Practical machine learning course project

Joaquim Villen Benseny

19 August 2018

Introduction

The following paper aims to predict a classification of excercises based on the utilisation of different accelerometers for different users.

Data exploration

Data reading

```
training <- read.csv('pml-training.csv', na.strings = c("NA", "#DIV/0!", ""))
crossval <- read.csv('pml-testing.csv', na.strings = c("NA", "#DIV/0!", ""))
```

Data cleaning

```
library(caret)
```

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

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

```
#remove entries with null values
training<- training[, colSums(is.na(training)) == 0]
crossval <- crossval[, colSums(is.na(crossval)) == 0]

#keep output
classe <- training$classe

#remove non relevant variables
training<- training[, !(grepl("^X|timestamp|window", names(training)))]
#remove non numeric variables
training<- training[, sapply(training, is.numeric)]
training$classe<- classe

#splitting data in train test split
inTrain <- createDataPartition(training$classe, p=0.60, list=F)
training <- training[inTrain, ]
testing <- training[-inTrain, ]

#keep vector of predictors
predictors <- colnames(testing)
predictors <- predictors[!(predictors %in% c("classe"))]
```

Data modeling

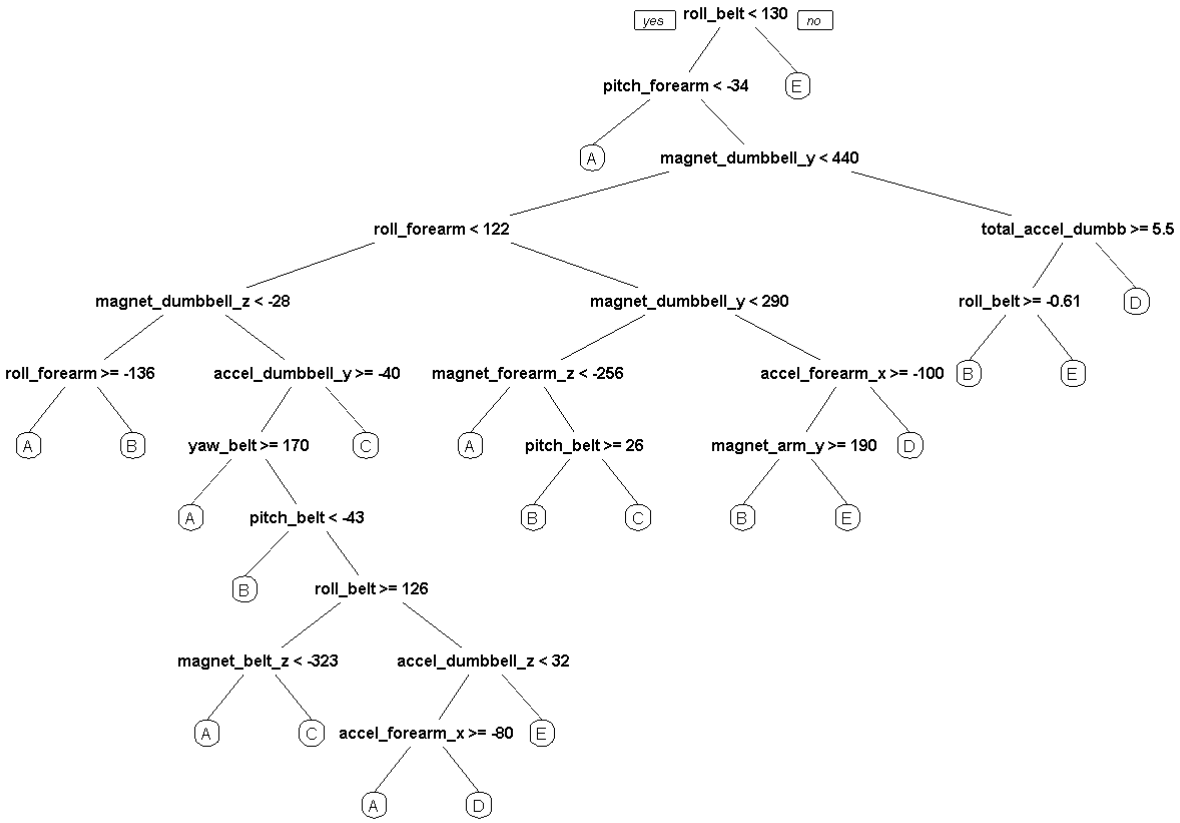
A couple of different machine learning algorithms will be trained below (random forest and gradient boosting).

```
library(rpart)
mdl1 <- train(classe~.,data = training, method = 'rf', ntree = 10)
mdl1
```

```
## Random Forest
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 11776, 11776, 11776, 11776, 11776, 11776, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9655410  0.9564075
##   27    0.9771770  0.9711330
##   52    0.9693404  0.9612184
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

Showing the tree:

```
library(rpart)
library(rpart.plot)
rtree<- rpart(classe ~ ., data=training, method="class")
prp(rtree)
```



Results and conclusions

Test error

```
pred <- predict(mdl1, testing)
confusionMatrix(pred,testing$classe)$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9997874      0.9997313      0.9988159      0.9999946      0.2806719
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

```
error <- 1 - as.numeric(confusionMatrix(testing$classe, pred)$overall[1])
```

The results predicted for the test dataset are provided below:

```
predict(mdl1,crossval[,predictors])
```

```
##  [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
```

Therefore it is possible to conclude that the out of sample error of the random forest model trained above is around 2.126302410^{-4}. The model accuracy is around 99.978737