] 9 4 6 4 6 0 4 6 2 9 1 4 6 2 0 6 0 1 0 7 4 6 2 9 1 4 2 9 1 2 0 6 8 8 4 7 9
## [3109] 6 4 4 6 6 0 4 4 6 1 5 1 4 7 8 1 1 4 0 4 7 8 1 1 4 7 9 0 7 4 6 5 8 5 3 1 2
## [3146] 9 0 5 0 0 5 3 1 1 5 9 2 4 4 4 2 3 1 4 0 4 4 8 1 5 4 4 5 9 2 9 9 7 1 2 4 9
## [3183] 7 2 1 9 7 0 7 5 9 7 3 0 9 9 8 3 9 0 9 8 0 2 9 0 9 8 0 5 9 8 0 3 1 6 7 4 3
## [3220] 6 1 7 6 7 3 5 6 7 2 0 1 6 7 8 4 6 1 4 8 9 6 7 8 6 4 2 1 2 0 4 1 3 1 9 0 4
## [3257] 4 6 2 9 1 2 9 1 0 0 1 0 4 6 2 8 3 8 3 0 3 9 9 7 7 5 6 7 9 0 1 8 1 0 9 6 7
## [3294] 0 1 9 6 8 1 8 5 0 5 0 1 5 2 4 0 5 0 2 0 8 5 0 0 8 9 1 0 9 8 9 4 0 6 8 9 1
```

