Peer-graded Assignment: Machine learning Course Project

Executive summary

The goal of your project is to predict the manner in which the participants did the exercise.

1. Question

In which manner they did the participants the exercise?

2. Input data

Weight Lifting Exercises Dataset. Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). The data for this project was generously offered by: http://groupware.les.inf.puc-rio.br/har.

Load data & tidy up

(Larger K=less bias, more variance; smaller k= more bias, less variance–> 20=accurate)

```
library(plyr)
library(dplyr)
library(ggplot2)
library(grid)
library(gridExtra)
library(caret)
library(rpart)
library(randomForest)

testing<-read.csv("pml-testing.csv")
training<-read.csv("pml-training.csv")
testing$classe
#-->Note: "classe" is not a variable of the testing dataset

training <- training[, colSums(is.na(training)) == 0]


# delete unnecessary data (timestamp etc)
training[3,1:10]
training <- training[, -c(1:7)]
```

Cross validation as there is no variable "classe" in the testing data set. Split: training data 60%, testing data 40%

```
#transform factors in numeric vectors
for(i in 1:85){
if (class(training[,i])=="factor")
    {training[,i]<-as.numeric(training[,i])}}

inTrain<-createDataPartition(y=training$classe,p=0.60,list=FALSE)
training_set<-training[inTrain,]
testing_set<-training[-inTrain,]
```

3. Algorithm

Classification tree

```
ModFitDecTree <- rpart(classe ~ ., data=training_set, method="class")
pred_decTree<-predict(ModFitDecTree,newdata = testing_set,type="class")
confusionMatrix(pred_decTree,testing_set$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2044  248   29   79   57
##          B   68  832   66   90  108
##          C   55  216 1096  186  161
##          D   24  118   78  813   85
##          E   41  104   99  118 1031
##
## Overall Statistics
##
##                Accuracy : 0.7413
##                  95% CI : (0.7314, 0.7509)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6717
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9158   0.5481   0.8012   0.6322   0.7150
## Specificity            0.9264   0.9475   0.9046   0.9535   0.9435
## Pos Pred Value         0.8319   0.7148   0.6394   0.7272   0.7401
## Neg Pred Value         0.9651   0.8973   0.9556   0.9297   0.9363
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2605   0.1060   0.1397   0.1036   0.1314
## Detection Prevalence   0.3132   0.1484   0.2185   0.1425   0.1775
## Balanced Accuracy      0.9211   0.7478   0.8529   0.7928   0.8292
```

