Practical_Machine_Learning_Project

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 (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

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

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

The data for this project come from this source: [http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)]. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Final Goal

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.

Reproducibility

Set your working directory

```
setwd("C:/Users/Alex/Desktop/Coursera/Practical Machine Learning")
```

install the following:

```
install.packages("caret")
install.packages("dplyr")
install.packages("randomForest")
install.packages("e1071")
```

Load the following packages

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.2
```

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

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.3.2
```

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

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

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

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.3.2
```

Creating Data set

Download the data from above in the “Data” section and copy into working directory

```
rawtrain <- read.csv("pml-training.csv",na.strings=c("NA","","#DIV/0!"))
```

splitting Data

Taking 70% for the training data and 30% for the test data

```
set.seed(555)
inTrain <- createDataPartition(y = rawtrain$classe, list = FALSE, p=0.7)
trainData <- rawtrain[inTrain,]
testData <- rawtrain[-inTrain,]
```

Identify N/A values

```
table(is.na(trainData))
```

```
##
##   FALSE    TRUE
##  851290 1346630
```

```
naprops <- colSums(is.na(trainData))/nrow(trainData)
mostNAs <- names(naprops[naprops > 0.75])
mostNACols <- which(naprops > 0.75)
```

Random sample from training data

Select a sample and remove N/A’s

```
set.seed(1256)
sampletrain <- trainData %>% tbl_df %>% sample_n(size=1000)
sampletrain <- sampletrain[,-mostNACols]
```

Remove row number and user name

```
sampletrain <- sampletrain[,-grep("X|user_name",names(sampletrain))]
```

Remove the cvtd_timestamp variable

```
sampletrain <- sampletrain[,-grep("cvtd_timestamp",names(sampletrain))]
```

Remove candidate

```
sampletrain <- sampletrain[,-nearZeroVar(sampletrain)]
```

List of candidate predictors

```
modelVars <- names(sampletrain)
modelVars1 <- modelVars[-grep("classe",modelVars)]
modelVars
```

```
##  [1] "raw_timestamp_part_1" "raw_timestamp_part_2" "num_window"
##  [4] "roll_belt"            "pitch_belt"           "yaw_belt"
##  [7] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
## [10] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [13] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [16] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [19] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [22] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [25] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [28] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [31] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [34] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [37] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [40] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [43] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [46] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [49] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [52] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [55] "magnet_forearm_z"     "classe"
```

Random Forest

```
set.seed(57)
cleanedTrainData <- trainData[,modelVars]
modelFit <- randomForest(classe ~., data=cleanedTrainData, type="class")
```

Error Estimates

```
predTrain <- predict(modelFit,newdata=trainData)
confusionMatrix(predTrain,trainData$classe)$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 3906    0    0    0    0
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
```

The in-sample error is high.

Now getting an out of sample error estimate

```
classe_col <- grep("classe",names(testData))
predTest <- predict(modelFit, newdata = testData[,-classe_col], type="class")

confusionMatrix(predTest,testData$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    2    0    0    0
##          B    0 1137    3    0    0
##          C    0    0 1023    2    0
##          D    0    0    0  961    1
##          E    0    0    0    1 1081
##
## Overall Statistics
##
##                Accuracy : 0.9985
##                  95% CI : (0.9971, 0.9993)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9981
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9982   0.9971   0.9969   0.9991
## Specificity            0.9995   0.9994   0.9996   0.9998   0.9998
## Pos Pred Value         0.9988   0.9974   0.9980   0.9990   0.9991
## Neg Pred Value         1.0000   0.9996   0.9994   0.9994   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1932   0.1738   0.1633   0.1837
## Detection Prevalence   0.2848   0.1937   0.1742   0.1635   0.1839
## Balanced Accuracy      0.9998   0.9988   0.9983   0.9983   0.9994
```

The model has an out of sample accuracy of: 0.998

Prediciting exercise activity using the model

Loading the pml-test Data

```
pmltest <- read.csv("pml-testing.csv",na.strings=c("NA","","#DIV/0!"))
```

Perform Prediction

```
predplmtest <- predict(modelFit, newdata = pmltest, type="class")
```

Answers not shown due to coursera honor code.

```
predplmtest
```