Coursera_PML_Project

Ashish Jha

2 September 2017

Reading data into R

```
library(readr)
library(caret)
```

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

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

```
training <- read_csv("pml-training.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_integer(),
##   user_name = col_character(),
##   raw_timestamp_part_1 = col_integer(),
##   raw_timestamp_part_2 = col_integer(),
##   cvtd_timestamp = col_character(),
##   new_window = col_character(),
##   num_window = col_integer(),
##   total_accel_belt = col_integer(),
##   kurtosis_roll_belt = col_character(),
##   kurtosis_picth_belt = col_character(),
##   kurtosis_yaw_belt = col_character(),
##   skewness_roll_belt = col_character(),
##   skewness_roll_belt.1 = col_character(),
##   skewness_yaw_belt = col_character(),
##   max_picth_belt = col_integer(),
##   max_yaw_belt = col_character(),
##   min_pitch_belt = col_integer(),
##   min_yaw_belt = col_character(),
##   amplitude_pitch_belt = col_integer(),
##   amplitude_yaw_belt = col_character()
##   # ... with 46 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
```

```
## Warning: 185 parsing failures.
## row # A tibble: 5 x 5 col     row               col expected  actual               file ex
pected   <int>             <chr>    <chr>   <chr>              <chr> actual 1  2231 kurtosis_
roll_arm a double #DIV/0! 'pml-training.csv' file 2  2231 skewness_roll_arm a double #DIV/0!
 'pml-training.csv' row 3  2255 kurtosis_roll_arm a double #DIV/0! 'pml-training.csv' col 4
 2255 skewness_roll_arm a double #DIV/0! 'pml-training.csv' expected 5  2282 kurtosis_roll_ar
m a double #DIV/0! 'pml-training.csv'
## ... .................. ... ..........................................................
 ........ .................................................................. ......
 ............................................................... ....
 ............................................................... ...
 ............................................................... ...
 ............................................................... ........
 ............................................................
## See problems(...) for more details.
```

```
pml_testing <- read_csv("pml-testing.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   X1 = col_integer(),
##   raw_timestamp_part_1 = col_integer(),
##   raw_timestamp_part_2 = col_integer(),
##   num_window = col_integer(),
##   roll_belt = col_double(),
##   pitch_belt = col_double(),
##   yaw_belt = col_double(),
##   total_accel_belt = col_integer(),
##   gyros_belt_x = col_double(),
##   gyros_belt_y = col_double(),
##   gyros_belt_z = col_double(),
##   accel_belt_x = col_integer(),
##   accel_belt_y = col_integer(),
##   accel_belt_z = col_integer(),
##   magnet_belt_x = col_integer(),
##   magnet_belt_y = col_integer(),
##   magnet_belt_z = col_integer(),
##   roll_arm = col_double(),
##   pitch_arm = col_double(),
##   yaw_arm = col_double()
##   # ... with 37 more columns
## )
## See spec(...) for full column specifications.
```

removing columns which have more than 90% missing values

```
training<- training[, -which(colMeans(is.na(training)) > 0.9)]
pml_testing<- pml_testing[, -which(colMeans(is.na(pml_testing)) > 0.9)]
```

Coverting classe in dataset as dummy variables because classe is of character dataset and would pose difficulty in further analysis.

```
dummy <- training$classe
dummy[dummy == "A"] <- 1
dummy[dummy == "B"] <- 2
dummy[dummy == "C"] <- 3
dummy[dummy == "D"] <- 4
dummy[dummy == "E"] <- 5

dummy <- as.numeric(dummy)

training$dummy <- dummy
```

Spliting dataset into training set and validation set

```
part <- createDataPartition(y=training$classe, p=0.7, list = FALSE)
train <- training[part,]
valid <- training[-part,]
```

