Elliott Machine Learning Final Class Project

John Elliott

September 3, 2016

## Background of the project:

This work is created for the final class project in the Practical Machine Learning Course, as part of the Coursera Data Science Track. To demonstrate machine learning techniques learned during the course data from <http://groupware.les.inf.puc-rio.br/har> is used to construct a model that can discern between the different classe type. The data is from fitbit type devices worn by weightlifters performing exercises where they purposely performed the exercise with a particular type of fault winch makes up the **classe** variable. The data experiment was published in the following paper: **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.** and you can read more about it here: <http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises>

## Execitive summary of the results:

To predict the outcome of the testing data, 3 models were developed. The first was a classification tree that used all predictor variables. It was not able to predict one of the classe types **"D"**. The classification tree was pruned by choosing predictor variables selected from feature plots where separation could be seen. The second model, the pruned classification tree was able to predict all classe types. The validation of the two classification trees was disappointing the first showed better accuracy but couldn't predict all classe types, both models had accuracy below 60%. A third model was created using a Random Forest method. This model yielded excellent results with the data set with an accuracy of 99%.

## Code:

#### Data Download:

Data is downloaded from the defined URLs. The code makes a check to see if the required data file is present in the working directory. If it is not found then the file is downloaded and placed into the working directory.

URL <- c("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",  
 "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")  
  
 required\_files <- c("pml-training.csv", "pml-testing.csv")  
  
 for(i in 1:length(URL)){  
 if(!all(lapply(required\_files[i],file.exists)==TRUE)){   
 print("Required Data Files Not Found!")  
 print("Please wait while the files are downloaded")  
 download.file(URL[i], destfile = required\_files[i], mode="wb")  
 }  
 }

#### Load Data Into Envioroment (Reading In Data)

A check for the required data is made and if not found it is read into the environment. Also "cache = TRUE" is set for the data chunk to provide data caching.

if(!exists("trainingData")){   
 print("Please Wait while dataset is loaded")  
 trainingData <- read.csv("pml-training.csv", header = TRUE, sep = "," ,  
 dec = ".",na.strings = c("", NA) )  
 }

[1] "Please Wait while dataset is loaded"

if(!exists("testingData")){   
 print("Please Wait while dataset is loaded")  
 testingData <- read.csv("pml-testing.csv", header = TRUE, sep = "," ,  
 dec = ".", na.strings = c("", NA) )  
 }

[1] "Please Wait while dataset is loaded"

#### Inspection of the dataset

First we take a quick look at the data set to see what we have.

[1] 20 160

X user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_timestamp  
3 3 jeremy 1322673075 342967 30/11/2011 17:11

'data.frame': 1 obs. of 5 variables:  
 $ X : int 3  
 $ user\_name : Factor w/ 6 levels "adelmo","carlitos",..: 5  
 $ raw\_timestamp\_part\_1: int 1322673075  
 $ raw\_timestamp\_part\_2: int 342967  
 $ cvtd\_timestamp : Factor w/ 11 levels "02/12/2011 13:33",..: 10

X user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_timestamp  
3 3 carlitos 1323084231 820366 05/12/2011 11:23

'data.frame': 1 obs. of 5 variables:  
 $ X : int 3  
 $ user\_name : Factor w/ 6 levels "adelmo","carlitos",..: 2  
 $ raw\_timestamp\_part\_1: int 1323084231  
 $ raw\_timestamp\_part\_2: int 820366  
 $ cvtd\_timestamp : Factor w/ 20 levels "02/12/2011 13:32",..: 9

We check for incomplete data columns with Na's in the two data sets, remove incomplete variables from the population and omit first 7 columns of descriptor variables.

testingData <- testingData[, colSums(is.na(testingData)) == 0]   
trainingData <- trainingData[, colSums(is.na(trainingData)) == 0]   
  
checknames <- is.na(match(colnames(trainingData), colnames(testingData)))  
  
trainingData <- trainingData[,7:60]  
testingData <- testingData[,7:60]

We create a validation subset from the training data

set.seed(1972)   
  
inTrain <- createDataPartition(trainingData$classe, p = 0.7, list = FALSE)  
trainingData <- trainingData[inTrain, ]  
validate <- trainingData[-inTrain, ]

Exploratory plots of the data are created to look for a subset of predictor variables. An example plot of 4 variables is shown, the process was repeated to find predictors that may improve the modeling accuracy. The list of predictors was placed in a variable named **training\_set**.

#### Model development of Classification Trees:

##### Model One:

The first model is a classification tree (method = "rpart") using all predictors. The model uses the default train control method of resampling, cross validation "cv", and a fold of 10.

set.seed(1972)  
  
model\_1 <- train(as.factor(classe) ~ . , method = "rpart", data = trainingData)  
  
 fancyRpartPlot(model\_1$finalModel)

The first model was not able to discern classe type **"D"**, the alternate set of predictors were used to create the second model a pruned version of the first classification tree.

