

Weight Lifting Exercise Prediction

Medini Jeere

2025-09-28

```
# -----  
# Load Libraries and Data  
# -----  
library(caret)  
library(randomForest)  
library(dplyr)  
  
train_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"  
test_url  <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"  
  
training <- read.csv(train_url, na.strings=c("NA", "#DIV/0!", ""))  
testing  <- read.csv(test_url, na.strings=c("NA", "#DIV/0!", ""))  
  
# -----  
# Data Cleaning  
# -----  
training <- training[, colSums(is.na(training)) == 0]  
testing  <- testing[, colSums(is.na(testing)) == 0]  
  
training <- training[, -(1:7)]  
testing  <- testing[, -(1:7)]  
  
training$classe <- factor(training$classe)  
  
# -----  
# Split Training and Validation Sets  
# -----  
set.seed(123)  
inTrain <- createDataPartition(training$classe, p=0.7, list=FALSE)  
trainData <- training[inTrain, ]  
validData <- training[-inTrain, ]  
  
# -----  
# Train Random Forest Model  
# -----  
set.seed(123)  
ctrl <- trainControl(method="cv", number=5)  
rfModel <- train(classe ~ ., data=trainData, method="rf", trControl=ctrl)  
  
print(rfModel)  
  
## Random Forest  
##
```

```
## 13737 samples
## 52 predictor
## 5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10990, 10990, 10988, 10990, 10990
## Resampling results across tuning parameters:
##
## mtry Accuracy Kappa
## 2 0.9895898 0.9868296
## 27 0.9910456 0.9886725
## 52 0.9856592 0.9818571
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
# -----
# Validate Model
# -----
pred_rf <- predict(rfModel, validData)
confMat <- confusionMatrix(pred_rf, validData$classe)
print(confMat)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    A    B    C    D    E
##           A 1673    7    0    0    0
##           B    1 1125    5    0    0
##           C    0    7 1019   10    4
##           D    0    0    2  954    4
##           E    0    0    0    0 1074
```

```
## Overall Statistics
```

```
##
##           Accuracy : 0.9932
##           95% CI : (0.9908, 0.9951)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##           Kappa : 0.9914
```

```
## McNemar's Test P-Value : NA
```

```
## Statistics by Class:
```

```
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity      0.9994  0.9877  0.9932  0.9896  0.9926
## Specificity      0.9983  0.9987  0.9957  0.9988  1.0000
## Pos Pred Value   0.9958  0.9947  0.9798  0.9937  1.0000
## Neg Pred Value    0.9998  0.9971  0.9986  0.9980  0.9983
## Prevalence       0.2845  0.1935  0.1743  0.1638  0.1839
## Detection Rate    0.2843  0.1912  0.1732  0.1621  0.1825
## Detection Prevalence 0.2855  0.1922  0.1767  0.1631  0.1825
```

```
## Balanced Accuracy      0.9989    0.9932    0.9944    0.9942    0.9963
```

```
# -----
```

```
# Predict on Test Data
```

```
# -----
```

```
pred_test <- predict(rfModel, testing)
```

```
pred_test
```

```
## [1] 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
```