Checking datatype of varibles in dataset since variables with character datatype won't work with randomforest

```
str(train)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    13737 obs. of  61 variables:
##  $ X                    : int  1 2 3 5 6 8 9 10 12 13 ...
##  $ user_name            : chr  "carlitos" "carlitos" "carlitos" "carlitos" ...
##  $ raw_timestamp_part_1: int  1323084231 1323084231 1323084231 1323084232 1323084232 13230
84232 1323084232 1323084232 1323084232 1323084232 ...
##  $ raw_timestamp_part_2: int  788290 808298 820366 196328 304277 440390 484323 484434 5283
16 560359 ...
##  $ cvtd_timestamp       : chr  "05/12/2011 11:23" "05/12/2011 11:23" "05/12/2011 11:23" "0
5/12/2011 11:23" ...
##  $ new_window           : chr  "no" "no" "no" "no" ...
##  $ num_window           : int  11 11 11 12 12 12 12 12 12 12 ...
##  $ roll_belt            : num  1.41 1.41 1.42 1.48 1.45 1.42 1.43 1.45 1.43 1.42 ...
##  $ pitch_belt           : num  8.07 8.07 8.07 8.07 8.06 8.13 8.16 8.17 8.18 8.2 ...
##  $ 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 ...
##  $ gyros_belt_x         : num  0 0.02 0 0.02 0.02 0.02 0.02 0.03 0.02 0.02 ...
##  $ gyros_belt_y         : num  0 0 0 0.02 0 0 0 0 0 0 ...
##  $ gyros_belt_z         : num  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 0 -0.02 0 ...
##  $ accel_belt_x         : int  -21 -22 -20 -21 -21 -22 -20 -21 -22 -22 ...
##  $ accel_belt_y         : int  4 4 5 2 4 4 2 4 2 4 ...
##  $ accel_belt_z         : int  22 22 23 24 21 21 24 22 23 21 ...
##  $ magnet_belt_x        : int  -3 -7 -2 -6 0 -2 1 -3 -2 -3 ...
##  $ magnet_belt_y        : int  599 608 600 600 603 603 602 609 602 606 ...
##  $ magnet_belt_z        : int  -313 -311 -305 -302 -312 -313 -312 -308 -319 -309 ...
##  $ 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 21.8 21.7 21.6 21.5 21.4 ...
##  $ 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 ...
##  $ gyros_arm_x          : num  0 0.02 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 ...
##  $ gyros_arm_y          : num  0 -0.02 -0.02 -0.03 -0.03 -0.02 -0.03 -0.03 -0.03 -0.02 ...
##  $ gyros_arm_z          : num  -0.02 -0.02 -0.02 0 0 0 -0.02 -0.02 0 -0.02 ...
##  $ accel_arm_x          : int  -288 -290 -289 -289 -289 -289 -288 -288 -288 -287 ...
##  $ accel_arm_y          : int  109 110 110 111 111 111 109 110 111 111 ...
##  $ accel_arm_z          : int  -123 -125 -126 -123 -122 -124 -122 -124 -123 -124 ...
##  $ magnet_arm_x         : int  -368 -369 -368 -374 -369 -372 -369 -376 -363 -372 ...
##  $ magnet_arm_y         : int  337 337 344 337 342 338 341 334 343 338 ...
##  $ magnet_arm_z         : int  516 513 513 506 513 510 518 516 520 509 ...
##  $ roll_dumbbell        : num  13.1 13.1 12.9 13.4 13.4 ...
##  $ pitch_dumbbell       : num  -70.5 -70.6 -70.3 -70.4 -70.8 ...
##  $ yaw_dumbbell         : num  -84.9 -84.7 -85.1 -84.9 -84.5 ...
##  $ total_accel_dumbbell: int  37 37 37 37 37 37 37 37 37 37 ...
##  $ gyros_dumbbell_x     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ gyros_dumbbell_y     : num  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
 ...
##  $ gyros_dumbbell_z     : num  0 0 0 0 0 0 0 0 0 -0.02 ...
##  $ accel_dumbbell_x     : int  -234 -233 -232 -233 -234 -234 -232 -235 -233 -234 ...
##  $ accel_dumbbell_y     : int  47 47 46 48 48 46 47 48 47 48 ...
##  $ accel_dumbbell_z     : int  -271 -269 -270 -270 -269 -272 -269 -270 -270 -269 ...
##  $ magnet_dumbbell_x    : int  -559 -555 -561 -554 -558 -555 -549 -558 -554 -552 ...
##  $ magnet_dumbbell_y    : int  293 296 298 292 294 300 292 291 291 302 ...
##  $ magnet_dumbbell_z    : int  -65 -64 -63 -68 -66 -74 -65 -69 -65 -69 ...
##  $ roll_forearm         : num  28.4 28.3 28.3 28 27.9 27.8 27.7 27.7 27.5 27.2 ...
##  $ pitch_forearm        : num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 -63.8 -63.9
 ...
##  $ yaw_forearm          : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -151 ...
##  $ total_accel_forearm : int  36 36 36 36 36 36 36 36 36 36 ...
```

```
##  $ gyros_forearm_x     : num  0.03 0.02 0.03 0.02 0.02 0.02 0.03 0.02 0.02 0 ...
##  $ gyros_forearm_y     : num  0 0 -0.02 0 -0.02 -0.02 0 0 0.02 0 ...
##  $ gyros_forearm_z     : num  -0.02 -0.02 0 -0.02 -0.03 0 -0.02 -0.02 -0.03 -0.03 ...
##  $ accel_forearm_x     : int  192 192 196 189 193 193 193 190 191 193 ...
##  $ accel_forearm_y     : int  203 203 204 206 203 205 204 205 203 205 ...
##  $ accel_forearm_z     : int  -215 -216 -213 -214 -215 -213 -214 -215 -215 -215 ...
##  $ magnet_forearm_x    : int  -17 -18 -18 -17 -9 -9 -16 -22 -11 -15 ...
##  $ magnet_forearm_y    : int  654 661 658 655 660 660 653 656 657 655 ...
##  $ magnet_forearm_z    : int  476 473 469 473 478 474 476 473 478 472 ...
##  $ classe              : chr  "A" "A" "A" "A" ...
##  $ dummy               : num  1 1 1 1 1 1 1 1 1 1 ...
```

Using random forest to train the using randomforest with 500 trees. Used randomforest as it is work horse , tried and tested. RandomForest works well all the time.

```
fit_train <- randomForest(formula = dummy ~ ., data = train[,-
c(1,2,5,6,60)],importance=TRUE,na.action=na.omit,ntree = 500,mtry=100)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within
## valid range
```

Prediciting on validation set

```
pred_valid <- predict(fit_train, newdata = valid[,-c(1,2,5,6,60)])
pred_valid_round <- round(pred_valid, digits = 0)
table(pred_valid_round, valid$dummy)
```

```
##
## pred_valid_round    1    2    3    4    5
##                1 1670    1    1    0    0
##                2    3 1137    8    3    0
##                3    1    1 1017    5    2
##                4    0    0    0  955    8
##                5    0    0    0    1 1072
```

Test results were overwhelming with accuracy of 99.67% . Hence i will accept this model for final prediction

```
pred_test <- predict(fit_train, newdata= pml_testing[,-c(1,2,5,6,60)])
pred_test_round <- round(pred_test, digits = 0)
pred_test_round
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  2  1  2  1  1  5  4  2  1  1  2  3  2  1  5  5  1  2  2  2
```