Practical Machine Learning Final Assignment

Simon Grasdal

2024-12-23

```
knitr::opts_chunk$set(echo = TRUE)
library(e1071)
library(caret)
```

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

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

```
library(rpart)
library(rpart.plot)
library(randomForest)
```

```
## 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
```

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

Background

One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Introduction

Because of a group of movement enthusiasts, the quantified self movement have recorded an astounding amount of data to determine how often movement are performed. Using devices from brands like Jawbone Up, Nike FuelBand, and Fitbit this data can be precisely measured and stored. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it.

In this assignment, we use accelerometer data from the wait (belt), forearm, are and dumbbells of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Using a machine learnign algorithm, we attempt to predict the *manner* in which each participant doe the excercise.

Data Cleaning/Preparation

Importing the Data

```
trainUrl <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trainFile <- "./data/pml-training.csv"
testFile  <- "./data/pml-testing.csv"
if (!file.exists("./data")) {
  dir.create("./data")
}
if (!file.exists(trainFile)) {
  download.file(trainUrl, destfile=trainFile, method="curl")
}
if (!file.exists(testFile)) {
  download.file(testUrl, destfile=testFile, method="curl")
}
```

Reading and CLeaning the Data

```
trainRaw <- read.csv("./data/pml-training.csv")
testRaw <- read.csv("./data/pml-testing.csv")
dim(trainRaw)
```

```
## [1] 19622   160
```

```
dim(testRaw)
```

```
## [1]  20 160
```

```
# Identifying Complete Cases
sum(complete.cases(trainRaw))
```

```
## [1] 406
```

```
# Removing Na's
trainRaw <- trainRaw[, colSums(is.na(trainRaw)) == 0]
testRaw <- testRaw[, colSums(is.na(testRaw)) == 0]

# Removing Columns that do not contribute to performace
classe <- trainRaw$classe
trainRemove <- grepl("^X|timestamp|window", names(trainRaw))
trainRaw <- trainRaw[, !trainRemove]
trainCleaned <- trainRaw[, sapply(trainRaw, is.numeric)]
trainCleaned$classe <- classe
testRemove <- grepl("^X|timestamp|window", names(testRaw))
testRaw <- testRaw[, !testRemove]
testCleaned <- testRaw[, sapply(testRaw, is.numeric)]
```

The training set rendered has 19,220 observations and 53 total variables. The Testing set contains 20 observations and likewise 53 variables.

Slicing the Data

The cleaned training set is split into a validation data set (30%) and a pure training data set (70%). The validation data set assists in conducting cross validation.

```
set.seed(20420) # For reproducible purpose
inTrain <- createDataPartition(trainCleaned$classe, p=0.70, list=F)
trainData <- trainCleaned[inTrain, ]
testData <- trainCleaned[-inTrain, ]
```

Modeling the Predictive Algorithm

Using a **Random Forest** algorithm we fit a predictive model for activity recognition. This algorithm was chosen because it automatically selects key variables and is robust. The *5-fold cross-validation* is used when applying the algorithm.

```
controlRf <- trainControl(method="cv", 5)
modelRf <- train(classe ~ ., data=trainData, method="rf", trControl=controlRf, ntree=250)
modelRf
```

```
## 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: 10989, 10989, 10990, 10990, 10990
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
```

```
##    2    0.9903182  0.9877515
##   27    0.9905368  0.9880297
##   52    0.9839853  0.9797423
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
predictRf <- predict(modelRf, newdata = testData)
#Testing accuracy
confusionMatrix(table(predictRf, testData$classe))
```

```
## Confusion Matrix and Statistics
##
##
## predictRf    A    B    C    D    E
##         A 1673   15    0    0    0
##         B    1 1117    4    1    0
##         C    0    7 1018   12    3
##         D    0    0    4  951    3
##         E    0    0    0    0 1076
##
## Overall Statistics
##
##                Accuracy : 0.9915
##                  95% CI : (0.9888, 0.9937)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9893
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9807   0.9922   0.9865   0.9945
## Specificity            0.9964   0.9987   0.9955   0.9986   1.0000
## Pos Pred Value         0.9911   0.9947   0.9788   0.9927   1.0000
## Neg Pred Value         0.9998   0.9954   0.9983   0.9974   0.9988
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1898   0.1730   0.1616   0.1828
## Detection Prevalence   0.2868   0.1908   0.1767   0.1628   0.1828
## Balanced Accuracy      0.9979   0.9897   0.9938   0.9925   0.9972
```

The quality of model fit for the prediction can be determined by calculating the values of accuracy and the out-of-sample Root Mean Square Error (RSME). In the interests of cross-validation, the RSME was normalized to aid in interpreting how well the prediction model fitted the test data.

