Quantified Self Movement Activity Recognition Model

11/23/2014

```
## Loading required package: lattice
## Loading required package: ggplot2
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

Executive Summary

Using devices such as Jawbone Up, Nike Fuel-band, and Fit-bit 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. This report was to uses the data from belt, forearm, arm, and dumbbell accelerometers of six participants to predict how well they were doing the exercises.

Input and Cleaning of Data

The two files containing the training and test data were downloaded from: https://d396qusza40orc.cloudfront.net/predmachlearn/

Missing values (NA,empty,blank) and the first eight columns that acted as identifiers (e.g. name, timestamps, ...) are then removed.

```
# options
opts_chunk$set(cache = FALSE)
opts_knit$set(root.dir=normalizePath('.'))

# csv file for training loaded here
training_data <- read.csv(training_file_local, na.strings= c("NA","","  "))

# clean the data by removing columns with NAs, empty strings and blanks
training_data_na <- apply(training_data, 2, function(x) {sum(is.na(x))})
training_data_cleaned <- training_data[,which(training_data_na == 0)]

# first eight columns that acted as identifiers: i.e. name, timestamps ...
training_data_cleaned <- training_data_cleaned[8:length(training_data_cleaned)]
```

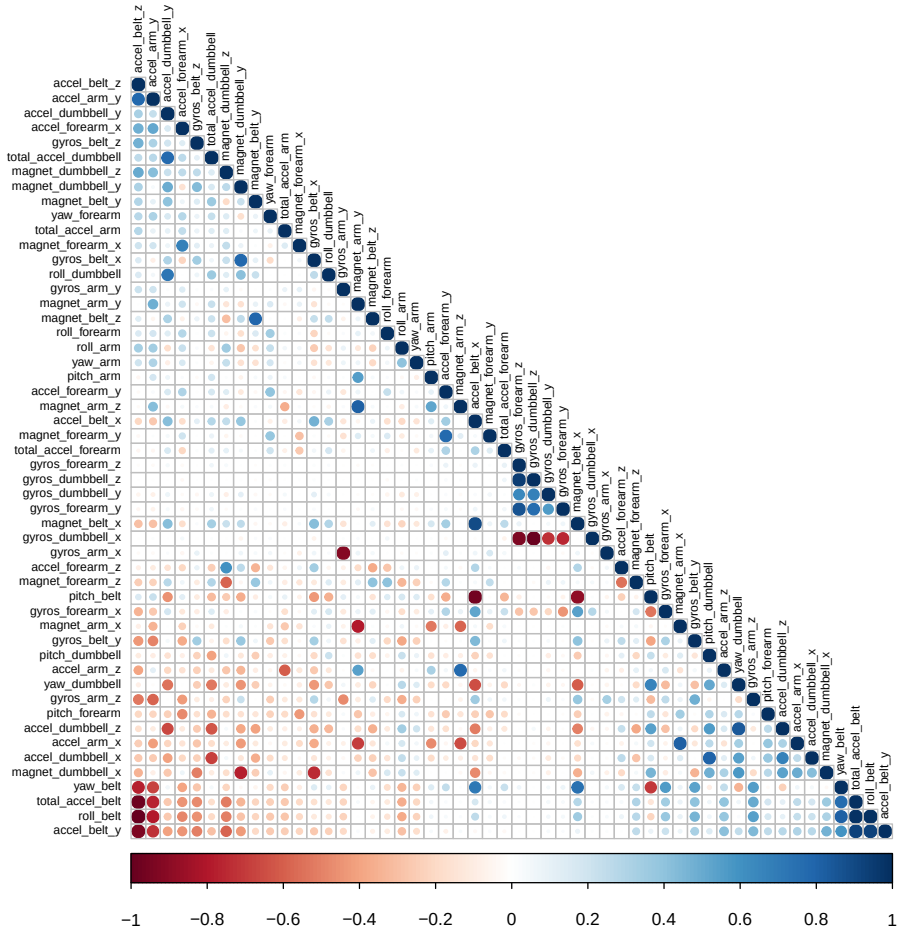
Model Fit and Cross Validation

The test data set was split up into training and cross validation sets in a 75:25 ratio in order to train the model and then test the model fit using data that it was not fitted to.