Random Forest

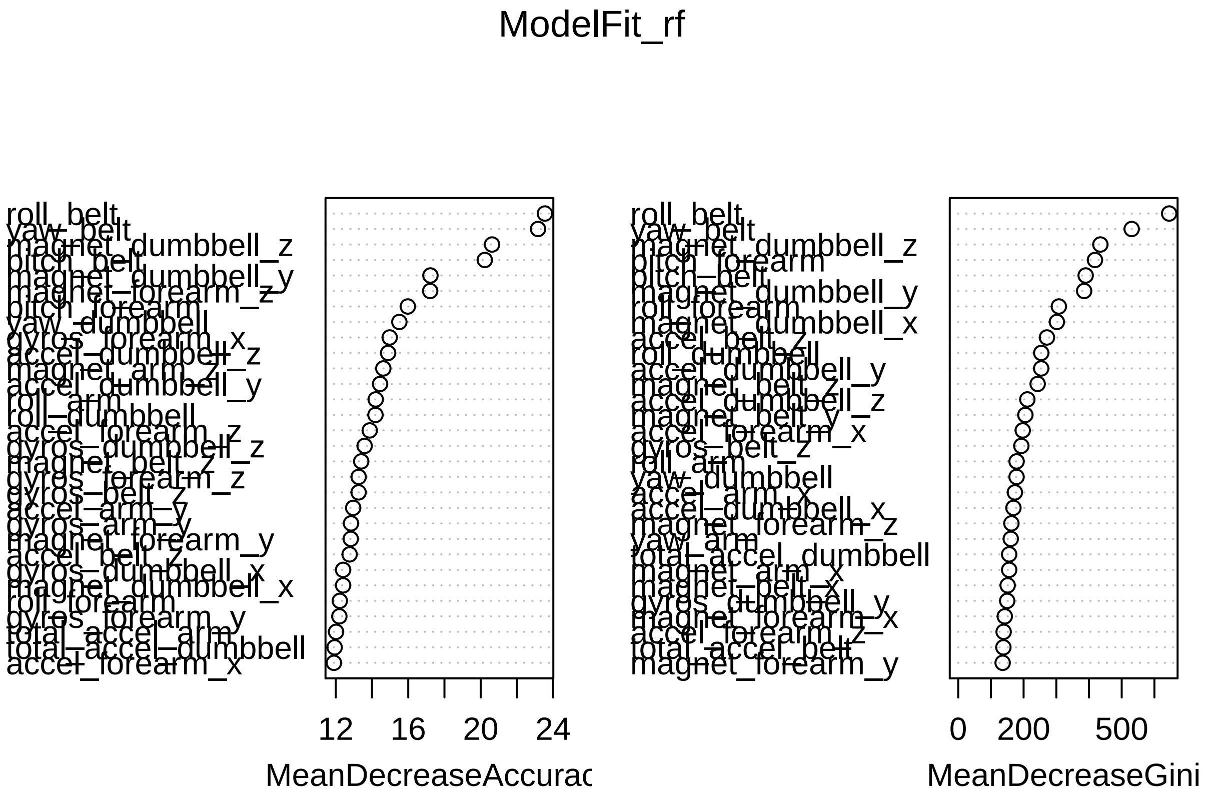
```r
set.seed(3141600)
ModelFit_rf <- randomForest(classe~., data=training_set, importance=TRUE, ntree=100)

pred_rf<-predict(ModelFit_rf,newdata = testing_set)
confusionMatrix(pred_rf,testing_set$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2231   17    0    0    0
##          B    0 1491   10    0    0
##          C    0   10 1355   23    0
##          D    0    0    3 1259    3
##          E    1    0    0    4 1439
##
## Overall Statistics
##
##                Accuracy : 0.991
##                  95% CI : (0.9886, 0.9929)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9886
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9996   0.9822   0.9905   0.9790   0.9979
## Specificity            0.9970   0.9984   0.9949   0.9991   0.9992
## Pos Pred Value         0.9924   0.9933   0.9762   0.9953   0.9965
## Neg Pred Value         0.9998   0.9957   0.9980   0.9959   0.9995
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1900   0.1727   0.1605   0.1834
## Detection Prevalence   0.2865   0.1913   0.1769   0.1612   0.1840
## Balanced Accuracy      0.9983   0.9903   0.9927   0.9890   0.9986
```

As there are some variables in the testing data , which just just “NAs”, we will use not all variables to answer the QUIZ. To find the most important variables, we are using varImpPlot.

```r
varImpPlot(ModelFit_rf)
```



```
set.seed(3141600)

ModelFit_rf2 <- randomForest(classe~yaw_belt+roll_belt+magnet_dumbbell_z+pitch_belt+pitch_forearm+gyros_

pred_rf2<-predict(ModelFit_rf2,newdata = testing_set)
confusionMatrix(pred_rf2,testing_set$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2202   28   11    4    4
##          B   11 1441   12    4   15
##          C   10   38 1333   21    3
##          D    5    8   12 1251    4
##          E    4    3    0    6 1416
##
## Overall Statistics
##
##                Accuracy : 0.9741
##                  95% CI : (0.9704, 0.9775)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9673
##  Mcnemar's Test P-Value : 7.543e-05
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9866   0.9493   0.9744   0.9728   0.9820
## Specificity            0.9916   0.9934   0.9889   0.9956   0.9980
## Pos Pred Value         0.9791   0.9717   0.9488   0.9773   0.9909
## Neg Pred Value         0.9946   0.9879   0.9946   0.9947   0.9959
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2807   0.1837   0.1699   0.1594   0.1805
## Detection Prevalence   0.2866   0.1890   0.1791   0.1631   0.1821
## Balanced Accuracy      0.9891   0.9713   0.9817   0.9842   0.9900
```

