Course 8

Daniel Cialdella - dcialdella@gmail.com

October 17, 2016

Objetives

The objetive of this tasks is use machine learning procedures to predict in which manner the "users" used "Ascelerometers" and the quality of data collected from devices. Using "training data" to generate a prediction and later compare it to the "testing data". Build a model, how use "cross validations", show "expectatives" and the expected error of the testing. Then use the prediction model with 20 different test cases.

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. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data used for Training and Testing.

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

Downloaded from the webpage and stored in the standard folder (rstudio).

Storing and cleaning data.

```
library(caret)   # needed for other functions
```

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

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

```
library(ggplot2)
library(randomForest)
```

```
## randomForest 4.6-12
```

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

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

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

```
set.seed(32768) # Seed for random values, not sure if needed but just in case

train <- read.csv("pml-training.csv", header=TRUE, sep=",", na.strings=c("NA","")) # Read Data from DIS
test  <- read.csv("pml-testing.csv" , header=TRUE, sep=",", na.strings=c("NA","")) # Read Data from DIS

train2 <- train[, -(nearZeroVar(train)) ]  # Remove data with Near Zero variability
test2 <-   test[, -(nearZeroVar(test))  ]  # Remove data with Near Zero variability
remove(train)
remove(test)

allna <- sapply(train2, function(x) mean(is.na(x))) > 0.95 # Identify Columns where MEANs are > 0.95
train3 <- train2[, allna==F]  # Reduce to 59 columns, removing some columns where NA is > 95%
train4 <- train3[ , -(1:6)]   # Remove first 6 cols, seams not important data
remove(train2)
remove(train3)

allnas <- sapply(test2, function(x) mean(is.na(x))) > 0.95 # Identify Columns where MEANs are > 0.95
test3 <- test2[, allnas==F]  # Reduce to 59 columns, removing some columns where NA is > 95%
test4 <- test3[ , -(1:6)]    # Remove first 6 cols, seams not important data
remove(test2)
remove(test3)

# summary(train4)  # verify variability and dispersion of data
# summary(test4)

# THIS ARE THE DATA CLEANED and VALID for OPERATIONS (xxxx0)
dim(train4)  # 19622 / 53 - last col is CLASSE
```

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

```
dim(test4)   # 20 / 53    - last col is PROBLEM_ID
```

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

```
# Now, seams easier to compare data or take actions

# To fix a future issue related to RandomForest with this field
train4$classe <- factor(train4$classe)     # Fix an error
```

In this point, review data provided by Summary in all fields (53) and check if other column could be deleted. CLASSE col is c("A", "B", "C", "D", "E")

Slice Data from TRAIN in TRAIN0 and VALID0

```
# Slice the data for TRAINING in TRAIN0 and VALID0 (for validation)
slicing   <- createDataPartition(train4$classe, p=0.66, list=FALSE)
```

```
train0   = train4[ slicing ,]    # 66% used for TRAINING         - 12953
valid0   = train4[ -slicing,]    # 34% used for VALIDATION MODEL - 6669
# remove(train4)   # may be needed for future tasks
```

Use RandomForest as MODEL Testing 1

Why RF ? (I prefer it) Easy one, no need for other validation or additional test, errors and other data generated during the Generation of Model. I will use 66% of TRAINING data to obtain a Model, then compare it with the other 34% of data. And later, rebuild the Model using the 100% of data from TRAINING.

```
ModelForTraining1 <- randomForest(classe ~ . , data=train0)
ModelForTraining1 # read data from RandomForest
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = train0)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.6%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3678    4    0    0    1 0.001357589
## B   18 2484    5    0    0 0.009174312
## C    0   14 2242    3    0 0.007525454
## D    0    0   21 2100    2 0.010833726
## E    0    0    3    7 2371 0.004199916
```

```
# Evaluate model with ConfusionMatrix (agains valid0)
ModelForTraining2 <- confusionMatrix(predict( ModelForTraining1 , newdata=valid0[,-ncol(valid0)]),valid
ModelForTraining2  # review results
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1897    7    0    0    0
##          B    0 1280   12    0    0
##          C    0    3 1151   20    0
##          D    0    0    0 1073    0
##          E    0    0    0    0 1226
##
## Overall Statistics
##
##                Accuracy : 0.9937
##                  95% CI : (0.9915, 0.9955)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.992
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9922   0.9897   0.9817   1.0000
## Specificity            0.9985   0.9978   0.9958   1.0000   1.0000
## Pos Pred Value         0.9963   0.9907   0.9804   1.0000   1.0000
## Neg Pred Value         1.0000   0.9981   0.9978   0.9964   1.0000
## Prevalence             0.2845   0.1934   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1919   0.1726   0.1609   0.1838
## Detection Prevalence   0.2855   0.1937   0.1760   0.1609   0.1838
## Balanced Accuracy      0.9993   0.9950   0.9928   0.9909   1.0000
```

Use TRAINING data and validate with VALIDATION data

Use model based in RF with TRAINING DATA, and using ConfusionMatrix compared with VALIDATION DATA

```
Comparatives <- c( as.numeric( predict( ModelForTraining1, newdata=valid0[,-ncol(valid0)])==valid0$class
Exactitude <- ( sum(Comparatives) / nrow(valid0) ) * 100
# 99.35 %
```

Exactitude of the model 99.35%

Rebuild the model, now with the 100% of data from Training.

```
ModelForTraining1 <- randomForest(classe ~ . , data=train0)
ModelForTraining1 # read data from RandomForest
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = train0)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.56%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3677    4    1    0    1 0.001629107
## B   14 2490    3    0    0 0.006781013
## C    0   16 2241    2    0 0.007968127
## D    0    0   21 2100    2 0.010833726
## E    0    0    2    7 2372 0.003779924
```

Predict the manner in which they did the exercise: the variable classe is used to classify how a subject performed an exercise, as documented in the paper at http://groupware.les.inf.puc-rio.br/work.jsf?p1=11201. Students should use it as the dependent variable in the machine learning model.

Now using TESTING data.

You will predict 20 test cases: The data distributed with the assignment includes a test file of 20 observations that do not have the classe variable, so you won't be able to tell whether the predictions are accurate until you use them in a quiz. Once you build your predictive model on the training data, run it against the test data to obtain predicted values for each of the 20 cases. You will use theses predictions to answer a quiz that is part of the assignment.

```
# Remove last column with a ID, not needed, build TEST0 data
test0 <- test4 [ , -ncol(test4) ]  # col nro 53 remover

# Create a new object with the Predicted values for NEW col.
ColToAdd <- predict( ModelForTraining1, newdata=test0)

# Add ColToAdd as a new COL to TEST0.
MagicHere <- cbind(test0, ColToAdd)
```

Prepare HTML version of DOC (RMD) GitHub repo with the RMD, and MD file.

EOF