Practical Machine Learning Course Project

W.X.

August 22, 2015

Background

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.

What you should submit

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

1. Your submission should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-).
2. You should also apply your machine learning algorithm to the 20 test cases available in the test data above. Please submit your predictions in appropriate format to the programming assignment for automated grading. See the programming assignment for additional details.

Preliminary Work

Reproduceability

An overall pseudo-random number generator seed was set at 1234 for all code. In order to reproduce the results below, the same seed should be used. Different packages were downloaded and installed, such as caret and randomForest. These should also be installed in order to reproduce the results below (please see code below for ways and syntax to do so).

How the model was built

Our outcome variable is classe, a factor variable with 5 levels. For this data set, "participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in 5 different fashions:

```
# exactly according to the specification (Class A)
# throwing the elbows to the front (Class B)
# lifting the dumbbell only halfway (Class C)
# lowering the dumbbell only halfway (Class D)
# throwing the hips to the front (Class E)
```

Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes." [1] Prediction evaluations will be based on maximizing the accuracy and minimizing the out-of-sample error. All other available variables after cleaning will be used for prediction. Two models will be tested using decision tree and random forest algorithms. The model with the highest accuracy will be chosen as our final model.

Cross-validation

Cross-validation will be performed by subsampling our training data set randomly without replacement into 2 subsamples: subTraining data (75% of the original Training data set) and subTesting data (25%). Our models will be fitted on the subTraining data set, and tested on the subTesting data. Once the most accurate model is choosen, it will be tested on the original Testing data set.

Expected out-of-sample error

The expected out-of-sample error will correspond to the quantity: 1-accuracy in the cross-validation data. Accuracy is the proportion of correct classified observation over the total sample in the subTesting data set. Expected accuracy is the expected accuracy in the out-of-sample data set (i.e. original testing data set). Thus, the expected value of the out-of-sample error will correspond to the expected number of missclassified observations/total observations in the Test data set, which is the quantity: 1-accuracy found from the cross-validation data set.

Reasons for my choices

Our outcome variable "classe" is an unordered factor variable. Thus, we can choose our error type as 1-accuracy. We have a large sample size with N= 19622 in the Training data set. This allow us to divide our Training sample into subTraining and subTesting to allow cross-validation. Features with all missing values will be discarded as well as features that are irrelevant. All other features will be kept as relevant variables.Decision tree and random forest algorithms are known for their ability of detecting the features that are important for classification [2]. Feature selection is inherent, so it is not so necessary at the data preparation phase. Thus, there won't be any feature selection section in this report.

Code and Results

Packages, Libraries, Seed

Installing packages, loading libraries, and setting the seed for reproduceability:

```
## Warning: package 'caret' was built under R version 3.1.3
```

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

```
## Warning: package 'lattice' was built under R version 3.1.3
```

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

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
library(randomForest) #Random forest for classification and regression
```

```
## Warning: package 'randomForest' was built under R version 3.1.2
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(rpart) # Regressive Partitioning and Regression trees
```

```
## Warning: package 'rpart' was built under R version 3.1.2
```

```
library(rpart.plot) # Decision Tree plot
```

```
## Warning: package 'rpart.plot' was built under R version 3.1.2
```

```
# setting the overall seed for reproduceability
set.seed(1234)
```

Loading data sets and preliminary cleaning

First we want to load the data sets into R and make sure that missing values are coded correctly. Irrelevant variables will be deleted. Results will be hidden from the report for clarity and space considerations.

```
# After saving both data sets into my working directory
# Some missing values are coded as string "#DIV/0!" or "" or "NA" - these will be changed to NA.
# We notice that both data sets contain columns with all missing values - these will be deleted.

# Loading the training data set into my R session replacing all missing with "NA"
trainingset <- read.csv("~/Desktop/pml-training.csv", na.strings=c("NA","#DIV/0!", ""))

# Loading the testing data set
testingset <- read.csv('~/Desktop/pml-testing.csv', na.strings=c("NA","#DIV/0!", ""))

# Check dimensions for number of variables and number of observations
dim(trainingset)
```

```
## [1] 19622   160
```

```r
dim(testingset)
```

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

```r
# Delete columns with all missing values
trainingset<-trainingset[,colSums(is.na(trainingset)) == 0]
testingset <-testingset[,colSums(is.na(testingset)) == 0]

