Data Analysis of Weight Lifting Exercises Dataset

Introduction

The purpose of the analysis is to analyze and predict how well an activity is performed by the wearer. The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. Six young health participants are asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Data are collected from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. We would like to build machine learning algorithm to predict classe according to the predictors.

Data Processing

First we load the data.

```
data <- read.csv("pml-training.csv")
dim(data)
```

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

```
names(data)[1:5]
```

```
## [1] "X"                    "user_name"            "raw_timestamp_part_1"
## [4] "raw_timestamp_part_2" "cvtd_timestamp"
```

The first 5 variables in the dataset record the ID of the observation, time stamp and name of the participant. As this is not a time series prediction, we regard these variables for information only and will exclude them in our analysis.

Except the factorial variable “classe”, all other variables are numeric. However, we notice that there are a lot of missing values in the dataset. This may be because the variables are only applicable to certain classe while not to the others. Since the variables are all numeric, we simply replace missing values with 0.

The dataset is split into a training set training set (75%) and a testing set (25%).

```
library(caret)
```

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

```
data <- data[,-5:-1]
data[, 2:154] <- sapply(data[, 2:154], as.numeric)
data[is.na(data)] <- 0
inTrain <- createDataPartition(y=data$classe, p=0.75, list=FALSE)
training <- data[inTrain, ]
testing <- data[-inTrain, ]
```

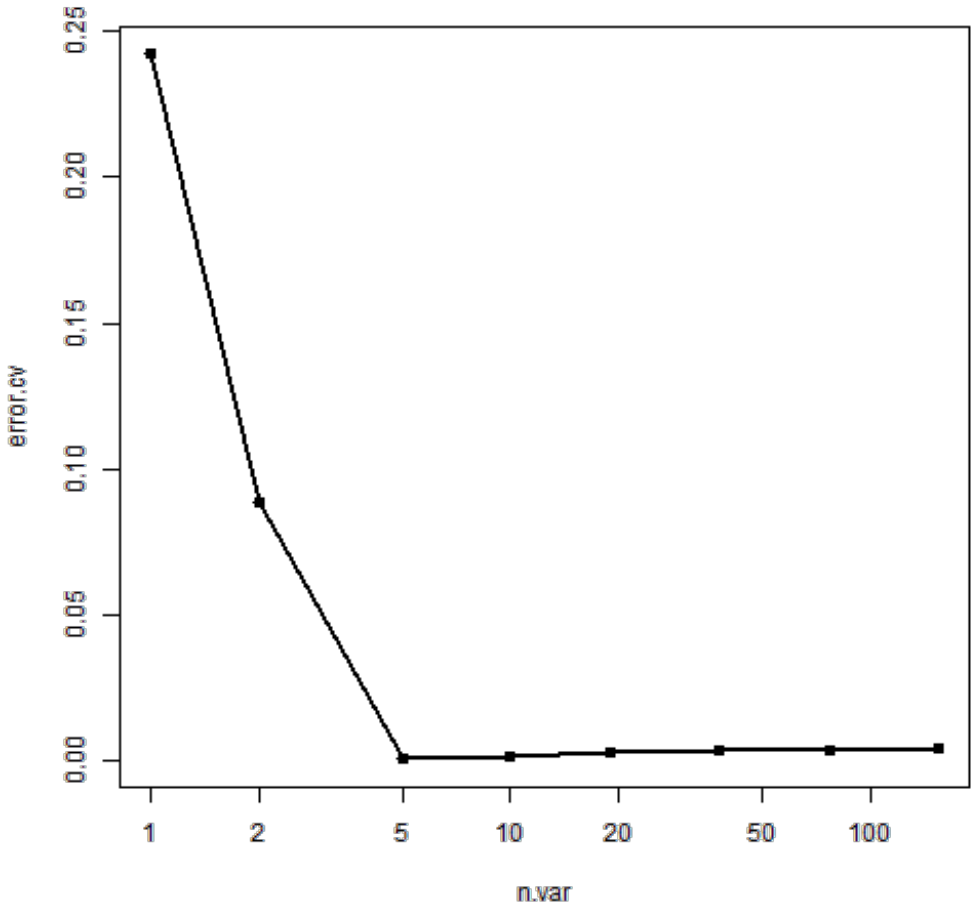
Machine Learning Model

This is a multi-class classification problem. We use out-of-bag random forest model to build our machine learning algorithm. There are 154 predictors in total. If all predictors are used, the model may overfit. Therefore, we would like to select the most important features via cross validation. Reducing the number of variables in the model also helps model interpretation.

