Practical Machine Learning Project

Felipe Llaugel

August 22, 2015

Background

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 dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Objective of the project

The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Data

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har).

Model construction

The prediction model was built using the submitted training data. After some preprocessing, a decision tree was adjusted to the data. The training set was applied to the model using the trained decision tree. The Random Forest algorithnm was also applied to the data, getting improvement in the classification

matrix.

Cross-Validation

Cross-validation was performed by subsampling the training data set randomly without replacement into 2 subsamples: subTraining data (75% of the original Training data set) and subTesting data (25%). Our models will be fitted on the subTraining data set, and tested on the subTesting data. Once the most accurate model is choosen, it will be tested on the original Testing data set.

Expected out-of-sample error

The expected out-of-sample error will correspond to the quantity: 1-accuracy in the cross-validation data. Accuracy is the proportion of correct classified observation over the total sample in the subTesting data set. Expected accuracy is the expected accuracy in the out-of-sample data set (i.e. original testing data set). Thus, the expected value of the out-of-sample error will correspond to the expected number of missclassified observations/total observations in the Test data set, which is the quantity: 1-accuracy found from the cross-validation data set.

Loading and preprocessing the data

```
# loading necesary libraries
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(rpart)
library(rpart.plot)
set.seed(5628)

# Loading the training and testing data set  replacing missing values with "NA"

trainingset <- read.csv("C:/misdatos/md4/DATA SCIENCE/PRACTICAL MACHINE LEARNING/Projec
t/pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testingset <- read.csv("C:/misdatos/md4/DATA SCIENCE/PRACTICAL MACHINE LEARNING/Project/p
ml-testing.csv", na.strings=c("NA","#DIV/0!", ""))

# Check dimensions for number of variables and number of observations

dim(trainingset)
```

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

```
dim(testingset)
```

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

```
# Delete columns with missing values

trainingset<-trainingset[,colSums(is.na(trainingset)) == 0]
testingset <-testingset[,colSums(is.na(testingset)) == 0]

# Deleting variables irrelevant variables to the project: user_name, raw_timestamp_par
t_1, raw_timestamp_part_,2 cvtd_timestamp, new_window, and  num_window (columns 1 to 7).
We can delete these variables.

trainingset   <-trainingset[,-c(1:7)]
testingset <-testingset[,-c(1:7)]

# New datasets:

dim(trainingset)
```

```
## [1] 19622    53
```

```
dim(testingset)
```

```
## [1] 20 53
```

```
# Partitioning de data sets

subsamples <- createDataPartition(y=trainingset$classe, p=0.75, list=FALSE)
subTraining <- trainingset[subsamples, ]
subTesting <- trainingset[-subsamples, ]
dim(subTraining)
```

```
## [1] 14718    53
```