# Some variables are irrelevant to our current project: user_name, raw_timestamp_part_1, raw_timestamp_p
trainingset <-trainingset[,-c(1:7)]
testingset <-testingset[,-c(1:7)]

# and have a look at our new datasets:
dim(trainingset)
```

```
## [1] 19622    53
```

```r
dim(testingset)
```

```
## [1] 20 53
```

```r
head(trainingset)
```

```
##   roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x gyros_belt_y
## 1      1.41       8.07    -94.4                3         0.00         0.00
## 2      1.41       8.07    -94.4                3         0.02         0.00
## 3      1.42       8.07    -94.4                3         0.00         0.00
## 4      1.48       8.05    -94.4                3         0.02         0.00
## 5      1.48       8.07    -94.4                3         0.02         0.02
## 6      1.45       8.06    -94.4                3         0.02         0.00
##   gyros_belt_z accel_belt_x accel_belt_y accel_belt_z magnet_belt_x
## 1        -0.02          -21            4           22            -3
## 2        -0.02          -22            4           22            -7
## 3        -0.02          -20            5           23            -2
## 4        -0.03          -22            3           21            -6
## 5        -0.02          -21            2           24            -6
## 6        -0.02          -21            4           21             0
##   magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm
## 1           599          -313     -128      22.5    -161              34
## 2           608          -311     -128      22.5    -161              34
## 3           600          -305     -128      22.5    -161              34
## 4           604          -310     -128      22.1    -161              34
## 5           600          -302     -128      22.1    -161              34
## 6           603          -312     -128      22.0    -161              34
##   gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## 1        0.00        0.00       -0.02        -288         109        -123
## 2        0.02       -0.02       -0.02        -290         110        -125
## 3        0.02       -0.02       -0.02        -289         110        -126
## 4        0.02       -0.03        0.02        -289         111        -123
## 5        0.00       -0.03        0.00        -289         111        -123
## 6        0.02       -0.03        0.00        -289         111        -122
```

```
##   magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell pitch_dumbbell
## 1         -368          337          516      13.05217      -70.49400
## 2         -369          337          513      13.13074      -70.63751
## 3         -368          344          513      12.85075      -70.27812
## 4         -372          344          512      13.43120      -70.39379
## 5         -374          337          506      13.37872      -70.42856
## 6         -369          342          513      13.38246      -70.81759
##   yaw_dumbbell total_accel_dumbbell gyros_dumbbell_x gyros_dumbbell_y
## 1    -84.87394                   37                0            -0.02
## 2    -84.71065                   37                0            -0.02
## 3    -85.14078                   37                0            -0.02
## 4    -84.87363                   37                0            -0.02
## 5    -84.85306                   37                0            -0.02
## 6    -84.46500                   37                0            -0.02
##   gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z
## 1             0.00             -234               47             -271
## 2             0.00             -233               47             -269
## 3             0.00             -232               46             -270
## 4            -0.02             -232               48             -269
## 5             0.00             -233               48             -270
## 6             0.00             -234               48             -269
##   magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z roll_forearm
## 1              -559               293               -65         28.4
## 2              -555               296               -64         28.3
## 3              -561               298               -63         28.3
## 4              -552               303               -60         28.1
## 5              -554               292               -68         28.0
## 6              -558               294               -66         27.9
##   pitch_forearm yaw_forearm total_accel_forearm gyros_forearm_x
## 1         -63.9        -153                  36            0.03
## 2         -63.9        -153                  36            0.02
## 3         -63.9        -152                  36            0.03
## 4         -63.9        -152                  36            0.02
## 5         -63.9        -152                  36            0.02
## 6         -63.9        -152                  36            0.02
##   gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y
## 1            0.00           -0.02             192             203
## 2            0.00           -0.02             192             203
## 3           -0.02            0.00             196             204
## 4           -0.02            0.00             189             206
## 5            0.00           -0.02             189             206
## 6           -0.02           -0.03             193             203
##   accel_forearm_z magnet_forearm_x magnet_forearm_y magnet_forearm_z
## 1            -215              -17              654              476
## 2            -216              -18              661              473
## 3            -213              -18              658              469
## 4            -214              -16              658              469
## 5            -214              -17              655              473
## 6            -215               -9              660              478
##   classe
## 1      A
## 2      A
## 3      A
## 4      A
```

