Practical machine Learning Project **Curtis Kaminski**

**Introduction*

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. 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 here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here:

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

The test data are available here:

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

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

** Load libraries** `{r load_libraries} library(caret) library(randomForest) library(e1071) library(rattle) library(knitr)`

Download data files "'{r download_data_files} trainFile <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv" trainFileName <- "trainingdata.csv"

testFile <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv" testFileName <- "testdata.csv"

download.file(trainFile, destfile=trainFileName) download.file(testFile, destfile=testFileName) "'

** Read data sets into data frames** `{r read_data_sets} trainData <- read.csv("trainingdata.csv") testData <- read.csv("testdata.csv")`

Remove NAs `{r remove_NA} trainData <- trainData[, colSums(is.na(trainData)) == 0] testData <- testData[, colSums(is.na(trainData)) == 0]`

Get rid of columns with little effect on accelerometer "'{r fix_data} classe <- trainData$classe trainFix < -grepl("^X|timestamp|window", names(trainData)) trainData < -trainData[, !trainFix] trainData < -trainData[, sapply(trainData, is.numeric)] trainData$classe <- classe

classe <- testData$classe testFix < -grepl("^X|timestamp|window", names(testData)) testData < -testData[, !testFix] testData < -testData[, sapply(testData, is.numeric)] testData$classe <- classe "'

Partition TrainData to create a training set and a test set within the training data `{r create_training_sets} set.seed(333) # For reproducibile purpose inTrain <- createDataPartition(trainDat p=0.70, list=F) trainSet <- trainData[inTrain, ] testSet <- trainData[-inTrain, ]`

** Train Models** "'{r train_models} modFitRpart <- train(classe ~ ., data=trainSet, method="rpart")

Train Random Forest Tree

modFitRF <- train(classe ~ ., data=trainSet, method="rf") "'

Conclusion

As we can see looking at Figures 1 & 2, the Random Forrest Model is much more accurate at 98.8% so we will us it for our predictions.

Print the Prediction

```
predictRF <- predict(modFitRF, testData)
predictRF
```

Appendix Figures 1. Decision Tree Accuracy `{r print_decision_ree_ccurracy} print(modFitRpart)`

2. Random Forest Accuracy `{r print_random_forest_accuracy} print(modFitRF)`