```
## [3331] 2 2 7 0 1 4 9 7 8 0 0 7 0 1 6 1 7 0 8 2 1 7 0 6 5 8 7 0 4 4 8 5 7 5 7 0 5
## [3368] 7 4 3 5 5 7 7 0 7 1 9 7 1 0 9 8 7 1 0 7 8 7 7 9 0 0 1 8 9 1 1 9 8 9 1 0 8
## [3405] 8 9 0 1 5 8 9 7 0 1 9 9 5 2 9 9 8 0 2 9 9 6 4 5 9 9 8 0 1 9 9 5 7 2 9 9 8
## [3442] 2 1 9 9 7 0 5 9 6 8 1 3 9 6 0 8 6 7 2 1 8 6 7 1 1 4 7 9 9 6 6 7 9 9 7 7 9
## [3479] 9 7 7 7 9 9 6 6 7 9 9 6 6 7 9 9 7 5 7 9 9 7 7 9 9 7 5 7 9 9 7 7 7 9 9 7 7
## [3516] 7 9 9 6 6 7 9 9 1 2 7 9 9 7 7 7 9 9 6 7 9 9 7 5 7 9 9 7 7 9 0 0 6 0 6 6 6
## [3553] 1 4 6 6 1 0 1 6 6 1 1 2 6 6 4 4 2 6 6 0 3 6 3 4 9 5 6 1 8 4 0 0 0 1 6 4 5
## [3590] 0 2 6 5 8 0 8 6 6 6 5 1 0 2 6 4 0 1 5 4 8 0 1 6 5 1 0 2 6 9 9 5 0 5 8 3 8
## [3627] 6 4 8 3 8 6 4 0 1 7 8 7 1 5 6 0 1 5 7 1 1 7 5 5 1 8 2 5 5 1 8 2 5 1 1 1 5
## [3664] 5 1 6 4 1 4 1 1 0 9 4 0 8 0 1 9 8 2 9 4 1 3 9 5 5 1 8 9 4 1 2 8 9 4 1 1 4
## [3701] 9 4 1 1 5 1 4 1 6 7 9 6 9 0 0 0 1 9 5 1 1 0 1 9 0 0 0 4 8 0 5 1 9 8 6 9 6
## [3738] 3 4 6 9 3 1 0 0 0 0 1 9 0 5 7 9 4 9 1 6 2 3 3 0 7 9 9 7 7 5 1 9 0 9 1 6 0
## [3775] 9 0 6 8 2 0 4 3 2 1 1 4 7 6 0 8 7 2 3 7 5 6 7 0 5 5 3 9 4 0 8 5 2 0 5 2 7
## [3812] 3 6 7 8 6 4 0 2 8 2 2 6 2 6 4 6 0 6 4 9 4 5 2 7 3 6 5 2 7 3 6 1 4 3 0 4 0
## [3849] 9 2 8 9 5 3 6 2 2 7 3 6 5 2 7 3 6 5 5 3 9 4 1 4 1 4 2 1 7 1 4 0 0 6 2 2 9
## [3886] 4 4 1 2 3 2 8 1 3 4 4 8 0 8 4 4 5 8 0 1 0 5 0 1 0 0 0 3 9 9 5 0 4 3 3 9 5
## [3923] 0 8 5 5 3 1 2 9 8 1 4 8 3 3 7 2 0 3 3 5 5 0 2 8 4 5 9 1 4 1 2 1 7 0 8 6 0
## [3960] 5 4 2 2 1 2 6 8 5 2 7 3 6 0 5 5 0 1 0 0 0 1 9 0 7 1 2 0 0 1 2 1 6 9 0 0 7
## [3997] 1 9 1 3 0 9 1 0 0 6 0 8 7 5 2 7 3 6 4 9 4 1 3 5 5 3 9 3 7 9 9 7 5 2 7 4 1
## [4034] 6 0 2 8 7 2 4 0 3 4 3 4 4 7 8 0 3 1 2 7 3 6 3 3 5 4 7 3 5 3 8 8 2 0 4 3 0
## [4071] 2 6 3 3 4 3 1 0 6 4 4 1 3 0 5 0 3 5 4 4 8 1 9 0 6 6 5 4 2 4 4 8 2 7 1 6 3
## [4108] 0 3 0 5 8 5 3 6 5 7 1 3 6 0 4 5 2 3 0 3 3 4 9 7 4 6 2 9 1 1 8 5 5 0 7 0 4
## [4145] 0 7 8 0 7 9 4 9 1 4 8 0 6 4 3 2 7 6 9 0 3 2 1 2 6 8 6 0 6 8 0 5 5 7 7 4 4
## [4182] 1 4 7 7 2 6 6 9 3 5 2 3 6 1 8 7 4 4 9 0 9 3 0 5 5 0 1 0 0 0 2 7 0 4 5 7 7
## [4219] 8 4 0 0 3 8 0 1 7 6 5 4 3 4 1 2 2 0 7 4 3 0 8 4 0 7 4 9 4 1 3 2 3 0 2 0 9
## [4256] 3 6 0 1 1 3 2 7 3 0 4 1 0 2 2 6 0 1 3 9 9 9 4 8 0 2 9 6 4 1 1 2 3 3 1 7 6
## [4293] 2 1 7 7 9 3 1 7 0 1 7 0 0 9 0 7 0 7 6 5 5 3 4 9 0 2 4 1 5 2 7 3 6 1 4 2 2
