Practical Machine Language Project

Dom Fernandez

Saturday, June 21, 2014

Executive Summary

- Data for this analysis was downloaded from “Human Activity Recognition (HAR)” website. http://groupware.les.inf.puc-rio.br/har#wle_paper_section (http://groupware.les.inf.puc-rio.br/har#wle_paper_section)
- Test-data was retrieved from: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)
- Training data from: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)
- With this data I have tried to build model(s) to predict the type of `movement` of an individual.
- Accelerometers were worn by the `Test-group` to record their movements.

Environment Setup

```
library(datasets)
library(caret)
```

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

```
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-7
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(knitr)
set.seed(32343)
```

Data Preparation

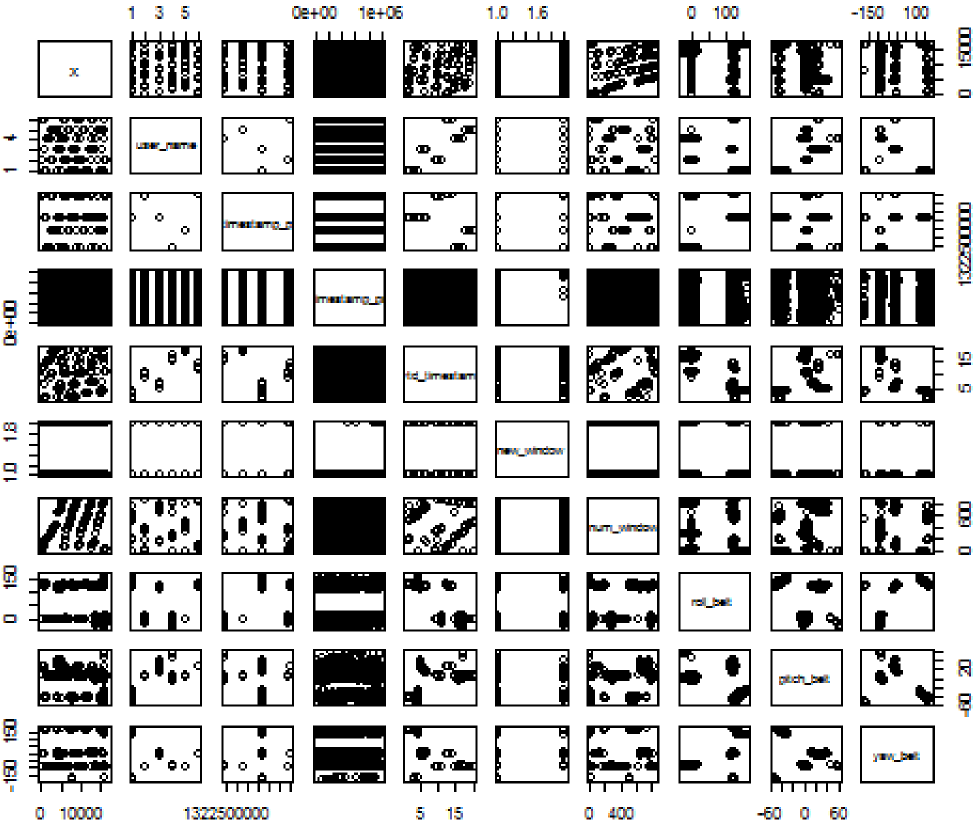
```
rm(list=ls())
loadCSV = read.csv("pml-training.csv")
inTrain <- createDataPartition(loadCSV$classe, p=0.60, list=FALSE)
training <- trainingCSV[inTrain, ]
validation <- trainingCSV[-inTrain, ]
```

Summary of data, viewing first-few records

```
summary(training)
head(training)
```

- Trying to create a plot with the full dataset created and error
- Hence the dataset was subsetted: [1:10000,1:10], [1:10000,11:20], [1:10000,21:30] (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

```
plot(training)
## Error: figure margins too large
pairs(training[1:10000,1:10])
```



Since most of the columns have no data, or predictive power, it might not be conducive to use them as-is. Therefore filtering out fields with a lot of (more than 60%) null values.

```
goodVar<-c((colSums(is.na(training[,-160])) >= 0.4*nrow(training)),160)
training<-training[,goodVar]
dim(training)
```

```
## [1] 11776    68
```

```
validation<-validation[,goodVar]
dim(validation)
```

```
## [1] 7846   68
```

```
testing<-testing[,goodVar]

training<-training[complete.cases(training),]
dim(training)
```

```
## [1] 11776    68
```

Training the model (RandomForest) on the training data set.

```
model <- randomForest(classe~.,data=training)
print(model)
```

