Peer graded assingment Practical machine learning

MvL

16 oktober 2017

In this report data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants will be used to predict in a test set of the (in)correctness of the movements. The participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

There are five classes indicating the performance of the excercise, Class A corresponds to the specified execution of the exercise, the others indicate common mistakes. The exercises were performed by six male participants aged between 20-28 years, with little weight lifting experience. The data for this project is kindly provided from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

Packages needed

```
#Load the packages needed to run this report.
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.2
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.4.2
```

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

```
## Warning: package 'rpart.plot' was built under R version 3.4.2
```

Data Exploration

The raw data is downloaded and processed for making a prediction on performance of weight lifting exercise on 6 participants.

```
# Download the dataset
trainUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

training <- read.csv(url(trainUrl), na.strings = c("NA", "#DIV/0!", ""))
testing <- read.csv(url(testUrl), na.strings = c("NA", "#DIV/0!", ""))
```

There is a lot of missing data in the datasets, in the next steps the columns with missing data and the first columns with data that can not be included to predict on, is filtered out of the datasets. A validation set and a training set are created from the training data to train and validate the models.

```
#delete missing variables with missing values in testset
training <- training[, colSums(is.na(training)) == 0]
testing <- testing[, colSums(is.na(testing)) == 0]
```

```r
#remove non informational columns 1-7
training_sel<- training[, -c(1:7)]
testing_sel<- testing[, -c(1:7)]
```

```r
#Split training set in two diferent sets
inTrain <- createDataPartition(y=training_sel$classe,p=0.7, list=FALSE)
training_data <- training_sel[inTrain,]
validation_data <- training_sel[-inTrain,]
dim(training_data); dim(validation_data)
```

```
## [1] 13737    53
```

```
## [1] 5885   53
```

Prediction models validation

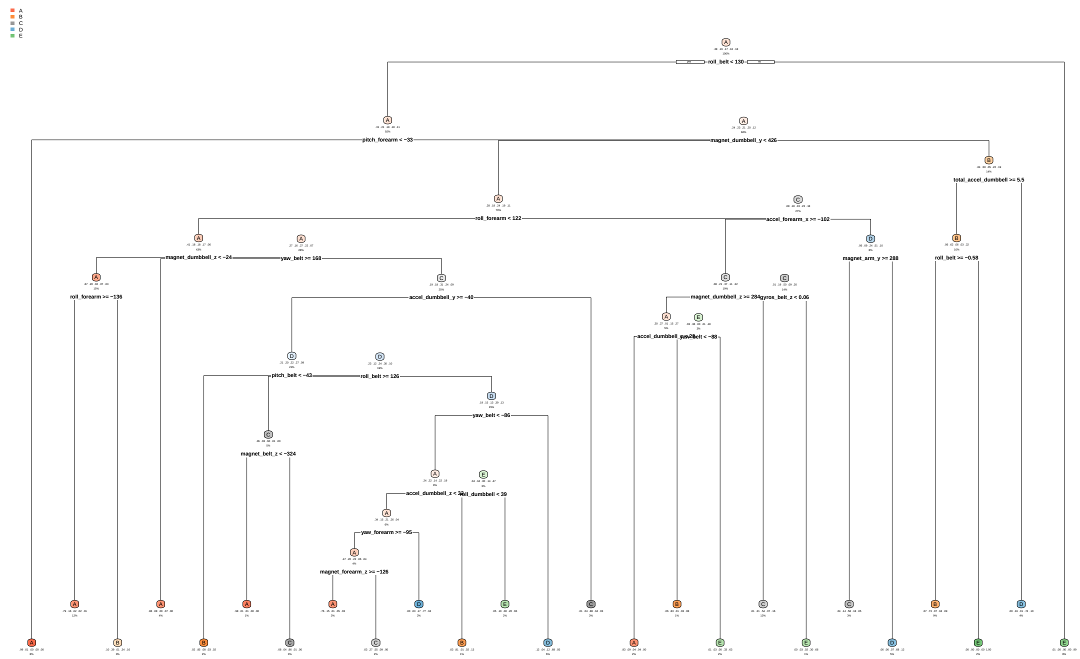
Two different models will be tested to predict class in the weight lifting experiment. First a decision tree model is trained and validated, on the other hand a random forest model will be executed.

```r
#Make a decision tree model
set.seed(2235)
modFitDT <- rpart(classe ~ ., data = training_data, method = "class")


#Predict on validation with decision tree
DT_pred <- predict(modFitDT, validation_data, type = "class")
#Show the decision tree and performance of the decision tree model
rpart.plot(modFitDT, main = "Decision Tree", under = T, faclen = 0)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

Decision Tree



```r
confusionMatrix(DT_pred, validation_data$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1504  164   29   34   14
##          B   59  643   54   71   95
##          C   42  223  861  111  126
##          D   57   84   81  667   87
##          E   12   25    1   81  760
##
## Overall Statistics
##
##                Accuracy : 0.7536
##                  95% CI : (0.7424, 0.7646)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6882
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8984   0.5645   0.8392   0.6919   0.7024
```

```
## Specificity            0.9428   0.9412   0.8967   0.9372   0.9752
## Pos Pred Value         0.8619   0.6974   0.6317   0.6834   0.8646
## Neg Pred Value         0.9589   0.9001   0.9635   0.9395   0.9357
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2556   0.1093   0.1463   0.1133   0.1291
## Detection Prevalence   0.2965   0.1567   0.2316   0.1658   0.1494
## Balanced Accuracy      0.9206   0.7529   0.8679   0.8146   0.8388
```