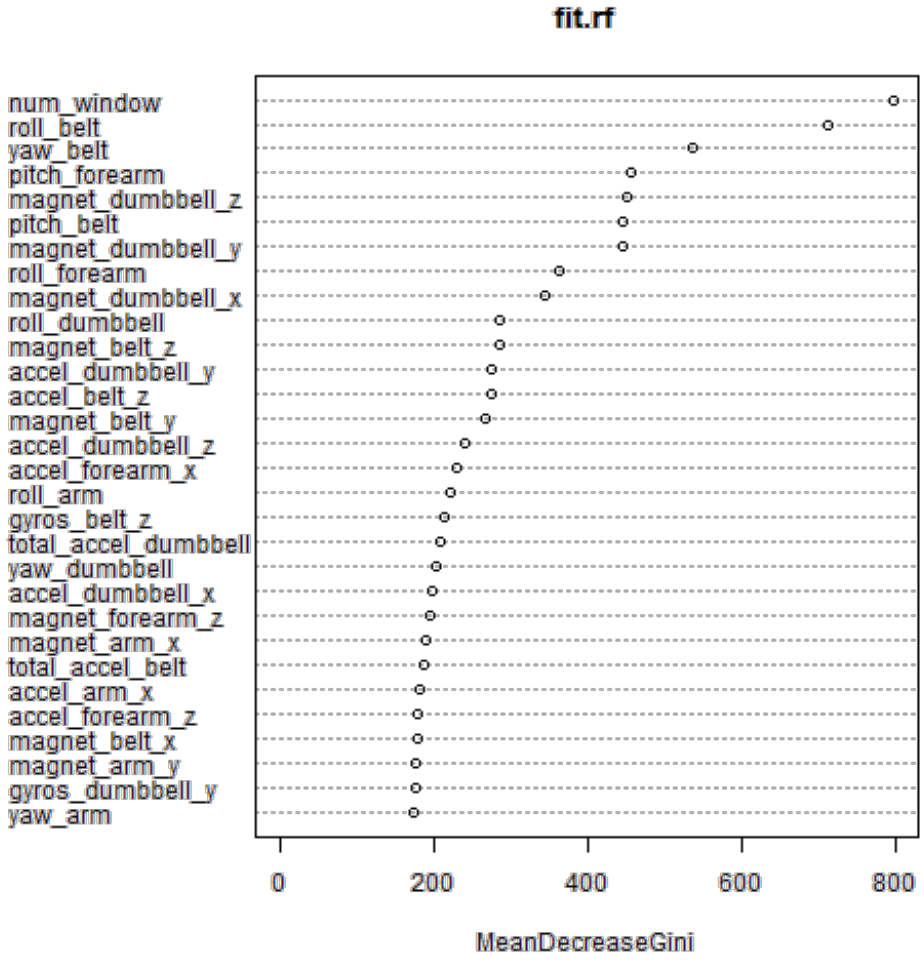
```
library(randomForest)
```

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

```
result <- rfcv(training[,-155], training[,155])
with(result, plot(n.var, error.cv, log="x", type="o", lwd=2))
```




```
fit.rf <- randomForest(classe ~ ., data=training, importantce=TRUE)
varImpPlot(fit.rf, sort=TRUE)
```




```
most5imp <- c("num_window", "roll_belt", "yaw_belt", "pitch_forearm", "magnet_dumbbell_z")
```

