Course Project

Fatemeh Abyarjoo

Friday, June 12, 2015

The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

LOADING THE DATA

```r
library(lattice)
library(ggplot2)
library(caret)
trainurl="http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testurl="http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

traindata= read.csv(url(trainurl), na.strings=c("NA","#DIV/0!",""))
testdata=read.csv(url(testurl), na.strings=c("NA","#DIV/0!",""))

dim(traindata)
```

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

```r
dim(testdata)
```

```
## [1]  20 160
```

```r
#names(traindata)
#head(traindata)
#head(testdata)
```

CLEAN THE DATA

```r
# eliminating near zeros
nearz= nearZeroVar(traindata)
traindata= traindata[, -nearz]

nearz= nearZeroVar(testdata)
testdata= testdata[, -nearz]
# eliminating irrelevant columns
traindata=traindata[,-c(1:7)]
testdata=testdata[,-c(1:7)]

# eliminating columns which are all NAs
traindata= traindata[,colSums(is.na(traindata)) == 0]
testdata= testdata[,colSums(is.na(testdata)) == 0]

dim(traindata)
```

```
## [1] 19622    52
```

PARTITIONING THE TRAINING DATA

```r
set.seed(2000)
data1= createDataPartition(y=traindata$classe, p=0.75, list=FALSE)
train_sub= traindata[data1, ]
test_sub= traindata[-data1, ]

#plot(train_sub$classe, col="green",xlab="classe", ylab="Frequency")
```

MODEL BUILDING: First model:Decision tree

```r
#install.packages("randomForest")
#install.packages("rpart")
#install.packages("rpart.plot")
#install.packages("rattle")
library(randomForest)
```

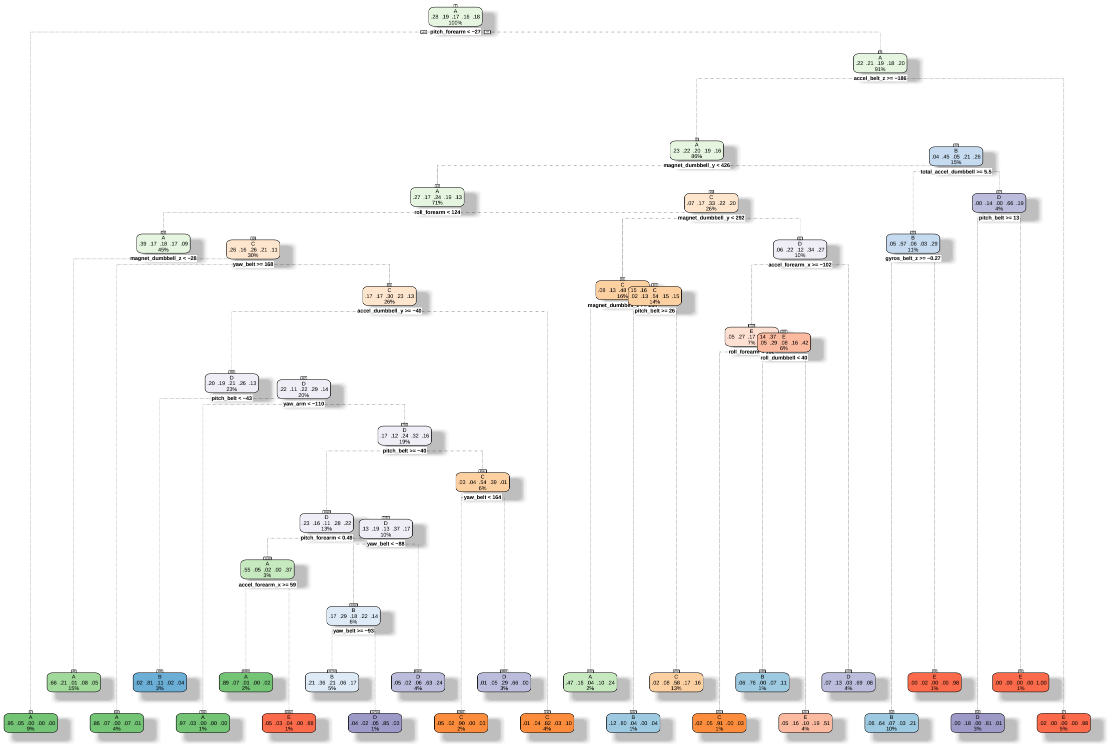
```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```r
library(rpart)
library(rpart.plot)
library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 3.4.1 Copyright (c) 2006-2014 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
firstmodel= rpart(classe ~ ., data=train_sub, method="class")
fancyRpartPlot(firstmodel)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2015−Jun−19 23:26:57 fatemeh

```r
firstpredict=predict(firstmodel, test_sub, type = "class")

dtree= confusionMatrix(firstpredict, test_sub$classe)
dtree
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1232  204   19   66   65
##          B  104  570   80   37  164
##          C   18   64  668  107   98
##          D   18   61   72  535   61
##          E   23   50   16   59  513
##
## Overall Statistics
##
##                Accuracy : 0.7174
##                  95% CI : (0.7045, 0.7299)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6407
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
```

```
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8832   0.6006   0.7813   0.6654   0.5694
## Specificity            0.8991   0.9027   0.9291   0.9483   0.9630
## Pos Pred Value         0.7768   0.5969   0.6995   0.7162   0.7761
## Neg Pred Value         0.9509   0.9040   0.9526   0.9353   0.9086
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2512   0.1162   0.1362   0.1091   0.1046
## Detection Prevalence   0.3234   0.1947   0.1947   0.1523   0.1348
## Balanced Accuracy      0.8911   0.7516   0.8552   0.8069   0.7662
```

Second model: Random forest

```
secondmodel=randomForest(classe ~ ., data=train_sub)
secondpredict= predict(secondmodel, test_sub, type = "class")

rforest= confusionMatrix(secondpredict, test_sub$classe)
rforest
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1392    5    0    0    0
##          B    2  942    6    0    0
##          C    0    1  848    7    0
##          D    0    0    1  796    1
##          E    1    1    0    1  900
## 
## Overall Statistics
## 
##                Accuracy : 0.9947
##                  95% CI : (0.9922, 0.9965)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9933
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9978   0.9926   0.9918   0.9900   0.9989
## Specificity            0.9986   0.9980   0.9980   0.9995   0.9993
## Pos Pred Value         0.9964   0.9916   0.9907   0.9975   0.9967
## Neg Pred Value         0.9991   0.9982   0.9983   0.9981   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2838   0.1921   0.1729   0.1623   0.1835
## Detection Prevalence   0.2849   0.1937   0.1746   0.1627   0.1841
## Balanced Accuracy      0.9982   0.9953   0.9949   0.9948   0.9991
```