Course 8 Prediction Assignment

Jianjun Ge

August 11, 2017

Summary

This is the course 8 assignment poject. The goal of this project is to use data from accelerometers on the belt, forearm, arm, etc to predict the manner in which six participants did the excise.

Load & Explore Data

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.3.2
```

```
## randomForest 4.6-12
```

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

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.3
```

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

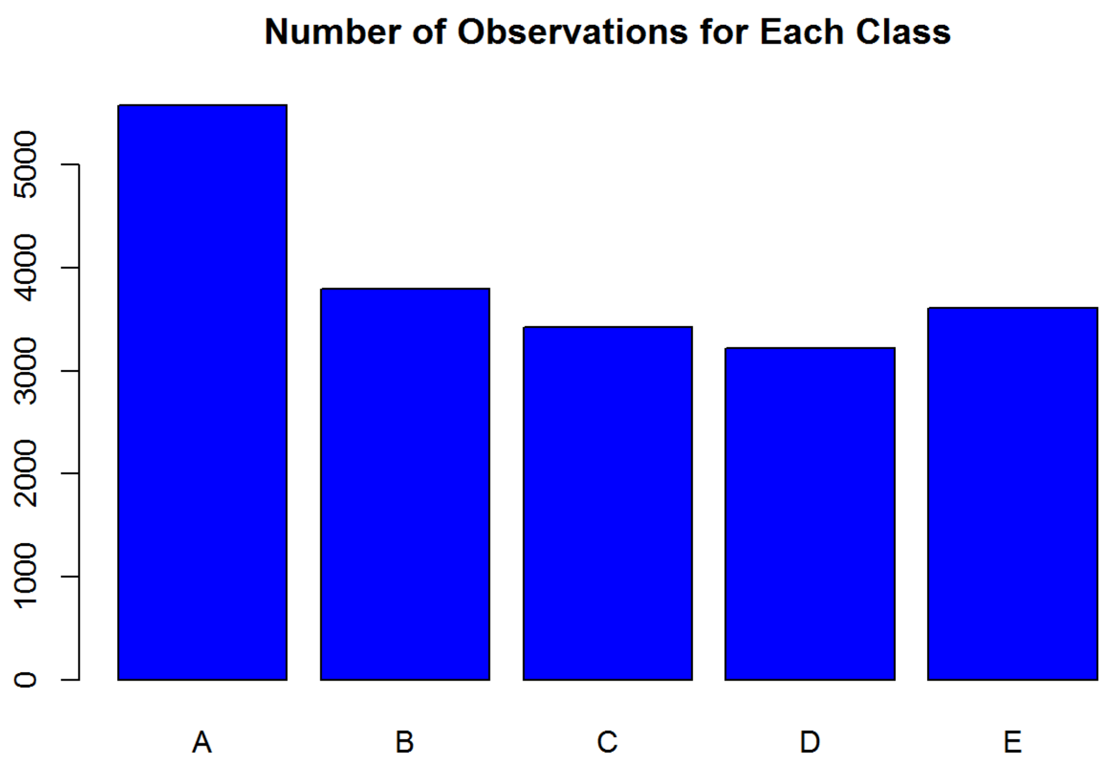
```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
##
## Attaching package: 'ggplot2'
```

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

```
pml.training <- read.csv("../08/pml-training.csv", header = TRUE)
dim(pml.training)
str(pml.training)
barplot(table(pml.training$classe),
        main = "Number of Observations for Each Class", col = "blue")
```



Preprocessing Data

The trainning data appear to have lots of NA and empty observations. The first step is to find out the number of NA or empty obs in each column.

```
pml.training[pml.training == ""] <- NA
pml.na <- apply(pml.training, 2, function(y) sum(is.na(y)))
```

Then, columns with more than 1000 NA/empty are removed. The beginning 6 columns are also removed for modeling because user_name and recording time related variables are not likely to be useful for predicting excercise manners.

```
pml.naremov <- pml.training[, pml.na < 1000]
pml.new <- pml.naremov[,-(1:6)]
dim(pml.new)
```

```
## [1] 19622    54
```

Random Forest Modeling

Random forest is chosen because of the relatively large number of observations and features of this dataset. RF is easy to train and can provide relative importance ranking of each predictors. RF is often used as benchmark models.

The trainning dataset is first broken into two parts: trainning (70%) and validation (30%).

```
train.ind <- sample(c(1:2), dim(pml.new)[1], replace = T, prob = c(0.7,0.3))
pml.new.train <- pml.new[train.ind == 1, ]
pml.new.valid <- pml.new[train.ind == 2, ]
table(pml.new.valid$classe)
```

```
##
##    A    B    C    D    E
## 1699 1148 1029  982 1118
```

