Human Activity Recognition

Zaher Haydar

July 28, 2017

Executive Summary

Using fitness devices, it is now possible to collect a large amount of data about personal activity relatively inexpensively. 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: 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). The data used in this project is available at: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv).

In this project, we've split the data into training and testing sets (70/30.) We then applied 4 different machine learning algorithms, namely Linear Discriminant Analysis, Gradient Boosting, Random Forest and Decision Tree. RF turns out to be the highest accuracy model for this use case (out of sample error rate of 0.0018692.)

Analysis

The following sections show how this anaysis was performed. As usual, we start by loading necessary libraries required by the code chunks. Then onto loading the data and performing some exploratory data analysis.

```
require(ggplot2)
require(caret)
require(stats)
require(graphics)
require(VGAM)
require(dplyr)
require(gbm)
require(rpart)
require(rattle)
require(rpart.plot)
require(randomForest)
```

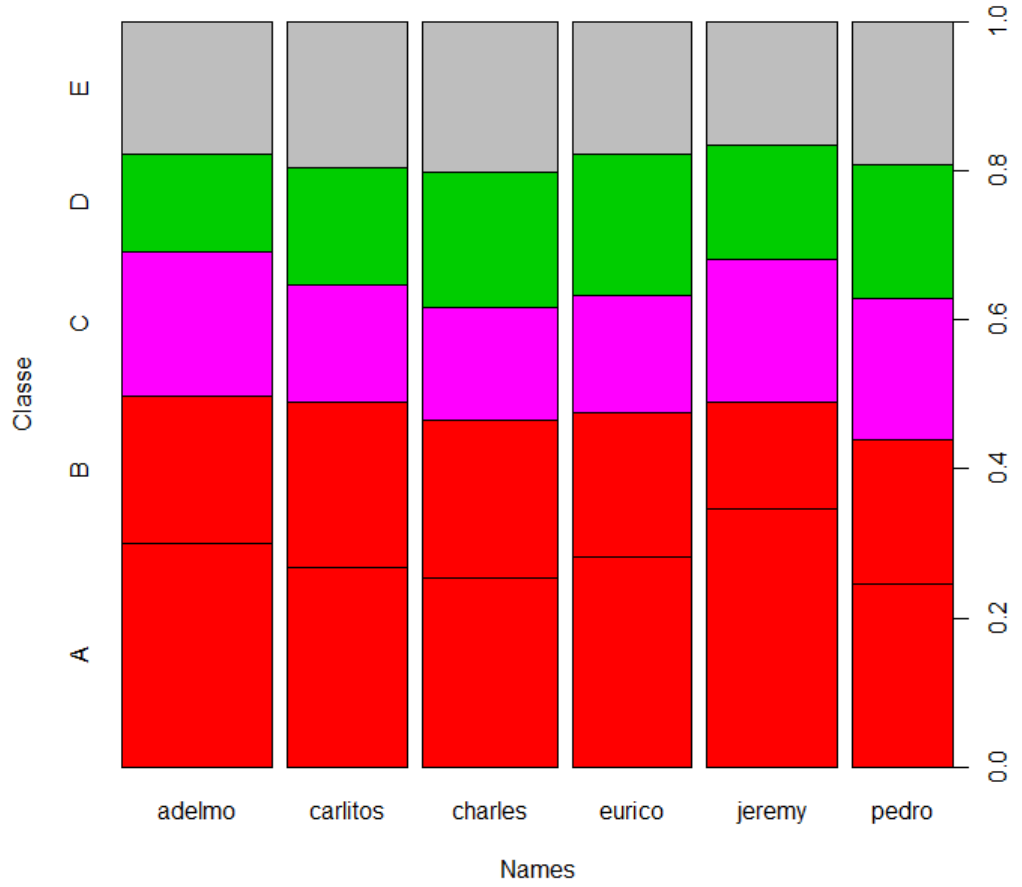
Exploratory Data Analysis

pmlTraining has 19622 obs of 160 vars. pmlTesting has only 20 rows and will be used at the end for the project quiz. We split training into training (70%) and testing (30%.)

