Practical Machine Learning - Course Project

Otavio Cals

Introduction

In this project we explored and analyzed the Human Activity Recognition project data gathered by the Groupware@LES team about personal activity. Our goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to quantify how well they do their physical activities. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways, those being:
* A: exactly according to the specification
* B: throwing the elbows to the front
* C: lifting the dumbbell only halfway
* D: lowering the dumbbell only halfway
* E: throwing the hips to the front

We will use a train set and a cross-validation set to train two models and choose the best one to predict our test set.

Loading the Data

First we load the date from the HAR project. The Training Data is avaiable here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The Testing Data is avaiable here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```
downcsv <- function(url, nas) {
    temp <- tempfile()
    download.file(url, temp, method = "curl")
    data <- read.csv(temp, na.strings = nas)
    unlink(temp)
    return(data)
}

trainurl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
train <- downcsv(trainurl, c("", "NA", "#DIV/0!"))

testurl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
test <- downcsv(testurl, c("", "NA", "#DIV/0!"))
```

Now we analize the number of observations and features, and the distribution of the measured stances A,B,C,D,E:

```
dim(train)
```

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

```
table(train$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Preprocessing

we start by loading the libraries that we will use in this analisys

```
library(caret)
```

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

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

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

```
## randomForest 4.6-12
```

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

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

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

Then we continue by splitting our training data into a training set and a validation set so that we can validate our model. For reproducibility, we will set a seed value. We will remove columns with all missing values and some variables are irrelevant to our current project.

```
set.seed(115687)

train<-train[,colSums(is.na(train)) == 0]
test <-test[,colSums(is.na(test)) == 0]

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

train_set <- createDataPartition(train$classe, p = 0.75, list = FALSE)
trainingset <- train[train_set, ]
validationset <- train[-train_set, ]
```

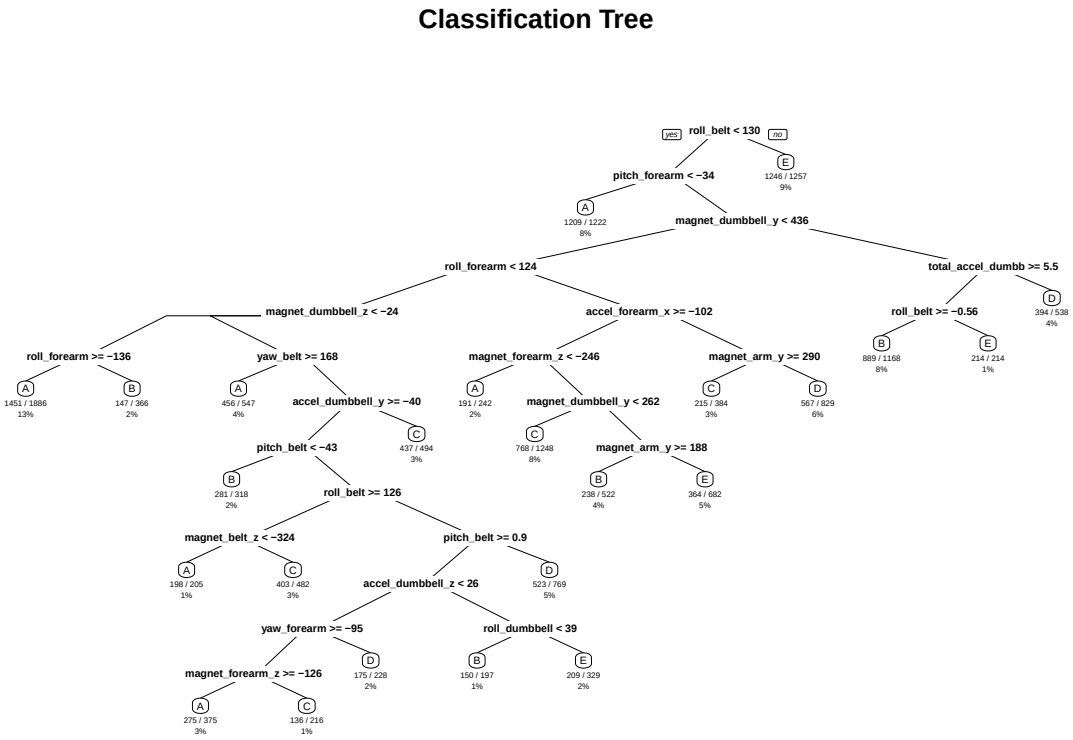
Model Training

Training: Decision Tree Model

We will train a Descision Tree Model using the rpart library and then plot it using rpart.plot

```
model1 <- rpart(classe ~ ., data=trainingset, method="class")
prediction1 <- predict(model1, validationset, type = "class")

rpart.plot(model1, main="Classification Tree", extra=102, under=TRUE, faclen=0)
```



Now we test our model using by calculating it's confusion matrix.

```
confusionMatrix(prediction1, validationset$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1248  148   21   38    9
##          B   44  564   95   85   87
##          C   34  129  642   75   83
##          D   58   56   68  539   70
##          E   11   52   29   67  652
##
```

```
## Overall Statistics
##
##                Accuracy : 0.7433
##                  95% CI : (0.7308, 0.7555)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6748
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8946   0.5943   0.7509   0.6704   0.7236
## Specificity            0.9384   0.9214   0.9207   0.9385   0.9603
## Pos Pred Value         0.8525   0.6446   0.6667   0.6814   0.8039
## Neg Pred Value         0.9573   0.9044   0.9460   0.9356   0.9392
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2545   0.1150   0.1309   0.1099   0.1330
## Detection Prevalence   0.2985   0.1784   0.1964   0.1613   0.1654
## Balanced Accuracy      0.9165   0.7578   0.8358   0.8045   0.8420
```

Training: Random Forest Model

Now we train a Random Forest Model and calculate it's confusion matrix.

```
model2 <- randomForest(classe ~. , data=trainingset, method="class")
prediction2 <- predict(model2, validationset, type = "class")

confusionMatrix(prediction2, validationset$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    9    0    0    0
##          B    0  940    8    0    0
##          C    0    0  845    9    0
##          D    0    0    2  794   10
##          E    0    0    0    1  891
##
## Overall Statistics
##
##                Accuracy : 0.992
##                  95% CI : (0.9891, 0.9943)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9899
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9905   0.9883   0.9876   0.9889
## Specificity            0.9974   0.9980   0.9978   0.9971   0.9998
## Pos Pred Value         0.9936   0.9916   0.9895   0.9851   0.9989
## Neg Pred Value         1.0000   0.9977   0.9975   0.9976   0.9975
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1917   0.1723   0.1619   0.1817
## Detection Prevalence   0.2863   0.1933   0.1741   0.1644   0.1819
## Balanced Accuracy      0.9987   0.9942   0.9930   0.9923   0.9943
```

Picking a Model

We verify now that the Decision Tree Model presented a 0.7433 accuracy rate on the cross-validation set while the Random Tree Model presented a 0.992 accuracy rate on the cross-validation set. Therefore we will use the Random Forest Model to predict our test set.

Predicting

We now perform our prediction on the test set using the chosen model and conclude that the classes of each of the 20 individuals of the test set are:

```
final_prediction <- predict(model2, test, type="class")
final_prediction
```

```
##  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
```