MLassignment

1. Synopsis

Problem: The objective of this assignment is to analyze data from from accelerometers on the belt, forearm, arm, and dumbell of 6 participants who were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Participants performed the exercises: 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) and throwing the hips to the front (Class E). The goal of the project was to predict the manner in which they did the exercise based on collected body sensor data.

More information about the data set is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Training data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) Test data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Result: Using a random forest model on a final set of 54 variables, the classifier achieved a prediction accuracy of 99.2%.

2. Set environment.

```
remove(list=ls())
set.seed(83749)

library(ggplot2)
library(caret)
library(randomForest)
library(rpart)
library(rpart.plot)
```

3. Data processing.

Load data then remove variables with excess NA values or near zero values.

```
data_train <- read.csv("pml-training.csv")
data_valid <- read.csv("pml-testing.csv")

#remove near zero variables (reduces to 100 variables)
nzv <- nearZeroVar(data_train)
data_train <- data_train[,-nzv]
data_valid <- data_valid[,-nzv]
dim(data_train)
```

```
## [1] 19622   100
```

```
dim(data_valid)
```

```
## [1]  20 100
```

```
#remove NA-variables (reduces to 59 variables)
navars <- sapply(data_train, function(x) mean(is.na(x))) > 0.95
data_train <- data_train[,navars==FALSE]
data_valid <- data_valid[,navars==FALSE]
dim(data_train)
```

```
## [1] 19622    59
```

```
dim(data_valid)
```

```
## [1] 20 59
```

```
#finally adjust output variables and remove label columns (reduces to 53)
data_train$classe <- factor(data_train$classe)
data_train <- data_train[,7:59]
data_valid <- data_valid[,7:59]
dim(data_train)
```

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

```
dim(data_valid)
```

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

4. Prediction Model

Implementing random forest model since this is a well behaving algorithm for a braod range of multi-class data.

```
#partition training data in training and testing set, so that we can arrive at a fair assessment
inTrain  <- createDataPartition(data_train$classe, p=0.6, list=FALSE)
trainset <- data_train[inTrain,]
testset <- data_train[-inTrain,]
dim(trainset)
```

```
## [1] 11776    53
```

```
dim(testset)
```

```
## [1] 7846   53
```

```
#train model
trControl = trainControl(method = "cv", number = 3, verboseIter = TRUE, allowParallel = TRUE)
modFit <- train(classe ~ ., data = trainset, method = "rf", trControl = trControl)
```

