

MLs

Gabre Hughes

6/11/2020

##Week 4 Assignment: Predicting Exercise Classe from Wearable Devices

##Instructions

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.

##Review criteria

- 1) Your submission for the Peer Review portion should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-).
- 2) Apply your machine learning algorithm to the 20 test cases available in the test data above and submit your predictions in appropriate format to the Course Project Prediction Quiz for automated grading.

###Library Setup

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.1
```

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

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

```
library("randomForest")
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

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

```
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

###Gathering Data

```
download.file(url =
'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv',
              destfile = 'train.csv', , quiet = TRUE)

download.file(url =
'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv',
              destfile = 'test.csv', , quiet = TRUE)

trainw <- read.csv('train.csv')
testw <- read.csv('test.csv')
```

##Preprocessing

First look at the data for each column and remove variables unrelated to exercise (column number and time stamps)

```
str(trainw)

## 'data.frame':    19622 obs. of  160 variables:
## $ X                      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ user_name              : Factor w/ 6 levels "adelmo","carlitos",...: 2
2 2 2 2 2 2 2 2 2 ...
## $ raw_timestamp_part_1   : int  1323084231 1323084231 1323084231
1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232
...
## $ raw_timestamp_part_2   : int  788290 808298 820366 120339 196328
304277 368296 440390 484323 484434 ...
## $ cvtd_timestamp        : Factor w/ 20 levels "02/12/2011 13:32",...: 9
9 9 9 9 9 9 9 9 9 ...
## $ new_window            : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1
1 1 1 ...
## $ num_window            : int  11 11 11 12 12 12 12 12 12 12 ...
## $ roll_belt             : num  1.41 1.41 1.42 1.48 1.48 1.45 1.42 1.42
1.43 1.45 ...
## $ pitch_belt            : num  8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13
8.16 8.17 ...
## $ yaw_belt              : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -
```

```

94.4 -94.4 -94.4 -94.4 ...
## $ total_accel_belt      : int   3 3 3 3 3 3 3 3 3 3 ...
## $ kurtosis_roll_belt    : Factor w/ 397 levels "", "-0.016850", ...: 1 1 1
1 1 1 1 1 1 1 ...
## $ kurtosis_pitch_belt   : Factor w/ 317 levels "", "-0.021887", ...: 1 1 1
1 1 1 1 1 1 1 ...
## $ kurtosis_yaw_belt     : Factor w/ 2 levels "", "#DIV/0!": 1 1 1 1 1 1
1 1 1 1 ...
## $ skewness_roll_belt    : Factor w/ 395 levels "", "-0.003095", ...: 1 1 1
1 1 1 1 1 1 1 ...
## $ skewness_roll_belt.1  : Factor w/ 338 levels "", "-0.005928", ...: 1 1 1
1 1 1 1 1 1 1 ...
## $ skewness_yaw_belt     : Factor w/ 2 levels "", "#DIV/0!": 1 1 1 1 1 1
1 1 1 1 ...
## $ max_roll_belt         : num   NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_belt        : int   NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_belt          : Factor w/ 68 levels "", "-0.1", "-0.2", ...: 1 1
1 1 1 1 1 1 1 1 ...
## $ min_roll_belt         : num   NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_belt        : int   NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_belt          : Factor w/ 68 levels "", "-0.1", "-0.2", ...: 1 1
1 1 1 1 1 1 1 1 ...
## $ amplitude_roll_belt   : num   NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_belt  : int   NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_belt    : Factor w/ 4 levels "", "#DIV/0!", "0.00", ...: 1
1 1 1 1 1 1 1 1 1 ...
## $ var_total_accel_belt  : num   NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_belt         : num   NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_belt      : num   NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_belt         : num   NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_belt        : num   NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_belt     : num   NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_belt        : num   NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_belt          : num   NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_belt       : num   NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_belt          : num   NA NA NA NA NA NA NA NA NA NA ...
## $ gyros_belt_x          : num   0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02
0.03 ...
## $ gyros_belt_y          : num   0 0 0 0 0.02 0 0 0 0 0 ...
## $ gyros_belt_z          : num   -0.02 -0.02 -0.02 -0.03 -0.02 -0.02 -
0.02 -0.02 -0.02 0 ...
## $ accel_belt_x          : int   -21 -22 -20 -22 -21 -21 -22 -22 -20 -21
...
## $ accel_belt_y          : int   4 4 5 3 2 4 3 4 2 4 ...
## $ accel_belt_z          : int   22 22 23 21 24 21 21 21 24 22 ...
## $ magnet_belt_x         : int   -3 -7 -2 -6 -6 0 -4 -2 1 -3 ...
## $ magnet_belt_y         : int   599 608 600 604 600 603 599 603 602 609
...
## $ magnet_belt_z         : int   -313 -311 -305 -310 -302 -312 -311 -313
-312 -308 ...

```

