Machine Learning Assignment

Gregory Papaioannou

January 28, 2016

```
library(caret)
```

Quantified Self Movement Data Analysis Report

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

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

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

```
## Warning: package 'rpart.plot' was built under R version 3.2.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.2.3
```

```
## 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
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.2.3
```

```
library(parallel)
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.2.3
```

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

```
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)
```

Parallel computing

```
train <- read.csv('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv')
test <- read.csv('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv')
```

Load the data

```
# dim(trainRaw)
# dim(testRaw)
# summary(train)
# summary(test)
# sum(complete.cases(train))
```

Get a first glimpse of the data

```
# Drop the empty columns
train <- train[, colSums(is.na(train)) == 0]
test <- test[, colSums(is.na(test)) == 0]

# Drop the columns that that do not contribute much to the accelerometer measurements
train <- train[, !grepl("^X|timestamp|window", names(train))]
test <- test[, !grepl("^X|timestamp|window", names(test))]

# Keep seperate the categorical variables
train_user_name <- train$user_name
test_user_name <- test$user_name
classe <- train$classe

# Some more cleaning
train <- train[, sapply(train, is.numeric)]
test <- test[, sapply(test, is.numeric)]

train$user_name <- train_user_name
train$classe <- classe
test$user_name <- test_user_name
test <-subset(test, select=-problem_id)
```

Start cleaning

```r
set.seed(13)
inTrain <- createDataPartition(train$classe, p=0.75, list=F)
trainData <- train[inTrain, ]
testData <- train[-inTrain, ]
```

Cross validation sample

```r
# 5 folds
cv <- trainControl(method="cv", 5, allowParallel = TRUE)
model <- train(classe ~ ., data=trainData, method="rf", trControl=cv, ntree=250)
```

Create model with the Stochastic Gradient Boosting algorithm

```r
prediction <- predict(model, testData)
confusionMatrix(testData$classe, prediction)
```

Performance of the model on the validation data set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    4  945    0    0    0
##          C    0    7  847    1    0
##          D    0    0    9  795    0
##          E    0    0    1    2  898
##
## Overall Statistics
##
##                Accuracy : 0.9951
##                  95% CI : (0.9927, 0.9969)
##     No Information Rate : 0.2853
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9938
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9971   0.9926   0.9883   0.9962   1.0000
## Specificity            1.0000   0.9990   0.9980   0.9978   0.9993
## Pos Pred Value         1.0000   0.9958   0.9906   0.9888   0.9967
## Neg Pred Value         0.9989   0.9982   0.9975   0.9993   1.0000
## Prevalence             0.2853   0.1941   0.1748   0.1627   0.1831
## Detection Rate         0.2845   0.1927   0.1727   0.1621   0.1831
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9986   0.9958   0.9932   0.9970   0.9996
```

```
postResample(prediction, testData$classe)
```

Accuracy

```
## Accuracy    Kappa
## 0.995106 0.993809
```

```
1 - as.numeric(confusionMatrix(testData$classe, prediction)$overall[1])
```

Estimated out-of-sample error

```
## [1] 0.004893964
```

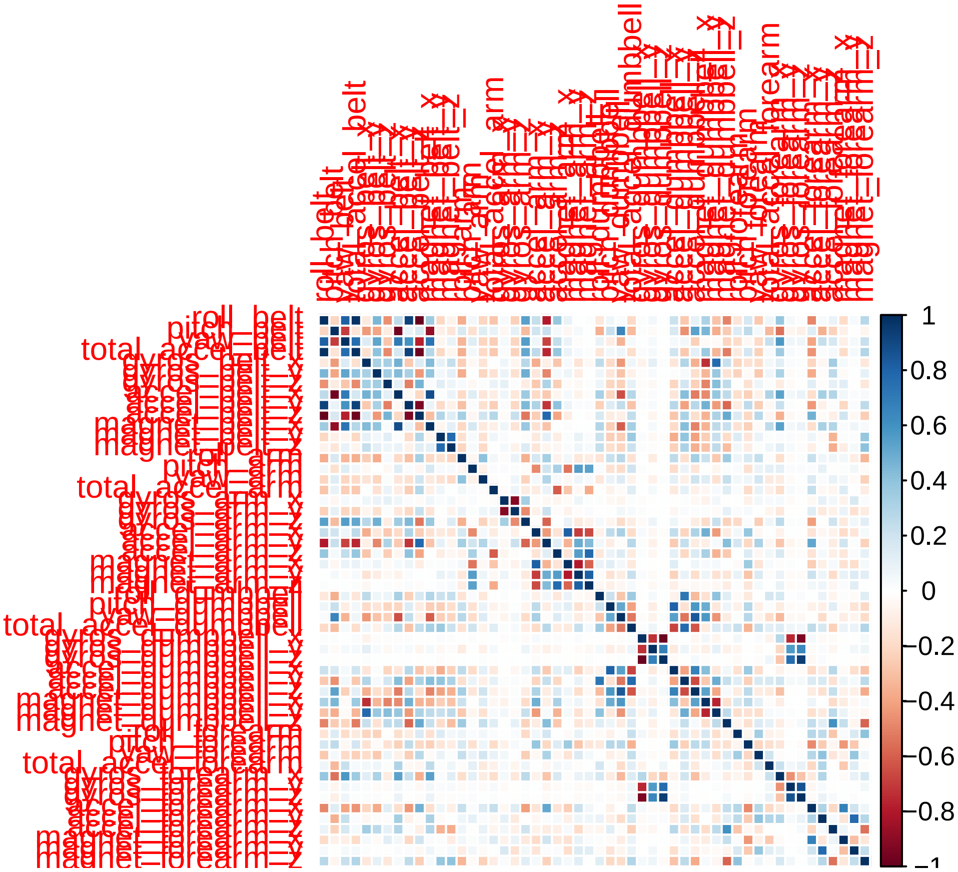
```
predict(model, test)
```

Prediction for Test Data

