Weight Lifting Exercise Classification

Mohammad Sohail.H

2025-11-08

Contents

Introduction

This project aims to predict the manner in which people perform weight lifting exercises. The dataset comes from accelerometers placed on the belt, forearm, arm, and dumbbell of study participants.

The target variable **classe** indicates 5 different ways the exercise was performed: - **A**: Correct execution - **B–E**: Common types of mistakes

We will clean the data, train a machine learning model, evaluate accuracy, and make final predictions on the test set.

Load Libraries and Data

```
library(caret)
library(randomForest)

set.seed(123)

train <- read.csv("pml-training.csv", na.strings = c("", "NA", "#DIV/0!"))
test <- read.csv("pml-testing.csv", na.strings = c("", "NA", "#DIV/0!"))
train <- train[, colSums(is.na(train)) == 0]
test   <- test[, colSums(is.na(test)) == 0]

train <- train[, -(1:7)]
test  <- test[, -(1:7)]

train$classe <- as.factor(train$classe)
inTrain <- createDataPartition(train$classe, p = 0.7, list = FALSE)
trainSet <- train[inTrain, ]
valSet   <- train[-inTrain, ]
model <- randomForest(classe ~ ., data = trainSet)
pred <- predict(model, valSet)
confusionMatrix(pred, valSet$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    4    0    0    0
##          B    0 1131    4    0    0
##          C    0    4 1022    7    4
##          D    0    0    0  957    4
##          E    0    0    0    0 1074
##
## Overall Statistics
##
##                Accuracy : 0.9954
##                  95% CI : (0.9933, 0.997)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9942
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9930   0.9961   0.9927   0.9926
## Specificity            0.9991   0.9992   0.9969   0.9992   1.0000
## Pos Pred Value         0.9976   0.9965   0.9855   0.9958   1.0000
## Neg Pred Value         1.0000   0.9983   0.9992   0.9986   0.9983
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1922   0.1737   0.1626   0.1825
## Detection Prevalence   0.2851   0.1929   0.1762   0.1633   0.1825
## Balanced Accuracy      0.9995   0.9961   0.9965   0.9960   0.9963
```

```
testPred <- predict(model, test)
testPred
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```