To get a feel of the classe (A, B, C, D, E) distribution within the dataset, we plot classe vs user names in the whole pmlTraining dataset. At a high level, it seems there is dependency of classe on time of day: users perform the exercises with A or B quality about 50% of the time and during specific times of day.

```
pmlTraining = read.csv("pml-training.csv")
pmlTesting = read.csv("pml-testing.csv")
inTrain <- createDataPartition(y=pmlTraining$classe, p=0.7, list=FALSE)
training <- pmlTraining[inTrain,]
testing <- pmlTraining[-inTrain,]

plot(pmlTraining$user_name,pmlTraining$classe, col=pmlTraining$raw_timestamp_part_2, xlab="Names", ylab="Classe")
```



training has 13737 obs of 160 vars and testing has 5885 obs of 160 vars, but both have empty and NA cells. We first remove columns with NA cells, we are left with 93 vars (training2.) Then we remove columns with empty cells and we are left with 60 vars (training3.)

```
training2 <- training[ , colSums(is.na(training)) == 0]
training3 <- training2[ , colSums(training2 == "") == 0]
testing2 <- testing[ , colSums(is.na(testing)) == 0]
testing3 <- testing2[ , colSums(testing2 == "") == 0]
```

Next we try to eliminate vars that are highly correlated (> 0.9) excluding non-numeric vars. We choose to remove gyros_forearm_z, gyros_dumbbell_z, gyros_arm_y, accel_belt_z and accel_belt_y. We also choose to remvoe vars X, user_name, cvtd_timestamp and new_window since by their nature they shouldn't be significant predictors of classe. We are left with 51 vars (training4.)

We also check for nearZeroVars and we don't get any.

```
harCorr <- as.data.frame(cor(training3[,c(3:4,7:59)]))
which(colSums(abs(harCorr) >= 0.9) > 1)
```

```
##        roll_belt       pitch_belt total_accel_belt     accel_belt_x
##                4                5                7               11
##     accel_belt_y     accel_belt_z      gyros_arm_x      gyros_arm_y
##               12               13               21               22
```

```
training4 <- subset(training3, select= -
c(X,user_name,cvtd_timestamp,new_window,gyros_forearm_z,gyros_dumbbell_z,gyros_arm_y,accel_belt_z,accel_belt_y))
testing4 <- subset(testing3, select= -c(X,user_name,cvtd_timestamp,new_window,gyros_forearm_z,gyros_dumbbell_z,gyros_arm_y,a
ccel_belt_z,accel_belt_y))

nzv <- nearZeroVar(training4,saveMetrics=TRUE)
```

Model Fitting

We now fit four different models onto the cleaned up training4 dataset: lda, gbm, rf and rpart.

```
set.seed(918273645)

modFit1 <- train(classe ~ ., data=training4, method="lda")
pred1 <- predict(modFit1, testing4)
confM1 <- confusionMatrix(pred1, testing4$classe)

cont2 <- trainControl(method = "repeatedcv", number=5, repeats=1)
modFit2  <- train(classe ~ ., training4, method="gbm", trControl=cont2, verbose=FALSE)
pred2 <- predict(modFit2, testing4)
confM2 <- confusionMatrix(pred2, testing4$classe)

modFit3 <- randomForest(classe ~ ., data=training4)
pred3 <- predict(modFit3, testing4)
confM3 <- confusionMatrix(pred3, testing4$classe)

modFit4 <- rpart(classe ~ ., data=training4, method="class")
pred4 <- predict(modFit4, testing4, type = "class")
confM4 <- confusionMatrix(pred4, testing4$classe)
```

Best Model Selection

Based on the results above, we get the following out of sample error rates for the 4 attempted models:

- lda model out of sample error rate: 0.2987256
- gbm out of sample error rate: 0.0028887
- rf out of sample error rate: 0.0018692
- rpart out of sample error rate: 0.1619371

```
confM1$overall[1]
```

```
##  Accuracy
## 0.6949873
```

```
confM2$overall[1]
```

```
##  Accuracy
## 0.9967715
```

```
confM3$overall[1]
```

```
##  Accuracy
## 0.9986406
```

