---

title: "Class Project - PracMachLearn - first try"

author: "Mark Abad"

date: "April 3, 2016"

output: pdf\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

## Class Project - Coursera Practical Machine Learning

This is an Rmd file containing the code and explanation for how a predictive function was created and chosen to make calls on the 20 unknown data sets of kinetic measures to see which correlate with the 5 categories of motion during dumb bell excercises. More information about this study can be found at "http://groupware.les.inf.puc-rio.br/har".

### Introduction

A training data set of human movement parameters ("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv") is used to try different strategies for building a prediction model. The model needs to correctly identify which of the 5 different motion categories (A, B, C, D, or E) correspond to the 20 unknowns in the test data set ("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv").

We tried using a boosted trees model ("gbm"), a random forest model ("rf"), and a linear discriminant analysis model ("lda"). The results of the accuracy for each of these 3 models were compared to select the best model, which was then used to analyze the 20 unknowns in the test data set.

### Synopsis of our approach

- read the training data set and test data set into R

- 70 percent of the data in the training set is used for training the prediciton models and 30 percent is used for cross validation to test the accuracy of each prediciton model and to rank order them.

- a subset of the original training data set is chosen to eliminate variables that have low variance or that have a majority of missing data.

- the number of variables are cut down using principal component analysis

- a boosted trees model is fitted with the 70% training data subset to build a prediction model and it is tested using the 30% training data subset (testing set) to assess accuracy

- the same is done again, but next fitting a linear discrimiant analysis model and assessing its accuracy

- the process is repeated a third time, fitting a random forest model and assessing its accuracy

- the model that performs best is subsequently used to predict which of the 5 motion categories of dumb bell lifting is best associated with the 20 sets of performance observatons in the test data set.

- we report each of the 20 predicted outcomes as either motion category A, B, C, D, E (the factor labelled "classe" in the training data set).

### Data upload

```{r, echo=TRUE}

data <- read.csv("pml-training.csv")

colnames(data)

summary(data)

```

### Splitting the training set data 70/30 for training and testing subsets

```{r, echo=TRUE}

library(caret)

set.seed(1111)

train <- createDataPartition(y=data$classe,p=.70,list=F)

training <- data[train,]

testing <- data[-train,]

```

### Eliminating low quality data from the training set

```{r, echo=TRUE}

#get rid of variables that cannot contribute to the predictive power of the models

Cl <- grep("name|timestamp|window|X", colnames(training), value=F)

trainingCl <- training[,-Cl]

#select variables with high (over 95%) missing data --> exclude them from the analysis

trainingCl[trainingCl==""] <- NA

NArate <- apply(trainingCl, 2, function(x) sum(is.na(x)))/nrow(trainingCl)

trainingCl <- trainingCl[!(NArate>0.95)]

summary(trainingCl)

```

### Principal Component Analysis is used to whittle down the number of variables and to get to a threshold of 95% coverage.

```{r, echo=TRUE}

preProc <- preProcess(trainingCl[,1:52],method="pca",thresh=.8) #12 components are required

preProc <- preProcess(trainingCl[,1:52],method="pca",thresh=.9) #18 components are required

preProc <- preProcess(trainingCl[,1:52],method="pca",thresh=.95) #25 components are required

preProc <- preProcess(trainingCl[,1:52],method="pca",pcaComp=25)

preProc$rotation

trainingPC <- predict(preProc,trainingCl[,1:52])

```

### Boosted Trees Model (gbm)

```{r, echo=TRUE}

library(gbm)

modFitGBM <- suppressMessages(train(trainingCl$classe~., data=trainingPC, method="gbm", verbose=FALSE))

print(modFitGBM) # view results

```

### Check accuracy for gbm model

```{r, echo=TRUE}

testingCl <- testing[,-Cl]

testingCl[testingCl==""] <- NA

NArate <- apply(testingCl, 2, function(x) sum(is.na(x)))/nrow(testingCl)

testingCl <- testingCl[!(NArate>0.95)]

testingPC <- predict(preProc,testingCl[,1:52])

confusionMatrix(testingCl$classe,predict(modFitGBM,testingPC))

## accuracy for gbm is only in the 80s %

```

### Check accuracy for lda model

```{r, echo=TRUE}

library(lda)

modFitLDA <- suppressMessages(train(trainingCl$classe~., data=trainingPC, method="lda"))

print(modFitLDA) # view results

testingCl <- testing[,-Cl]

testingCl[testingCl==""] <- NA

NArate <- apply(testingCl, 2, function(x) sum(is.na(x)))/nrow(testingCl)

testingCl <- testingCl[!(NArate>0.95)]

testingPC <- predict(preProc,testingCl[,1:52])

confusionMatrix(testingCl$classe,predict(modFitLDA,testingPC))

## accuracy for lda model is even worse, only in the 50s%

```

### Check accuracy for rf model

```{r, echo=TRUE}

library(randomForest)

modFitRF <- randomForest(trainingCl$classe ~ ., data=trainingPC, do.trace=F)

print(modFitRF) # view results

importance(modFitRF) # to show the importance of each predictor

testingCl <- testing[,-Cl]

testingCl[testingCl==""] <- NA

NArate <- apply(testingCl, 2, function(x) sum(is.na(x)))/nrow(testingCl)

testingCl <- testingCl[!(NArate>0.95)]

testingPC <- predict(preProc,testingCl[,1:52])

confusionMatrix(testingCl$classe,predict(modFitRF,testingPC))

## accuracy for rf model is very good! (>95%)

```

## Use Random Forest Method, because clearly, the rf model outperformed both of the others and will now be used to make the predicitons.

### Predict which category of motion, A through E, is best associated with the 20 unknowns in the test data set.

```{r, echo=TRUE}

testdata <- read.csv("pml-testing.csv")

testdataCl <- testdata[,-Cl]

testdataCl[testdataCl==""] <- NA

NArate <- apply(testdataCl, 2, function(x) sum(is.na(x)))/nrow(testdataCl)

testdataCl <- testdataCl[!(NArate>0.95)]

testdataPC <- predict(preProc,testdataCl[,1:52])

testdataCl$classe <- predict(modFitRF,testdataPC)

testdataCl$classe

## prints out each letter of the category in order for unknowns 1 through 20

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

```

## Discussion

In this analyses, 19622 observations from weight lifting exercise were used to analyze and predict correct body movement from others during the exercise. 70% of the total observations (13737 observations) were used to build three models by boosted trees method (gbm), linear discriminant method (lda) and random forest method (rf). The rest of 30% of the observations (5885 observations) were used for validating the models. The statistics showed that the rf model had the overall accuracy of 97% for the testing set, which is not overlapping with observations used to built the model. The sensitivity was in between 92%-99% and the specificity was over 99% for all classes, A through E.

## Additional Reading Material

Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers’ Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6\_6.

Read more: http://groupware.les.inf.puc-rio.br/har#weight\_lifting\_exercises#ixzz3jOpnStGb