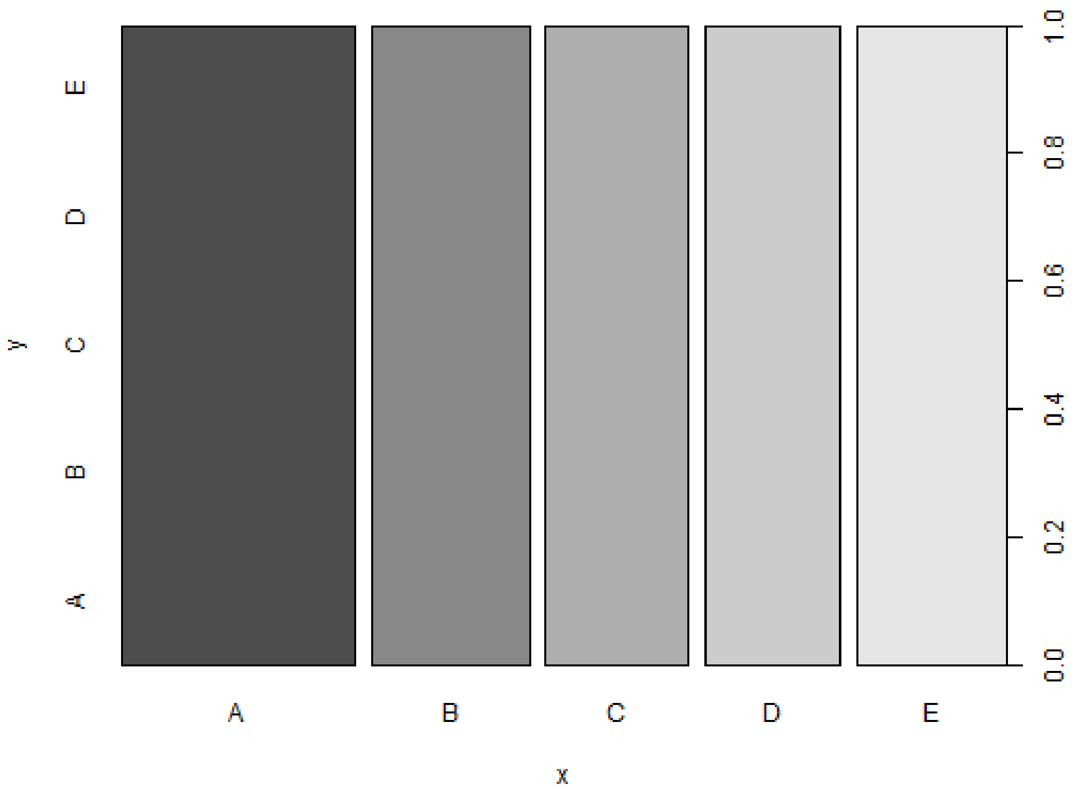
```
##
## Call:
##  randomForest(formula = classe ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 0.01%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3348    0    0    0    0   0.0000000
## B    1 2278    0    0    0   0.0004388
## C    0    0 2054    0    0   0.0000000
## D    0    0    0 1930    0   0.0000000
## E    0    0    0    0 2165   0.0000000
```

```
head(importance(model))
```

```
##     MeanDecreaseGini
## X              130.2
## X.1            158.0
## X.2            145.2
## X.3            163.0
## X.4            156.5
## X.5            141.8
```

Evaluating the model on the evaluation dataset.

```
plot(predict(model,newdata=validation[,-ncol(validation)]),validation$classe)
```



```
confusionMatrix(predict(model,newdata=validation[,-ncol(validation)]),validation$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    1    0    0    0
##          B    0 1517    1    0    0
##          C    0    0 1367    1    0
##          D    0    0    0 1285    1
##          E    0    0    0    0 1441
## 
## Overall Statistics
## 
##                Accuracy : 0.999
##                  95% CI : (0.999, 1)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
## 
##                   Kappa : 0.999
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    0.999    0.999    0.999    0.999
## Specificity             1.000    1.000    1.000    1.000    1.000
## Pos Pred Value          1.000    0.999    0.999    0.999    1.000
## Neg Pred Value          1.000    1.000    1.000    1.000    1.000
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.284    0.193    0.174    0.164    0.184
## Detection Prevalence    0.285    0.193    0.174    0.164    0.184
## Balanced Accuracy       1.000    1.000    1.000    1.000    1.000
```

```
accurate<-c(as.numeric(predict(model,newdata=validation[,-ncol(validation)])==validation$cla
sse))
accuracy<-sum(accurate)*100/nrow(validation)
message("Expected out of sample error using cross-validation is = " , format(round(100-accur
acy, 2), nsmall = 2), "%")
```

```
## Expected out of sample error using cross-validation is = 0.05%
```

Predicting the new values in the testing csv provided.

```
testing =  read.csv("pml-testing.csv")
dim(testing)
```

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

```
testing<-testing[,goodVar]
dim(testing)
```

```
## [1] 20 68
```

```
predictions<-predict(model,newdata=testing)
predictions
```

<!-- dynamically load mathjax for compatibility with --self-contained -->