Prediction Assigment

Manu Srivastav

August 23, 2015

Data Cleaning

To clean the data, the first row index and all colomuns with NA were removed. The traing and testing data were saved as traing2.csv and testing2.csv.

```
# Remove everything in current working library
rm(list = ls())
# Read cleaned training and testing data
training <- read.table(file = "pml-training2.csv",
                       header = TRUE, sep = ",", quote = "")
testing <- read.table(file = "pml-testing2.csv",
                      header = TRUE, sep = ",", quote = "")
# Change the numeric type to integer type to make sure
# the same data type in training data and testing data
training$magnet_dumbbell_z <- as.integer(training$magnet_dumbbell_z)
training$magnet_forearm_y <- as.integer(training$magnet_forearm_y)
training$magnet_forearm_z <- as.integer(training$magnet_forearm_z)
# Change the
levels(testing$new_window) <- levels(training$new_window)
```

Exploratory Data Analysis

Cross Validation was performed to find the out of sample errors.

```
# Install randomForest package
# install.packages("randomForest")
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.1.3

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

# install.packages("caret")
library(caret)

## Warning: package 'caret' was built under R version 3.1.3

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

## Warning: package 'ggplot2' was built under R version 3.1.3
```

Exploratory Data Analysis

```
set.seed(111)
# Define cross-validation experiment
fitControl = trainControl( method = "cv", number = 2)
# Perform the cross validation
cv <- train(classe ~ ., data = training, method = "rf",
  trControl = fitControl)
cv$bestTune$mtry
```

```
## [1] 28
```

Exploratory Data Analysis

```
cv
```

```
## Random Forest
##
## 19622 samples
##    54 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 9811, 9811
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD   Kappa SD
##    2    0.9924574  0.9904582  0.0008648743  0.001094362
##   28    0.9957701  0.9946489  0.0019459671  0.002462580
##   54    0.9940883  0.9925214  0.0027387685  0.003465625
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 28.
```

Build random forest model with full training model

Best Tune of number of variable randomly sampled is: 28

```
RandomForest = randomForest(classe ~ ., data = training,
                            mtry = cv$bestTune$mtry)
PredictForTrain = predict(RandomForest)
table(PredictForTrain, training$classe)
```

```
##
## PredictForTrain    A    B    C    D    E
##               A 5578    4    0    0    0
##               B    1 3790    4    0    0
##               C    0    2 3418    7    0
##               D    0    1    0 3208    4
##               E    1    0    0    1 3603
```

Predict testing data

```
PredictForest = predict(RandomForest, newdata = testing)
PredictForest
```

```
##  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
```

Write the Prediction to files

```
# Function to write a vector to files
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_", i ,".txt")
    write.table(x[i], file = filename, quote = FALSE,
                row.names = FALSE, col.names = FALSE)
  }
}
# Call the function
pml_write_files(PredictForest)
```