Practical Machine Learning - Project

Arun Panangatt

20 June 2015

Project Summary

The project is with reference to Human Activity Recognition(HAR). The project details how participants were asked to do barbell lifts 5 different ways-

1.Sitting 2.Sitting-down 3.Standing 4.Standing-up 5.Walking

The resultant data was collected using devices like Jawbone Up, Nike FuelBand, and Fitbit worn by the paricipants. It is acknowledged that the classification of the quality of the exercises done by the participants may or may not have been correct.

The obejctive is to build a machine learning algorithm to predict activity quality from activity sensor monitors and train a model based on the data received through various sensor values, which could later be used to predict the Classe variable,i.e, the manner in which the participants of HAR performed the exercises.

Setting up the Environment

```
library(caret)
```

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

```
library(doParallel)
```

```
## Loading required package: foreach
## Loading required package: iterators
## Loading required package: parallel
```

```
library(randomForest)
```

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

```
set.seed(20150135)
knitr::opts_chunk$set(echo=TRUE, fig.width=12, fig.height=12)
```

Downloading data

Data was downloaded from the following locations to the working directory

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

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

Cleaning Data

The pml-training.csv data is used to devise training and testing sets during fitting of the model. The pml-test.csv data is used to submit 20 test cases based on the fitted model.

All blank "", #DIV/0 and NA values are converted to NA.

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

It is prudent to remove columns with too many "NA" values.

```
tidydata   <- which((colSums(!is.na(cleantrain)) >= 0.6*nrow(cleantrain)))
cleantrain <- cleantrain[,tidydata]
cleantest    <- cleantest[,tidydata]
```

A few minor corrections to test set are needed to perform optimally with random forests.This deals with removal of problematic ids and correcting factor levels.

```
cleantest <- cleantest[-ncol(cleantest)]
cleantest$new_window <- factor(cleantest$new_window, levels=c("no","yes"))
```

The X and cvtd_timestamp columns are removed from the dataset as they are not relevant.

```
cleantrain <- cleantrain[,-c(1,5)]
cleantest    <- cleantest[,-c(1,5)]
```

Division of data into Training and Test data sets

Data is divided as follows - Training :60% and Test :40%

```
Traindata  <- createDataPartition(cleantrain$classe, p = 0.6, list = FALSE)
training    <- cleantrain[Traindata, ]
testing     <- cleantrain[-Traindata, ]
```

Fitting Random Forests

The output variable is named class and other columns are in the dataframe named data

```
class <- training$classe
data  <- training[-ncol(training)]
```

The Parallel Random Forest algorithm will be used to fit the model.A 5 fold cross validation will be applied in the algorithm.

```
registerDoParallel()
rf <- train(data, class, method="parRF",
    tuneGrid=data.frame(mtry=3),
    trControl=trainControl(method="cv",5))
rf
```

```
## Parallel Random Forest
##
## 11776 samples
##    57 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 9421, 9420, 9421, 9421, 9421
##
## Resampling results
##
##   Accuracy  Kappa      Accuracy SD  Kappa SD
##   0.995669  0.9945214  0.002131851  0.002696925
##
## Tuning parameter 'mtry' was held constant at a value of 3
##
```

A plot depicting the relative importance of the model variables is given in the Appendix (Diagram 1)

Confusion Matrix for the Tesing set

The next step is to predict on the Tesing set and generate the confusion matrix for the Testing set.

```
testingPredictions <- predict(rf, newdata=testing)
confMatrix <- confusionMatrix(testingPredictions,testing$classe)
confMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    8    0    0    0
##          B    0 1508    1    0    0
##          C    0    2 1365    4    0
##          D    0    0    2 1282    6
##          E    0    0    0    0 1436
##
## Overall Statistics
##
##                Accuracy : 0.9971
##                  95% CI : (0.9956, 0.9981)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9963
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9934   0.9978   0.9969   0.9958
## Specificity            0.9986   0.9998   0.9991   0.9988   1.0000
## Pos Pred Value         0.9964   0.9993   0.9956   0.9938   1.0000
## Neg Pred Value         1.0000   0.9984   0.9995   0.9994   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1922   0.1740   0.1634   0.1830
## Detection Prevalence   0.2855   0.1923   0.1747   0.1644   0.1830
## Balanced Accuracy      0.9993   0.9966   0.9984   0.9978   0.9979
```

Conclusion

Verification of Accuracy

```
confMatrix$overall[1]
```

```
##  Accuracy
## 0.9970686
```

