Practical Machine Learning Course Project Write Up

Eric Lu

4/7/2017

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Loading and preprocessing the data

We load the data sets that we already downloaded.

```
setwd("~/Documents")
trainingData <- read.csv("pml-training.csv")
testingData <- read.csv("pml-testing.csv")
## table(trainingData$classe)
```

We separate the training data into a training set and a validation set.

```
library(caret)
set.seed(999)
inTrain <- createDataPartition(trainingData$classe, p = 0.75, list = FALSE)
trainingSet <- trainingData[inTrain, ]
validationSet <- trainingData[-inTrain, ]
```

Feature selection

Then we clean up near-zero-variance variables and columns with missing values more than 70%.

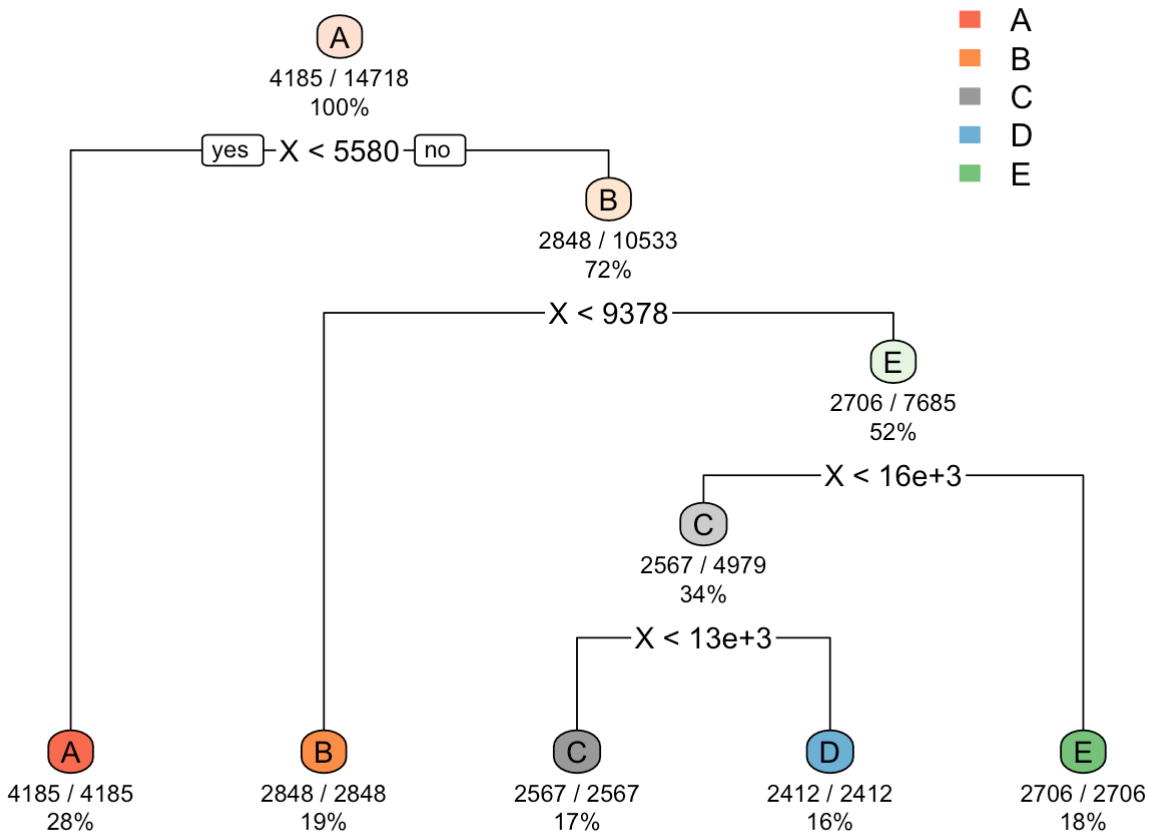
```
NZVcol <- nearZeroVar(trainingSet)
trainingSet <- trainingSet[, -NZVcol]
treshold <- dim(trainingSet)[1] * 0.7
goodCol <- !apply(trainingSet, 2, function(x) sum(is.na(x)) > treshold  || sum(x=="") >
treshold)
trainingSet <- trainingSet[, goodCol]
table(trainingSet$classe)
```

```
##
##    A    B    C    D    E
## 4185 2848 2567 2412 2706
```

Model comparison

First, we will use the decision tree model.

```
library(rpart)
library(rpart.plot)
mod1 <- rpart(classe ~., data=trainingSet, method="class")
rpart.plot(mod1, extra=102, under=TRUE, faclen=0)
```




```
pred1 <- predict(mod1, validationSet, type = "class")
confusionMatrix(pred1, validationSet$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    0    0    0    0
##          B    1  949    0    0    0
##          C    0    0  855    0    0
##          D    0    0    0  803    0
##          E    0    0    0    1  901
##
## Overall Statistics
##
##                Accuracy : 0.9996
##                  95% CI : (0.9985, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9995
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9993   1.0000   1.0000   0.9988   1.0000
## Specificity            1.0000   0.9997   1.0000   1.0000   0.9998
## Pos Pred Value         1.0000   0.9989   1.0000   1.0000   0.9989
## Neg Pred Value         0.9997   1.0000   1.0000   0.9998   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1935   0.1743   0.1637   0.1837
## Detection Prevalence   0.2843   0.1937   0.1743   0.1637   0.1839
## Balanced Accuracy      0.9996   0.9999   1.0000   0.9994   0.9999
```

We see that accuracy for the decision tree model is 99.96% with a 95% confidence interval of (0.9985, 1).

Next, we will use the random forest model.

```
library(randomForest)
mod2 <- randomForest(classe ~., data=trainingSet)
pred2 <- predict(mod2, validationSet, type = "class")
confusionMatrix(pred2, validationSet$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    0  949    0    0    0
##          C    0    0  855    0    0
##          D    0    0    0  804    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9992, 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   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Based on the results above, we see that we have reached an accuracy of 100% with a 95% confidence interval of (0.9992, 1) by using the random forest model. Therefore, we stop the comparison here and move on with the random forest model.

Test data predictions

Now we apply the model to the testing set.

```
pred3 <- predict(mod2, testingData)
print(pred3)
```

```
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)
  }
}
pml_write_files(pred3)
```