Practical Machine Learning

```
library(data.table)
library(caret)
```

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

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

```
library(e1071)
library(rmarkdown)
```

preparing data

```
# setwd("/Users/LIU/Desktop/Machine_Learning_R/project/Course_project/data/")
pml_train <- fread("pml-training.csv")

## view data globaux information
# str(pml_train)
# dim(pml_train)

## drop the NAs
Count_NA <- data.frame( Nb_NA=apply(pml_train, 2, function(x) sum(is.na(x))) )
pml_train <- pml_train[, (Count_NA$Nb_NA==0), with=FALSE]
pml_train <- na.omit(pml_train) # most lines are still there, so I keep this way
pml_train <- pml_train[, -c(1:7)]

pml_train$classe <- as.factor(pml_train$classe )
```

Modelling

```
inTrain = createDataPartition(pml_train$classe, p = 0.75)[[1]]
training <- pml_train[inTrain]
testing <- pml_train[-inTrain]
rm(pml_train)

##--- svm : the fastest
set.seed(12345)
mod_svm <- svm(classe ~ ., data = training)
# summary(mod_svm)
pred_svm_train <- predict(mod_svm, training)
table(pred_svm_train, training$classe)
```

```
##
## pred_svm_train    A    B    C    D    E
##               A 4180  174    0    3    0
##               B    1 2644   53    0    6
##               C    4   29 2495  218   50
##               D    0    0   15 2190   55
##               E    0    1    4    1 2595
```

```
pred_svm_test <- predict(mod_svm, testing)
accuracy_svm <- confusionMatrix(pred_svm_test, testing$classe)$overall[1]

##--- random fororest
set.seed(12345)
controlRF <- trainControl(method="cv", number=3, verboseIter=FALSE)
mod_rf <- train(classe ~ ., data=training, method="rf", trControl=controlRF)
# mod_rf$finalModel
pred_rf_test <- predict(mod_rf, testing)
accuracy_rf <- confusionMatrix(pred_rf_test, testing$classe)$overall[1]
# varImpPlot(fit,type=2)

##--- gbm
set.seed(12345)
controlGbm <- trainControl(method = "repeatedcv", number = 3, repeats = 1)
mod_gbm  <- train(classe ~ ., data=training, method = "gbm", trControl = controlGb
m, verbose = FALSE)
mod_gbm$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 52 predictors of which 39 had non-zero influence.
```

```
pred_gbm_test <- predict(mod_gbm, testing)
accuracy_gbm <- confusionMatrix(pred_gbm_test, testing$classe)$overall[1]
```

Accuracy values of the 3 methodes :

```
print( paste0( "SVM :  ", accuracy_svm ) )
```

```
## [1] "SVM :  0.948817292006525"
```

```
print( paste0( "Rf :  ", accuracy_rf ) )
```

```
## [1] "Rf :  0.994290375203915"
```

```
print( paste0( "Gbm :  ", accuracy_gbm ) )
```

```
## [1] "Gbm :  0.965130505709625"
```

So the best methode is the random forest 0.99, and I use it to do the prediction.

Prediction

```
pml_test <- fread("pml-testing.csv")
pml_test <- pml_test[, (Count_NA$Nb_NA==0), with=FALSE]
pml_test <- na.omit(pml_test)
pml_test <- pml_test[, -c(1:7)]
pred_rf_test_res <- predict(mod_rf, pml_test)
```

The predicted values are:

```
pred_rf_test_res
```

```
##  [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
```

```
## save the resultat in a csv file
pml_test_result <- cbind(pml_test, predict = pred_rf_test_res)
fwrite(pml_test_result, file= "pml_test_result.csv" )
```