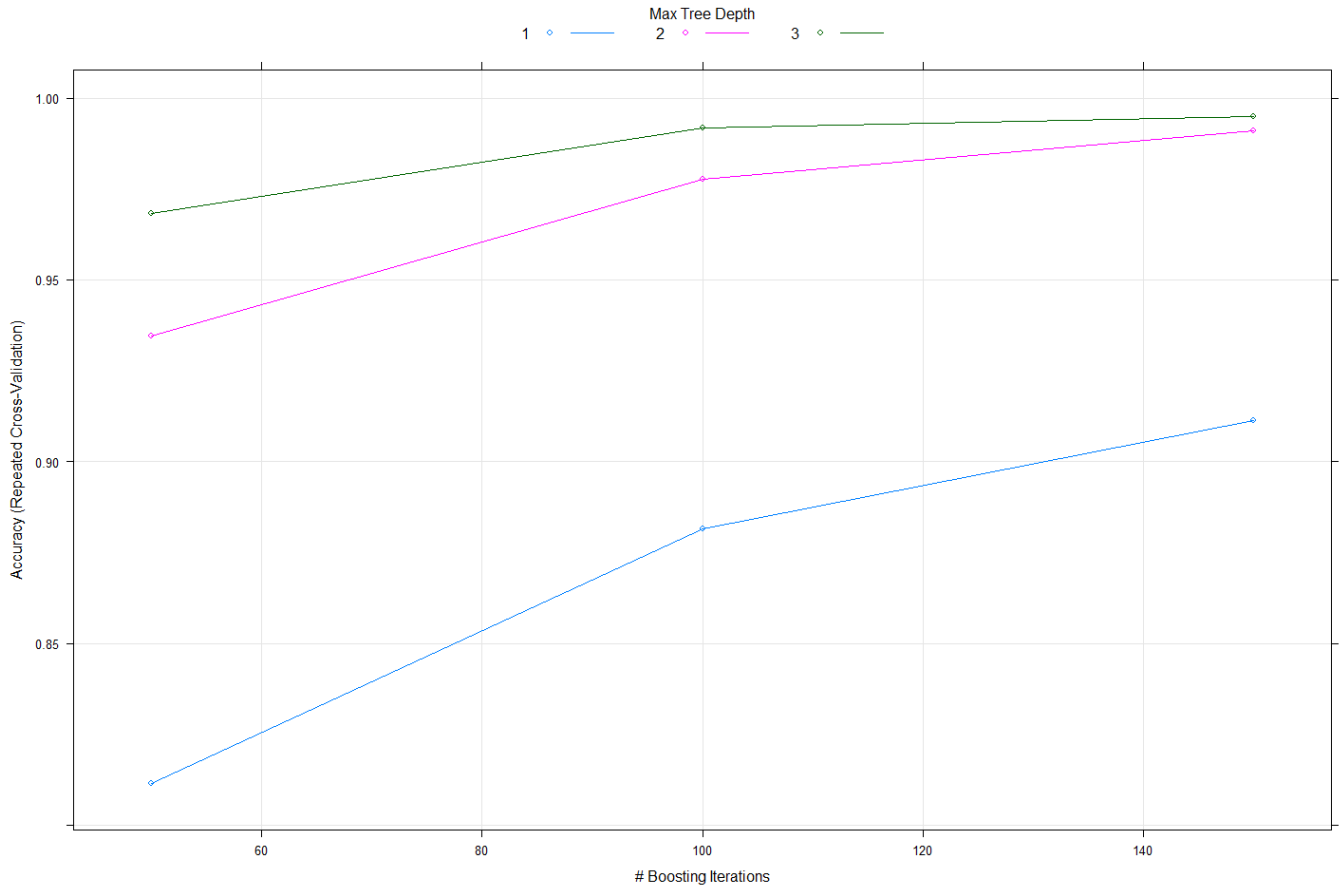
```
confM4$overall[1]
```

```
##  Accuracy
## 0.8285472
```