```

## $ roll_arm          : num  -128 -128 -128 -128 -128 -128 -128 -128
-128 -128 ...
## $ pitch_arm         : num   22.5 22.5 22.5 22.1 22.1 22 21.9 21.8
21.7 21.6 ...
## $ yaw_arm           : num   -161 -161 -161 -161 -161 -161 -161 -161
-161 -161 ...
## $ total_accel_arm   : int    34 34 34 34 34 34 34 34 34 34 ...
## $ var_accel_arm     : num    NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_arm   : num    NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_arm     : num    NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_arm  : num    NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_arm     : num    NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_arm    : num    NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
## $ gyros_arm_x       : num    0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02
...
## $ gyros_arm_y       : num    0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -
0.02 -0.03 -0.03 ...
## $ gyros_arm_z       : num   -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -
0.02 ...
## $ accel_arm_x       : int   -288 -290 -289 -289 -289 -289 -289 -289
-288 -288 ...
## $ accel_arm_y       : int    109 110 110 111 111 111 111 111 109 110
...
## $ accel_arm_z       : int   -123 -125 -126 -123 -123 -122 -125 -124
-122 -124 ...
## $ magnet_arm_x      : int   -368 -369 -368 -372 -374 -369 -373 -372
-369 -376 ...
## $ magnet_arm_y      : int    337 337 344 344 337 342 336 338 341 334
...
## $ magnet_arm_z      : int    516 513 513 512 506 513 509 510 518 516
...
## $ kurtosis_roll_arm : Factor w/ 330 levels "", "-0.02438",...: 1 1 1
1 1 1 1 1 1 1 ...
## $ kurtosis_pitch_arm : Factor w/ 328 levels "", "-0.00484",...: 1 1 1
1 1 1 1 1 1 1 ...
## $ kurtosis_yaw_arm   : Factor w/ 395 levels "", "-0.01548",...: 1 1 1
1 1 1 1 1 1 1 ...
## $ skewness_roll_arm  : Factor w/ 331 levels "", "-0.00051",...: 1 1 1
1 1 1 1 1 1 1 ...
## $ skewness_pitch_arm : Factor w/ 328 levels "", "-0.00184",...: 1 1 1
1 1 1 1 1 1 1 ...
## $ skewness_yaw_arm   : Factor w/ 395 levels "", "-0.00311",...: 1 1 1
1 1 1 1 1 1 1 ...
## $ max_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_arm        : int    NA NA NA NA NA NA NA NA NA NA ...

```

```
## $ min_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_arm       : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_arm  : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ roll_dumbbell     : num   13.1 13.1 12.9 13.4 13.4 ...
## $ pitch_dumbbell    : num   -70.5 -70.6 -70.3 -70.4 -70.4 ...
## $ yaw_dumbbell      : num   -84.9 -84.7 -85.1 -84.9 -84.9 ...
## $ kurtosis_roll_dumbbell : Factor w/ 398 levels "", "-0.0035", "-0.0073",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ kurtosis_pitch_dumbbell : Factor w/ 401 levels "", "-0.0163", "-0.0233",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ kurtosis_yaw_dumbbell  : Factor w/ 2 levels "", "#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
## $ skewness_roll_dumbbell : Factor w/ 401 levels "", "-0.0082", "-0.0096",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ skewness_pitch_dumbbell : Factor w/ 402 levels "", "-0.0053", "-0.0084",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ skewness_yaw_dumbbell  : Factor w/ 2 levels "", "#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
## $ max_roll_dumbbell     : num   NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_dumbbell    : num   NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_dumbbell      : Factor w/ 73 levels "", "-0.1", "-0.2",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ min_roll_dumbbell     : num   NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_dumbbell    : num   NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_dumbbell      : Factor w/ 73 levels "", "-0.1", "-0.2",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ amplitude_roll_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
## [list output truncated]

train <- trainw[, 6:ncol(trainw)]
```

Split the data into 70% training and 30% testing set

```
set.seed(23954)
inTrain <- createDataPartition(y = train$classe, p = 0.7, list = F)
training <- train[inTrain, ]
testing <- train[-inTrain, ]
```

Remove the variables with a lot of similarities

```
nzv <- nearZeroVar(train, saveMetrics = T)
keepFeat <- row.names(nzv[nzv$nzv == FALSE, ])
training <- training[, keepFeat]
```

This is a rather stringent cutoff but there is still >50 features after removal

```
training <- training[, colSums(is.na(training)) == 0]
dim(training)
```

```
## [1] 13737    54
```

```
##Model
```

Set up 5-fold cross validation for training

```
modCtl <- trainControl(method = 'cv', number = 5)
```

Fit a model with random forests

```
set.seed(2384)
modRf <- train(classe ~., data = training, method = 'rf', trControl =
modCtl)
```

Read the summary of the model built with random forests

```
predRf <- predict(modRf, newdata = testing)
confusionMatrix(predRf, testing$classe)$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##           A 1673    4    0    0    0
##           B    1 1132    1    0    0
##           C    0    2 1025    2    0
##           D    0    1    0  962    0
##           E    0    0    0    0 1082
```

Predict with the validation set and check the confusion matrix and accuracy

```
confusionMatrix(predRf, testing$classe)$overall[1]
```

```
## Accuracy
## 0.9981308
```

###The accuracy is ~99.6% under 5-fold cross validation

Fit a model with gradient boosting method

```
modGbm <- train(classe ~., data = training, method = 'gbm', trControl =
modCtl, verbose = F)
```

Read the summary of the model built with gbm

```
modGbm$finalModel

## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 53 predictors of which 53 had non-zero influence.

predRfTest <- predict(modRf, newdata = testw)
predRfTest

## [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
```

```
predGbmTest <- predict(modGbm, newdata = testw)
table(predRfTest, predGbmTest)
```

```
##           predGbmTest
## predRfTest A B C D E
##           A 7 0 0 0 0
##           B 0 8 0 0 0
##           C 0 0 1 0 0
##           D 0 0 0 1 0
##           E 0 0 0 0 3
```