

Practical Machine Learning Project October 2015

SYNOPSIS

“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 dumbbell 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).”

Given both training and test data from the following study:

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.

The goal of this project is to “predict the manner in which they did the exercise.”

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>

Need to load the appropriate packages and set the seed for reproduceable results.

```
library(AppliedPredictiveModeling)
```

```
## Warning: package 'AppliedPredictiveModeling' was built under R version  
## 3.2.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.2
```

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

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.2.2
```

```
## Loading required package: RGtk2
```

```
## Warning: package 'RGtk2' was built under R version 3.2.2
```

```
## Rattle: A free graphical interface for data mining with R.  
## Version 3.5.0 Copyright (c) 2006-2015 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.2.2
```

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

```
## Warning: package 'rpart.plot' was built under R version 3.2.2
```

```
library(RColorBrewer)
```

```
## Warning: package 'RColorBrewer' was built under R version 3.2.2
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.2.2
```

```
## randomForest 4.6-12  
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(12345)
```

QUESTION

By processing data gathered from accelerometers on the belt, forearm, arm, and dumbbell of the participants in a machine learning algorithm, the question is can the appropriate activity quality (class A-E) be predicted?

INPUT DATA

The first step is to import the data.

```
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"  
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"  
training <- read.csv(url(trainUrl), na.strings=c("NA", "#DIV/0!", ""))  
testing <- read.csv(url(testUrl), na.strings=c("NA", "#DIV/0!", ""))
```

FEATURES

Partitioned the training data set into two data sets. Reduced the number of features by removing variables with nearly zero variance. Created another subset without NZV variables.

```

inTrain <- createDataPartition(y=training$classe, p=0.6, list=FALSE)
myTraining <- training[inTrain, ]; myTesting <- training[-inTrain, ]
dim(myTraining); dim(myTesting)

## [1] 11776    160

## [1] 7846    160

myNZVvars <- names(myTraining) %in% c("new_window", "kurtosis_roll_belt", "kurtosis_picth_belt",
"kurtosis_yaw_belt", "skewness_roll_belt", "skewness_roll_belt.1", "skewness_yaw_belt",
"max_yaw_belt", "min_yaw_belt", "amplitude_yaw_belt", "avg_roll_arm", "stddev_roll_arm",
"var_roll_arm", "avg_pitch_arm", "stddev_pitch_arm", "var_pitch_arm", "avg_yaw_arm",
"stddev_yaw_arm", "var_yaw_arm", "kurtosis_roll_arm", "kurtosis_picth_arm",
"kurtosis_yaw_arm", "skewness_roll_arm", "skewness_pitch_arm", "skewness_yaw_arm",
"max_roll_arm", "min_roll_arm", "min_pitch_arm", "amplitude_roll_arm", "amplitude_pitch_arm",
"kurtosis_roll_dumbbell", "kurtosis_picth_dumbbell", "kurtosis_yaw_dumbbell", "skewness_roll_dumbbell",
"skewness_pitch_dumbbell", "skewness_yaw_dumbbell", "max_yaw_dumbbell", "min_yaw_dumbbell",
"amplitude_yaw_dumbbell", "kurtosis_roll_forearm", "kurtosis_picth_forearm", "kurtosis_yaw_forearm",
"skewness_roll_forearm", "skewness_pitch_forearm", "skewness_yaw_forearm", "max_roll_forearm",
"max_yaw_forearm", "min_roll_forearm", "min_yaw_forearm", "amplitude_roll_forearm",
"amplitude_yaw_forearm", "avg_roll_forearm", "stddev_roll_forearm", "var_roll_forearm",
"avg_pitch_forearm", "stddev_pitch_forearm", "var_pitch_forearm", "avg_yaw_forearm",
"stddev_yaw_forearm", "var_yaw_forearm")
myTraining <- myTraining[!myNZVvars]
#To check the new N?? of observations
dim(myTraining)

## [1] 11776    100

```

ALGORITHM

Cleaned variables with too many NAs. Left out variables that have more than a 60% threshold of NAs. Coerced the data into the same type to support use of Decision Trees and Random Forest.

```

myTraining <- myTraining[c(-1)]
trainingV3 <- myTraining
for(i in 1:length(myTraining)) {
  if( sum( is.na( myTraining[, i] ) ) /nrow(myTraining) >= .6 ) {
    for(j in 1:length(trainingV3)) {
      if( length( grep(names(myTraining[i]), names(trainingV3)[j]) ) ==1) {
        trainingV3 <- trainingV3[ , -j]
      }
    }
  }
}
dim(trainingV3)

## [1] 11776    58

```

```

myTraining <- trainingV3
rm(trainingV3)
clean1 <- colnames(myTraining)
clean2 <- colnames(myTraining[, -58])
myTesting <- myTesting[clean1]
testing <- testing[clean2]

