Practical Machine Learning - Submission

Objective

The goal of this work is to predict from data collected with quantified self devices if people are performing barbell lifts correctly or incorrectly.

Librairies & seed

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.2
```

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

```
set.seed(1234)
```

Collect data and build data sets

```
url_csv_train="http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
url_csv_test="http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

raw_data_train <- read.csv(url_csv_train)
raw_data_test <- read.csv(url_csv_test)



inTrain <- createDataPartition(raw_data_train$classe, p = 0.7, list = FALSE)
training <- raw_data_train[ inTrain,]
testing <- raw_data_train[-inTrain,]
```

Features

By looking at the data (using str), we can see that a majority of variables are filled with a large majority of NA, or are empty. We decide to eliminate: - these variables with a large majority of NA or empty values - the first columns (X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, new_window, num_window) which may not give information on how well the exercise is performed

```
features=c("roll_belt","pitch_belt","yaw_belt","total_accel_belt","gyros_belt_x","gyros_belt_y","g
yros_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","magnet_arm_y","magnet_arm_z","roll_du
mbbell","pitch_dumbbell","yaw_dumbbell","total_accel_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","total_accel
_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")
```

At the end, we keep the variables which seems to be the measures: - roll - pitch - yaw - total accel - gyros x - gyros y - gyros z - accel x - accel y - accel z - magnet x - magnet y - magnet z

for different parts/captors: - belt - arm - dumbbell - forearm

Model

We choose “randomForest” algorithm and to perform 5-fold cross-validation.

```
control1 <- trainControl(method = "cv", number = 5, allowParallel = TRUE)
modelFit <- train(classe~.,data=training[,append(features,"classe")],trControl = control1,method="
rf")
```

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

```
## Warning: package 'randomForest' was built under R version 3.1.2
```

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

Test - out of sample error

We use our own testing data set to determine the out of sample error.

```
predictions<-predict(modelFit,newdata=testing)
confusionMatrix(predictions,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    9    0    0    0
##          B    1 1129    7    0    0
##          C    0    1 1018   10    0
##          D    0    0    1  953    1
##          E    0    0    0    1 1081
##
## Overall Statistics
##
##                Accuracy : 0.995
##                  95% CI : (0.993, 0.996)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.993
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.999    0.991    0.992    0.989    0.999
## Specificity             0.998    0.998    0.998    1.000    1.000
## Pos Pred Value          0.995    0.993    0.989    0.998    0.999
## Neg Pred Value          1.000    0.998    0.998    0.998    1.000
## Prevalence              0.284    0.194    0.174    0.164    0.184
## Detection Rate          0.284    0.192    0.173    0.162    0.184
## Detection Prevalence    0.286    0.193    0.175    0.162    0.184
## Balanced Accuracy       0.999    0.995    0.995    0.994    0.999
```

Predict the values for the test set

```
answers=predict(modelFit,newdata=raw_data_test)
answers
```

```
##  [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
```

```
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(answers)
```