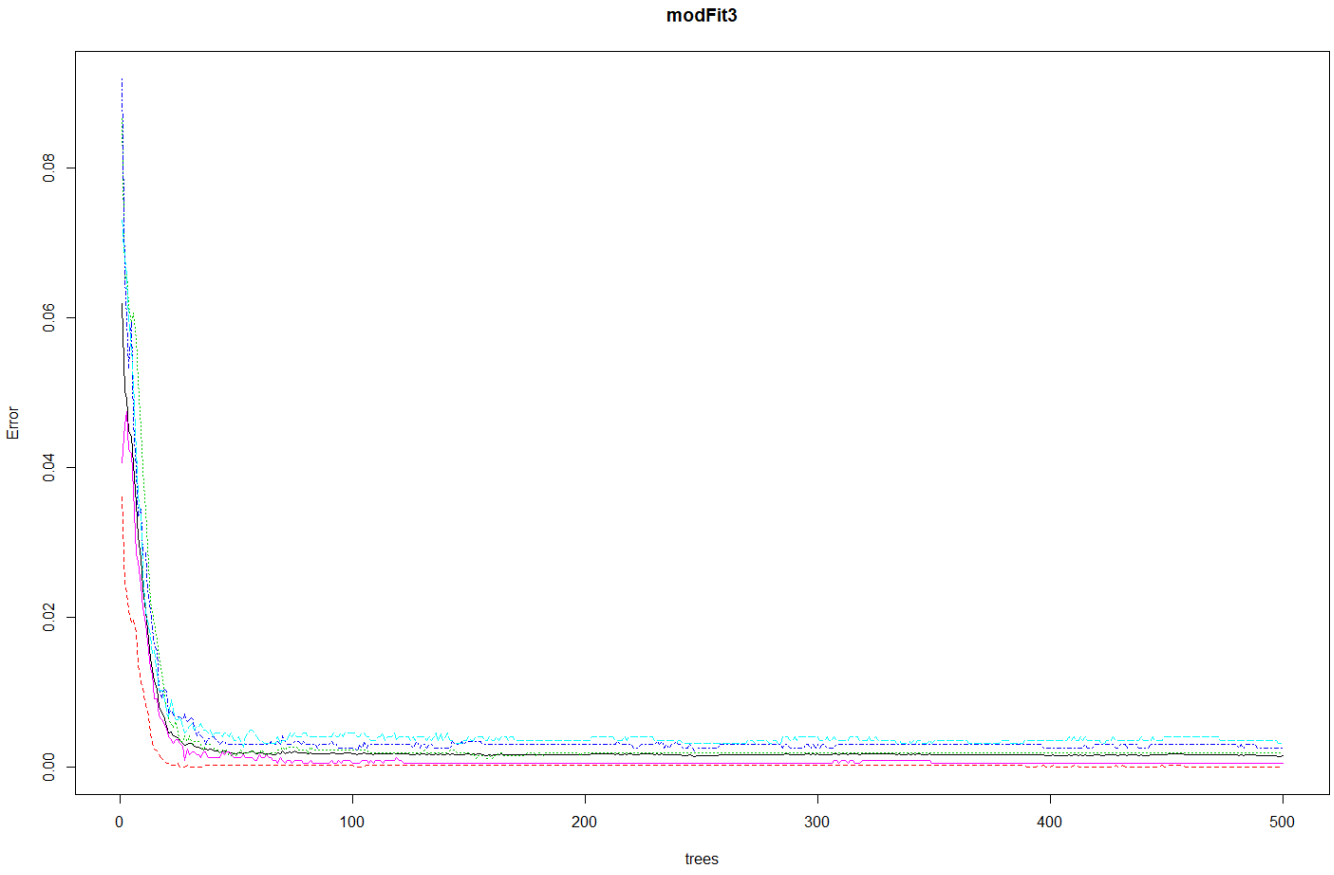
So we select rf as the best model even though gbm would have been equally good. Given the high accuracy of both (> 0.99) we don't think there is need for combining two or more models together.