```
# split the cleaned testing data into training and cross validation
inTrain <- createDataPartition(y = training_data_cleaned$classe, p = 0.75, list = FALSE)
training_data <- training_data_cleaned[inTrain, ]
crossvalidation_data <- training_data_cleaned[-inTrain, ]
```

A random forest model was selected to predict the classification because it known to produce good results for this type of data. A correlation plot was produced in order to see how strong the variables relate to each other to access if keeping all the predictors makes sense.

```
# correlation matrix plot
correlationMatrix <- cor(training_data[, -length(training_data)])
corrplot(correlationMatrix, order = "FPC", method = "circle", type = "lower", tl.cex = 0.6,  tl.col = rg
```



The dark red and blue colors indicate a highly negative and positive relationship respectively between the variables. There doesn't appear to be highly correlated predictors which means that all of them can be included in the model.

Model was then fitted with the outcome set to "classe" with all the other variables used to fit the model.

```
# fit a model to predict the classe using everything else as a predictor
modelFit <- randomForest(classe ~ ., data = training_data)
modelFit
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training_data)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.45%
```

```
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4181    3    0    0    1   0.0009558
## B   13 2832    3    0    0   0.0056180
## C    0   15 2549    3    0   0.0070121
## D    0    0   20 2390    2   0.0091211
## E    0    0    2    4 2700   0.0022173
```

The model produced a OOB error rate of .45% which seemed low enough to progress into testing phase.

The model was then used to classify the remaining 25% of data. The results were placed in a confusion matrix along with the actual classifications in order to determine the accuracy of the model.

```
# crossvalidate the model using the remaining 25% of data
predictCrossValidation <- predict(modelFit, crossvalidation_data)
confusionMatrix(crossvalidation_data$classe, predictCrossValidation)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1389    6    0    0    0
##          B    2  946    1    0    0
##          C    0    3  852    0    0
##          D    0    0    9  795    0
##          E    0    0    0    6  895
##
## Overall Statistics
##
##                Accuracy : 0.994
##                  95% CI : (0.992, 0.996)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.993
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.999    0.991    0.988    0.993    1.000
## Specificity             0.998    0.999    0.999    0.998    0.999
## Pos Pred Value          0.996    0.997    0.996    0.989    0.993
## Neg Pred Value          0.999    0.998    0.998    0.999    1.000
## Prevalence              0.284    0.195    0.176    0.163    0.183
## Detection Rate          0.283    0.193    0.174    0.162    0.183
## Detection Prevalence    0.284    0.194    0.174    0.164    0.184
## Balanced Accuracy       0.998    0.995    0.994    0.995    0.999
```

This model yielded a 99.4% prediction accuracy. This model appears adequate to predict new data.

Predictions on Test Data

A separate data set was loaded and cleaned in the same manner as training data. The model fit was then used to predict the classifications of the 20 results of this new data.

```
# apply the same treatment to test data
test_data <- read.csv(testing_file_local, na.strings= c("NA","","  "))
test_data_na <- apply(test_data, 2, function(x) {sum(is.na(x))})
test_data_cleaned <- test_data[,which(test_data_na == 0)]
test_data_cleaned <- test_data_cleaned[8:length(test_data_cleaned)]

# predict the classes of the test set
predicted <- predict(modelFit, test_data_cleaned)
predicted
```

```
##  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
```

Summary

Devices such as Jawbone Up, Nike Fuel-band, and Fit-bit collect a large amount of data that can be used to accurately predict how well a person is preforming an exercise using a properly trained Machine Learning model.