---

title: "MLProject"

author: "RT"

date: "Tuesday, May 19, 2015"

output: html\_document

---

**Summary**

The purpose of this project was to test machine learning algorithm on accuracy of predicting activity from activity monitors. Data was loaded, cleaned, partitioned, trained, and cross validated. The random forest model outperformed the decision tree model.

**Data Sources**

The training data for this project are available here:

<https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv>

The test data are available here:

<https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv>

The data for this project comes from this original source: <http://groupware.les.inf.puc-rio.br/har>

**Reproducibility**

The research must follow several methods in order to reproduce these results. First, a seed was set at 9876 for modeling.

> # Set seed for reproducibility

> set.seed(9876)

Second, the researcher must use the same R packages to replicate results. Please see details below.

Third, the researcher must follow the procedures on model building methods.

The dependent variable was a five-level categorical variable called classe. Please find brief descriptions here:

Class A - exactly according to the specification

Class B -  throwing the elbows to the front

Class C - lifting the dumbbell only halfway

Class D lowering the dumbbell only halfway

Class E throwing the hips to the front

Model evaluation was based on accuracy and minimizing the out-of-sample error.

**Install, load packages**

The researcher must install and load the appropriate packages in order to conduct this analysis.

>getwd()

>install.packages("caret")

>install.packages("randomForest")

>install.packages("rpart")

> # Loading required package: lattice

> library(lattice)

> # Loading required package: ggplot2

> library(ggplot2)

> #Classification and regression

> library(caret)

> library(randomForest)

> # Regressive Partitioning and Regression trees

> library(rpart)

> # Decision Tree plot

> library(rpart.plot)

**Get and Clean Data**

> #Load and clean data

>

> # Load training data set into my R session replacing all missing with "NA"

> trainingset <- read.csv("C:/Users/thomasr/Documents/trainingdata.csv", na.strings=c("NA","#DIV/0!", ""))

>

> # Load testing data set

> testingset <- read.csv('C:/Users/thomasr/Documents/testingdata.csv', na.strings=c("NA","#DIV/0!", ""))

> # Delete columns with all missing values

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

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

># Summary statistics

>dim(trainingset)

>head(trainingset)

>dim(testingset)

>head(testingset)

**Cross-validation and Data Partitioning**

> #In order to perform cross-validation, the training data set is partionned into 2 sets: subTraining (75%) and subTest (25%) random subsampling without replacement.

>

> subsamples <- createDataPartition(y=trainingset$classe, p=0.75, list=FALSE)

> subTraining <- trainingset[subsamples, ]

> subTesting <- trainingset[-subsamples, ]

> dim(subTraining)

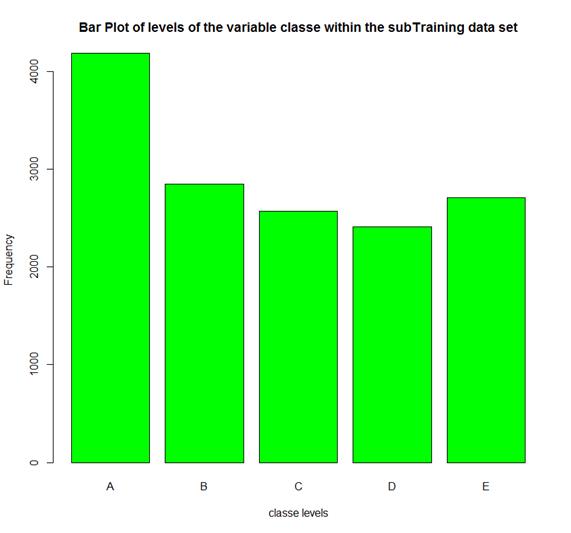
> dim(subTesting)

Our models will be fitted on the subTraining data set, and tested on the subTesting data. Once the most accurate model is choosen, it will be tested on the original Testing data set.

> #Explore the data

> summary(subTraining)

> plot(subTraining$classe, col="green", main="Bar Plot of levels of the variable classe within the subTraining data set", xlab="classe levels", ylab="Frequency")



> #Classe A is the most frequent, D is the least frequent.