```
## 5      A
## 6      A
```

```r
head(testingset)
```

```
##   roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x gyros_belt_y
## 1    123.00      27.00    -4.75               20        -0.50        -0.02
## 2      1.02       4.87   -88.90                4        -0.06        -0.02
## 3      0.87       1.82   -88.50                5         0.05         0.02
## 4    125.00     -41.60   162.00               17         0.11         0.11
## 5      1.35       3.33   -88.60                3         0.03         0.02
## 6     -5.92       1.59   -87.70                4         0.10         0.05
##   gyros_belt_z accel_belt_x accel_belt_y accel_belt_z magnet_belt_x
## 1        -0.46          -38           69         -179           -13
## 2        -0.07          -13           11           39            43
## 3         0.03            1           -1           49            29
## 4        -0.16           46           45         -156           169
## 5         0.00           -8            4           27            33
## 6        -0.13          -11          -16           38            31
##   magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm
## 1           581          -382     40.7    -27.80     178              10
## 2           636          -309      0.0      0.00       0              38
## 3           631          -312      0.0      0.00       0              44
## 4           608          -304   -109.0     55.00    -142              25
## 5           566          -418     76.1      2.76     102              29
## 6           638          -291      0.0      0.00       0              14
##   gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## 1       -1.65        0.48       -0.18          16          38          93
## 2       -1.17        0.85       -0.43        -290         215         -90
## 3        2.10       -1.36        1.13        -341         245         -87
## 4        0.22       -0.51        0.92        -238         -57           6
## 5       -1.96        0.79       -0.54        -197         200         -30
## 6        0.02        0.05       -0.07         -26         130         -19
##   magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell pitch_dumbbell
## 1         -326          385          481     -17.73748       24.96085
## 2         -325          447          434      54.47761      -53.69758
## 3         -264          474          413      57.07031      -51.37303
## 4         -173          257          633      43.10927      -30.04885
## 5         -170          275          617    -101.38396      -53.43952
## 6          396          176          516      62.18750      -50.55595
##   yaw_dumbbell total_accel_dumbbell gyros_dumbbell_x gyros_dumbbell_y
## 1    126.23596                    9             0.64             0.06
## 2    -75.51480                   31             0.34             0.05
## 3    -75.20287                   29             0.39             0.14
## 4   -103.32003                   18             0.10            -0.02
## 5    -14.19542                    4             0.29            -0.47
## 6    -71.12063                   29            -0.59             0.80
##   gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z
## 1            -0.61               21              -15               81
## 2            -0.71             -153              155             -205
## 3            -0.34             -141              155             -196
## 4             0.05              -51               72             -148
## 5            -0.46              -18              -30               -5
## 6             1.10             -138              166             -186
```

```
##   magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z roll_forearm
## 1               523              -528               -56          141
## 2              -502               388               -36          109
## 3              -506               349                41          131
## 4              -576               238                53            0
## 5              -424               252               312         -176
## 6              -543               262                96          150
##   pitch_forearm yaw_forearm total_accel_forearm gyros_forearm_x
## 1         49.30       156.0                  33            0.74
## 2        -17.60       106.0                  39            1.12
## 3        -32.60        93.0                  34            0.18
## 4          0.00         0.0                  43            1.38
## 5         -2.16       -47.9                  24           -0.75
## 6          1.46        89.7                  43           -0.88
##   gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y
## 1           -3.34           -0.59            -110             267
## 2           -2.78           -0.18             212             297
## 3           -0.79            0.28             154             271
## 4            0.69            1.80             -92             406
## 5            3.10            0.80             131             -93
## 6            4.26            1.35             230             322
##   accel_forearm_z magnet_forearm_x magnet_forearm_y magnet_forearm_z
## 1            -149             -714              419              617
## 2            -118             -237              791              873
## 3            -129              -51              698              783
## 4             -39             -233              783              521
## 5             172              375             -787               91
## 6            -144             -300              800              884
##   problem_id
## 1          1
## 2          2
## 3          3
## 4          4
## 5          5
## 6          6
```

Partitioning the training data set to allow cross-validation

The training data set contains 53 variables and 19622 obs. The testing data set contains 53 variables and 20 obs. In order to perform cross-validation, the training data set is partionned into 2 sets: subTraining (75%) and subTest (25%). This will be performed using random subsampling without replacement.

