Applied ML Project

Boris Borodyansky

23 сентября 2017 г

Intro and Sinopsis

This paper deals with creating machine learnign algorythm, that should predict how certain people perform physycal exercices based on how they performed them earlier.

We will use caret and random forest package and a dataset provided by Coursera.

About the data set and summaries

So, let's perform some xploratory analyses before we move forward. What we are interested in are: - NAs - How variables are structrured all in all

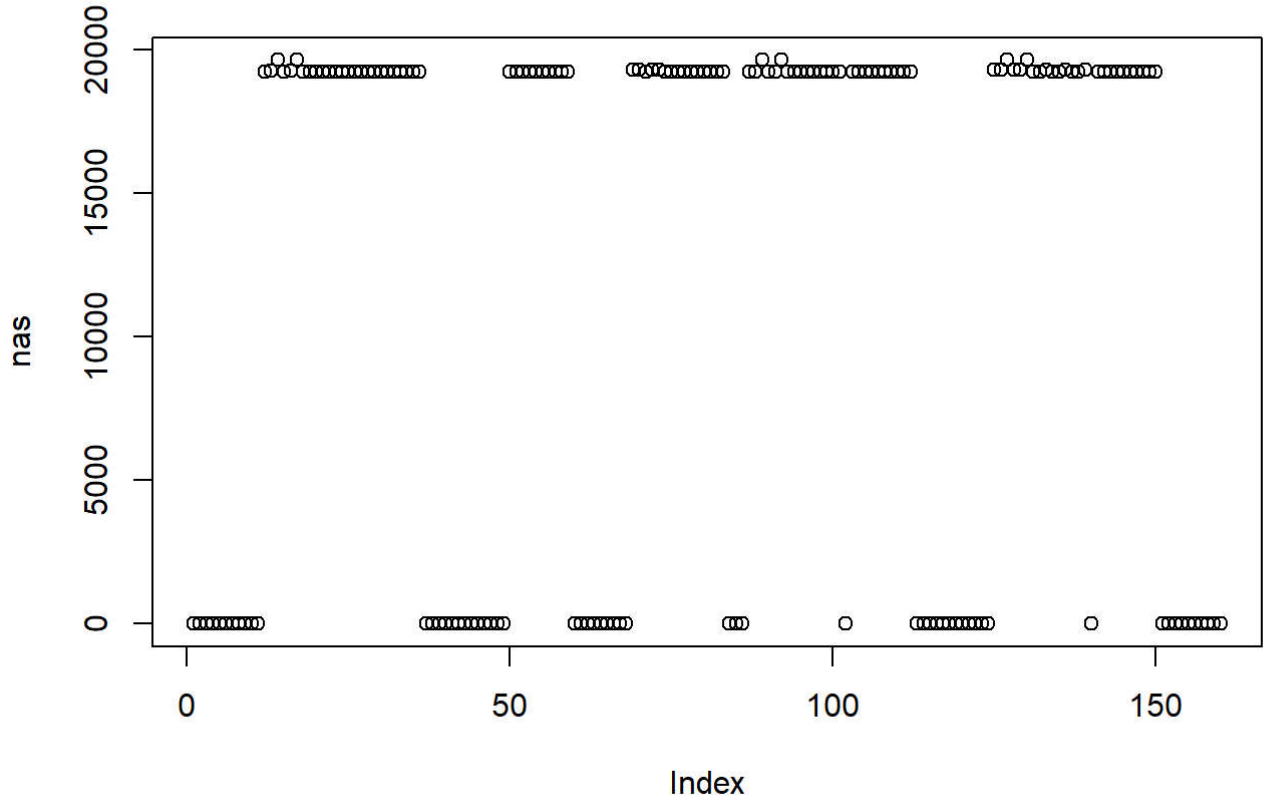
Here is some data on amount of NAs in different columns of the data set.

```
nas = c()
for(i in 1:length(names(data_set)))
  nas = c(nas, length(which(is.na(data_set[,i])==TRUE)))

table(nas)
```

```
## nas
##     0 19216 19217 19218 19220 19221 19225 19226 19227 19248 19293 19294
##    60    67     1     1     1     4     1     4     2     2     1     1
## 19296 19299 19300 19301 19622
##     2     1     4     2     6
```

```
plot(nas)
```



We can see, that there are variables that have no NAs, and those that do now have almost any meaningful values. So we cannot substitute those meaningsless variables, and have to get rid of them.

Thus we create a new data set with all meaningful columns.

