Irisk Lab report1

Ruibo Hou

2024-02-02

```r
# Download the first 2000 observations of the MNIST dataset
mnist <- read.csv("https://pjreddie.com/media/files/mnist_train.csv", nrows = 2000)
colnames(mnist) <- c("Digit", paste("Pixel", seq(1:784), sep = ""))
save(mnist, file = "mnist_first2000.RData")
```

1 knn

```r
# Load the required packages
library(caret)
```

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

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

```r
library(class)

# Load the MNIST dataset
load("mnist_first2000.RData")

# Split the dataset into training data (first 1000 rows) and testing data (last 1000 rows)
train_data <- mnist[1:1000,]
test_data <- mnist[1001:2000,]

# Train a KNN model using 5-fold cross-validation
train_control <- trainControl(method="cv", number=5)
tune_grid <- expand.grid(.k=1:20)
knn_model <- train(as.factor(Digit)~., data=train_data, method="knn", tuneGrid=tune_grid, trControl=trai

# Select the value of K with the lowest cross-validation error
best_k <- knn_model$bestTune$k
print(best_k)
```

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

```r
# Use the KNN algorithm for classification
knn_fit <- knn(train=train_data[,1:785], test=test_data[,1:785], cl=train_data$Digit, k=best_k)
```

```r
# Calculate the prediction error and confusion matrix
error_rate <- mean(knn_fit != test_data$Digit)
confusion_matrix <- table(knn_fit, test_data$Digit)
print(confusion_matrix)
```

```
##
## knn_fit   0   1   2   3   4   5   6   7   8   9
##       0  89   0   0   1   0   1   1   0   2   1
##       1   1 103   4   1   6   0   0   1   1   1
##       2   0   0  86   4   1   0   0   0   0   0
##       3   0   0   0  76   0   6   0   0   1   1
##       4   0   0   1   0  81   0   1   0   1   5
##       5   0   0   0  11   1  76   1   2   3   0
##       6   2   0   0   0   2   3 103   0   3   0
##       7   0   1   5   1   1   0   0 101   0   5
##       8   0   0   3   2   1   1   0   0  70   0
##       9   1   0   0   2  16   2   0   3   4  97
```

```r
# Print the prediction classification error
print(paste("Prediction classification error:", error_rate))
```

```
## [1] "Prediction classification error: 0.118"
```

2 logistic regression

```r
# Load the glmnet package
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
# Load the MNIST dataset
load("mnist_first2000.RData")

# Split the data into training and testing sets
train_data <- mnist[1:1000,]
test_data <- mnist[1001:2000,]

# Extract the predictors and response variables
x_train <- train_data[,2:785]
y_train <- train_data[, 1]
x_test <- test_data[,2:785]
y_test <- test_data[, 1]

# Convert non-numeric columns to numeric using model.matrix()
x_train <- model.matrix(~., data=train_data[, -1])
x_test  <- model.matrix(~., data=test_data[, -1])
# Fit a multi-class logistic regression model with Lasso penalty
```

```r
fit <- cv.glmnet(x_train, y_train, family="multinomial", alpha=1, type.measure="class")

# Select the best tuning parameter
best_lambda <- fit$lambda.min

# Make predictions on the testing data
pred <- predict(fit, newx=x_test, s=best_lambda, type="class")

# Compute the confusion matrix
confusion_matrix <- table(pred, y_test)

# Compute the prediction classification error
error_rate <- mean(pred != y_test)

# Print the confusion matrix and prediction classification error
print(confusion_matrix)
```

```
##     y_test
## pred  0  1  2  3  4  5  6  7  8  9
##    0 83  0  1  0  0  2  2  0  3  4
##    1  0 90  2  1  1  0  1  0  3  0
##    2  0  1 73 12  3  0  2  0  2  0
##    3  1  1  1 63  1  1  0  2  1  2
##    4  1  0  2  0 81  3  1  3  0 10
##    5  5  0  1 10  2 74  3  1  1  1
##    6  0  0  5  1  3  1 96  0  1  0
##    7  1  1  6  2  3  2  0 92  2  7
##    8  2 11  5  5  2  4  1  2 68  2
##    9  0  0  3  4 13  2  0  7  4 84
```

```r
print(paste("Prediction classification error:", error_rate))
```

```
## [1] "Prediction classification error: 0.196"
```