**Expected out-of-sample error**

The expected out-of-sample error will correspond to the quantity: 1-Sensitivity in the cross-validation data. Also known as Accuracy, this is the proportion of correct classified observation over the total sample in the subTesting data set. Expected accuracy is the expected accuracy in the out-of-sample data set (i.e. original testing data set). Thus, the expected value of the out-of-sample error will correspond to the expected number of misclassified observations/total observations in the Test data set, which is the quantity: 1-accuracy found from the cross-validation data set.

**ML Algorithm: Decision Trees**

> #First prediction model: Decision Tree with rpart

> model1 <- rpart(classe ~ ., data=subTraining, method="class")

>

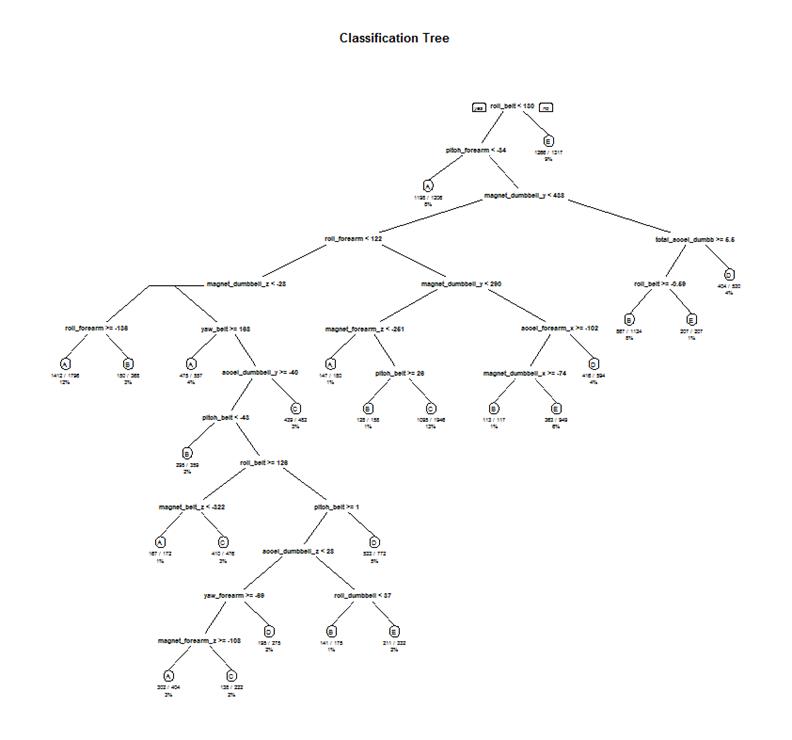
> # Predicting:

> prediction1 <- predict(model1, subTesting, type = "class")