dim(myTesting)

```

```
## [1] 7846 58
```

```
dim(testing)
```

```
## [1] 20 57
```

```

for (i in 1:length(testing) ) {
  for(j in 1:length(myTraining)) {
    if( length( grep(names(myTraining[i]), names(testing)[j]) ) ==1) {
      class(testing[j]) <- class(myTraining[i])
    }
  }
}
testing <- rbind(myTraining[2, -58] , testing)
testing <- testing[-1,]

```

PARAMETERS

Classification Tree and Random Forest.

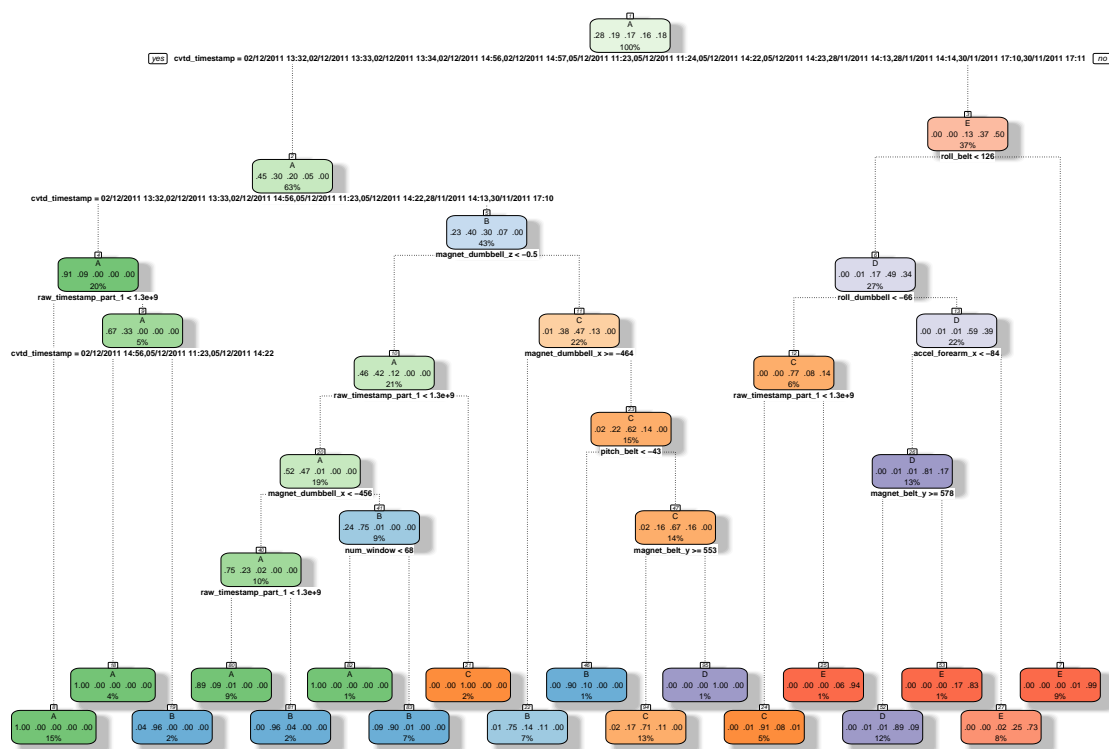
Classification Tree

Used Confusion Matrix to test results. Accuracy was 0.879 with an out-of-sample error of 12.1%.

```

modFitA1 <- rpart(classe ~ ., data=myTraining, method="class")
fancyRpartPlot(modFitA1)

```



Rattle 2015-Oct-25 12:16:05 Daddy 2

```
predictionsA1 <- predict(modFitA1, myTesting, type = "class")
confusionMatrix(predictionsA1, myTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A    B    C    D    E
##           A 2150   60    7    1    0
##           B   61 1260   69   64    0
##           C   21  188 1269  143    4
##           D    0   10   14  857   78
##           E    0    0    9  221 1360
##
## Overall Statistics
##
##           Accuracy : 0.8789
##           95% CI : (0.8715, 0.8861)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.8468
##           McNemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##                               Class: A Class: B Class: C Class: D Class: E
## Sensitivity                   0.9633   0.8300   0.9276   0.6664   0.9431
## Specificity                   0.9879   0.9693   0.9450   0.9845   0.9641
## Pos Pred Value                0.9693   0.8666   0.7809   0.8936   0.8553
## Neg Pred Value                0.9854   0.9596   0.9841   0.9377   0.9869
## Prevalence                    0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate                0.2740   0.1606   0.1617   0.1092   0.1733
## Detection Prevalence         0.2827   0.1853   0.2071   0.1222   0.2027
## Balanced Accuracy             0.9756   0.8997   0.9363   0.8254   0.9536
```

Random Forest

Random Forests provided better results. Used Confusion Matrix to test results. Accuracy was 0.999 with an out-of-sample error of approximately of 0.1%.

```
modFitB1 <- randomForest(classe ~. , data=myTraining)
predictionsB1 <- predict(modFitB1, myTesting, type = "class")
confusionMatrix(predictionsB1, myTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##           A 2231    2    0    0    0
##           B    1 1516    2    0    0
##           C    0    0 1366    3    0
##           D    0    0    0 1282    2
##           E    0    0    0    1 1440
##
## Overall Statistics
##
##           Accuracy : 0.9986
##           95% CI : (0.9975, 0.9993)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.9982
##           Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                               Class: A Class: B Class: C Class: D Class: E
## Sensitivity                   0.9996   0.9987   0.9985   0.9969   0.9986
## Specificity                   0.9996   0.9995   0.9995   0.9997   0.9998
## Pos Pred Value                0.9991   0.9980   0.9978   0.9984   0.9993
## Neg Pred Value                0.9998   0.9997   0.9997   0.9994   0.9997
## Prevalence                    0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate                0.2843   0.1932   0.1741   0.1634   0.1835
## Detection Prevalence         0.2846   0.1936   0.1745   0.1637   0.1837
## Balanced Accuracy             0.9996   0.9991   0.9990   0.9983   0.9992
```

MAKING TEST SET PREDICTIONS

Used the model fit on Training to predict the observations in Testing and write the predictions to files for the automatic evaluation.

```
predictionsB2 <- predict(modFitB1, testing, type = "class")
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictionsB2)
```