# Practical Machine Learning Project:

## Objective

The goal of the project is to build a Model to predict how well individuals do barbell lifts using as input data from accelerometers on the belt, forearm, arm, and dumbbell of six participants.

## Data

The Weight Lifting Exercises Dataset for this project come from this source:[http://groupware.les.inf.puc-rio.br/har](http://groupware.les.inf.puc-rio.br/har).

- The training data for this project are available here:

[https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv](https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

- The test data are available here:

[https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv](https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

For detailed information see this paper:

*Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.*

## Steps

### 0. Reading data and creating of two datasets for training and testing model

The dataset “pml-training.csv” is splitted into a training and test set. The training set (pmlTrain) contains 75% of the observations while testing set(pmlTest) contains the remaining 25%.

```
pml_Training <- read.csv("pml-training.csv")
library(caret)
intrain <- createDataPartition(y=pml_Training$classe, p=0.75, list=FALSE)
pmlTrain <- pml_Training[intrain,]
pmlTest <- pml_Training[-intrain,]
```

### 1. Predictors selection: choose the best set of variables for the fit model

- First, it was removed any variable which contained missing values (NA).

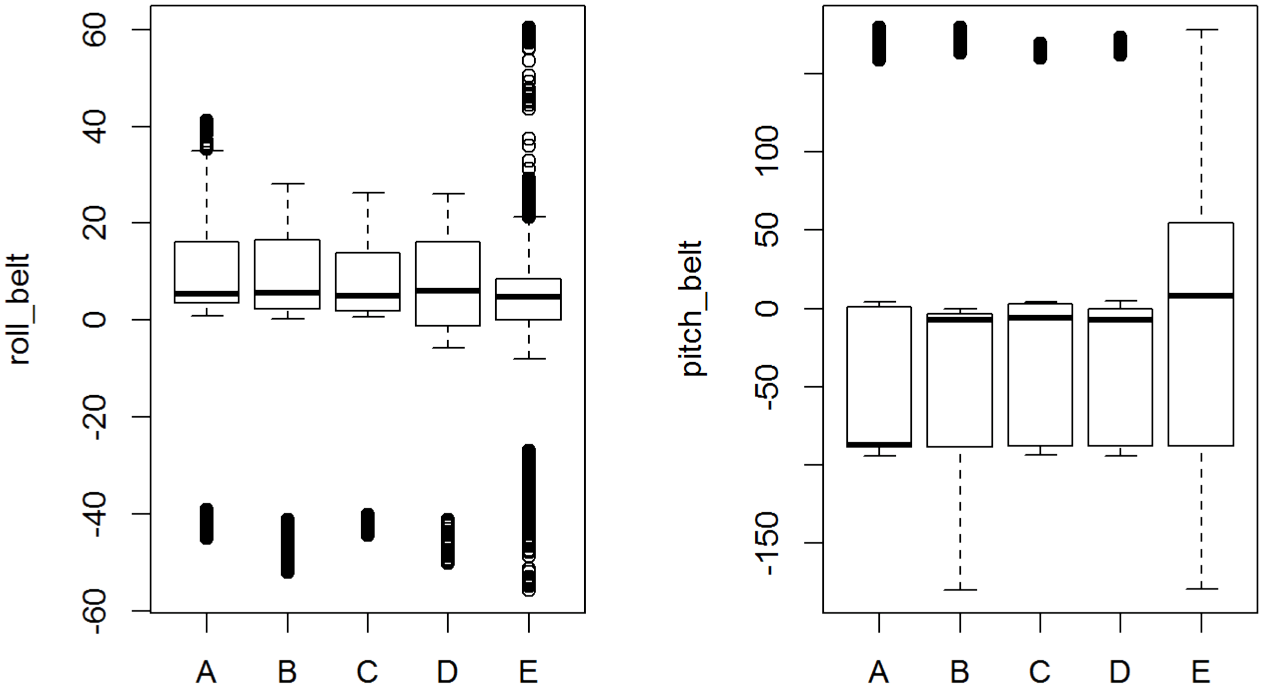
- Second, the descriptive variables that do not influence the final result were discarded: user name,the first three columns which recording time, new_window and num_window.

- Third, a visual inspection of the variables was performed using boxplots. In general, the predictive power of each variable is weak separately.

