Practical Machine Learning Prediction Assignment

Melvin Yap

Executive Summary

Given the training dataset, the objective of this assignment is to develop a predictive model for target variable **classe** that represents one of the 5 ways that the barbell lifts are performed by the participants. The model is then tested on the test dataset to determine its accuracy.

Data Munging

The training and test datasets are first downloaded and loaded as data frames.

```
library(RCurl)
```

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

```
urlTrain <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
urlTest <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
x <- getURL(urlTrain)
y <- getURL(urlTest)
dfTrain <- as.data.frame(read.csv(textConnection(x)))
dfTest <- as.data.frame(read.csv(textConnection(y)))
```

```
#Take note of the number of columns of the datasets
dim(dfTrain)
```

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

```
dim(dfTest)
```

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

Columns with near-zero variances are then removed as they are not only non-informative, but may also affect the accuracy of the model.

```
library(caret)
```

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

```
nzv_cols <- nearZeroVar(dfTrain)
if(length(nzv_cols) > 0) dfTrain <- dfTrain[, -nzv_cols]
nzv_cols <- nearZeroVar(dfTest)
if(length(nzv_cols) > 0) dfTest <- dfTest[, -nzv_cols]
```

In addition, columns with irrelevant data, e.g. IDs, participant names and timestamps are removed from the dataset as they do not value-add to the predictive model. In both training and testing datasets, the first six columns have been identified to be irrelevant and are removed.

```
dfTrain <- dfTrain[7:length(dfTrain)]
dfTest <- dfTest[7:length(dfTest)]
```

Lastly, columns with *NA* values are removed.

```
trainNAs <- apply(dfTrain, 2, function(x) {sum(is.na(x))})
dfTrain <- dfTrain[,which(trainNAs == 0)]
testNAs <- apply(dfTest, 2, function(x) {sum(is.na(x))})
dfTest <- dfTest[,which(testNAs == 0)]
```

```
#Take note of the trimmed number of columns of the datasets
dim(dfTrain)
```

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

```
dim(dfTest)
```

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

Partitioning the Data

The training data is partitioned into both training (70%) and cross-validation (30%) datasets. The purpose of partitioning is to train the chosen model and validate it against the data it was not specifically fitted to in order to determine the model's accuracy.

```
set.seed(1234)
idxTrain <- createDataPartition(y=dfTrain$classe, p=0.7, list=F)
partTrain <- dfTrain[idxTrain,]
partValidate <- dfTrain[-idxTrain,]
```

Model Development

Selecting the Model

Random Forest is an ensemble of decision trees and is selected for its ability in finding a natural balance in biased datasets without much tuning, and yet produce fairly strong predictions.

Fitting the Model

The Random Forest model is first fitted with the partitioned training data, and

```
library(randomForest)
```

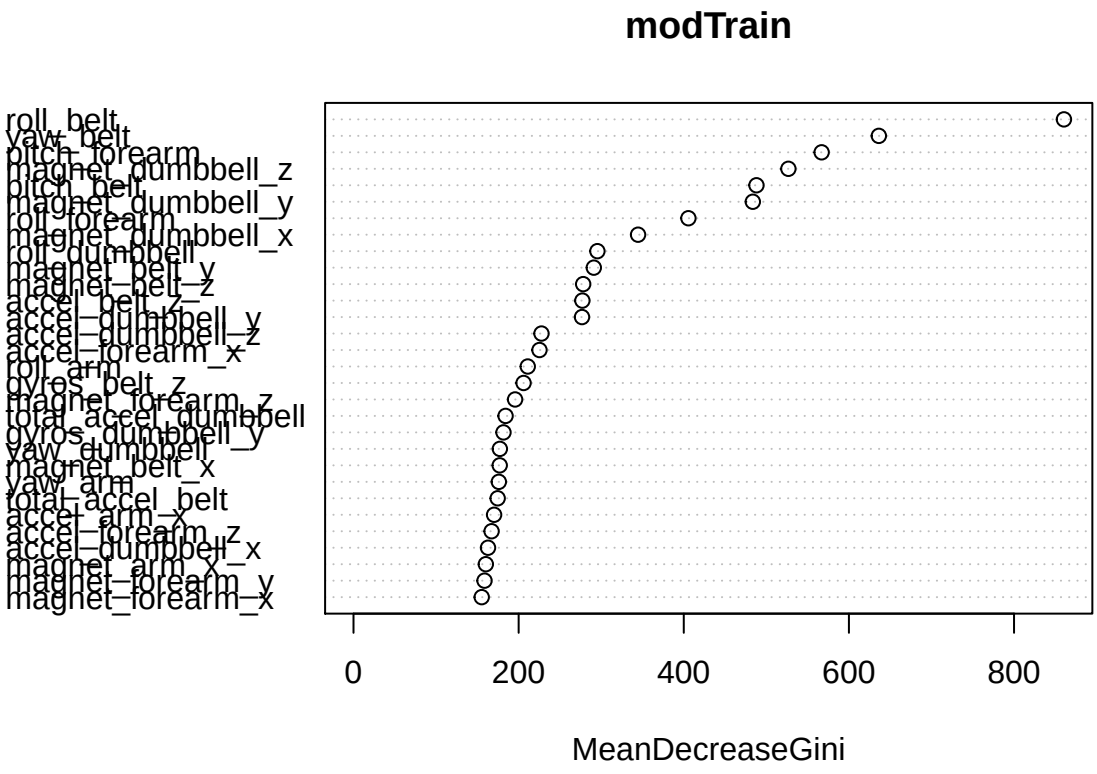
```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
modTrain <- randomForest(classe~., data=partTrain)
```

Determining the Out-of-Sample Error and Variable Importance

From the output of the training model *modTrain*, it is observed that the out-of-sample error is represented by the out-of-bag (OOB) error rate of 0.51%, which is small. The following plot identifies the variables in order of their importance towards the predictive function. **Roll-belt** is listed as the most important predictor.

```
varImpPlot(modTrain)
```



Cross-Validation

The model is used to test against the 30% validation dataset, and the respective **classe** variables are compared to determine its accuracy.

```
predictValidate <- predict(modTrain, partValidate)
conMatrix <- confusionMatrix(partValidate$classe, predictValidate)
conMatrix$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9966015      0.9957006      0.9947562      0.9979229      0.2858114
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

It is noted that the model has an accuracy of **99.66%** within the 95% confidence interval.

Testing the Model

The model is now applied to the test dataset and the following prediction result is obtained, which fits the expected results of this assignment.

```
predictTest <- predict(modTrain, dfTest)
predictTest #answers for submission of 20 files
```

```
##  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
```