Prediction Assignment Writeup

Muhamad Noorazizi

12/22/2020

Overview:

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, the 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://web.archive.org/web/20161224072740/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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har.

Preparing Libraries

```r
set.seed(9397)
library("caret")
library("lattice")
library("ggplot2")
library("rpart")
library("randomForest")
library("RColorBrewer")
library("rattle")
library("rpart.plot")
library("corrplot")
library("e1071")
library("gbm")
```

Configure Parallel Processing

```r
library("parallel")
library("doParallel")
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)
```

Preparing Data

```r
#train <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
train_csv<-"C:/Users/azizi/Desktop/Prediction Assignment Writeup/pml-traininig.csv"

#test <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
test_csv<-"C:/Users/azizi/Desktop/Prediction Assignment Writeup/pml-testing.csv"

# Downloading the Datasets
if(!file.exists(train_csv))
{
    download.file(train, destfile = train_csv)
}
train_data <- read.csv(train_csv, na.strings = c("", "NA"))

if(!file.exists(test_csv))
{
    download.file(test, destfile = test_csv)
}
test_data <- read.csv(test_csv, na.strings = c("", "NA"))

# Creating a Partition of the training dataset with a ratio 70 to 30
inTrain  <- createDataPartition(train_data$classe, p=0.7, list=FALSE)

train_set <- train_data[inTrain, ]
test_set <- train_data[-inTrain, ]

dim(train_set)
```

```
## [1] 13737   160
```

```r
dim(test_set)
```

```
## [1] 5885  160
```

Cleaning Data

```r
# Removing near-zero-variance (NZV) variables
nzv <- nearZeroVar(train_set)

train_set <- train_set[ , -nzv]
test_set <- test_set[ , -nzv]

dim(train_set)
```

```
## [1] 13737   122
```

```r
dim(test_set)
```

```
## [1] 5885  122
```

```r
# Removing NA values with a 95% threshold
nav <- sapply(train_set, function(x) mean(is.na(x))) > 0.95

train_set <- train_set[ , nav == FALSE]
test_set <- test_set[ , nav == FALSE]

dim(train_set)
```

```
## [1] 13737    59
```

```r
dim(test_set)
```

```
## [1] 5885   59
```

```r
# Removing identification variables in column 1 to 5
train_set <- train_set[, -(1:5)]
test_set <- test_set[, -(1:5)]

dim(train_set)
```

```
## [1] 13737    54
```

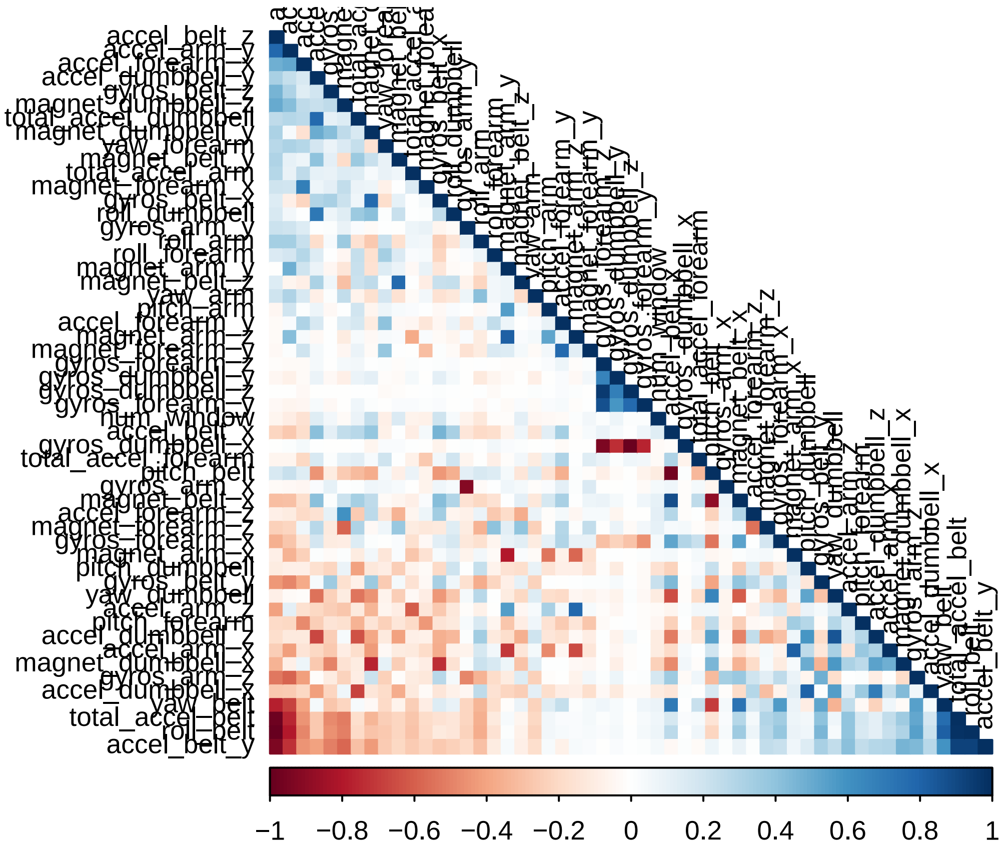
```r
dim(test_set)
```

```
## [1] 5885   54
```

