Practical Machine Learning Project

Practical Machine Learning Course Project Report

Background

The use of devices such as Jawbone Up, Nike FuelBand, and Fitbit is increasing amongst those who are involved in quantified self movement. These people regularly quantify particular activities. In this project, the goal is to use data from belt, forearm, arm, accelerometers and dumbbells. 6 participants were included, and were required to perform barbell lifts correctly and incorrectly in 5 different ways.

Data Sources

The training data for this project is available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)
The test data is available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)
The data for this project comes from this original source: http://groupware.les.inf.puc-rio.br/har
(http://groupware.les.inf.puc-rio.br/har).

Objective

The goal of this project is to predict the manner in which they did the exercise.

```
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(caret)
```

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

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

```
library(rpart)
library(rpart.plot)
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:rattle':
##
##     importance
```

```
library(RColorBrewer)
```

```
set.seed(56789)
```

Data import

Datasets are downloaded as follows:

```
trainUrl <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trainFile <- "./data/pml-training.csv"
testFile  <- "./data/pml-testing.csv"
if (!file.exists("./data")) {
  dir.create("./data")
}
if (!file.exists(trainFile)) {
  download.file(trainUrl, destfile = trainFile, method = "curl")
}
if (!file.exists(testFile)) {
  download.file(testUrl, destfile = testFile, method = "curl")
}
rm(trainUrl)
rm(testUrl)
```

Data reading

```
trainRaw <- read.csv(trainFile)
testRaw <- read.csv(testFile)
dim(trainRaw)
```

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

```
dim(testRaw)
```

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

```
rm(trainFile)
rm(testFile)
```

Data cleaning

1. Exclude near zero variance variables.

```
NZV <- nearZeroVar(trainRaw, saveMetrics = TRUE)
head(NZV, 20)
```

```
##                         freqRatio percentUnique zeroVar   nzv
## X                        1.000000  100.00000000   FALSE FALSE
## user_name                1.100679    0.03057792   FALSE FALSE
## raw_timestamp_part_1     1.000000    4.26562022   FALSE FALSE
## raw_timestamp_part_2     1.000000   85.53154622   FALSE FALSE
## cvtd_timestamp           1.000668    0.10192641   FALSE FALSE
## new_window              47.330049    0.01019264   FALSE  TRUE
## num_window               1.000000    4.37264295   FALSE FALSE
## roll_belt                1.101904    6.77810621   FALSE FALSE
## pitch_belt               1.036082    9.37722964   FALSE FALSE
## yaw_belt                 1.058480    9.97349913   FALSE FALSE
## total_accel_belt         1.063160    0.14779329   FALSE FALSE
## kurtosis_roll_belt    1921.600000    2.02323922   FALSE  TRUE
## kurtosis_picth_belt    600.500000    1.61553358   FALSE  TRUE
## kurtosis_yaw_belt       47.330049    0.01019264   FALSE  TRUE
## skewness_roll_belt    2135.111111    2.01304658   FALSE  TRUE
## skewness_roll_belt.1   600.500000    1.72255631   FALSE  TRUE
## skewness_yaw_belt       47.330049    0.01019264   FALSE  TRUE
## max_roll_belt            1.000000    0.99378249   FALSE FALSE
## max_picth_belt           1.538462    0.11211905   FALSE FALSE
## max_yaw_belt           640.533333    0.34654979   FALSE  TRUE
```

```
training01 <- trainRaw[, !NZV$nzv]
testing01 <- testRaw[, !NZV$nzv]
dim(training01)
```

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

```
dim(testing01)
```

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

```
rm(trainRaw)
rm(testRaw)
rm(NZV)
```

2. Non applicable variables removed.

```
regex <- grepl("^X|timestamp|user_name", names(training01))
training <- training01[, !regex]
testing <- testing01[, !regex]
rm(regex)
rm(training01)
rm(testing01)
dim(training)
```

```
## [1] 19622    95
```

```
dim(testing)
```

