Practical Machine Learning Course Project

T K

2025-08-06

Executive Summary

This project involves the analysis of data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. The objective is to predict the manner in which they did the exercise.

Two models were compared in order to determine the best methodology to predict the classes for the test dataset: the decision tree model and the random forest model. Following testing with the training dataset, it appeared that the random forest model was more accurate. The random forest model was then used to predict the exercise performance of the test dataset.

Loading Data/Libraries

Loading relevant data processing libraries and importing test/train data. Additional data filtering of missing values was performed for efficiency. The training data was further split into training and testing datasets.

```r
set.seed(1)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.5.1
```

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

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

```
## Warning: package 'lattice' was built under R version 4.5.1
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.5.1
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(rpart)

training <- read.csv("pml-training.csv", na.strings = c("", "NA"))
testing <- read.csv("pml-testing.csv", na.strings = c("", "NA"))
training$classe <- as.factor(training$classe)

train_filter <- training[, colSums(is.na(training)) == 0]
test_filter <- testing[, colSums(is.na(testing)) == 0]
```

```
training_data <- train_filter[, -c(1:7)]
testing_data <- test_filter[, -c(1:7)]

inTrain <- createDataPartition(training_data$classe, p = 0.7, list = FALSE)
training_subset <- training_data[inTrain, ]
validation_subset <- training_data[-inTrain, ]
```

Selecting Models

Deciding between the decision tree model or the random forest model by using the partitioned training data.

```
dt <- train(classe ~ ., data = training_subset, method = "rpart", trControl = trainControl(method = "cv
dt_prediction <- predict(dt, validation_subset)
confusionMatrix(validation_subset$classe, dt_prediction)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1523   23  124    0    4
##          B  498  350  291    0    0
##          C  489   30  507    0    0
##          D  412  169  383    0    0
##          E  174  139  280    0  489
##
## Overall Statistics
##
##                Accuracy : 0.4875
##                  95% CI : (0.4747, 0.5004)
##     No Information Rate : 0.5261
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3297
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4919  0.49226  0.31987       NA  0.99189
## Specificity            0.9459  0.84751  0.87930   0.8362  0.89002
## Pos Pred Value         0.9098  0.30729  0.49415       NA  0.45194
## Neg Pred Value         0.6265  0.92394  0.77814       NA  0.99917
## Prevalence             0.5261  0.12082  0.26933   0.0000  0.08377
## Detection Rate         0.2588  0.05947  0.08615   0.0000  0.08309
## Detection Prevalence   0.2845  0.19354  0.17434   0.1638  0.18386
## Balanced Accuracy      0.7189  0.66989  0.59959       NA  0.94095
```

```
postResample(dt_prediction, validation_subset$classe)
```

```
##  Accuracy     Kappa
## 0.4875106 0.3297412
```

```
oose_dt <- 1 - postResample(dt_prediction, validation_subset$classe)[1]
oose_dt
```

```
##  Accuracy
## 0.5124894
```

```
rf <- randomForest(classe ~ ., data=training_subset, control = trainControl(method = "cv", number = 3),
rf_prediction <- predict(rf, validation_subset)
confusionMatrix(validation_subset$classe, rf_prediction)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    2    0    0    0
##          B    1 1138    0    0    0
##          C    0    8 1013    5    0
##          D    0    0   16  947    1
##          E    0    0    0    2 1080
##
## Overall Statistics
##
##                Accuracy : 0.9941
##                  95% CI : (0.9917, 0.9959)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9925
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9913   0.9845   0.9927   0.9991
## Specificity            0.9995   0.9998   0.9973   0.9966   0.9996
## Pos Pred Value         0.9988   0.9991   0.9873   0.9824   0.9982
## Neg Pred Value         0.9998   0.9979   0.9967   0.9986   0.9998
## Prevalence             0.2843   0.1951   0.1749   0.1621   0.1837
## Detection Rate         0.2841   0.1934   0.1721   0.1609   0.1835
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9995   0.9955   0.9909   0.9946   0.9993
```

```
postResample(rf_prediction, validation_subset$classe)
```

```
##  Accuracy     Kappa
## 0.9940527 0.9924772
```

```
oose_rf <- 1 - postResample(rf_prediction, validation_subset$classe)[1]
oose_rf
```

```
##    Accuracy
## 0.005947324
```

Based on the results, the random forest model works better. It has an accuracy rate of 0.9940527 compared to the 0.4875106 of the decision tree model. The out of sample errors for both models are 0.5124894 and 0.005947324, respectively.

Predictions for Test Cases

Using the random forest method to categorize the test data.

```
predictions <- predict(rf, testing_data)
predictions
```

```
##  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
```