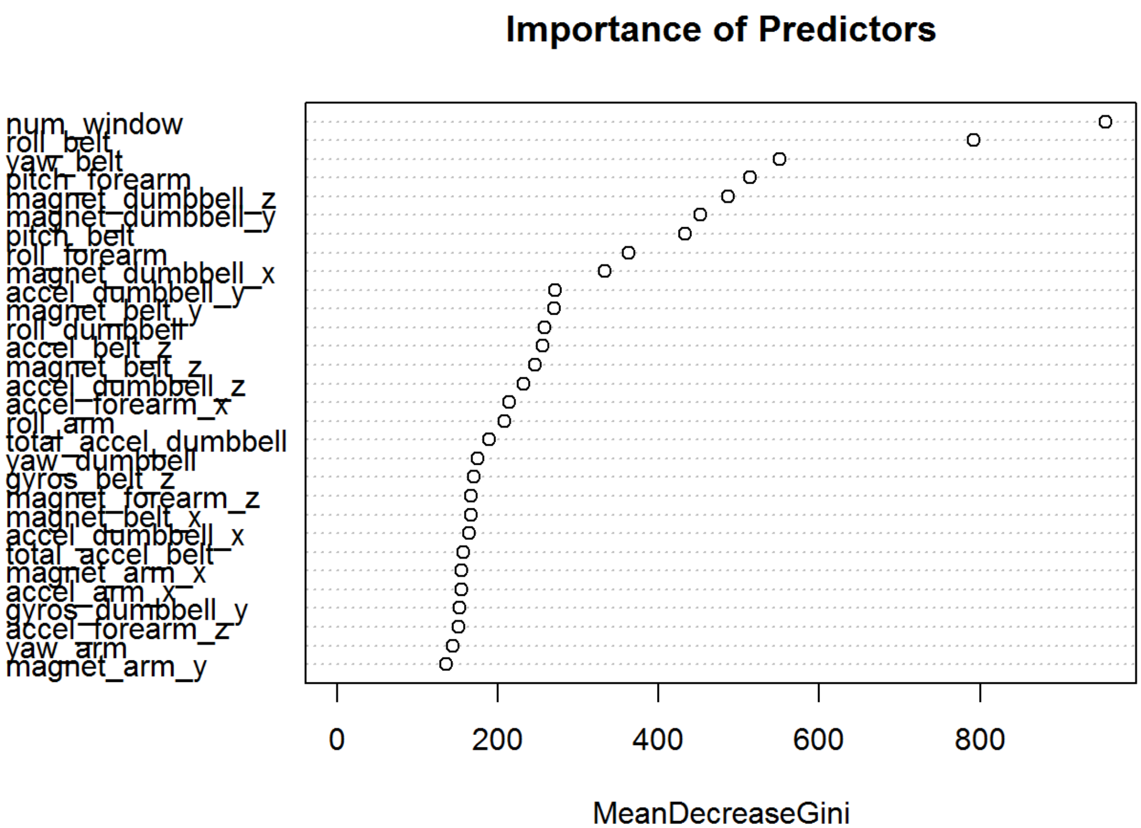
Run RF on trainning data:

```
rf.1 <- randomForest(classe ~ ., data=pml.new.train)
rf.1
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = pml.new.train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.36%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3880    1    0    0    0 0.0002576656
## B    6 2641    2    0    0 0.0030200076
## C    0    9 2383    1    0 0.0041788550
## D    0    0   22 2211    1 0.0102954342
## E    0    0    0    7 2482 0.0028123744
```

Importance of predictors from this RF model:

```
varImpPlot(rf.1, main = "Importance of Predictors")
```



This plot shows that "num_window", "roll_belt", and "yaw_belt" are top three most important variables in predicting exercise manner classes.

Cross validation using the rest of the trainning data (30%):

```
pred <- predict(rf.1, pml.new.valid)
table(pred, pml.new.valid$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1698    3    0    0    0
##    B    0 1144    3    0    0
##    C    0    1 1026    2    0
##    D    0    0    0  980    2
##    E    1    0    0    0 1116
```

```
confusionMatrix(pred, pml.new.valid$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1698    3    0    0    0
##          B    0 1144    3    0    0
##          C    0    1 1026    2    0
##          D    0    0    0  980    2
##          E    1    0    0    0 1116
##
## Overall Statistics
##
##                Accuracy : 0.998
##                  95% CI : (0.9965, 0.999)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9975
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9965   0.9971   0.9980   0.9982
## Specificity            0.9993   0.9994   0.9994   0.9996   0.9998
## Pos Pred Value         0.9982   0.9974   0.9971   0.9980   0.9991
## Neg Pred Value         0.9998   0.9992   0.9994   0.9996   0.9996
## Prevalence             0.2843   0.1921   0.1722   0.1643   0.1871
## Detection Rate         0.2841   0.1914   0.1717   0.1640   0.1867
## Detection Prevalence   0.2846   0.1919   0.1722   0.1643   0.1869
## Balanced Accuracy      0.9994   0.9979   0.9982   0.9988   0.9990
```

The Out Of Bag estimate of error rate is 0.28%, which is very low. Using cross-validation, the overall accuracy is 0.9976, which is very high. The reported Kappa coefficient is also very high 0.997.

Prediction Using 20 Test Cases

```
pml.testing <- read.csv("../08/pml-testing.csv", header = TRUE)
pred.2 <- predict(rf.1, pml.testing)
pred.2
```

```
##  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
```