## [4330] 0 7 0 0 2 1 4 0 1 5 5 3 6 5 1 4 7 7 9 1 3 6 4 6 6 0 3 6 5 7 0 3 3 8 0 4 4
## [4367] 5 1 2 7 0 7 8 5 4 8 0 2 4 5 5 4 6 0 3 3 0 2 0 2 0 4 2 3 4 7 9 9 6 6 2 1 2
## [4404] 6 8 1 6 7 3 2 3 0 8 1 4 2 0 3 7 5 0 7 4 4 5 4 9 7 3 8 8 7 2 5 2 7 3 6 3 5
## [4441] 8 0 9 0 8 0 4 5 6 0 0 4 4 2 2 2 1 2 6 8 0 2 3 7 0 6 1 6 1 6 8 5 4 4 2 1 2
## [4478] 2 8 8 7 7 5 3 6 2 1 2 6 8 3 3 0 2 1 3 2 2 4 4 4 5 0 4 0 0 2 1 1 6 8 0 9 0
## [4515] 8 5 2 2 2 8 3 3 3 6 1 3 3 3 5 2 9 9 0 2 0 0 8 8 3 0 3 3 0 6 9 0 8 7 8 5 5
## [4552] 4 6 0 7 9 9 6 6 4 4 2 5 8 3 2 8 0 8 9 1 0 8 0 8 5 3 2 2 3 0 7 0 3 0 9 4 9
## [4589] 2 8 3 2 9 5 8 7 7 0 8 4 2 4 3 6 3 1 6 4 4 1 2 0 8 0 6 8 2 8 2 5 2 8 6 0 1
## [4626] 5 5 3 9 9 2 0 7 8 2 2 9 1 0 8 5 4 1 3 3 4 4 0 3 0 1 3 3 3 2 4 5 7 4 0 8 0
## [4663] 7 7 3 3 4 7 7 2 3 4 5 6 3 9 6 1 7 4 3 8 2 3 2 0 1 0 8 5 4 1 4 6 2 4 0 0 9
## [4700] 8 7 7 0 7 7 6 0 0 6 1 3 6 0 1 3 2 6 7 2 5 5 3 6 5 7 8 3 5 8 3 3 0 6 5 4 4
## [4737] 2 2 1 3 3 6 0 9 1 4 1 7 3 0 8 0 2 9 4 0 0 6 8 8 0 0 6 4 0 4 3 2 0 5 5 1 9
## [4774] 0 0 8 0 6 0 6 0 9 0 2 9 2 2 0 1 6 0 6 4 6 6 3 1 4 1 3 3 2 5 3 0 0 6 0 6 9
## [4811] 3 0 2 9 6 2 7 1 1 4 7 0 2 6 4 0 8 8 5 7 7 9 9 1 2 4 0 2 1 0 6 0 9 0 2 5 5
## [4848] 4 4 0 4 8 9 1 0 2 1 2 0 7 3 3 5 9 0 5 5 3 5 6 3 1 0 5 1 4 4 0 8 7 7 5 0 4
## [4885] 2 4 5 4 6 9 0 7 7 0 3 3 4 0 3 2 8 6 0 1 0 2 0 2 1 3 3 8 5 2 0 8 5 0 4 6 5
## [4922] 8 0 2 7 3 6 3 2 0 3 4 4 4 9 0 5 6 1 5 5 4 6 0 6 0 6 1 1 3 4 4 1 2 3 7 0 3
## [4959] 1 4 2 0 7 0 6 0 3 8 7 9 9 6 6 6 8 1 4 7 3 3 6 5 1 9 4 1 9 5 5 3 9 4 4 7 4
## [4996] 0 1 7 2 9 1 3 3 8 4 0 1 4 4 1 1 1 0 5 5 0 1 9 1 0 7 9 9 7 5 0 7 0 7 9 3 3
## [5033] 6 3 0 6 0 6 3 2 5 5 1 1 1 1 2 1 8 0 3 5 9 0 3 6 6 6 2 6 4 4 4 4 5 5 3 9 4
## [5070] 1 3 1 2 7 4 0 1 1 8 7 5 4 4 0 1 6 0 2 7 2 6 9 6 2 0 0 4 3 2 1 0 5 5 0 9 6
## [5107] 5 3 3 4 1 7 3 3 9 0 7 3 9 0 1 4 1 4 6 5 1 0 2 1 4 2 4 0 8 1 2 1 3 3 9 3 6
## [5144] 1 3 8 8 6 2 6 7 3 7 0 2 7 2 3 7 0 3 5 5 3 9 9 2 8 1 0 3 3 4 8 0 8 0 9 3 1
## [5181] 0 5 3 3 3 3 1 2 4 0 5 0 4 9 4 2 6 7 6 0 1 7 3 3 5 4 4 5 5 6 0 0 5 5 0 1 8 9
## [5218] 1 2 4 3 3 5 5 5 1 8 2 0 6 0 1 6 3 3 5 9 5 1 4 2 0 2 9 2 8 0 7 9 9 7 7 0 4
## [5255] 0 9 6 5 5 3 9 4 0 7 6 4 4 0 7 9 3 6 9 1 3 6 7 1 4 1 7 3 4 9 4 1 3 4 2 4 0
## [5292] 7 1 4 4 4 4 4 6 3 3 9 3 1 7 2 7 5 7 7 7 5 7 3 4 9 4 1 3 6 0 7 0 4 4 5 1 2 0
```