```r
subsamples <- createDataPartition(y=trainingset$classe, p=0.75, list=FALSE)
subTraining <- trainingset[subsamples, ]
subTesting <- trainingset[-subsamples, ]
dim(subTraining)
```

```
## [1] 14718    53
```

```r
dim(subTesting)
```

```
## [1] 4904   53
```

```r
head(subTraining)
```

```
##   roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x gyros_belt_y
## 2      1.41       8.07    -94.4                3         0.02         0.00
## 3      1.42       8.07    -94.4                3         0.00         0.00
## 4      1.48       8.05    -94.4                3         0.02         0.00
## 5      1.48       8.07    -94.4                3         0.02         0.02
## 6      1.45       8.06    -94.4                3         0.02         0.00
## 7      1.42       8.09    -94.4                3         0.02         0.00
##   gyros_belt_z accel_belt_x accel_belt_y accel_belt_z magnet_belt_x
## 2        -0.02          -22            4           22            -7
## 3        -0.02          -20            5           23            -2
## 4        -0.03          -22            3           21            -6
## 5        -0.02          -21            2           24            -6
## 6        -0.02          -21            4           21             0
## 7        -0.02          -22            3           21            -4
##   magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm
## 2           608          -311     -128      22.5    -161              34
## 3           600          -305     -128      22.5    -161              34
## 4           604          -310     -128      22.1    -161              34
## 5           600          -302     -128      22.1    -161              34
## 6           603          -312     -128      22.0    -161              34
## 7           599          -311     -128      21.9    -161              34
##   gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## 2        0.02       -0.02       -0.02        -290         110        -125
## 3        0.02       -0.02       -0.02        -289         110        -126
## 4        0.02       -0.03        0.02        -289         111        -123
## 5        0.00       -0.03        0.00        -289         111        -123
## 6        0.02       -0.03        0.00        -289         111        -122
## 7        0.00       -0.03        0.00        -289         111        -125
##   magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell pitch_dumbbell
## 2         -369          337          513      13.13074      -70.63751
## 3         -368          344          513      12.85075      -70.27812
## 4         -372          344          512      13.43120      -70.39379
## 5         -374          337          506      13.37872      -70.42856
## 6         -369          342          513      13.38246      -70.81759
## 7         -373          336          509      13.12695      -70.24757
##   yaw_dumbbell total_accel_dumbbell gyros_dumbbell_x gyros_dumbbell_y
## 2    -84.71065                   37                0            -0.02
## 3    -85.14078                   37                0            -0.02
## 4    -84.87363                   37                0            -0.02
## 5    -84.85306                   37                0            -0.02
## 6    -84.46500                   37                0            -0.02
## 7    -85.09961                   37                0            -0.02
##   gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z
## 2             0.00             -233               47             -269
## 3             0.00             -232               46             -270
## 4            -0.02             -232               48             -269
## 5             0.00             -233               48             -270
## 6             0.00             -234               48             -269
## 7             0.00             -232               47             -270
##   magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z roll_forearm
## 2              -555               296               -64         28.3
```

```
## 3              -561               298               -63          28.3
## 4              -552               303               -60          28.1
## 5              -554               292               -68          28.0
## 6              -558               294               -66          27.9
## 7              -551               295               -70          27.9
##   pitch_forearm yaw_forearm total_accel_forearm gyros_forearm_x
## 2         -63.9        -153                  36            0.02
## 3         -63.9        -152                  36            0.03
## 4         -63.9        -152                  36            0.02
## 5         -63.9        -152                  36            0.02
## 6         -63.9        -152                  36            0.02
## 7         -63.9        -152                  36            0.02
##   gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y
## 2            0.00           -0.02             192             203
## 3           -0.02            0.00             196             204
## 4           -0.02            0.00             189             206
## 5            0.00           -0.02             189             206
## 6           -0.02           -0.03             193             203
## 7            0.00           -0.02             195             205
##   accel_forearm_z magnet_forearm_x magnet_forearm_y magnet_forearm_z
## 2            -216              -18              661              473
## 3            -213              -18              658              469
## 4            -214              -16              658              469
## 5            -214              -17              655              473
## 6            -215               -9              660              478
## 7            -215              -18              659              470
##   classe
## 2      A
## 3      A
## 4      A
## 5      A
## 6      A
## 7      A
```