Plots for the different attempted models

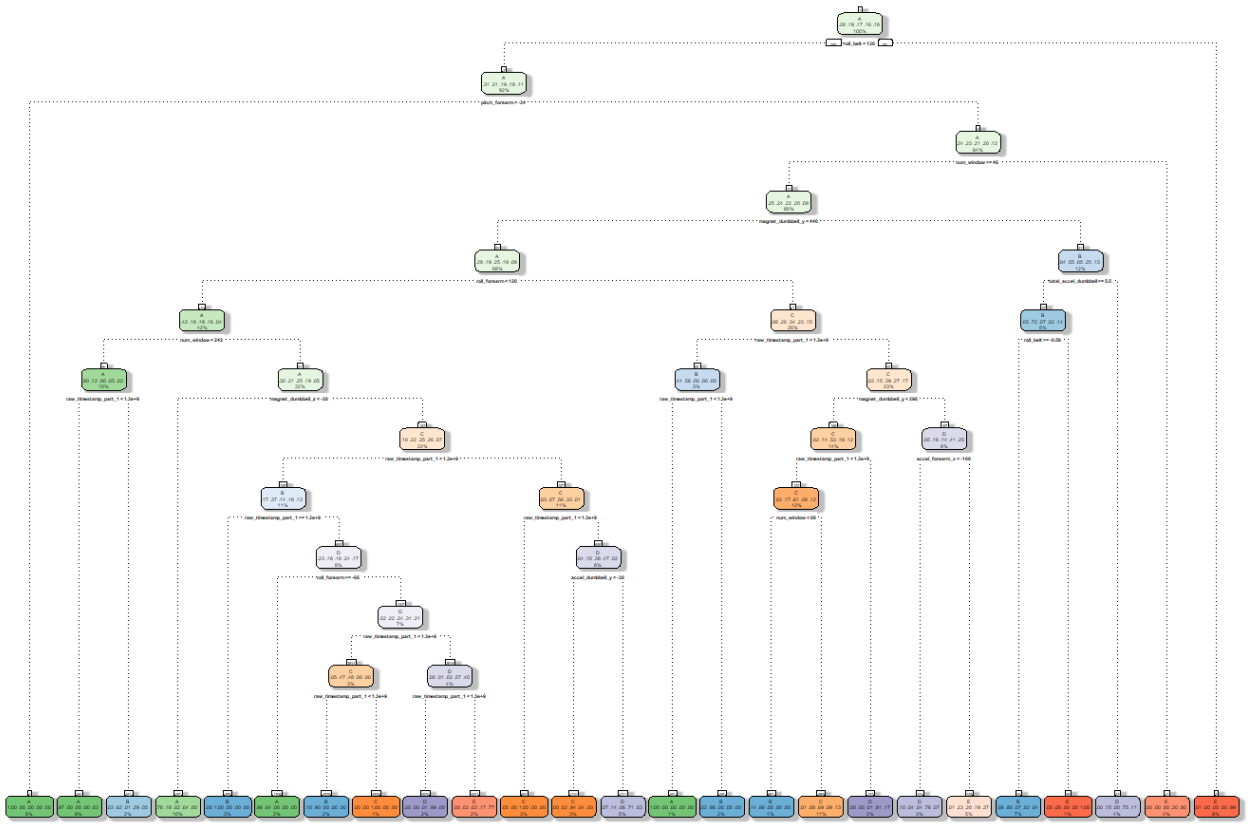
```
plot(modFit2)
```




```
plot(modFit3)
```



```
fancyRpartPlot(modFit4)
```



Rattle 2017-Jul-28 10:47:31 Zaher

Project Quiz

For testing, we do the same cleanup done previously to training and testing. We also remove problem_id since it's an added var with no relation to the data. We end up with pmlTesting4 with 50 vars.

Models 2 (gbm) and 3 (rf) give exactly the same result and they both have accuracy > 0.99. So the quiz ansers will are answered using the output from these models and they are 100% correct.

```
#par(mfrow = c(2,2)); plot(lmfitfinal)

pmlTesting2 <- pmlTesting[ , colSums(is.na(testing)) == 0]
pmlTesting3 <- pmlTesting2[ , colSums(testing2 == "") == 0]
pmlTesting4 <- subset(pmlTesting3, select= -c(X,user_name,cvtd_timestamp,new_window,gyros_forearm_z,gyros_dumbbell_z,gyros_a
rm_y,accel_belt_z,accel_belt_y,problem_id))

predict(modFit1, newdata=pmlTesting4)
```

```
##  [1] B A B A A C D D A A D A E A E A A B B B
## Levels: A B C D E
```

```
predict(modFit2, newdata=pmlTesting4)
```

```
##  [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
```

```
predict(modFit3, newdata=pmlTesting4)
```

```
##  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
```

```
predict(modFit4, pmlTesting4, type = "class")
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  E  A  A  C  D  E  A  A  C  C  B  A  E  E  A  A  B  B
## Levels: A B C D E
```