Coursera Project Practical Machine Learning: Prediction Assignment Writeup

Thorsten

2020-01-09

Overview

The goal of the project is to predict the manner in which they did the exercise. Following data will be available:

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

Applying the machine learning algorithm to the 20 test cases available in the test data for checking the model.

Loading data

```
trainset <- read.csv("pml-training.csv")
```

Cleaning data

```
trainset <- trainset[ , colSums(is.na(trainset)) == 0] # selecting only columns that do not have NAs
trainset <- trainset[ , -nearZeroVar(trainset)] # removing columns with near zero variance
trainset <- trainset[ , -c(1:6)] # removing variables for row number, username, timestamp, numwindow
```

Devide trainset into train/test for Prediction

```
partition <- createDataPartition(y=trainset$classe, p=0.8, list=FALSE)
trainset.Train <- trainset[partition,]
trainset.Test <- trainset[-partition,]
```

Prediction

Parallel Processing

```
cl <- makePSOCKcluster(3)  # use three cores
registerDoParallel(cl)  # do not forget to deregister via stopCluster(cl)
```

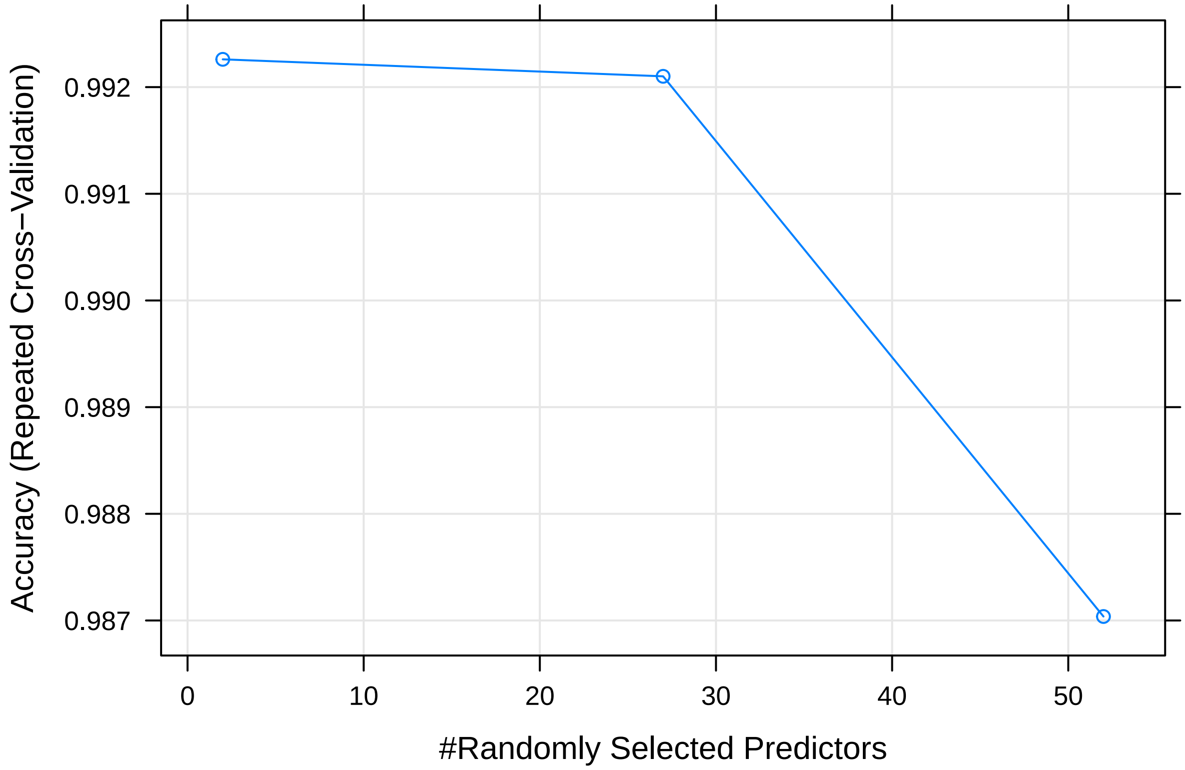
```
theControl <- trainControl(method = "repeatedcv", number = 4, repeats = 2, allowParallel = TRUE, verbos
```