- Finally, chosen variables were extracted from the training and testing set (into pmlTrain_1 and pmlTest_1) to build and test the model.

```
vChoice <- c(8:11,37:49,60:68,84:86,102,113:124,140,151:159,160)
pmlTrain_1 <- pmlTrain[,vChoice]
pmlTest_1 <- pmlTest[,vChoice]
```

Examples of boxplots to see the data range of each variable associated with each class (A,B,C,D,E)



## 2. Model Selection and Cross validation

The model was fitted with a gradient boosted machine (gbm) using the selected predictors in the training set (pmlTrain_1), along with a K-fold cross-validation without repetition for estimating the performance for the predictive model.

The ‘gbm’ method was chosen because the most predictors are weak and this method allows taking these variables and average them together with weights, in order to get a stronger predictor.

```
library(gbm)
fitControl <- trainControl(method = "cv", number = 10,repeats = 10)
pmlFit <-train(classe ~.,method="gbm",trControl=fitControl,verbose=FALSE,data=pmlTrain_1)
pmlFit
```

In this case, as the results are categorical, the expected level of fit of the model to a dataset is determined by its accuracy (0.962 with a standar deviation of 0.0051) and the concordance measure kappa (0.95 with a standard deviation of 0.0064).

```
Stochastic Gradient Boosting

14718 samples
   52 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 13244, 13246, 13247, 13249, 13246, 13247, ...

Resampling results across tuning parameters:

  interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD  Kappa SD
  3                  150      0.9620880  0.9520368  0.005076822  0.006421897

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were n.trees = 150, interaction.depth = 3 and shrinkage =
0.1.
```

## 3. Test of model with testing set

The model was tested with testing dataset (pmlTest_1)

```
pml_pred <- predict(pmlFit,pmlTest_1)
confusionMatrix(pmlTest_1$classe,pml_pred)
```

According to confusion Matrix, The overall accuracy rate is 96 percent along with a 95% confidence interval for this rate between 95% and 96%.

```
Confusion Matrix:

                    Observed
Prediction     A    B    C    D    E
         A  1375   14    2    3    1
         B    30  884   35    0    0
         C     0   24  820    9    2
         D     2    2   42  750    8
         E     1   12   10    8  870


Overall Statistics:

               Accuracy : 0.9582
                 95% CI : (0.9522, 0.9636)
    No Information Rate : 0.2871
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9471

"Mcnemar's Test" P-Value : 1.938e-07

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9766   0.9444   0.9021   0.9740   0.9875
Specificity            0.9943   0.9836   0.9912   0.9869   0.9923
Pos Pred Value         0.9857   0.9315   0.9591   0.9328   0.9656
Neg Pred Value         0.9906   0.9869   0.9780   0.9951   0.9973
Prevalence             0.2871   0.1909   0.1854   0.1570   0.1796
Detection Rate         0.2804   0.1803   0.1672   0.1529   0.1774
Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
Balanced Accuracy      0.9854   0.9640   0.9467   0.9805   0.9899
```

## 4. Validation of Model with 20 test cases

Finally, the machine learning algorithm was used to predict 20 different test cases (from “pml-testing.csv” file) . The results were very satisfactory with 20 out of 20 correct.

```
pmlTest3 <- read.csv("pml-testing.csv")
pmlTest3 <- pmlTest3[,vChoice]
pred <- predict(pmlFit,pmlTest3)
observed <- c("B","A","B","A","A","E","D","B","A","A","B","C",
             "B","A","E","E","A","B","B","B")
confusionMatrix(observed,pred)
```

The confusion Matrix between observed and predicted data shows an 100% accuracy along with a confidence interval betwen 0.83 and 1. For these 20 test cases the model showed a 100% sensitivity and 100% Specificity.

```
Confusion Matrix:

              observed
Prediction  A  B  C  D  E
         A  7  0  0  0  0
         B  0  8  0  0  0
         C  0  0  1  0  0
         D  0  0  0  1  0
         E  0  0  0  0  3

Overall Statistics

               Accuracy : 1
                 95% CI : (0.8316, 1)
    No Information Rate : 0.4
    P-Value [Acc > NIR] : 1.1e-08
```

```
                  Kappa : 1

 "Mcnemar's Test" P-Value : NA

 Statistics by Class:

                      Class: A Class: B Class: C Class: D Class: E
 Sensitivity              1.00      1.0     1.00     1.00     1.00
 Specificity              1.00      1.0     1.00     1.00     1.00
 Pos Pred Value           1.00      1.0     1.00     1.00     1.00
 Neg Pred Value           1.00      1.0     1.00     1.00     1.00
 Prevalence               0.35      0.4     0.05     0.05     0.15
 Detection Rate           0.35      0.4     0.05     0.05     0.15
 Detection Prevalence     0.35      0.4     0.05     0.05     0.15
 Balanced Accuracy        1.00      1.0     1.00     1.00     1.00
```