Pre-processing

```r
# Correlation Analysis

ca <- cor(train_set[, -54])
corrplot(ca, order = "FPC", method = "color", type = "lower",
         tl.cex = 0.8, tl.col = rgb(0, 0, 0))
```



Prediction Model

```r
# Model 1 - Random Forest

rf_control <- trainControl(method = "cv", number = 5, allowParallel=TRUE)
rf <- train(classe ~ ., data = train_set, method = "rf", trControl = rf_control, verbose = FALSE)
rf$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, verbose = FALSE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.27%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3904    1    0    0    1 0.0005120328
## B    6 2650    2    0    0 0.0030097818
## C    0    3 2392    1    0 0.0016694491
## D    0    0   14 2238    0 0.0062166963
## E    0    1    0    8 2516 0.0035643564
```

```r
# Confusion Matrix
```

```
confusionMatrix(rf)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 28.4  0.1  0.0  0.0  0.0
##          B  0.0 19.3  0.1  0.0  0.0
##          C  0.0  0.0 17.4  0.1  0.0
##          D  0.0  0.0  0.0 16.3  0.1
##          E  0.0  0.0  0.0  0.0 18.3
##
##  Accuracy (average) : 0.9967
```

```
# Model 2 - Linear Discriminant Analysis

lda_control <- trainControl(method = "cv", number = 5, allowParallel=TRUE)
lda <- train(classe ~ ., data = train_set, method = "lda", trControl = lda_control, verbose = FALSE)

# Confusion Matrix

confusionMatrix(lda)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 23.5  2.8  1.7  0.9  0.7
##          B  0.7 12.6  1.7  0.7  2.8
##          C  1.9  2.3 11.6  2.1  1.7
##          D  2.1  0.8  1.9 12.0  1.7
##          E  0.2  0.9  0.5  0.6 11.5
##
##  Accuracy (average) : 0.7125
```

```
# Model 3 - Generalized Boosted Model (GBM)

gbm_control <- trainControl(method = "cv", number = 5, allowParallel=TRUE)
gbm <- train(classe ~ ., data = train_set, method = "gbm", trControl = gbm_control, verbose = FALSE)

# Confusion Matrix
confusionMatrix(gbm)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 28.4  0.2  0.0  0.0  0.0
##          B  0.1 18.9  0.2  0.1  0.1
##           C  0.0  0.2 17.2  0.2  0.0
##           D  0.0  0.0  0.1 16.1  0.2
##           E  0.0  0.0  0.0  0.0 18.1
##
##  Accuracy (average) : 0.9873
```

Results of the three model

Model 1 : Random Forest (Accuracy = 0.9967) . . . Model 2 : Linear Discriminant Analysis (Accuracy 0.7125) . . . Model 3 : Generalized Boosted Model (Accuracy 0.9873)

As we can observe from the selected model above, Random Forest gave the highest accuracy. Therefore, we will use Random Forest to predict our Original Test Data.

Using Random Forest Model to Predict Test Data

```r
pred_given_test_data <- predict(rf, newdata = test_data)
pred_given_test_data
```

```
##  [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
```

Stopping Clustering

```r
stopCluster(cluster)
registerDoSEQ()
```