The estimated accuracy of the model is 99.79% and the Normalized out-of-sample error (RMSE) has a low value between 1 and 0 (0.00815633). These results indicate a high degree of fit for the prediction of the dataset.

```
accuracy <- postResample(table(predictRf), table(testData$classe))
accuracy
```

```
##       RMSE   Rsquared        MAE
## 12.0000000  0.9983683 11.2000000
```

```
oose <- 1 - as.numeric(confusionMatrix(table(testData$classe, predictRf))$overall[1])
oose
```

```
## [1] 0.008496177
```

Final Model (Top 20 Predictor Variables)

```
modelRf$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 250, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 250
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.76%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3900    4    1    0    1 0.001536098
## B   23 2629    4    1    1 0.010910459
## C    0   12 2374   10    0 0.009181970
## D    0    1   25 2224    2 0.012433393
## E    0    1    5   14 2505 0.007920792
```

```
varImp(modelRf)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 52)
##
##                      Overall
## roll_belt            100.000
## pitch_forearm         65.019
## yaw_belt              56.395
## magnet_dumbbell_z     47.733
## magnet_dumbbell_y     44.852
## pitch_belt            44.212
## roll_forearm          43.800
## accel_dumbbell_y      23.559
## roll_dumbbell         20.202
## magnet_dumbbell_x     17.281
```

```
## accel_forearm_x       17.057
## magnet_belt_z         15.929
## accel_belt_z          15.312
## magnet_belt_y         14.964
## total_accel_dumbbell  13.817
## magnet_forearm_z      13.797
## accel_dumbbell_z      13.479
## yaw_arm               11.224
## gyros_belt_z          11.176
## magnet_belt_x          9.506
```

Conclusion

A machine learning (ML) model predicted the manner of participant exercise, which was Classe 'A'. Accelerometer data located on participants' belt, forearm, arm, and dumbell, from an Exercise dataset, was cleaned, and split into training and test datasets. The predictive model successfully identified the Classe. RSME calculations contributed to the anticipated strength of the model developped for this data.

Appendix

##Figures - Data Visualization

Figure 1. - Correlation Matrix Visualization

```
corrPlot <- cor(trainData[, -length(names(trainData))])
corrplot(corrPlot, method="color")
```