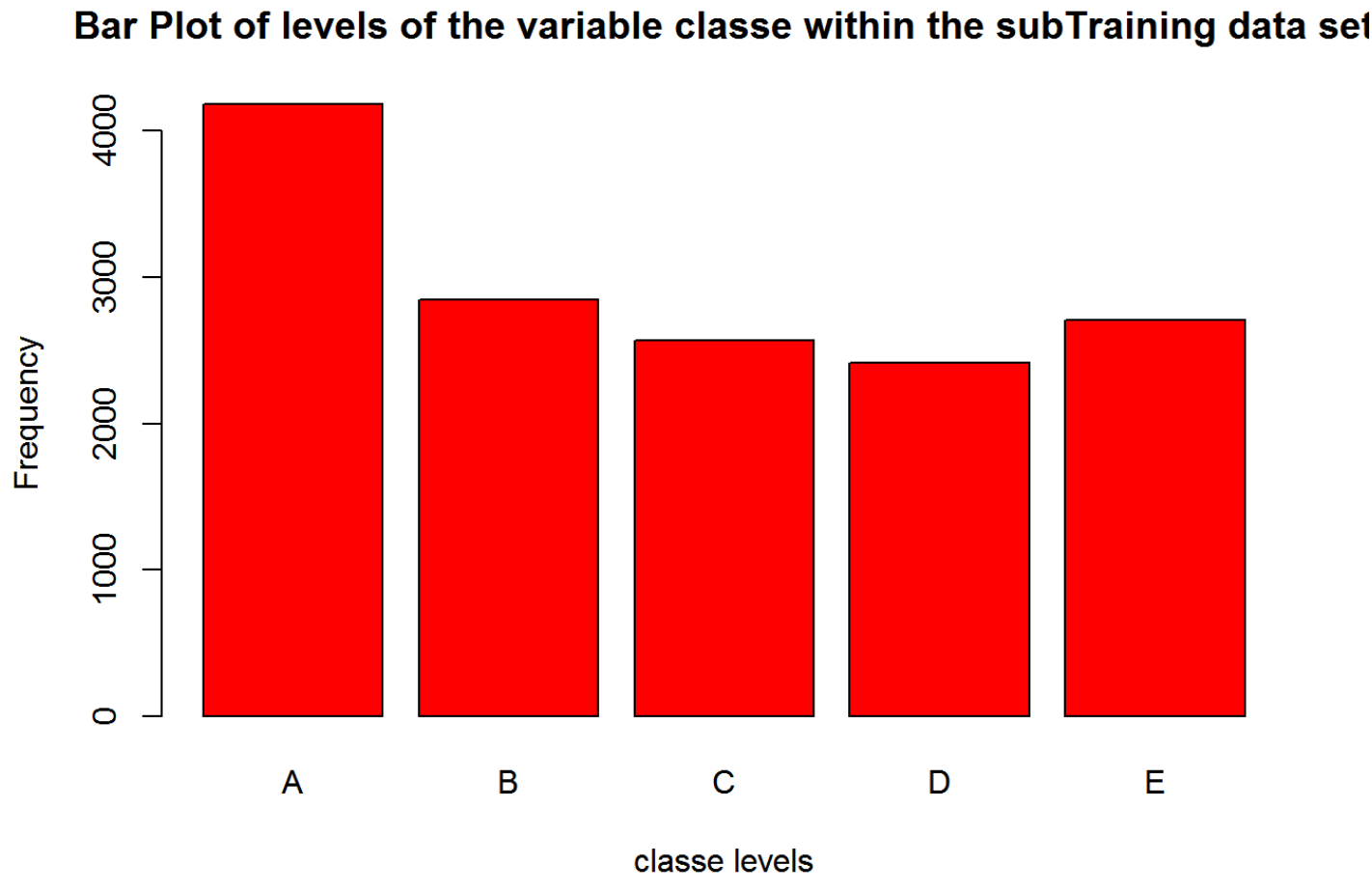
```
dim(subTesting)
```

```
## [1] 4904   53
```

Model training and prediction using decision tree

```
# Bar plot of variable Classe

plot(subTraining$classe, col="red", main="Bar Plot of levels of the variable classe withi
n the subTraining data set", xlab="classe levels", ylab="Frequency")
```