```
##  [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
```

Figures

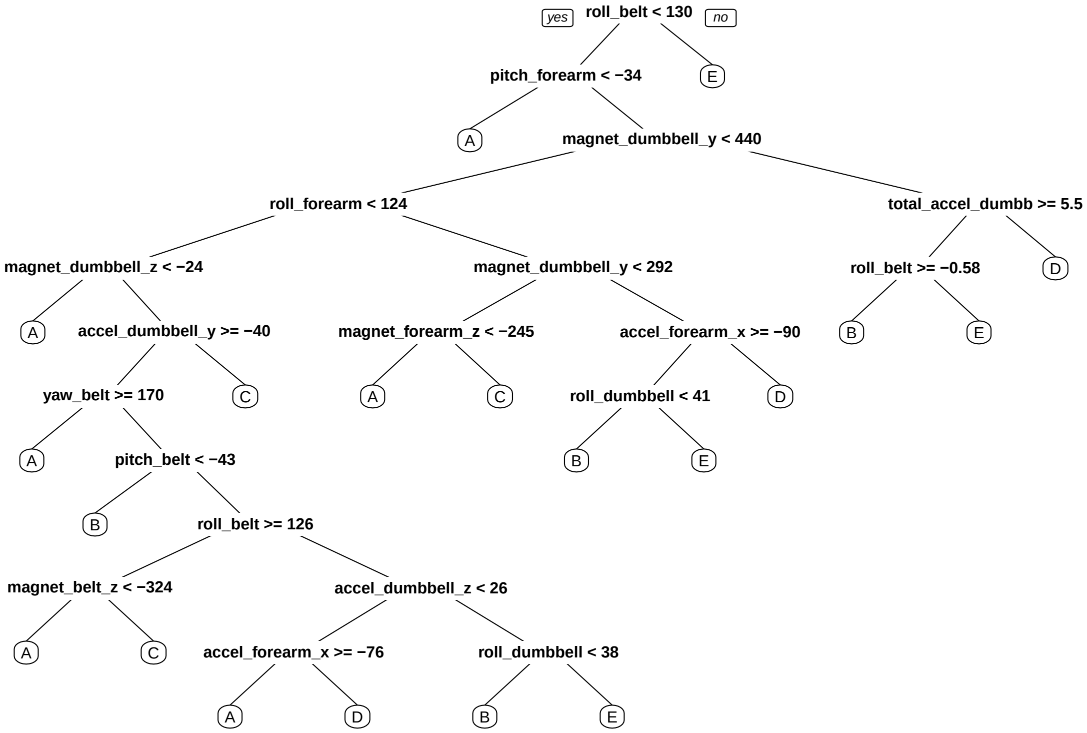
```
correl <- cor(trainData[, sapply(train, is.numeric)])
corrplot(correl, method="color")
```



Correlation Matrix

#### Tree

```
treeModel <- rpart(classe ~ ., data=trainData, method="class")
prp(treeModel)
```


yes
roll_belt < 130
no
pitch_forearm < −34
E
A
magnet_dumbbell_y < 440
roll_forearm < 124
total_accel_dumbb >= 5.5
magnet_dumbbell_z < −24
magnet_dumbbell_y < 292
roll_belt >= −0.58
D
A
accel_dumbbell_y >= −40
magnet_forearm_z < −245
accel_forearm_x >= −90
B
E
yaw_belt >= 170
C
A
C
roll_dumbbell < 41
D
A
pitch_belt < −43
B
E
B
roll_belt >= 126
magnet_belt_z < −324
accel_dumbbell_z < 26
A
C
accel_forearm_x >= −76
roll_dumbbell < 38
A
D
B
E