```
## [5329] 1 3 0 1 0 8 6 5 0 4 4 1 9 3 3 9 2 0 5 3 1 1 0 6 3 1 0 4 6 4 1 1 9 6 3 1 6
## [5366] 0 6 3 1 1 4 6 0 0 5 7 5 0 7 4 7 5 2 3 1 9 9 6 0 3 9 9 5 0 4 8 3 4 4 1 1 3
## [5403] 3 8 8 0 1 5 0 6 6 2 5 2 7 3 6 8 1 0 0 4 8 1 0 0 9 2 9 9 0 5 5 8 3 1 6 6 6
## [5440] 8 1 3 1 8 5 4 1 7 9 9 6 6 0 4 1 0 4 7 0 1 1 3 1 4 1 0 6 4 1 9 9 4 4 1 0 6
## [5477] 9 9 9 0 1 4 2 1 1 1 2 1 3 1 4 2 1 2 1 4 2 1 2 1 4 2 6 1 1 2 2 7 1 4 2 2 2
## [5514] 1 4 2 2 2 1 4 2 6 1 4 2 1 4 1 4 2 4 0 1 4 2 2 1 1 4 2 2 0 1 4 2 2 0 3 0 0
## [5551] 1 4 2 1 3 1 4 2 1 4 1 4 2 1 8 1 4 2 1 4 1 4 2 1 4 4 1 2 4 2 3 6 4 2 2 2 3
## [5588] 4 2 2 2 3 0 5 7 6 3 0 5 6 0 2 0 5 8 7 3 2 6 1 4 9 0 2 9 1 0 8 7 1 0 2 9 0
## [5625] 1 0 2 9 2 0 2 9 0 3 9 9 9 0 0 2 9 3 0 3 0 3 2 0 3 4 4 7 6 6 2 1 2 6 6 6 1
## [5662] 4 6 3 1 0 1 6 3 1 2 2 8 3 0 3 8 0 3 2 3 5 9 2 0 4 1 0 2 9 0 0 4 3 9 0 0 0
## [5699] 1 7 5 2 1 4 7 5 2 2 2 5 0 4 3 2 6 6 8 8 3 9 5 6 0 8 3 6 0 5 7 5 4 2 1 3 2
## [5736] 4 2 3 8 3 2 2 2 1 0 6 7 3 1 1 6 7 3 5 0 5 7 4 8 0 2 7 3 1 1 3 7 1 3 0 1 1
## [5773] 4 1 1 6 0 7 0 0 7 9 9 7 7 7 7 0 3 5 7 9 9 7 5 7 6 0 1 7 7 9 9 7 3 7 9 9 7
## [5810] 7 8 9 1 1 8 0 4 1 0 7 9 6 7 9 7 9 6 8 1 8 5 7 3 1 9 5 7 6 3 8 7 7 8 0 3 1
## [5847] 0 4 8 1 2 1 6 6 2 1 0 9 9 7 0 4 5 3 6 2 8 0 5 1 5 8 5 3 7 7 5 3 8 0 6 1 1
## [5884] 7 0 1 0 6 8 5 4 4 6 8 5 4 4 3 9 5 6 0 3 9 1 2 4 4 1 3 0 4 4 1 0 4 1 4 1 0
## [5921] 6 3 9 5 3 0 5 8 7 7 3 5 8 5 0 2 5 9 7 1 2 6 3 1 1 6 3 1 0 5 6 4 1 1 1 0 3
## [5958] 0 4 7 5 2 1 7 7 5 2 3 4 7 5 2 5 8 7 5 2 6 5 5 1 9 0 5 5 4 4 0 5 4 0 0 0 9
## [5995] 9 5 5 4 8 5 5 5 5 5 4 1 5 5 5 4 1 4 1 9 6 8 6 3 3 3 3 1 6 2 8 9 9 3 3 3 1
## [6032] 2 3 3 3 1 9 3 3 3 0 2 3 3 3 0 9 3 3 3 1 2 1 9 5 7 3 3 2 5 3 3 3 3 4 3 8 8
## [6069] 6 6 3 0 3 3 2 3 0 3 1 0 3 0 3 0 3 0 3 5 9 3 8 2 4 3 2 3 0 9 0 7 3 3 0 2 3
## [6106] 0 0 6 7 1 4 1 0 6 0 4 1 0 2 3 3 3 0 2 3 3 3 0 2 3 3 1 8 4 4 1 0 2 4 4 1 1
## [6143] 0 4 4 1 0 2 4 4 1 1 2 8 0 3 2 2 5 5 1 8 5 9 0 7 5 2 1 4 3 4 2 5 3 0 3 2 5
## [6180] 3 3 5 7 0 0 6 0 0 7 6 0 2 0 8 5 2 7 3 6 3 8 0 5 4 3 7 7 6 3 1 3 8 4 0 2 1
## [6217] 8 4 2 3 0 1 3 0 9 0 6 5 0 5 3 1 4 1 1 7 9 4 9 1 1 0 5 6 5 2 9 9 0 2 2 9 9
## [6254] 2 8 2 8 2 3 4 2 8 2 0 2 6 3 1 8 6 3 1 5 0 0 0 7 1 8 2 5 0 1 8 2 0 0 1 3 9
## [6291] 5 6 4 8 9 0 1 5 5 2 7 3 6 5 2 7 3 6 5 2 