title: “Machine learning course project”
author: “Sergei Keidzh”
date: “12 July 2018”

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here:http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

loading libraries

```
setwd("D:/Game")
library(caret)
library(ggplot2)
library(randomForest)
library(rpart)
set.seed(123456)
```

Downloading the training and testing data

```
library(data.table)
training1 <- read.csv("D:/Game/pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testing1 <- read.csv('D:/Game/pml-testing.csv', na.strings=c("NA","#DIV/0!", ""))
str(training1)
```

To continue our analysis and for future predicting, we subset training.data for cross validation

```
summary(training1$classe)
training1<-training1[,colSums(is.na(training1)) == 0]
```

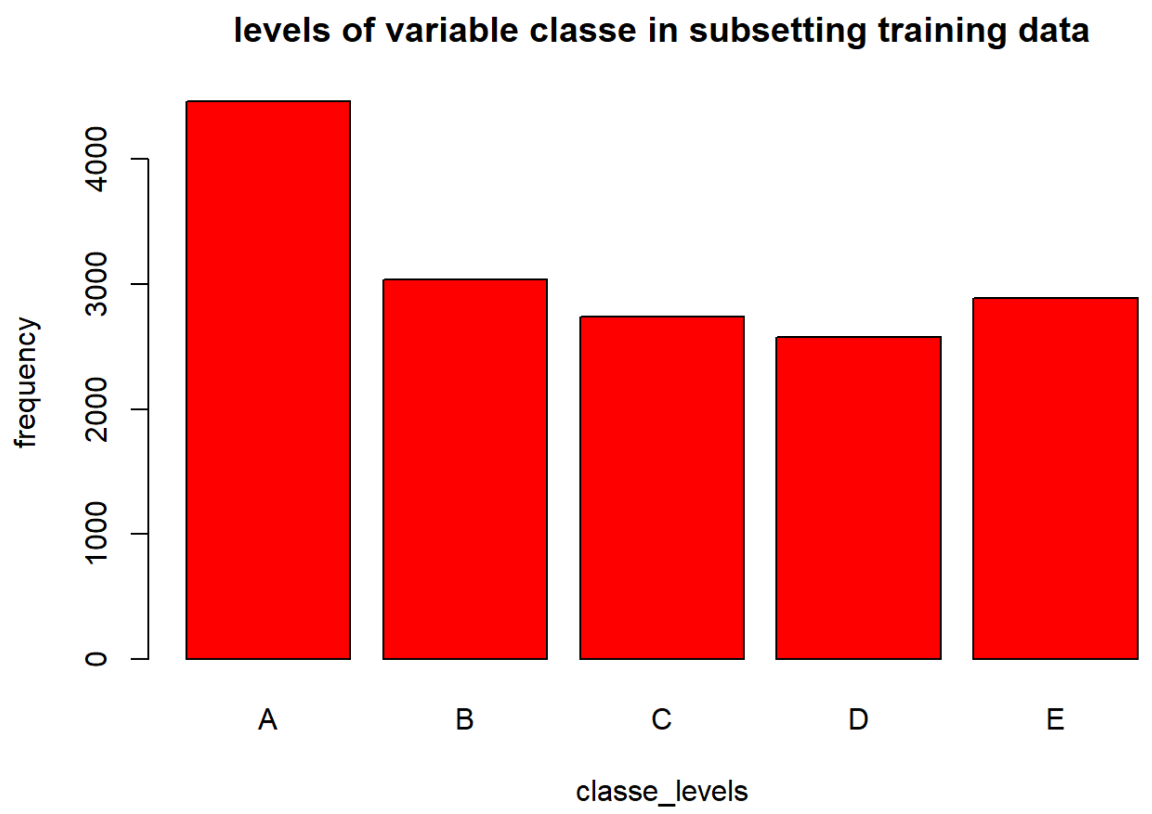
```
testing1<-testing1[,colSums(is.na(testing1)) == 0]
training1   <-training1[,-c(1:7)]
testing1 <-testing1[,-c(1:7)]
dim(testing1)
## [1] 20 53
dim(training1)
## [1] 19622    53
```

Above we splitted the training.data and testing.data

```
inTrain<- createDataPartition(y=training1$classe, p=0.8, list=FALSE)
subtraining <- training1[inTrain, ]
subtesting <- training1[-inTrain, ]
```

Now take a look at the data

```
plot(subtraining$classe, col="red", main="levels of variable classe in subset
ting training data", xlab="classe_levels", ylab="frequency")
```



Create a prediction

```
pred1<-rpart(classe ~ ., data=subtraining, method="class")
```

```r
prediction<-predict(pred1, subtesting, type = "class")
```

Now use the Matrix

```r
confusionMatrix(prediction,subtesting$classe)
```

```
## Confusion Matrix and Statistics
##           Reference
## Prediction    A    B    C    D    E
##          A 1046  157   12   81   26
##          B   21  416   74   38   57
##          C   24   97  524   53   51
##          D   15   50   55  421   54
##          E   10   39   19   50  533
## Overall Statistics
##                Accuracy : 0.7494
##                  95% CI : (0.7356, 0.7629)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##                   Kappa : 0.681
##  Mcnemar's Test P-Value : < 2.2e-16
## Statistics by Class:
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9373   0.5481   0.7661   0.6547   0.7393
## Specificity            0.9017   0.9399   0.9305   0.9470   0.9631
## Pos Pred Value         0.7912   0.6865   0.6996   0.7076   0.8187
## Neg Pred Value         0.9731   0.8966   0.9496   0.9333   0.9425
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2666   0.1060   0.1336   0.1073   0.1359
## Detection Prevalence   0.3370   0.1545   0.1909   0.1517   0.1659
## Balanced Accuracy      0.9195   0.7440   0.8483   0.8008   0.8512
```

Create the second prediction, this time using randomForest

```r
pred2 <- randomForest(classe ~. , data=subtraining,importance = TRUE, method=
"class",ntrees = 10, na.action = na.pass)
prediction.random<- predict(pred2, subtesting, type = "class")
confusionMatrix(prediction.random, subtesting$classe)
```

```
## Confusion Matrix and Statistics
##           Reference
## Prediction    A    B    C    D    E
```

```
##           A 1116    6    0    0    0
##           B    0  752    5    0    0
##           C    0    1  679    4    0
##           D    0    0    0  639    3
##           E    0    0    0    0  718
## Overall Statistics
##                Accuracy : 0.9952
##                  95% CI : (0.9924, 0.9971)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##                   Kappa : 0.9939
##  Mcnemar's Test P-Value : NA
## Statistics by Class:
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9908   0.9927   0.9938   0.9958
## Specificity            0.9979   0.9984   0.9985   0.9991   1.0000
## Pos Pred Value         0.9947   0.9934   0.9927   0.9953   1.0000
## Neg Pred Value         1.0000   0.9978   0.9985   0.9988   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1917   0.1731   0.1629   0.1830
## Detection Prevalence   0.2860   0.1930   0.1744   0.1637   0.1830
## Balanced Accuracy      0.9989   0.9946   0.9956   0.9964   0.9979
```

As we see those two predictions have different results. To compare with first prediction Random forest shows better algorithm. Accuracy of first prediction is 0.739, for Random Forest is 0.995. It is better to choose RandomForest model. Out-of-sample error is estimated at 0.005, (it means 0.5%).

The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set.Testing data set comprises 20 cases. With an accuracy above 99% on our cross-validation data, we can expect that just some , or none, of the test samples will be missclassified

Try to using randomForest algorithm in the original testing data

```
pred3 <- predict(pred2, testing1, type="class")
pred3
```

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

Submission

```
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(pred3)
```