```
head(subTesting)
```

```
##    roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x
## 1       1.41       8.07    -94.4                3         0.00
## 21      1.60       8.10    -94.4                3         0.02
## 22      1.57       8.09    -94.4                3         0.02
## 23      1.56       8.10    -94.3                3         0.02
## 25      1.53       8.11    -94.4                3         0.03
## 26      1.55       8.09    -94.4                3         0.02
##    gyros_belt_y gyros_belt_z accel_belt_x accel_belt_y accel_belt_z
## 1          0.00        -0.02          -21            4           22
## 21         0.00        -0.02          -20            1           20
## 22         0.02        -0.02          -21            3           21
## 23         0.00        -0.02          -21            4           21
## 25         0.00         0.00          -19            4           21
## 26         0.00         0.00          -21            3           22
##    magnet_belt_x magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm
## 1             -3           599          -313     -128      22.5    -161
## 21           -10           607          -304     -129      20.9    -161
## 22            -2           604          -313     -129      20.8    -161
```

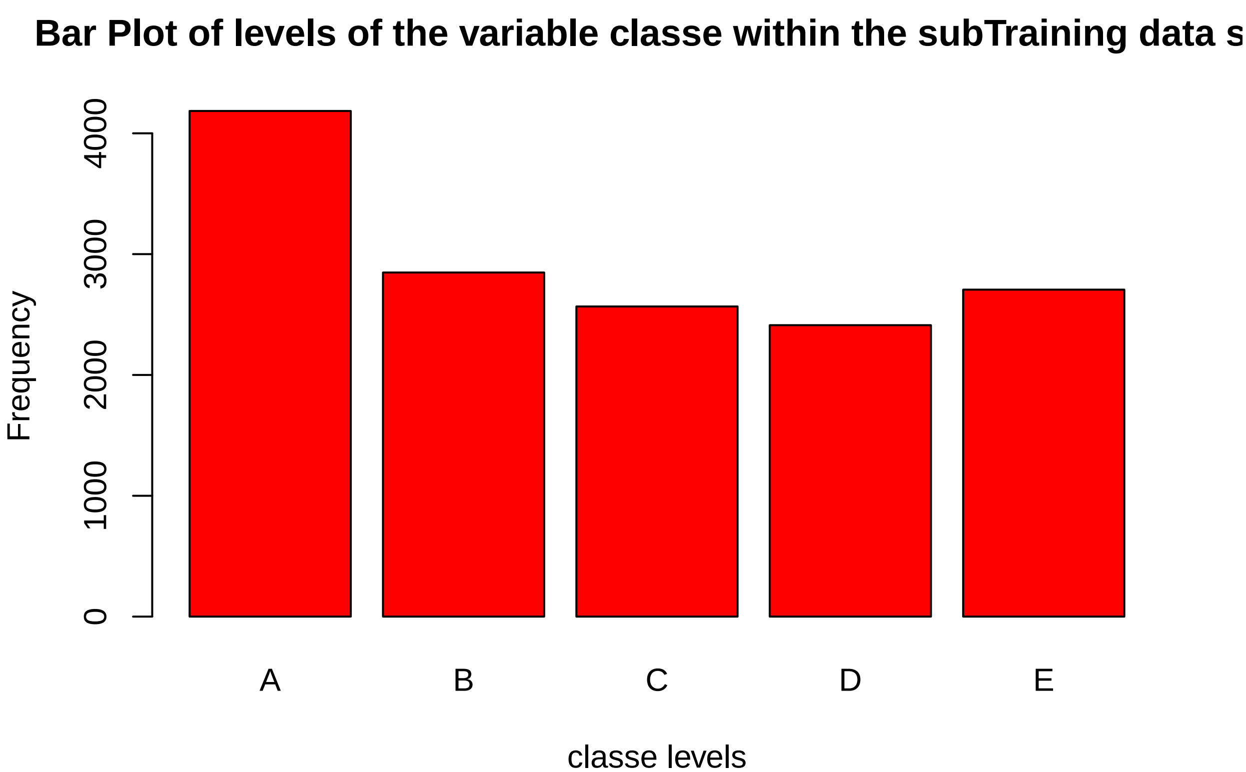
```
## 23           -4           606           -311     -129      20.7    -161
## 25           -8           605           -319     -129      20.7    -161
## 26          -10           601           -312     -129      20.7    -161
##    total_accel_arm gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x
## 1               34        0.00        0.00       -0.02        -288
## 21              34        0.03       -0.02       -0.02        -288
## 22              34        0.03       -0.02       -0.02        -289
## 23              34        0.02       -0.02       -0.02        -290
## 25              34       -0.02       -0.02        0.00        -289
## 26              34       -0.02       -0.02       -0.02        -290
##    accel_arm_y accel_arm_z magnet_arm_x magnet_arm_y magnet_arm_z
## 1          109        -123         -368          337          516
## 21         111        -124         -375          337          513
## 22         111        -123         -372          338          510
## 23         110        -123         -373          333          509
## 25         109        -123         -370          340          512
## 26         108        -123         -366          346          511
##    roll_dumbbell pitch_dumbbell yaw_dumbbell total_accel_dumbbell
## 1       13.05217      -70.49400    -84.87394                   37
## 21      13.38246      -70.81759    -84.46500                   37
## 22      13.37872      -70.42856    -84.85306                   37
## 23      13.35451      -70.63995    -84.64919                   37
## 25      13.05217      -70.49400    -84.87394                   37
## 26      12.80060      -70.31305    -85.11886                   37
##    gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x
## 1                 0            -0.02             0.00             -234
## 21                0            -0.02             0.00             -234
## 22                0            -0.02             0.00             -233
## 23                0            -0.02             0.00             -234
## 25                0            -0.02             0.00             -234
## 26                0            -0.02            -0.02             -233
##    accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y
## 1                47             -271              -559               293
## 21               48             -269              -554               299
## 22               48             -270              -554               301
## 23               48             -270              -557               294
## 25               47             -271              -555               290
## 26               46             -271              -563               294
##    magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm
## 1                -65         28.4         -63.9        -153
## 21               -72         26.9         -63.9        -151
## 22               -65         27.0         -63.9        -151
## 23               -69         26.9         -63.8        -151
## 25               -68         27.1         -63.7        -151
## 26               -72         27.0         -63.7        -151
##    total_accel_forearm gyros_forearm_x gyros_forearm_y gyros_forearm_z
## 1                   36            0.03            0.00           -0.02
## 21                  36            0.03           -0.03           -0.02
## 22                  36            0.02           -0.03           -0.02
## 23                  36            0.02           -0.02           -0.02
## 25                  36            0.05           -0.03            0.00
## 26                  36            0.03            0.00            0.00
##    accel_forearm_x accel_forearm_y accel_forearm_z magnet_forearm_x
## 1              192             203            -215              -17
```