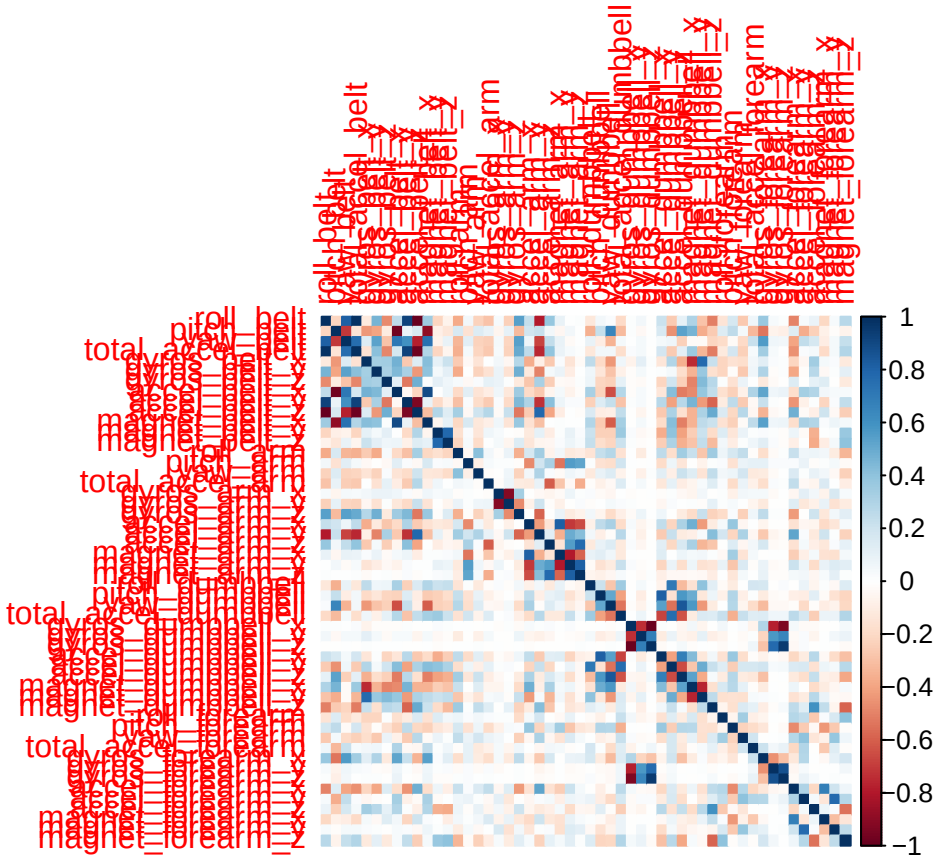

Figure 2. - Decision Tree Visualization

```
treeModel <- rpart(classe ~ ., data=trainData, method="class")
prp(treeModel) # fast plot
```

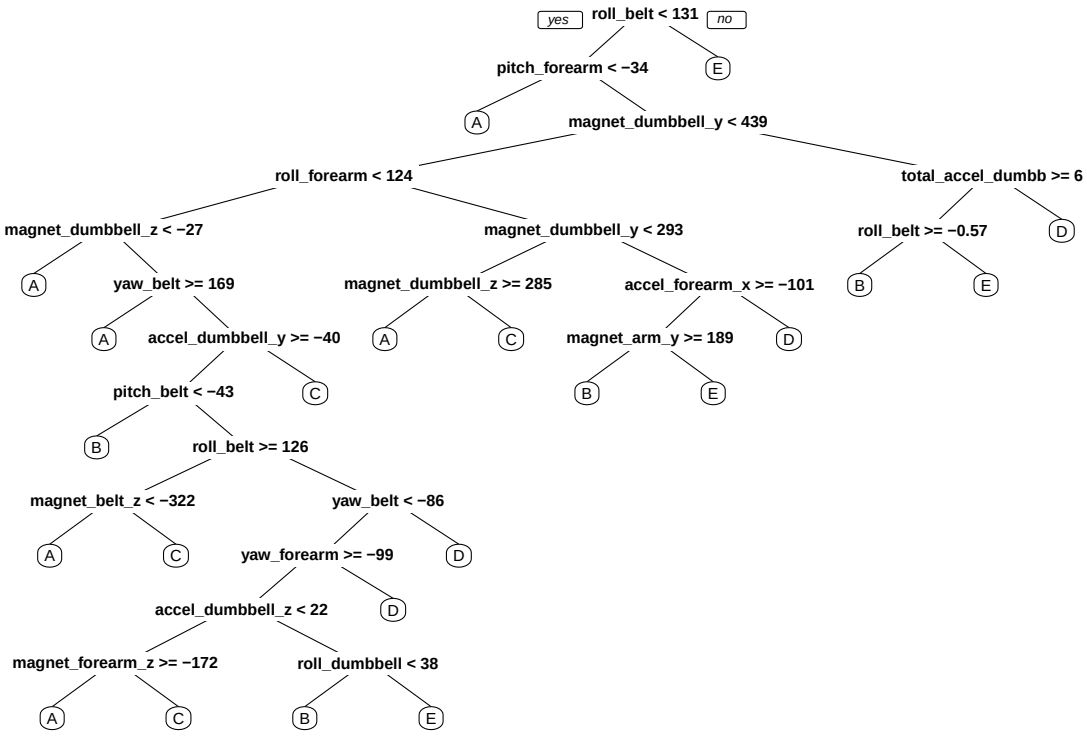

Figure 3. - Prediction Model

The prediction model was applied to the original testing data set, downloaded from the data source.

```
result <- predict(modelRf, testCleaned[, -length(names(testCleaned))])
result
```

```
##  [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
```