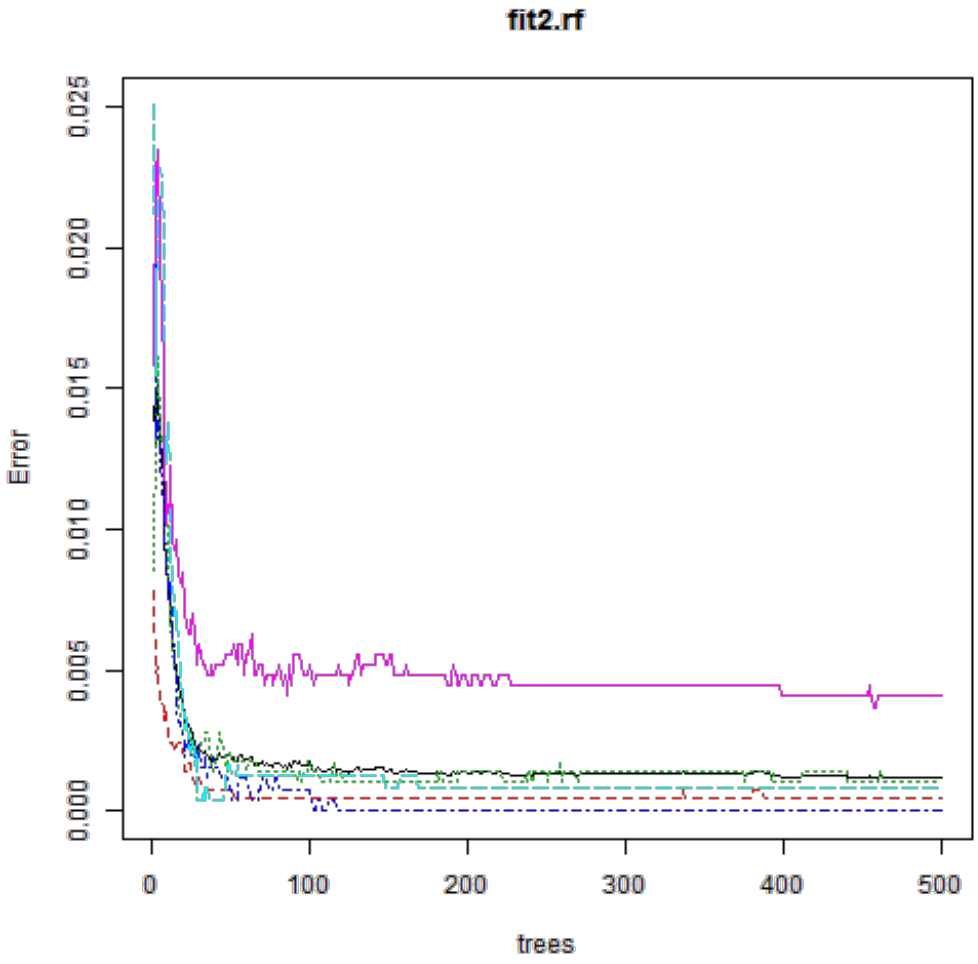
From the plot we can see that that with 5 predictors, the model error rate is reduced to almost zero. We run random forest model with full predictors, and rank the variable importance. We can see that the most important 5 predictors are “num_window”, “roll_belt”, “yaw_belt”, “pitch_forearm”, “magnet_dumbbell_z”. We will use these variables as predictors of the prediction model.

The out-of-bag random forest model already incorporates the cross validation concept. Each tree is constructed using a different bootstrap sample from the original data, and about one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree. The classification result is determined by the majority votes of all trees.

```
set.seed(1)
training_fivefeatures <- training[, c(most5imp, "classe")]
fit2.rf <- randomForest(classe ~ ., data=training_fivefeatures, importance=TRUE)
fit2.rf$confusion
```

```
##      A    B    C    D    E class.error
## A 4183    2    0    0    0   0.0004779
## B    2 2845    1    0    0   0.0010534
## C    0    0 2567    0    0   0.0000000
## D    0    0    1 2410    1   0.0008292
## E    0    2    1    8 2695   0.0040650
```

```
plot(fit2.rf)
```



From the confusion matrix and the plot, we can see that the error rate of the model is very low.

Then we use the model to test the testing dataset.

```
pred2 <- predict(fit2.rf, testing)
table(pred2, testing$classe)
```

```
##
## pred2    A    B    C    D    E
##     A 1395    2    0    0    0
##     B    0  947    0    0    0
##     C    0    0  855    0    1
##     D    0    0    0  804    3
##     E    0    0    0    0  897
```

```
err_rate <- length(pred2[!pred2==testing$classe])/nrow(testing)
err_rate
```

```
## [1] 0.001223
```

From the matrix we can see the model correctly predict most classe values in the test dataset. The out-of-sample error rate is 0.122%.

Summary

In this project, we analyze the Weight Lifting Exercises dataset and build machine learning algorithm to predict “how well” an activity is performed. We use out-of-bag random forest model, and build the algorithm with the most 5 important variables. From testing result we can see the model performs very well in terms of prediction accuracy.