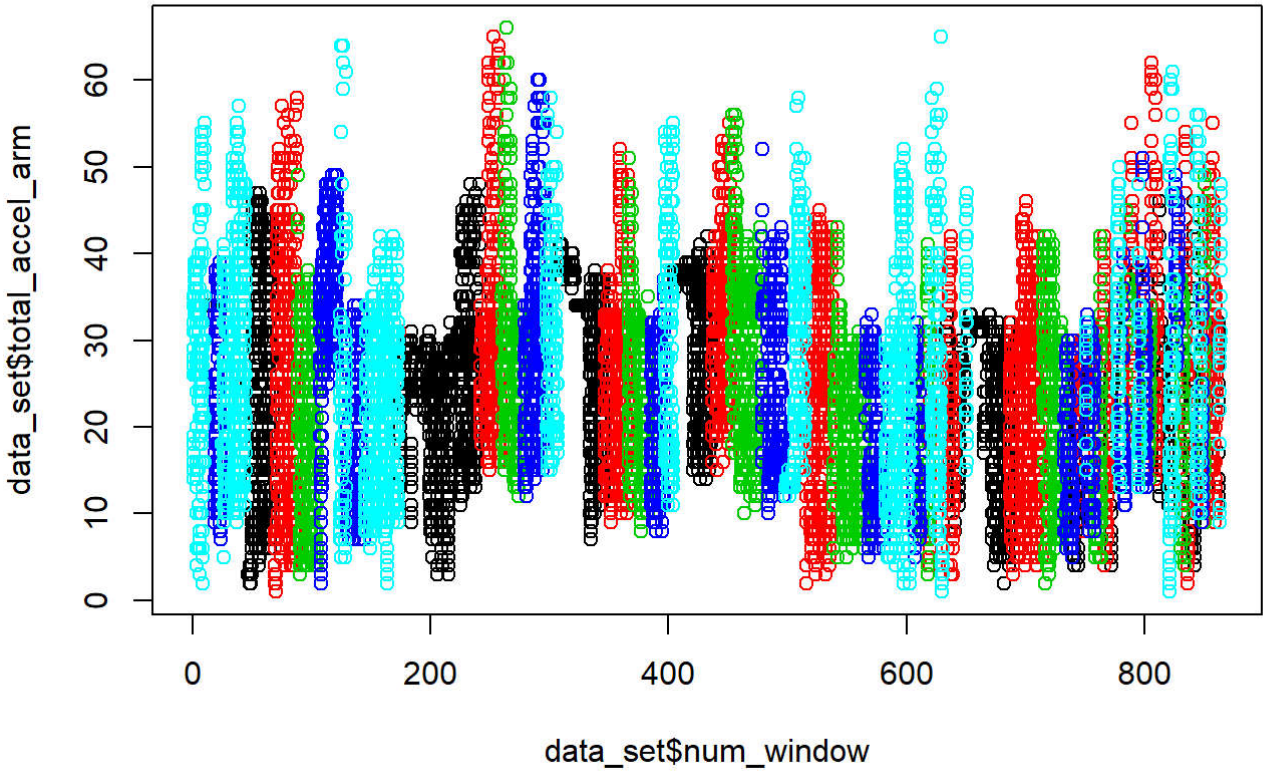
```
# get rid of NAs
no_na = c()
for(i in 1:length(names(data_set))){
  if(length(data_set[is.na(data_set[,i]),i])>(length(data_set$X)*0.60))
    no_na = c(no_na,FALSE)
  else
    no_na = c(no_na, TRUE)
}

data_set = data_set[,no_na]
```

More exploratory analyses

Let's see if the data that's left has any graphically noticable pattern.

```
plot(data_set$num_window,data_set$total_accel_arm, col = factor(data_set$classe))
```



It seems, that linear regression or other simple techiques won't be able to train and predict a model based on this data. So we'd rather stick to more complex models, like random forest.

Let's have a look at the variables that's left.

```
# exploratory analysis of variables
sort(names(data_set))
```

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

It seems, that not all variables form “complete” sets of snsors (like, pitch for arm, forearm, belt and dumbbel). Forthe model I consider choosinf those that have complete sets, and those thar do not deal with other aspects (time, name, id and so on)

Process data for ML model

Now we are going to create data partitions and choose column for model. I decided to choose 32 columns with full sets of sensors and no total. We will also use control method of reputation.

```
# preparing for analysys

inTrain = createDataPartition(y=data_set$classe, p=0.6, list = FALSE)
training = data_set[inTrain,]
testing = data_set[-inTrain,]

# building up prediction model
ctrl = trainControl(method = "repeatedvc", number = 10, repeats = 3)
set.seed(7)

TrainD = training[,c(8:10,11:20,25:33,50:59)]
TrainC = training[,60]
```

Model training and prediction

Here we use random forest method for our model. We also predict it on the test set we created earlier.

```
model33 = train(TrainD, TrainC,method = "rf")
pred = predict(model33, testing)
```

Let’s check prediction model.

```
# check the prediction model
check_pred = c()
for(i in 1:length(pred))
  check_pred = c(check_pred,testing$classe[i] == pred[i])

summary(check_pred)
```

```
##    Mode   FALSE    TRUE    NA's
## logical     135    7711       0
```

It seems our model is trained alright.

Hope it works alright on the Test Quiz Also =)