```
## 21             194             208            -214              -11
## 22             191             206            -213              -17
## 23             194             206            -214              -10
## 25             191             202            -214              -14
## 26             190             203            -216              -16
##    magnet_forearm_y magnet_forearm_z classe
## 1               654              476      A
## 21              654              469      A
## 22              654              478      A
## 23              653              467      A
## 25              667              470      A
## 26              658              462      A
```

Data visualization

The variable "classe" contains 5 levels: A, B, C, D and E. A plot of the outcome variable will allow us to see the frequency of each levels in the subTraining data set and compare one another.

```
plot(subTraining$classe, col="red",
     main="Bar Plot of levels of the variable classe within the subTraining data set",
     xlab="classe levels", ylab="Frequency")
```



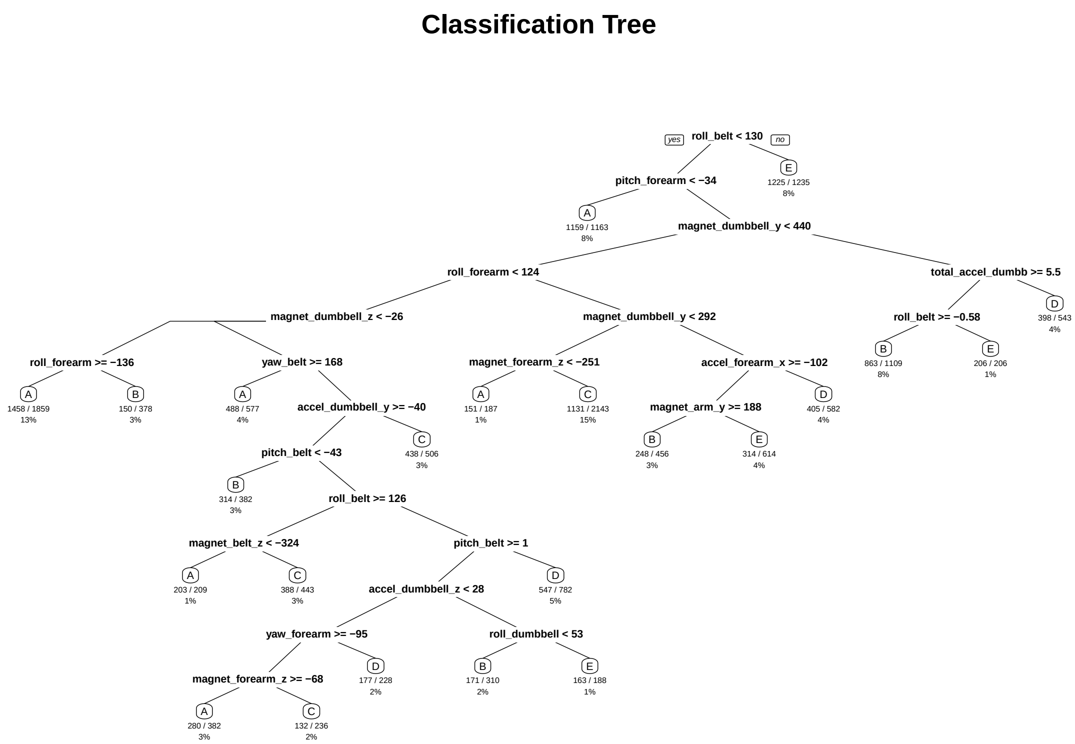
From the graph above, we can see that each level frequency is within the same order of magnitude of each other. Level A is the most frequent with more than 4000 occurrences while level D is the least frequent with about 2500 occurrences.

First prediction model: Using Decision Tree

