Prediction Assignment Writeup

Keng Yew Hoe

10/16/2020

Introduction

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. The goal of this project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set.

Loading the data

```
# Downloading the data
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(trainUrl, destfile = "./train.csv", method = "curl")
download.file(testUrl, destfile = "./test.csv", method = "curl")
# Reading the data and filling missing data
train <- read.csv("./train.csv",na.strings = c("NA", "#DIV/0!"))
test <- read.csv("./test.csv",na.strings = c("NA", "#DIV/0!"))
```

Reading the data

```
# Getting the dimensions of the dataframe
dim(train)
```

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

```
dim(test)
```

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

Cleaning data

Columns with mostly missing data are removed from the dataset. The first 7 columns are also remove as they are not needed in preparing the model.

```
# Remove variables with near zero variance
train<-train[,colSums(is.na(train)) == 0]
test <-test[,colSums(is.na(test)) == 0]

# Remove the first 7 columns which are not used in the prediction model
train   <-train[,-c(1:7)]
test <-test[,-c(1:7)]

# Dimensions of the data after cleaning
dim(train)
```

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

```
dim(test)
```

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

The cleaned training data now has 19622 rows and 53 columns while the cleaned test data has 20 rows and 53 columns.

Subsetting the training data

To build our model, we subset our training data to sub training data (75%) and sub test data (25%) for cross validation purpose. This also gets us the out-of-sample errors.

```
# Split the training data in training (75%) and testing (25%) data subset
library(caret)
inTrain <- createDataPartition(y=train$classe, p=0.75, list=FALSE)
subTrain <- train[inTrain, ]
subTest <- train[-inTrain, ]
dim(subTrain)
```

```
## [1] 14718    53
```

```
dim(subTest)
```

```
## [1] 4904   53
```

The sub training data has 14718 rows which is rougly 75% of the original training data.

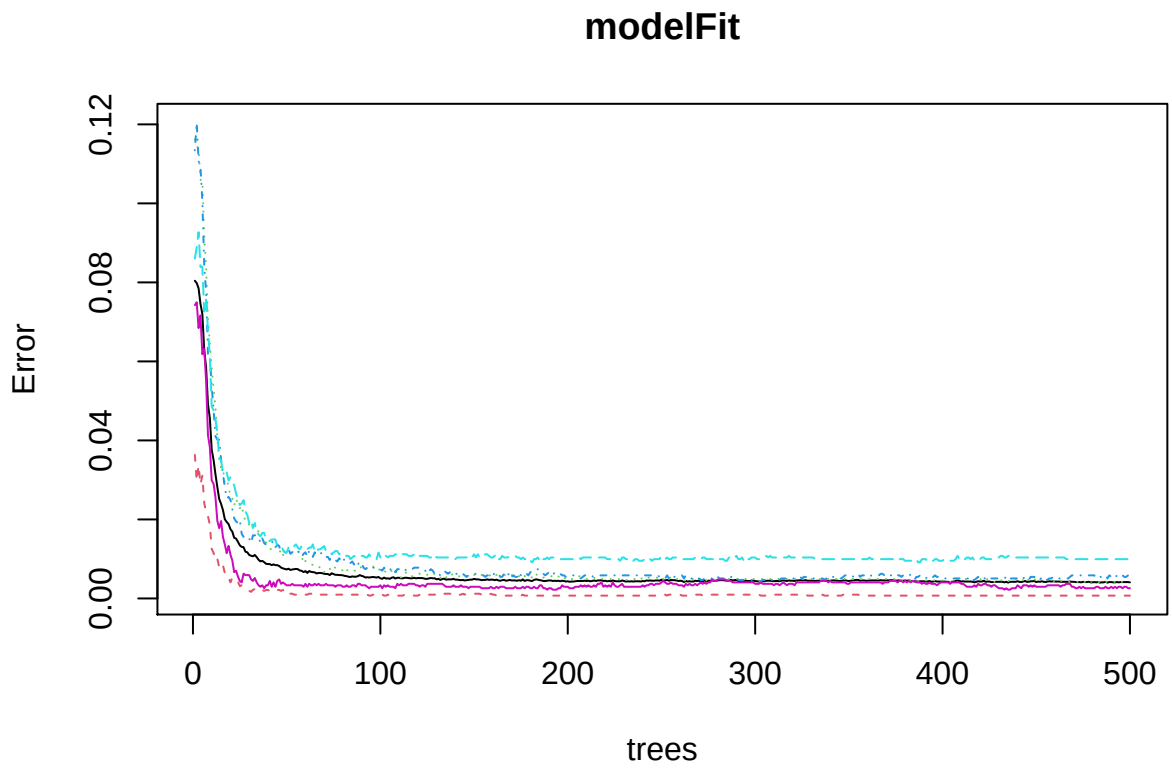
Building the Prediction Models

```
# Random Forest
library(rpart)
library(randomForest)
set.seed(555)
```

```
rfGrid <-  expand.grid(interaction.depth = c(1, 5, 9),
                       n.trees = (1:30)*50,
                       shrinkage = 0.1)
subTrain$classe <- as.factor(subTrain$classe)
modelFit <- randomForest(classe ~ .,  subTrain, tuneGrid = rfGrid)
print(modelFit)
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = subTrain, tuneGrid = rfGrid)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.41%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4182    3    0    0    0 0.0007168459
## B    7 2836    5    0    0 0.0042134831
## C    0   12 2553    2    0 0.0054538372
## D    0    0   23 2388    1 0.0099502488
## E    0    0    1    6 2699 0.0025868441
```

```
plot(modelFit)
```



This model has a very low classification error in all classes with errors all close to 0%.

Cross validation

```
knitr::opts_chunk$set(comment = NA)
# Testing the sub test data
predictions <- predict(modelFit, newdata = subTest)
subTest$classe <- as.factor(subTest$classe)
confusionMatrix(predictions, subTest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    2    0    0    0
##          B    0  946    2    0    0
##          C    0    1  852    6    0
##          D    0    0    1  797    0
##          E    0    0    0    1  901
##
## Overall Statistics
##
##                Accuracy : 0.9973
##                  95% CI : (0.9955, 0.9986)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9966
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9968   0.9965   0.9913   1.0000
## Specificity            0.9994   0.9995   0.9983   0.9998   0.9998
## Pos Pred Value         0.9986   0.9979   0.9919   0.9987   0.9989
## Neg Pred Value         1.0000   0.9992   0.9993   0.9983   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1929   0.1737   0.1625   0.1837
## Detection Prevalence   0.2849   0.1933   0.1752   0.1627   0.1839
## Balanced Accuracy      0.9997   0.9982   0.9974   0.9955   0.9999
```

The model is accepted because it has a high global accuracy of 0..9949 and Kappa of 0.9936 with high sensitivity and specificity for all cases

Testing the model

```
# Test validation sample
answers <- predict(modelFit, newdata = test, type = "response")
print(answers)
```

```
 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
```

The model succesfully predicted all 20 questions accurately with a score of 100% at the PML submission page