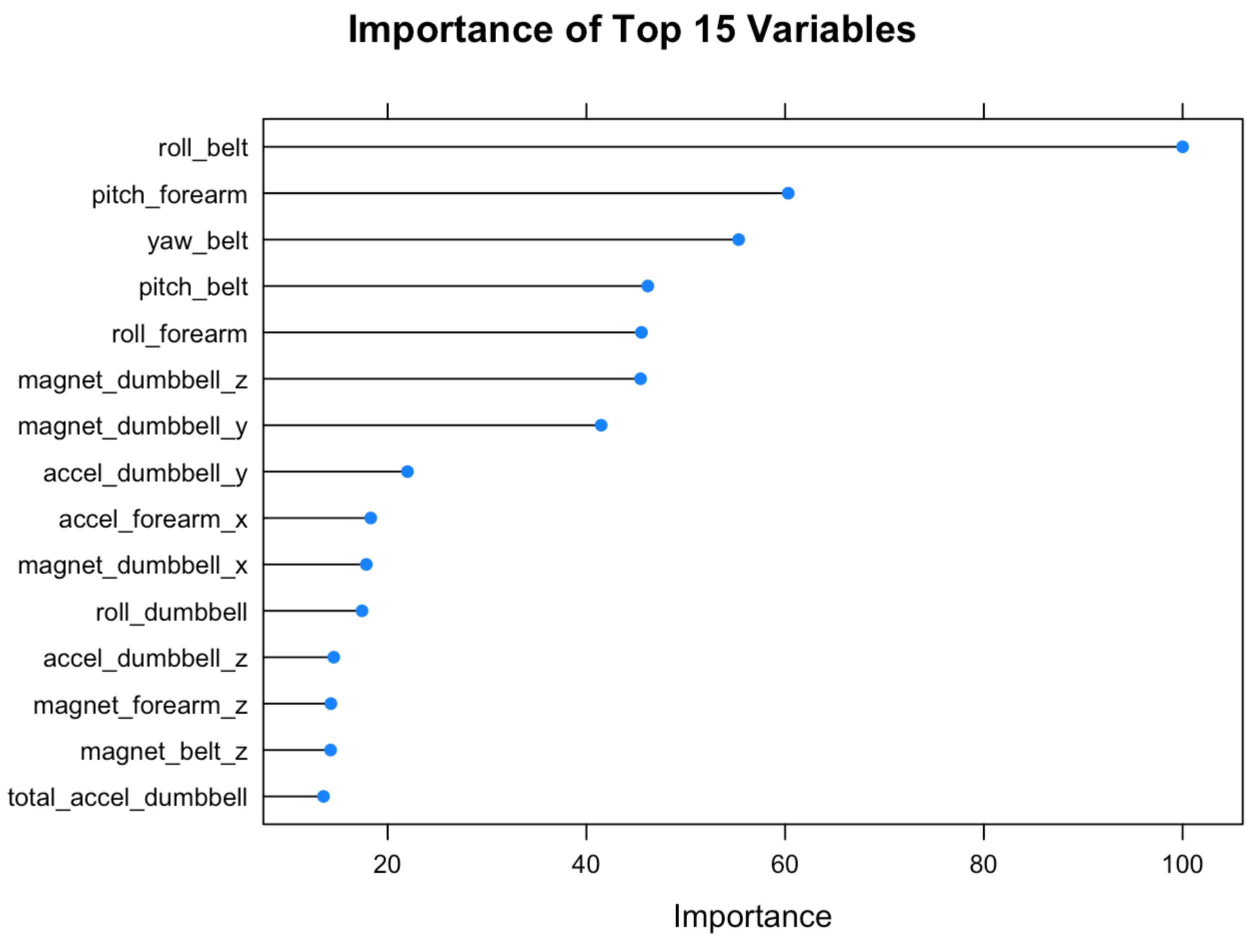
```
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=27
## - Fold1: mtry=27
## + Fold1: mtry=52
## - Fold1: mtry=52
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=27
## - Fold2: mtry=27
## + Fold2: mtry=52
## - Fold2: mtry=52
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=27
## - Fold3: mtry=27
## + Fold3: mtry=52
## - Fold3: mtry=52
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 27 on full training set
```

```
print(modFit, digits=3)
```

```
## Random Forest
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 7852, 7850, 7850
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    2    0.985     0.981
##   27    0.986     0.983
##   52    0.980     0.974
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The final model suggests greatest importance from the **roll_belt** and **pitch_frearm** variables, followed by **yaw_belt.**

```
#plot variable importance
varimp <- varImp(modFit)
plot(varimp, main = "Importance of Top 15 Variables", top = 15)
```



Model performance on test data was **99%**.

```
#test on reserved test set
pred <- predict(modFit, newdata=testset)
confMat <- confusionMatrix(pred, testset$classe)
print(confMat, digits=3)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2229   11    0    0    0
##          B    1 1498   14    2    0
##          C    0    9 1346   19    1
##          D    0    0    8 1264    3
##          E    2    0    0    1 1438
## 
## Overall Statistics
## 
##                Accuracy : 0.991
##                  95% CI : (0.989, 0.993)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
## 
##                   Kappa : 0.989
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.999    0.987    0.984    0.983    0.997
## Specificity             0.998    0.997    0.996    0.998    1.000
## Pos Pred Value          0.995    0.989    0.979    0.991    0.998
## Neg Pred Value          0.999    0.997    0.997    0.997    0.999
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.284    0.191    0.172    0.161    0.183
## Detection Prevalence    0.285    0.193    0.175    0.163    0.184
## Balanced Accuracy       0.998    0.992    0.990    0.991    0.998
```

Validation set

The final predictions on the validation cases were correctly identified.

```
finalPred <- predict(modFit, newdata = data_valid)
finalPred
```

```
##  [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
```

Reference

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.