##### Model Two:

Model two also uses the default train control method of resampling, cross validation "cv", and a fold of 10.

model\_2 <- train(as.factor(classe) ~ accel\_belt\_y + accel\_belt\_z + magnet\_belt\_x + magnet\_belt\_y + accel\_dumbbell\_z + magnet\_dumbbell\_x + magnet\_dumbbell\_y + magnet\_dumbbell\_z + roll\_forearm + pitch\_forearm + yaw\_forearm + total\_accel\_forearm + gyros\_forearm\_x , method = "rpart", data = trainingData)  
  
   
 fancyRpartPlot(model\_2$finalModel)

#### Validation of the Classification Trees

Validation of the two models was performed using the validation data that was subset ted from the training data.

mod1\_prediction <- predict(model\_1, validate)  
mod2\_prediction <- predict(model\_2, validate)  
  
mod1\_confusion <- as.matrix(confusionMatrix(validate$classe, mod1\_prediction))  
mod2\_confusion <- as.matrix(confusionMatrix(validate$classe, mod2\_prediction))  
  
accuracy\_mod1 <- confusionMatrix(validate$classe, mod1\_prediction)$overall[1]  
accuracy\_mod2 <- confusionMatrix(validate$classe, mod2\_prediction)$overall[1]

#### Predictions made with Classification Trees:

mod1\_predict <- predict(model\_1, testingData)  
mod2\_predict <- predict(model\_2, testingData)

Model 1 has better accuracy at 0.5661337 compared to model 2 accuracy of 0.4779554 . With both models having ac curacies below 60% another modelling method was explored.

#### Model development of Random Forests

The last model explored is a Random Forest (method = "rf") using all predictor variables.

##### Model Three:

For the training The train control used was k fold cross validation "cv", with a k fold value of 5.

myControl <- trainControl(method = "cv", number = 5)  
  
mod3 <- train(as.factor(classe) ~ .,method="rf",trControl=myControl,data=trainingData)

#### Validation of the Random Forest

Validation of the Random Forest model was performed using the validation data that was subset ted from the training data.

val\_predict <- predict(mod3, validate)  
My\_Confusion <- as.matrix(confusionMatrix(validate$classe, val\_predict))  
accuracy\_mod3 <- confusionMatrix(validate$classe, val\_predict)$overall[1]

The Random Forest model is able to predict all classe types and has an accuracy approaching 100% based on the validation data.

mod3\_predict <- predict(mod3, testingData)

#### Model Comparison

Tables of the accuracy and prediction values are created for model comparison.

Accuracy\_Table <- as.data.frame(matrix(nrow=3,ncol=2))  
Method <- c("Classification Tree", "Pruned Tree", "Random Forest")  
rownames(Accuracy\_Table)<-c("Model 1", "Model 2", "Model 3")  
colnames(Accuracy\_Table)<- c( "Method Used", "Accuracy")  
Accuracy\_Table[,1] <- as.data.frame(Method)  
Accuracy\_Table[1,2] <- as.data.frame(accuracy\_mod1)  
Accuracy\_Table[2,2] <- as.data.frame(accuracy\_mod2)  
Accuracy\_Table[3,2] <- as.data.frame(accuracy\_mod3)  
  
Predict\_Table <- as.data.frame(matrix(nrow=20,ncol=3))  
colnames(Predict\_Table) <-c("Model 1", "Model 2", "Model 3")  
Predict\_Table[,1] <- as.data.frame(mod1\_predict)  
Predict\_Table[,2] <- as.data.frame(mod2\_predict)  
Predict\_Table[,3] <- as.data.frame(mod3\_predict)  
  
(Accuracy\_Table)

Method Used Accuracy  
Model 1 Classification Tree 0.5661337  
Model 2 Pruned Tree 0.4779554  
Model 3 Random Forest 1.0000000

(Predict\_Table)

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

#### Plots of the Confusion Matrix for each model

library(reshape2)  
melted\_mod1\_confusion <- melt(mod1\_confusion)  
melted\_mod2\_confusion <- melt(mod2\_confusion)  
melted\_My\_Confusion <- melt(My\_Confusion)  
colnames( melted\_mod1\_confusion ) <- c("Classe", "Predicted","value")  
colnames( melted\_mod2\_confusion ) <- c("Classe", "Predicted","value")  
colnames( melted\_My\_Confusion ) <- c("Classe", "Predicted","value")  
library(gridExtra)  
g1 <-ggplot(data = melted\_mod1\_confusion, aes(x=Classe, y=Predicted, fill=value)) +   
 geom\_tile() + ggtitle("Model 1") + coord\_fixed(ratio=1)  
  
g2 <-ggplot(data = melted\_mod2\_confusion, aes(x=Classe, y=Predicted, fill=value)) +   
 geom\_tile() + ggtitle("Model 2") + coord\_fixed(ratio=1)  
  
g3 <- ggplot(data = melted\_My\_Confusion, aes(x=Classe, y=Predicted, fill=value)) +   
 geom\_tile() + ggtitle("Model 3") + coord\_fixed(ratio=1)  
  
grid.arrange(g1, g2, g3, ncol = 3)

The plots above of the confusion matrices, show how the Random Forest model performance (model 3) out performs the Classification Tree model. It can also be seen that Model 1 can not predict Classe type "D".

## Summary and Conclusions

The random Forest model out performed the Classification Tree and Pruned tree models by leaps and bounds with an accuracy over 99%, concluding for this data set it is the best model selection. An important note is that the results are only based on the data set that the models were trained on and it is expected the results could be different for data collected under different circumstances and conditions, such as new individuals who perform the exercises.