```r
model1 <- rpart(classe ~ ., data=subTraining, method="class")

# Predicting:
prediction1 <- predict(model1, subTesting, type = "class")

# Plot of the Decision Tree
rpart.plot(model1, main="Classification Tree", extra=102, under=TRUE, faclen=0)
```




```r
#install.packages('e1071', dependencies=TRUE)
# Test results on our subTesting data set:
confusionMatrix(prediction1, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1235  157   16   50   20
##          B   55  568   73   80  102
##          C   44  125  690  118  116
##          D   41   64   50  508   38
##          E   20   35   26   48  625
##
## Overall Statistics
##
##                Accuracy : 0.7394
##                  95% CI : (0.7269, 0.7516)
```

```
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6697
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8853   0.5985   0.8070   0.6318   0.6937
## Specificity            0.9307   0.9216   0.9005   0.9529   0.9678
## Pos Pred Value         0.8356   0.6469   0.6313   0.7247   0.8289
## Neg Pred Value         0.9533   0.9054   0.9567   0.9296   0.9335
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2518   0.1158   0.1407   0.1036   0.1274
## Detection Prevalence   0.3014   0.1790   0.2229   0.1429   0.1538
## Balanced Accuracy      0.9080   0.7601   0.8537   0.7924   0.8307
```

Second prediction model: Using Random Forest

```r
model2 <- randomForest(classe ~. , data=subTraining, method="class")

# Predicting:
prediction2 <- predict(model2, subTesting, type = "class")

# Test results on subTesting data set:
confusionMatrix(prediction2, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    3    0    0    0
##          B    1  944   10    0    0
##          C    0    2  843    6    0
##          D    0    0    2  798    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 0.9951
##                  95% CI : (0.9927, 0.9969)
##     No Information Rate : 0.2845
##     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            0.9993   0.9947   0.9860   0.9925   1.0000
```

```
## Specificity            0.9991   0.9972   0.9980   0.9995   1.0000
## Pos Pred Value         0.9979   0.9885   0.9906   0.9975   1.0000
## Neg Pred Value         0.9997   0.9987   0.9970   0.9985   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1925   0.1719   0.1627   0.1837
## Detection Prevalence   0.2849   0.1947   0.1735   0.1631   0.1837
## Balanced Accuracy      0.9992   0.9960   0.9920   0.9960   1.0000
```

Decision

As expected, Random Forest algorithm performed better than Decision Trees. Accuracy for Random Forest model was 0.995 (95% CI: (0.993, 0.997)) compared to 0.739 (95% CI: (0.727, 0.752)) for Decision Tree model. **The random Forest model is choosen**. The accuracy of the model is 0.995. The expected out-of-sample error is estimated at 0.005, or **0.5%**. The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set. Our Test data set comprises 20 cases. With an accuracy above 99% on our cross-validation data, we can expect that very few, or none, of the test samples will be missclassified.

Submission

```
# predict outcome levels on the original Testing data set using Random Forest algorithm
predictfinal <- predict(model2, testingset, type="class")
predictfinal
```

```
##  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
```

```
# Write files for submission
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictfinal)
```