Machine Learning Project

Beatriz Del Fiol

14 de outubro de 2019

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.

Getting data set

```
library(data.table)

TrainDataURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
TestDataURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

# Download and clean the datasets
TrainingData <- read.csv(url(TrainDataURL), na.strings=c("NA","#DIV/0!",""))
TestingData <- read.csv(url(TestDataURL), na.strings=c("NA","#DIV/0!",""))

# Check for the datasets dimemsions
dim(TrainingData)
```

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

```
dim(TestingData)
```

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

Preparing dataset

```
# Delete columns with missing values
TrainingData <-TrainingData[,colSums(is.na(TrainingData)) == 0]
TestingData <-TestingData[,colSums(is.na(TestingData)) == 0]

# Delete unused columns
TrainingData <-TrainingData[,-c(1:7)]
TestingData <-TestingData[,-c(1:7)]

# Check for the datasets dimemsions
dim(TrainingData)
```

```
## [1] 19622    53
```

```
dim(TestingData)
```

```
## [1] 20 53
```

Preparing to run models

Load necessary pachages knitr, caret, rpart.plot, rattle, randomForest, corrplot, gbm

```
library(knitr)
library(caret)
```

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

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

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

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:rattle':
##
##     importance
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
library(e1071)
set.seed(321)
```

Dividing dataset

70% training and 30% test

```
PartData <- createDataPartition(TrainingData$classe, p=0.7, list=FALSE)
TrainingSet <- TrainingData[PartData, ]
dim(TrainingSet)
```

```
## [1] 13737    53
```

```
#names(TrainingSet)

TestingSet <- TrainingData[-PartData, ]
dim(TestingSet)
```

```
## [1] 5885   53
```

```
#names(TestingSet)
```

Correlation matriz

TraingSet and TestingSet

Prediction model Building

It will predicting with tree, Bagging, Random Forests, Booting and verify the accuracy.

a. Random Forests

```
library(randomForest)
set.seed(321)
RFmodel <- randomForest(classe ~., data=TrainingSet, method="class")

# Predicting
prediction1 <- predict(RFmodel, TestingSet, Type="class")

#Testing
confusionMatrix(prediction1, TestingSet$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    3    0    0    0
##          B    0 1135    6    0    0
##          C    0    1 1017    9    0
##          D    0    0    3  952    1
##          E    1    0    0    3 1081
##
## Overall Statistics
##
##                Accuracy : 0.9954
##                  95% CI : (0.9933, 0.997)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9942
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9965   0.9912   0.9876   0.9991
## Specificity            0.9993   0.9987   0.9979   0.9992   0.9992
## Pos Pred Value         0.9982   0.9947   0.9903   0.9958   0.9963
## Neg Pred Value         0.9998   0.9992   0.9981   0.9976   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
```

```
## Detection Rate         0.2843   0.1929   0.1728   0.1618   0.1837
## Detection Prevalence   0.2848   0.1939   0.1745   0.1624   0.1844
## Balanced Accuracy      0.9993   0.9976   0.9946   0.9934   0.9991
```

b. Decision Tree

```
set.seed(321)
fitDT <- rpart(classe ~ ., data = TrainingSet, method="class")
predictDT <- predict(fitDT, newdata = TestingSet, type="class")
conf_matrixDT <- confusionMatrix(predictDT, TestingSet$classe)
conf_matrixDT
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1521  291   23  127   36
##          B   30  571   69   43   81
##          C   50  157  848   79   86
##          D   60   76   72  641   78
##          E   13   44   14   74  801
##
## Overall Statistics
##
##                Accuracy : 0.7446
##                  95% CI : (0.7333, 0.7557)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.675
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9086  0.50132   0.8265   0.6649   0.7403
## Specificity            0.8867  0.95301   0.9234   0.9419   0.9698
## Pos Pred Value         0.7613  0.71914   0.6951   0.6915   0.8467
## Neg Pred Value         0.9606  0.88843   0.9618   0.9349   0.9431
## Prevalence             0.2845  0.19354   0.1743   0.1638   0.1839
## Detection Rate         0.2585  0.09703   0.1441   0.1089   0.1361
## Detection Prevalence   0.3395  0.13492   0.2073   0.1575   0.1607
## Balanced Accuracy      0.8977  0.72717   0.8750   0.8034   0.8551
```

c. Generalized Boosted Model (GBM)

```
library(caret)
set.seed(321)
fitControl <- trainControl(method="repeatedcv", number=5, repeats=1)
GBMmodel <- train(classe ~., data=TrainingSet, method="gbm", trControl=fitControl, verbose=FALSE)

# Predicting
```

```r
prediction2 <- predict(GBMmodel, TestingSet)

# Testing
confusionMatrix(prediction2, TestingSet$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1656   52    0    3    2
##          B   14 1064   42    5    8
##          C    3   23  971   37    6
##          D    1    0   13  907   14
##          E    0    0    0   12 1052
##
## Overall Statistics
##
##                Accuracy : 0.9601
##                  95% CI : (0.9547, 0.9649)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9494
##
##  Mcnemar's Test P-Value : 6.006e-10
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9892   0.9342   0.9464   0.9409   0.9723
## Specificity            0.9865   0.9855   0.9858   0.9943   0.9975
## Pos Pred Value         0.9667   0.9391   0.9337   0.9701   0.9887
## Neg Pred Value         0.9957   0.9842   0.9886   0.9885   0.9938
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2814   0.1808   0.1650   0.1541   0.1788
## Detection Prevalence   0.2911   0.1925   0.1767   0.1589   0.1808
## Balanced Accuracy      0.9879   0.9598   0.9661   0.9676   0.9849
```

Test

Since Random Forest was chosen due to the best accuray, we will run the prediction test with the Test data file.

```r
predictTest <- predict(RFmodel, TestingData)
predictTest
```

```
##  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
```