Prediction Assignment Writeup

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.

Environment Setting

```
## free up memory
rm(list=ls())

## Loading required package
require(data.table)
install.packages('caret', dependencies = TRUE)
install.packages('corrplot', dependencies = TRUE)

## Loading library
library(knitr);
library(caret);
library(rpart);
library(ggplot2);
library(corrplot);
library(randomForest);
```

Data Preparation

```
## download data from URL
Url.Train <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
Url.Test  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

training <- read.csv(url(Url.Train))
testing  <- read.csv(url(Url.Test))

inTrain  <- createDataPartition(training$classe, p=0.7, list=FALSE)
```

```
TrainSet <- training[inTrain, ]
TestSet  <- training[-inTrain, ]
```

Data	
inTrain	int [1:13737, 1] 1 2 4 5 6 7 8 10 12 14 ...
testing	20 obs. of 160 variables
TestSet	5885 obs. of 160 variables
training	19622 obs. of 160 variables
TrainSet	13737 obs. of 160 variables

```
## View data
View(TrainSet)
View(TestSet)
```

Data Cleaning

```
## Remove variables with Nearly Zero Variance
NZV <- nearZeroVar(TrainSet)
TrainSet <- TrainSet[, -NZV]
TestSet  <- TestSet[, -NZV]

## Remove variables which are mostly NA
AllNA    <- sapply(TrainSet, function(x) mean(is.na(x))) > 0.95
TrainSet <- TrainSet[, AllNA==FALSE]
TestSet  <- TestSet[, AllNA==FALSE]

## Remove identification only variables (columns 1 to 5)
TrainSet <- TrainSet[, -(1:5)]
TestSet  <- TestSet[, -(1:5)]
```

Data	
inTrain	int [1:13737, 1] 1 2 4 5 6 7 8 10 12 14 ...
testing	20 obs. of 160 variables
TestSet	5885 obs. of 54 variables
training	19622 obs. of 160 variables
TrainSet	13737 obs. of 54 variables

View data

View(TrainSet)

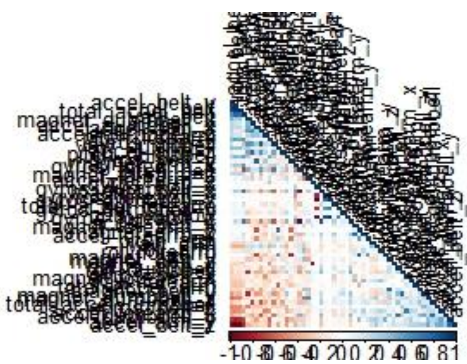
View(TestSet)

Training, testing & validation data

Data exploration

corMatrix <- cor(TrainSet[,-54])

corrplot(corMatrix, order = "FPC", method = "color", type = "lower", tl.cex = 0.8, tl.col = rgb(0, 0, 0))



Decision Trees

model_tree <- rpart(classe ~ ., data=TrainSet, method="class")

prediction_tree <- predict(model_tree, TestSet, type="class")

class_tree <- confusionMatrix(prediction_tree, TestSet$classe)

class_tree

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1529  257   46   96   56
         B   45  696   93   78  127
         C    9   56  830  111   69
         D   87  130   57  632  139
         E    4    0    0   47  691

Overall Statistics

               Accuracy : 0.7439
                 95% CI : (0.7326, 0.755)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6741
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9134   0.6111   0.8090   0.6556   0.6386
Specificity            0.8919   0.9277   0.9496   0.9161   0.9894
Pos Pred Value         0.7707   0.6699   0.7721   0.6048   0.9313
Neg Pred Value         0.9628   0.9086   0.9593   0.9314   0.9240
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2598   0.1183   0.1410   0.1074   0.1174
Detection Prevalence   0.3371   0.1766   0.1827   0.1776   0.1261
Balanced Accuracy      0.9027   0.7694   0.8793   0.7858   0.8140
```

```
## Random Forest

forest_model <- randomForest(classe ~ ., data=TrainSet, method="class")

prediction_forest <- predict(forest_model, TestSet, type="class")

random_forest <- confusionMatrix(prediction_forest, TestSet$classe)

random_forest
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1674    1    0    0    0
         B    0 1137    4    0    0
         C    0    1 1022    3    0
         D    0    0    0  961    3
         E    0    0    0    0 1079

Overall Statistics

               Accuracy : 0.998
                 95% CI : (0.9964, 0.9989)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9974
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9982   0.9961   0.9969   0.9972
Specificity            0.9998   0.9992   0.9992   0.9994   1.0000
Pos Pred Value         0.9994   0.9965   0.9961   0.9969   1.0000
Neg Pred Value         1.0000   0.9996   0.9992   0.9994   0.9994
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2845   0.1932   0.1737   0.1633   0.1833
Detection Prevalence   0.2846   0.1939   0.1743   0.1638   0.1833
Balanced Accuracy      0.9999   0.9987   0.9976   0.9981   0.9986
```

Data Prediction

Using Random Forest

prediction <- predict(forest_model, newdata=TestSet)

confusionMatrix(prediction, TestSet$classe)

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1674    1    0    0    0
         B    0 1137    4    0    0
         C    0    1 1022    3    0
         D    0    0    0  961    3
         E    0    0    0    0 1079

Overall Statistics

               Accuracy : 0.998
                 95% CI : (0.9964, 0.9989)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9974
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9982   0.9961   0.9969   0.9972
Specificity            0.9998   0.9992   0.9992   0.9994   1.0000
Pos Pred Value         0.9994   0.9965   0.9961   0.9969   1.0000
Neg Pred Value         1.0000   0.9996   0.9992   0.9994   0.9994
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2845   0.1932   0.1737   0.1633   0.1833
Detection Prevalence   0.2846   0.1939   0.1743   0.1638   0.1833
Balanced Accuracy      0.9999   0.9987   0.9976   0.9981   0.9986
>
```

prediction

```
 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
```

Other

sessionInfo()

```
R version 3.3.2 (2016-10-31)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

locale:
[1] LC_COLLATE=English_United States.1252  LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] corrplot_0.77       randomForest_4.6-12 caret_6.0-76        lattice_0.20-34     data.table_1.10.4
[6] ggplot2_2.2.1       rpart_4.1-10        knitr_1.15.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.10       magrittr_1.5       splines_3.3.2      MASS_7.3-45        munsell_0.4.3
 [6] colorspace_1.3-2   foreach_1.4.3      minqa_1.2.4        stringr_1.1.0      car_2.1-4
[11] plyr_1.8.4         tools_3.3.2        parallel_3.3.2     pbkrtest_0.4-7     nnet_7.3-12
[16] grid_3.3.2         gtable_0.2.0       nlme_3.1-128       mgcv_1.8-15        quantreg_5.29
[21] e1071_1.6-8        class_7.3-14       MatrixModels_0.4-1 iterators_1.0.8    lme4_1.1-13
[26] lazyeval_0.2.0     assertthat_0.1     tibble_1.2         Matrix_1.2-7.1     nloptr_1.0.4
[31] reshape2_1.4.2     ModelMetrics_1.1.0 codetools_0.2-15   stringi_1.1.2      scales_0.4.1
[36] stats4_3.3.2       SparseM_1.76
```