PractMachineLearning

Eric Pei

October 24, 2015

```
library(caret)
```

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

```
train <- read.csv(file="pml-training.csv", header= TRUE)
test <- read.csv(file="pml-testing.csv", header= TRUE)
summary(train$classe)
```

```
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

```
set.seed(12345)
inTrain <- createDataPartition(y=train$classe, p=0.6, list=FALSE)
training <- train[inTrain,]
testing <- train[-inTrain,]
dim(training)
```

```
## [1] 11776   160
```

```
dim(testing)
```

```
## [1] 7846  160
```

First, we preprocess our data in a couple of ways.

Step 1, we remove the variabels with too many NAs

Step 2, we remove the values with very little variance with the nearZeroVar command.

We will use the Random Forest method and since using class(method="rf") takes hours on my computer, I have installed the randomForest package.

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
##          B    0 1518    0    0    0
##          C    0    0 1368    0    0
##          D    0    0    0 1286    0
##          E    0    0    0    0 1442
```

```
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9995, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Here is our confusion matrix. Accuracy of 100%! No need to look further, lets apply this to our test.