7 3 6 2 7 3 6 6 7 6 2 2 9 9 0 5 7
## [6328] 3 5 2 3 5 4 0 1 3 0 2 4 5 3 1 9 0 6 7 6 0 1 2 0 7 1 8 5 1 7 7 9 0 7 6 3 1
## [6365] 0 7 6 1 6 1 0 1 1 5 0 6 1 1 5 2 0 5 0 3 5 0 0 3 1 9 2 8 2 2 6 2 8 2 3 4 1
## [6402] 1 2 2 1 3 2 8 2 2 8 1 3 7 5 6 0 2 9 0 1 1 1 6 5 9 6 8 1 4 6 9 1 6 0 6 9 1
## [6439] 6 0 8 1 0 2 8 9 1 1 5 6 4 1 5 2 6 1 7 8 0 2 9 0 8 8 3 7 0 2 9 6 7 4 4 8 2
## [6476] 8 0 1 8 2 8 0 1 8 7 1 2 4 8 7 1 1 7 6 8 1 7 5 5 9 4 0 2 5 3 8 0 7 3 1 9 4
## [6513] 7 3 1 0 6 2 9 2 1 6 0 2 2 9 5 7 7 2 9 9 2 8 5 3 6 9 0 0 0 5 5 2 7 3 6 0 5
## [6550] 8 1 9 6 6 2 0 8 5 7 0 3 3 0 5 0 3 2 4 2 0 8 3 7 0 1 3 8 1 3 8 7 1 1 8 6 0
## [6587] 6 1 8 3 0 6 1 0 7 4 4 1 9 9 4 4 1 4 4 4 4 4 1 1 2 5 5 4 6 0 2 6 5 0 5 2 6 1
## [6624] 8 4 2 6 5 0 6 2 6 5 0 5 7 2 0 8 6 7 4 0 7 7 5 2 7 3 6 8 9 1 1 9 8 9 1 0 2
## [6661] 8 9 5 0 6 7 9 9 6 6 7 7 0 7 1 8 5 1 7 7 9 0 7 0 9 4 5 0 3 5 0 8 9 1 1 7 8
## [6698] 9 1 1 4 8 9 1 1 5 7 5 2 7 5 7 5 2 6 6 7 5 2 2 8 5 5 4 1 1 5 9 9 3 7 8 3 2
## [6735] 1 8 7 1 1 2 8 8 0 0 1 8 5 4 0 3 6 8 0 0 9 7 2 1 7 7 5 7 2 2 1 5 5 8 8 9 6
## [6772] 7 4 0 8 6 0 2 6 7 4 2 0 6 6 0 3 1 4 3 2 8 4 6 0 1 1 4 3 0 4 8 9 7 0 1 8 9
## [6809] 4 3 1 7 4 1 3 3 7 3 1 0 6 7 3 1 2 0 7 4 3 6 1 2 8 2 1 1 5 5 4 4 0 5 0 0 0
## [6846] 5 5 4 6 0 0 4 1 0 1 0 4 1 0 1 0 4 1 0 1 1 4 1 0 4 1 4 0 5 0 0 1 5 0 3 0 0
## [6883] 3 7 2 0 0 3 7 0 0 8 2 0 0 1 8 7 1 1 2 5 8 1 0 3 5 8 7 0 1 5 6 5 3 1 5 3 6
## [6920] 7 3 2 7 7 9 6 8 7 0 1 6 8 1 7 2 0 0 5 6 8 0 0 9 6 9 1 6 6 6 2 0 1 6 6 6 1
## [6957] 1 9 6 8 6 3 8 1 7 6 4 1 1 3 6 3 1 6 4 0 6 1 0 8 8 2 4 0 1 8 2 0 0 9 8 2 4
## [6994] 3 5 8 2 5 0 1 8 2 8 2 4 0 1 8 2 6 0 4 8 6 4 9 8 2 5 2 0 8 2 5 8 2 0 7 1 8
## [7031] 3 0 0 1 8 4 1 4 3 9 7 0 2 3 9 0 4 2 3 9 3 1 1 7 1 4 5 9 0 2 0 7 0 8 0 7 7
## [7068] 0 0 5 2 0 2 3 9 0 5 0 1 2 0 0 0 9 0 2 9 9 2 5 5 4 8 0 0 9 0 2 7 5 8 8 2 2
## [7105] 0 8 7 1 2 4 7 2 0 2 3 7 2 4 0 1 7 2 8 3 0 8 4 1 2 1 8 3 4 4 0 5 8 1 0 9 5
## [7142] 8 5 4 8 8 5 0 1 6 4 1 4 1 6 3 1 6 4 5 7 1 9 7 5 7 3 2 5 7 1 1 7 6 0 0 4 7
## [7179] 4 7 2 2 1 0 5 5 4 6 0 1 4 0 9 7 5 4 0 5 4 0 3 2 9 0 6 3 7 9 9 6 6 1 1 0 6
## [7216] 1 0 3 0 6 1 8 3 0 6 1 0 4 0 6 1 4 3 0 6 1 5 8 2 0 1 5 9 2 7 0 5 9 4 0 4 8
## [7253] 3 4 0 2 8 3 3 4 0 5 0 3 3 6 5 0 3 5 0 5 0 3 0 9 3 3 3 0 3 3 3 3 1 1 3 3 3
## [7290] 0 1
```

