```
# options(rpubs.upload.method = 'internal')
```

Predicting the manner how people exercise

Executive Summary

The goal of this work is to develop a predictive model that would allow to estimate well the people do weight lifting exercises based on set of different activity monitors. The data is vailable at https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv . Anlother task was to make predictions for 2 sample acivities (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv).

For this task the predictive model based on the Random Forest algorithm was developed. Due to the computational difficulties, the model was trained only on the 6% of the trining set (more that 1000 observ ation). Even though, cross-validating model onthe remaining data (over 18,000 observations) gave 92% accuracy. Preductions on the 20 given samples gave 90% accuracy (18 correct results)

Exploratory Analysis

The data sets contain around 160 variables (acivitity monitors). However just by looking at them we see that 100+ of these variables have no values for most of observations as well as for all 20 test cases (pml-testing data set) we are supposed to predict. Evidently these variables will just create extra "noise" and are useless for prediction in our 20 cases. So I decided to exclude them.

As well I decided to exclude the new_window since it has value "no" for all records in the testing set. However, in this cade we also removed records with value "yes" from the consideration in the training set.

Finally, I decided not to use for prediction few more variables that seem to be kept for the "housekeeping" purpose rather than have predictive value (like timestamps, subject name).

Still after removing all these variables, there were around 50 predictors left.

Predictive Model

I decided to use the Randow Forest method that is proven to work well with large number of predictors.

The origial plan was to split the training set (training-pml) into training and testing subsets with aroun 70% of observations kept into training set, train the model on the training set and cross validate on the test set (there is a naming confusion; this testing set is different from 20 samples provided in pms-testing.csv).

However attempts to train the model on the 70% training set failed. As was suggested in the Discussion Forum, I decided to use smaller training test sets. I started with 1% and made several runs with larger sizes until 8%. In all cases cases, there was perfect fit for the training set (eveidently just because it was small); the error rate for the testing/cross-validation set decreased with the size: 1% set gave 31% error rate, it decreared to 18% for 2% size, 11% for 4% size and to 8% for the 6% sample size.

Since the computation time to train the model substantially grew with the size of the training set and the training set already included over 1000 observations, I decided to stop at this point even though further increase of the test set size very likely would allow to increase accuracy.

Applying the resulted model to 20 cases from pml-testing.csv gave 2 errors (90% accuracy) that is also consisted with 8% error rate (92% accuracy) estimated during cross-validation.

Below is the R code:

Read training set and subsetting on the new_window vsariable

```
training.set <- read.csv("pml-training.csv", header = TRUE, sep = ",")
training.1 <- subset(training.set, new_window == "no")
dim(training.1)
```

```
## [1] 19216   160
```

Selecting the variables we decide to keep

```
nm <- c ("X", ## "user_name",  ## "num_window",
"roll_belt",    "pitch_belt",   "yaw_belt", "total_accel_belt",
"gyros_belt_x", "gyros_belt_y",
"gyros_belt_z", "accel_belt_x", "accel_belt_y",
"accel_belt_z", "magnet_belt_x",    "magnet_belt_y",
"magnet_belt_z",    "roll_arm", "pitch_arm",
"yaw_arm",  "total_accel_arm",
"gyros_arm_x",  "gyros_arm_y",  "gyros_arm_z",
"accel_arm_x",  "accel_arm_y",  "accel_arm_z",
"magnet_arm_x", "roll_dumbbell",    "pitch_dumbbell",
"yaw_dumbbell", "gyros_dumbbell_x", "gyros_dumbbell_y",
"gyros_dumbbell_z", "accel_dumbbell_x", "accel_dumbbell_y",
"accel_dumbbell_z", "magnet_dumbbell_x",    "magnet_dumbbell_y",
"magnet_dumbbell_z",    "roll_forearm", "pitch_forearm",
"yaw_forearm",  "gyros_forearm_x",  "gyros_forearm_y",
"gyros_forearm_z",  "accel_forearm_x",  "accel_forearm_y",
"accel_forearm_z",  "magnet_forearm_x", "magnet_forearm_y",
"magnet_forearm_z", "classe")



col_list <- c(1)
for (i in 2:length( names(training.1)) ) {
if (names(training.1)[i] %in% nm)
  { col_list <- append(col_list,i)
 }
}

training.2 <- subset(training.1, select= col_list)

dim(training.2)
```

```
## [1] 19216    50
```

Splitting data set

```
library(caret)
```

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

```
library(kernlab)
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```


set.seed(2833)
trainIndex = createDataPartition(y = training.2$classe, p = 0.06, list = FALSE)
trainSet <- training.2[trainIndex, ]
testSet <- training.2[-trainIndex, ]
```

Training model

```
Sys.time()
```

```
## [1] "2014-08-24 14:18:43 PDT"
```

```
modFit <- train(classe ~ . - X, method = "rf", data = trainSet, preProcess = c("center",
    "scale"), prox = TRUE, importance = TRUE)
Sys.time()
```

```
## [1] "2014-08-24 14:29:36 PDT"
```

Estimations on the training set

```
importance(modFit$finalModel)[, 7]
```

```
##         roll_belt        pitch_belt          yaw_belt  total_accel_belt
##           111.463            44.671            46.620             6.700
##       gyros_belt_x      gyros_belt_y      gyros_belt_z      accel_belt_x
##             5.135             4.293             7.608             9.547
##       accel_belt_y      accel_belt_z     magnet_belt_x     magnet_belt_y
##             5.168            19.204            19.498            20.891
##     magnet_belt_z          roll_arm         pitch_arm           yaw_arm
##            19.987            13.454             7.747             8.828
##    total_accel_arm       gyros_arm_x       gyros_arm_y       gyros_arm_z
##             6.352             8.270             8.940             5.009
##        accel_arm_x       accel_arm_y       accel_arm_z      magnet_arm_x
##            11.541             6.724             8.110            10.810
##     roll_dumbbell    pitch_dumbbell      yaw_dumbbell  gyros_dumbbell_x
##            22.750            11.249            13.735             6.355
##   gyros_dumbbell_y  gyros_dumbbell_z  accel_dumbbell_x  accel_dumbbell_y
##            12.436             6.764             9.397            27.193
##   accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z
##            14.851            24.177            63.934            48.080
##       roll_forearm     pitch_forearm       yaw_forearm   gyros_forearm_x
##            53.605            72.226             8.566             7.763
##    gyros_forearm_y   gyros_forearm_z   accel_forearm_x   accel_forearm_y
##             6.107             6.108            19.604             8.160
##    accel_forearm_z  magnet_forearm_x  magnet_forearm_y  magnet_forearm_z
##            14.458            10.385            11.142            17.533
```

```
predictions <- predict(modFit, newdata = trainSet)
sum(trainSet$classe != predictions)/length(trainSet$classe)
```

```
## [1] 0
```

Cross validation on the test set

```
predictions <- predict(modFit, newdata = testSet)
sum(testSet$classe != predictions)/length(testSet$classe)
```

```
## [1] 0.08754
```

Predictions for the 20 cases from pml-testing,csv

```
testing.set <- read.csv("pml-testing.csv", header = TRUE, sep = ",")
predictions <- predict(modFit, newdata = testing.set)
```

Conclusion

This model allows to predict the manner in which people did the exercise with pretty hight degree of accuracy (92%). However, very likely the accuracy can be improved with using more computational resources.