Checking out of sample error

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

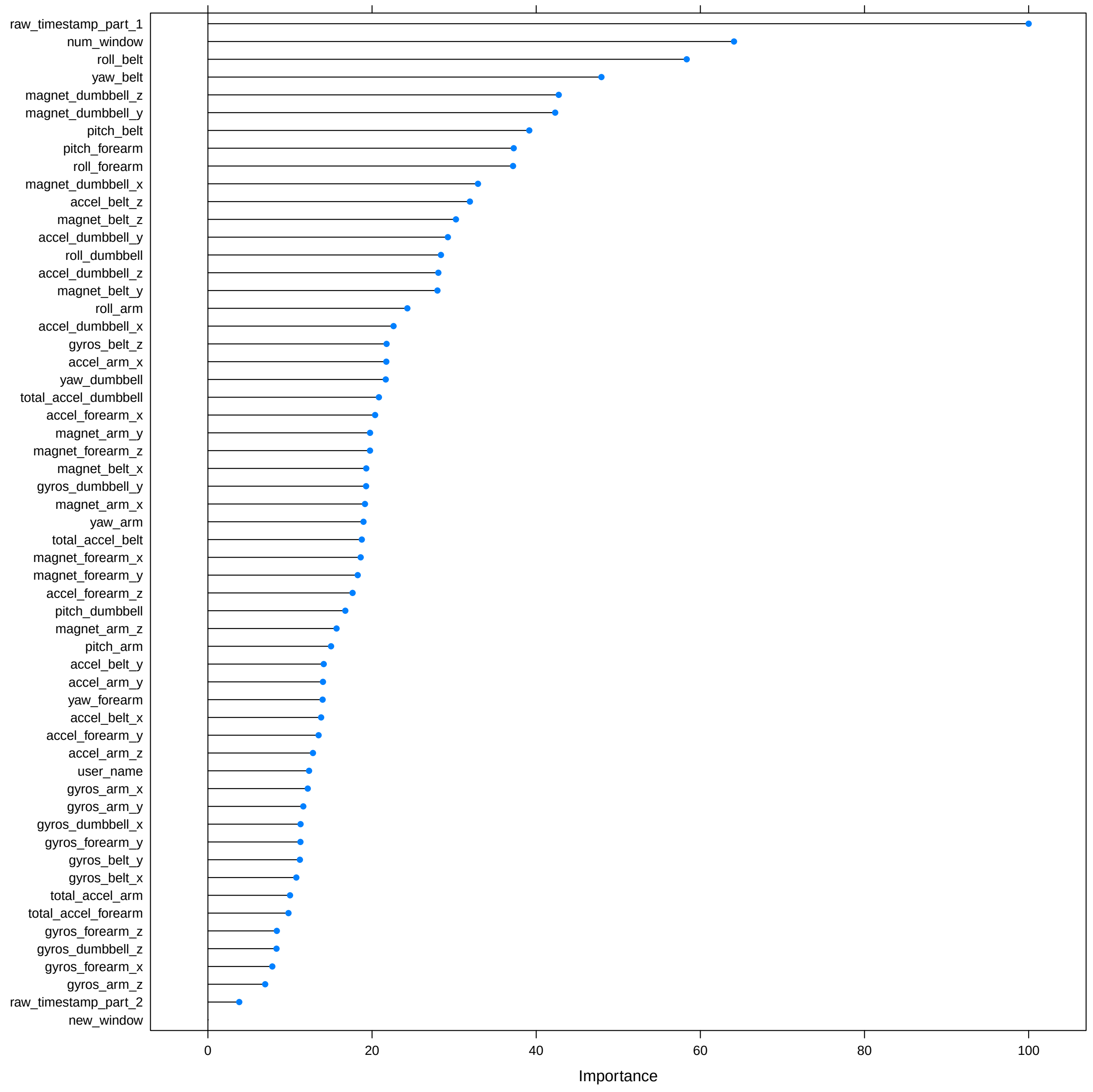
```
## [1] 0.00293143
```

It can be seen that the model predicts with an accuracy of 99.6% with a negligible out of sample error.

APPENDIX

DIAGRAM 1

```
plot(varImp(rf))
```


raw_timestamp_part_1
num_window
roll_belt
yaw_belt
magnet_dumbbell_z
magnet_dumbbell_y
pitch_belt
pitch_forearm
roll_forearm
magnet_dumbbell_x
accel_belt_z
magnet_belt_z
accel_dumbbell_y
roll_dumbbell
accel_dumbbell_z
magnet_belt_y
roll_arm
accel_dumbbell_x
gyros_belt_z
accel_arm_x
yaw_dumbbell
total_accel_dumbbell
accel_forearm_x
magnet_arm_y
magnet_forearm_z
magnet_belt_x
gyros_dumbbell_y
magnet_arm_x
yaw_arm
total_accel_belt
magnet_forearm_x
magnet_forearm_y
accel_forearm_z
pitch_dumbbell
magnet_arm_z
pitch_arm
accel_belt_y
accel_arm_y
yaw_forearm
accel_belt_x
accel_forearm_y
accel_arm_z
user_name
gyros_arm_x
gyros_arm_y
gyros_dumbbell_x
gyros_forearm_y
gyros_belt_y
gyros_belt_x
total_accel_arm
total_accel_forearm
gyros_forearm_z
gyros_dumbbell_z
gyros_forearm_x
gyros_arm_z
raw_timestamp_part_2
new_window
0
20
40
60
80
100
Importance