```
#Make a random forest model with 10-fold crossvalidation and 10 trees
set.seed(2236)
fitcontrol <- trainControl(method="cv",number=10,allowParallel = TRUE)
modFitRF <- train(classe ~ .,data=training_data,method="rf",trControl=fitcontrol, ntree=10)
```

```
## Warning: package 'randomForest' was built under R version 3.4.2
```

```
## 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
```

```
#Predict and show performance of random forest model
RF_pred <- predict(modFitRF, validation_data)
confusionMatrix(RF_pred, validation_data$classe)
```

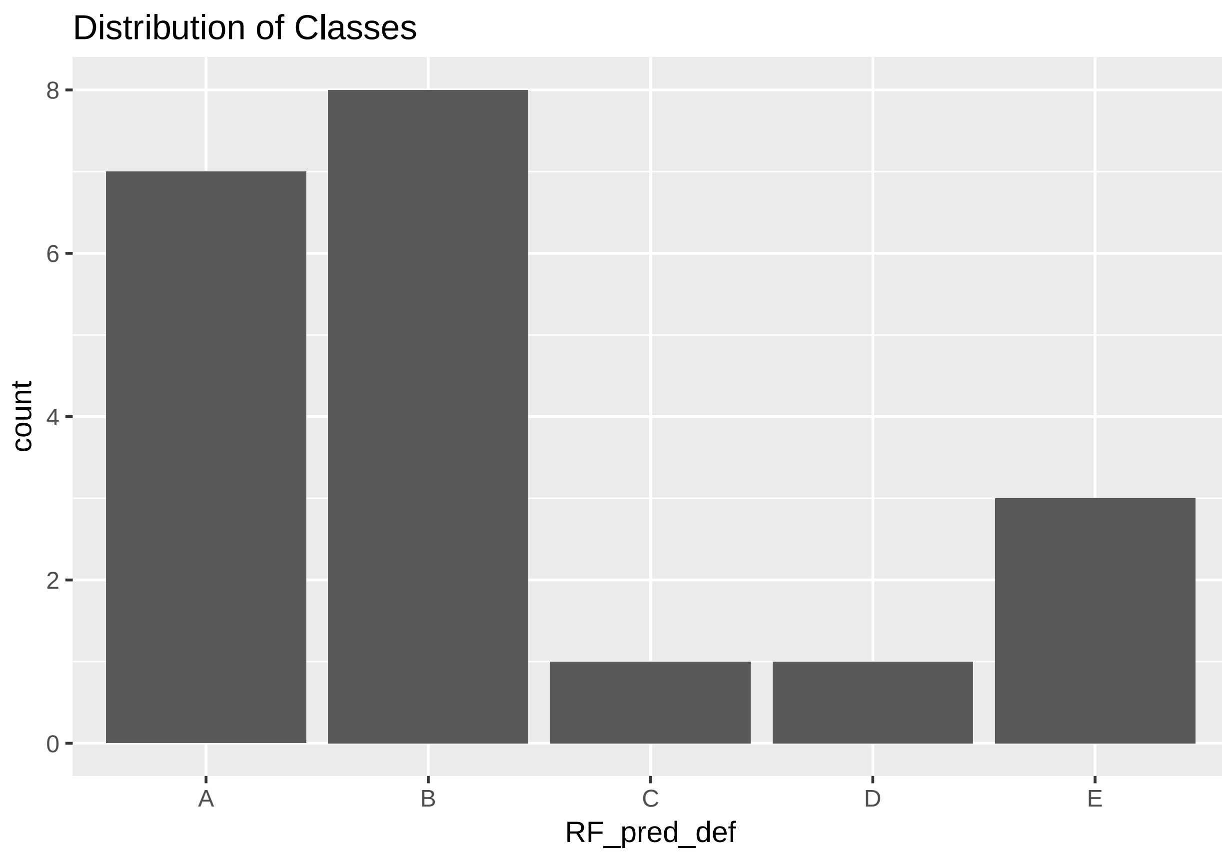
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1662    8    0    0    0
##          B    9 1118    6    2    3
##          C    0   11 1017   18    1
##          D    3    2    2  944   10
##          E    0    0    1    0 1068
##
## Overall Statistics
##
##                Accuracy : 0.9871
##                  95% CI : (0.9839, 0.9898)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9837
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9928   0.9816   0.9912   0.9793   0.9871
## Specificity            0.9981   0.9958   0.9938   0.9965   0.9998
## Pos Pred Value         0.9952   0.9824   0.9713   0.9823   0.9991
## Neg Pred Value         0.9972   0.9956   0.9981   0.9959   0.9971
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
```

```
## Detection Rate         0.2824   0.1900   0.1728   0.1604   0.1815
## Detection Prevalence   0.2838   0.1934   0.1779   0.1633   0.1816
## Balanced Accuracy      0.9955   0.9887   0.9925   0.9879   0.9934
```

Model selection

The random forest model performance has a higher accuracy than the decision tree model, i.e. 0.99 vs 0.74, therefore the random forest model is selected as final model to predict the correct execution of a weight lifting exercise. A large part of the dataset was not usefull to predict performance and not included in the prediction. However even excluding this data gives a near perfect prediction on the validation set.

```r
#Prediction on testing dataset
RF_pred_def <- predict(modFitRF, testing)
test_pred <-cbind(RF_pred_def, testing)
qplot(RF_pred_def, data=testing, main="Distribution of Classes")
```



##Conclusion The prediction shows that the predicted classes of the 20 test cases.

Reference

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.