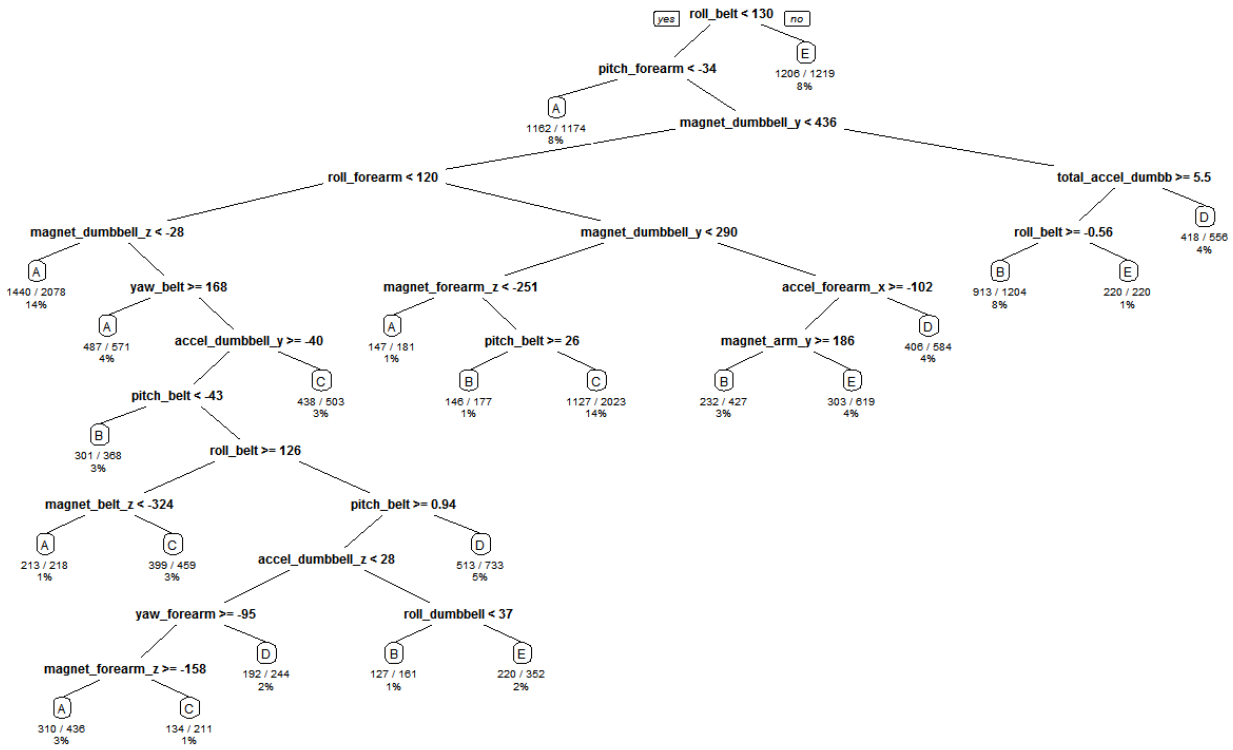


```
model1 <- rpart(classe ~ ., data=subTraining, method="class")

# Model prediction using decision tree
# Predicting:
prediction1 <- predict(model1, subTesting, type = "class")

# Plot of the Decision Tree
rpart.plot(model1, main="Decision Tree", extra=102, under=TRUE, faclen=0)
```

Decision Tree




```
# Test results on subTesting data set:
confusionMatrix(prediction1, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1247  195   13   88   28
##          B   40  555   80   27   55
##          C   40   91  681  123  117
##          D   47   73   53  511   45
##          E   21   35   28   55  656
##
## Overall Statistics
##
##                Accuracy : 0.7443
##                  95% CI : (0.7318, 0.7565)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6753
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8939   0.5848   0.7965   0.6356   0.7281
## Specificity            0.9077   0.9489   0.9084   0.9468   0.9653
## Pos Pred Value         0.7938   0.7332   0.6473   0.7010   0.8252
## Neg Pred Value         0.9556   0.9050   0.9548   0.9298   0.9404
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2543   0.1132   0.1389   0.1042   0.1338
## Detection Prevalence   0.3204   0.1544   0.2145   0.1487   0.1621
## Balanced Accuracy      0.9008   0.7669   0.8524   0.7912   0.8467
```

Model training and prediction using random forest

```
#Prediction Model Using Random Forest
model2 <- randomForest(classe ~. , data=subTraining, method="class")

# Predicting:
prediction2 <- predict(model2, subTesting, type = "class")

# Test results on subTesting data set:
confusionMatrix(prediction2, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    2    0    0    0
##          B    0  947    1    0    0
##          C    0    0  853    5    0
##          D    0    0    1  799    1
##          E    0    0    0    0  900
##
## Overall Statistics
##
##                Accuracy : 0.998
##                  95% CI : (0.9963, 0.999)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9974
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9979   0.9977   0.9938   0.9989
## Specificity            0.9994   0.9997   0.9988   0.9995   1.0000
## Pos Pred Value         0.9986   0.9989   0.9942   0.9975   1.0000
## Neg Pred Value         1.0000   0.9995   0.9995   0.9988   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1931   0.1739   0.1629   0.1835
## Detection Prevalence   0.2849   0.1933   0.1750   0.1633   0.1835
## Balanced Accuracy      0.9997   0.9988   0.9982   0.9966   0.9994
```

```
# predict outcome levels on the original Testing data set using Random Forest algorithm

predictfinal <- predict(model2, testingset, type="class")
predictfinal
```

```
##  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
```

Writing prediction file

```
# Write files for assingment submission

pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictfinal)
```

REFERENCES

1- 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.

2- Krzysztof Grabczewski and Norbert Jankowski. Feature Selection with Decision Tree Criterion.