Random Forest

```r
theModel <- train(classe ~ ., data = trainset.Train, method = "rf", trControl = theControl)
```

```
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 2 on full training set
```

```r
plot(theModel)
```



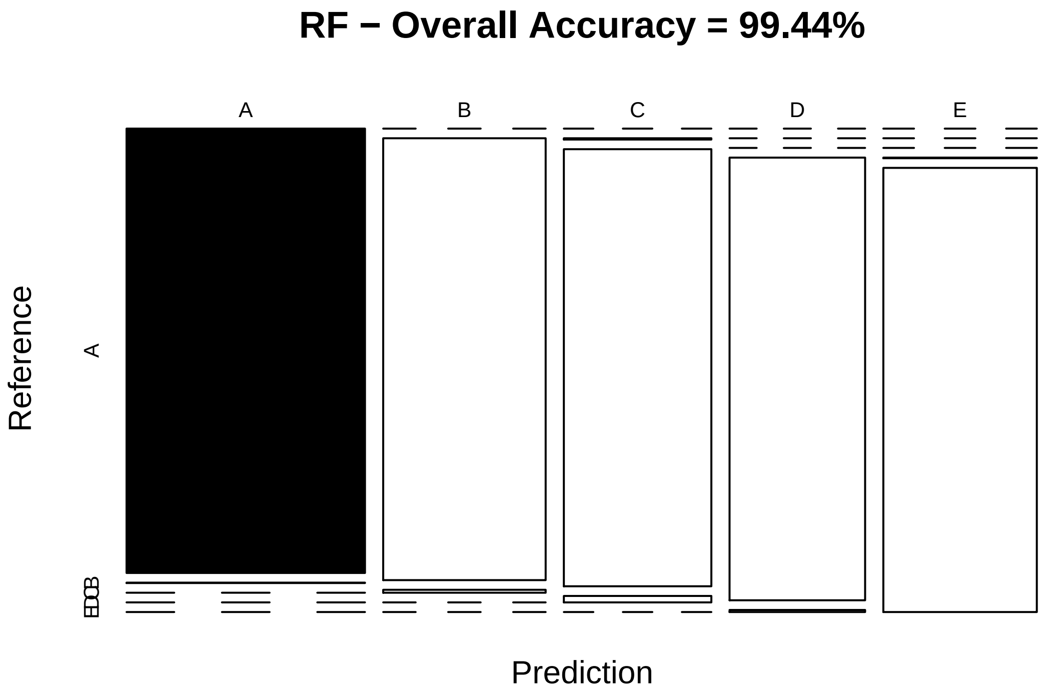
Stop Parallel Processing

```r
stopCluster(cl)
```

```r
thePredict <- predict(theModel, trainset.Test)
theConfMat <- confusionMatrix(thePredict, trainset.Test$classe)
theConfMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    1    0    0    0
##          B    0  756    5    0    0
##          C    0    2  679   10    0
##          D    0    0    0  632    3
##          E    0    0    0    1  718
##
## Overall Statistics
##
##                Accuracy : 0.9944
##                  95% CI : (0.9915, 0.9965)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9929
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9960   0.9927   0.9829   0.9958
## Specificity            0.9996   0.9984   0.9963   0.9991   0.9997
## Pos Pred Value         0.9991   0.9934   0.9826   0.9953   0.9986
## Neg Pred Value         1.0000   0.9991   0.9985   0.9967   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1927   0.1731   0.1611   0.1830
## Detection Prevalence   0.2847   0.1940   0.1761   0.1619   0.1833
## Balanced Accuracy      0.9998   0.9972   0.9945   0.9910   0.9978
```

```r
plot(theConfMat$table, col = theConfMat$byClass, main = paste("RF - Overall Accuracy = ", round(theConf
```



Prediction on test dataset

```
testset <- read.csv("pml-testing.csv")
testset <- testset[ , colSums(is.na(testset)) == 0]
testset <- testset[ , -nearZeroVar(testset)]

thePredictResult <- predict(theModel, testset)
thePredictResult
```

```
##  [1] 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
```