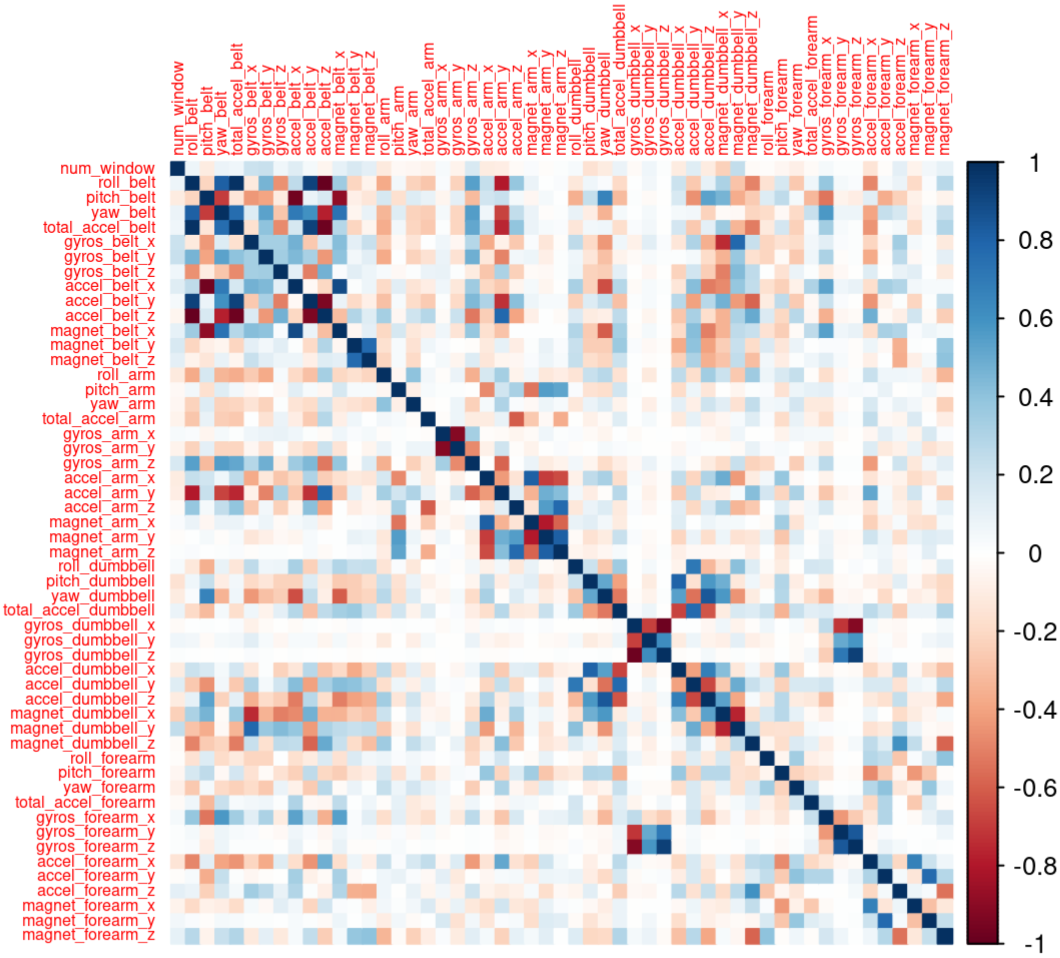
```
## [1] 20 95
```

3. Remove NAs

```
cond <- (colSums(is.na(training)) == 0)
training <- training[, cond]
testing <- testing[, cond]
rm(cond)
```

Correlation plot of training dataset

```
corrplot(cor(training[, -length(names(training))]), method = "color", tl.cex = 0.5)
```



Partitioning training dataset

Data split into training set (70%) and validation set (30%).

```
set.seed(56789) # For reproducibile purpose
inTrain <- createDataPartition(training$classe, p = 0.70, list = FALSE)
validation <- training[-inTrain, ]
training <- training[inTrain, ]
rm(inTrain)
```

Datasets now include:

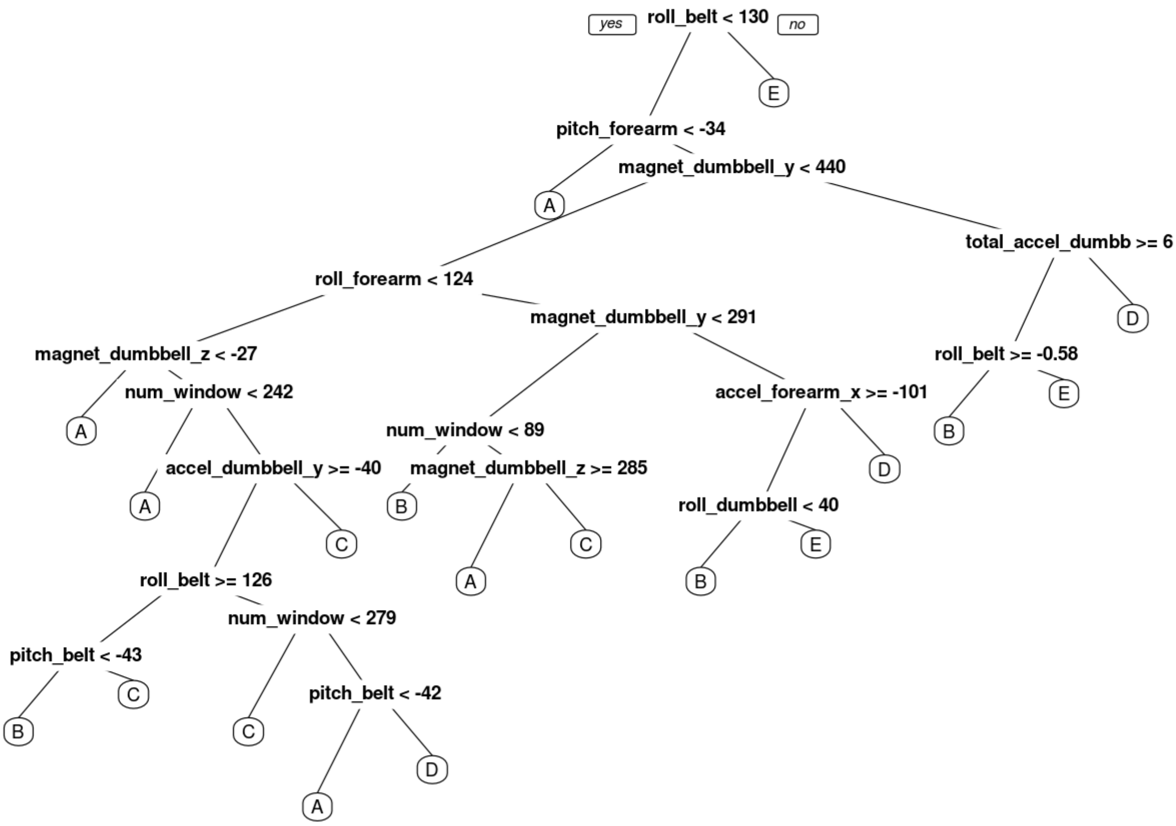
1. Training Data: 13737 observations.
2. Validation Data: 5885 observations.
3. Testing Data: 20 observations.

Data modelling

Decision Tree

A predictive model was developed in order to identify activity.

```
modelTree <- rpart(classe ~ ., data = training, method = "class")
prp(modelTree)
```



Model performance on the validation dataset.

```
predictTree <- predict(modelTree, validation, type = "class")
confusionMatrix(predictTree,as.factor(validation$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1492  270   55  116   84
##          B   37  551   32   17   89
##          C   10  120  818  117   61
##          D   84  134   49  655  140
##          E   51   64   72   59  708
##
## Overall Statistics
##
##                Accuracy : 0.7178
##                  95% CI : (0.7061, 0.7292)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6409
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8913  0.48376   0.7973   0.6795   0.6543
## Specificity            0.8753  0.96313   0.9366   0.9173   0.9488
## Pos Pred Value         0.7397  0.75895   0.7265   0.6168   0.7421
## Neg Pred Value         0.9529  0.88602   0.9563   0.9359   0.9242
## Prevalence             0.2845  0.19354   0.1743   0.1638   0.1839
## Detection Rate         0.2535  0.09363   0.1390   0.1113   0.1203
## Detection Prevalence   0.3427  0.12336   0.1913   0.1805   0.1621
## Balanced Accuracy      0.8833  0.72344   0.8669   0.7984   0.8016
```

```
accuracy <- postResample(predictTree, as.numeric(validation$classe))
rm(predictTree)
rm(modelTree)
```

The Estimated accuracy is `r accuracy[1]*100%

Random Forest

A predictive model was devised using the Random Forest algorithm, with a 5-fold cross validation

```
modelRF <- train(classe ~ ., data = training, method = "rf", trControl = trainControl(method
 = "cv", 5), ntree = 50)
modelRF
```

```
## Random Forest
##
## 13737 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10988, 10990, 10991, 10990, 10989
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9917018  0.9895015
##   27    0.9965791  0.9956728
##   53    0.9941038  0.9925418
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The model as tested on the validation dataset.

```
predictRF <- predict(modelRF, validation)
confusionMatrix(predictRF,as.factor(validation$classe))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    2    0    0    0
##          B    0 1134    1    0    0
##          C    0    2 1025    0    0
##          D    0    1    0  964    1
##          E    0    0    0    0 1081
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9988          
##                  95% CI : (0.9976, 0.9995)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9985          
##                                           
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9956   0.9990   1.0000   0.9991
## Specificity            0.9995   0.9998   0.9996   0.9996   1.0000
## Pos Pred Value         0.9988   0.9991   0.9981   0.9979   1.0000
## Neg Pred Value         1.0000   0.9989   0.9998   1.0000   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1927   0.1742   0.1638   0.1837
## Detection Prevalence   0.2848   0.1929   0.1745   0.1641   0.1837
## Balanced Accuracy      0.9998   0.9977   0.9993   0.9998   0.9995
```

```
accuracy <- postResample(predictRF, as.numeric(validation$classe))
rm(predictRF)
```

The model accuracy is estimated as NA%