```
## Levels: 0 1 2 3 4 5 6 7 8 9
```

```
############################## Q6 ######################################
# Extract labels and features from the test dataset
test_labels <- test_data[, 1]
test_features <- test_data[, -1]

# Convert labels to factor
test_labels <- as.factor(test_labels)

# Use the final k-NN model to make predictions on the test dataset
test_predictions <- knn(train_features, test_features, train_labels, k = final_k)

# Create a confusion matrix
conf_matrix <- table(Actual = test_labels, Predicted = test_predictions)

# Print the confusion matrix
cat('Confusion Matrix:\n')
```

```
## Confusion Matrix:
```

```
print(conf_matrix)
```

```
##       Predicted
## Actual   0   1   2   3   4   5   6   7   8   9
##      0 355   0   2   0   0   0   0   1   0   1
##      1   0 255   0   0   6   0   2   1   0   0
##      2   6   1 183   2   1   0   0   2   3   0
##      3   3   0   2 154   0   5   0   0   0   2
##      4   0   3   1   0 182   1   2   2   1   8
##      5   2   1   2   4   0 145   2   0   3   1
##      6   0   0   1   0   2   3 164   0   0   0
##      7   0   1   1   1   4   0   0 139   0   1
##      8   5   0   1   6   1   1   0   1 148   3
##      9   0   0   1   0   2   0   0   4   1 169
```

```
# Calculate the conditional test error using zero-one loss
conditional_test_error <- 1 - sum(diag(conf_matrix)) / sum(conf_matrix)
cat('\nConditional Test Error (Zero-One Loss):', conditional_test_error, '\n')
```

```
##
## Conditional Test Error (Zero-One Loss): 0.05630294
```