4. Results

The following Accuracy have our models:

Classification tree: 0.7552 Random Forest with all variables: 0.9926 Random Forest with the 6 most important variables: 0.9749

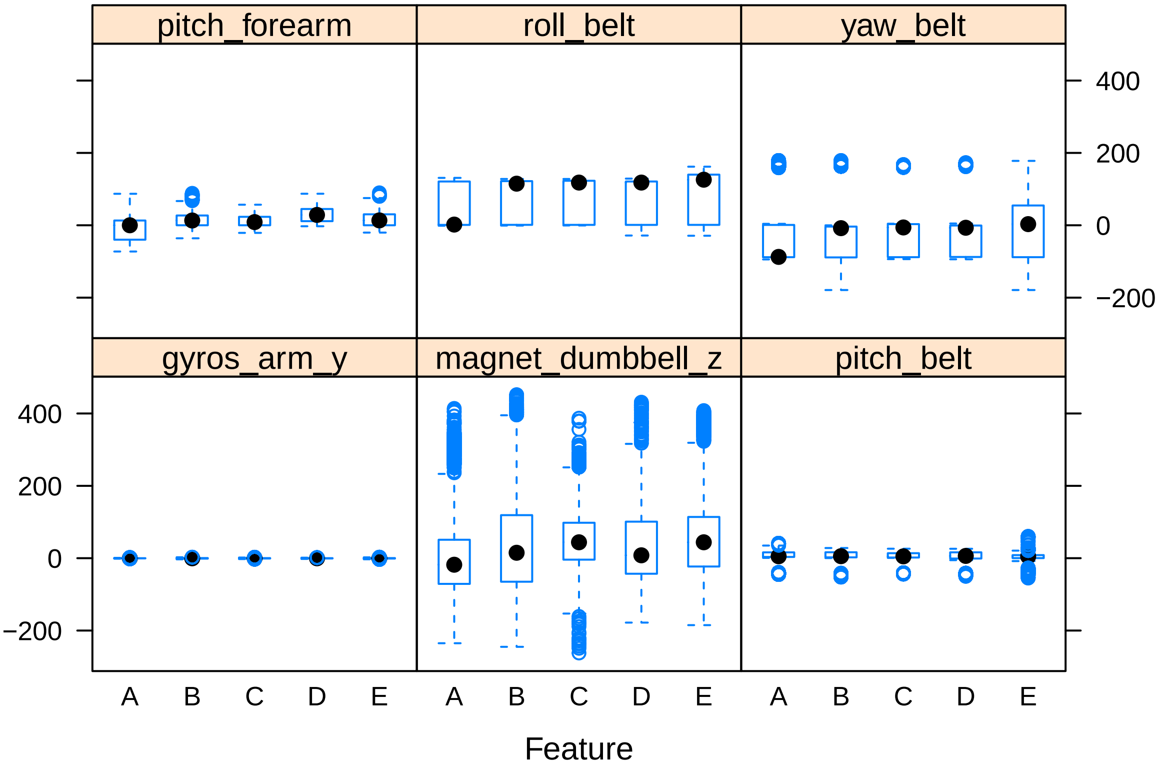
Therefore the Random Forest model with all variables is the best prediction model. But for the sprecific testing data the Random Forest with the 6 most important variables is the the accurate one.

```
pred<-predict(ModelFit_rf2,newdata = testing)
pred
```

```
##  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
```

APPENDIX:

```
#some additional analysis
featurePlot(x=training_set[,c("yaw_belt","roll_belt","magnet_dumbbell_z","pitch_belt","pitch_forearm","
```



```
data<-group_by(training_set, classe)
data1<-summarize_each(data,funs(mean(., na.rm = TRUE)))
```

Read more: http://groupware.les.inf.puc-rio.br/har#ixzz4rpKV18aZ