>

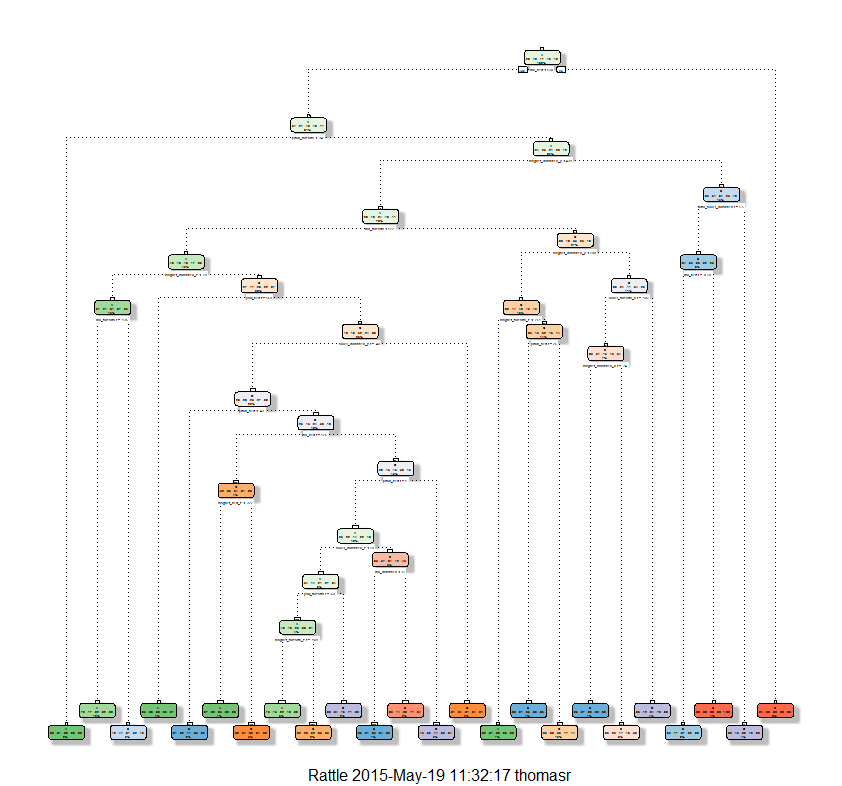
> # Plot of the Decision Tree

> rpart.plot(model1, main="Classification Tree", extra=102, under=TRUE, faclen=0)

****

> library(rattle)

> fancyRpartPlot(model1)



> # Test results on subTesting:

> confusionMatrix(prediction1, subTesting$classe)

Confusion Matrix and Statistics

          Reference

Prediction    A    B    C    D    E

         A 1212  127   10   46   11

         B   52  571   47   60   52

         C   40   93  695  126  101

         D   49   76   46  502   42

         E   42   82   57   70  695

Overall Statistics

               Accuracy : 0.7494

                 95% CI : (0.737, 0.7615)

    No Information Rate : 0.2845

    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6831

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E

Sensitivity            0.8688   0.6017   0.8129   0.6244   0.7714

Specificity            0.9447   0.9466   0.9111   0.9480   0.9373

Pos Pred Value         0.8620   0.7302   0.6588   0.7021   0.7347

Neg Pred Value         0.9477   0.9083   0.9584   0.9279   0.9480

Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837

Detection Rate         0.2471   0.1164   0.1417   0.1024   0.1417

Detection Prevalence   0.2867   0.1595   0.2151   0.1458   0.1929

Balanced Accuracy      0.9068   0.7742   0.8620   0.7862   0.8543

**ML Algorithm: Random Forest**

> #Second prediction model: Using Random Forest

>

> model2 <- randomForest(classe ~. , data=subTraining, method="class")

> # Predicting:

> prediction2 <- predict(model2, subTesting, type = "class")

> # Test results on subTesting data set:

> confusionMatrix(prediction2, subTesting$classe)

Confusion Matrix and Statistics

          Reference

Prediction    A    B    C    D    E

         A 1395    0    0    0    0

         B    0  948    5    0    0

         C    0    1  850    5    1

         D    0    0    0  799    1

         E    0    0    0    0  899

Overall Statistics

               Accuracy : 0.9973

                 95% CI : (0.9955, 0.9986)

    No Information Rate : 0.2845

    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9966

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E

Sensitivity            1.0000   0.9989   0.9942   0.9938   0.9978

Specificity            1.0000   0.9987   0.9983   0.9998   1.0000

Pos Pred Value         1.0000   0.9948   0.9918   0.9988   1.0000

Neg Pred Value         1.0000   0.9997   0.9988   0.9988   0.9995

Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837

Detection Rate         0.2845   0.1933   0.1733   0.1629   0.1833

Detection Prevalence   0.2845   0.1943   0.1748   0.1631   0.1833

Balanced Accuracy      1.0000   0.9988   0.9962   0.9968   0.9989

**Result**

Due to error equaling 1-accuracy, the random forest model performed better with 99.8% accuracy compared to 74.9% by the decision tree model.

The expected out-of-sample error is estimated at 0.005, or 0.5%. The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set. Our Test data set comprises 20 cases. With an accuracy above 99% on our cross-validation data, we can expect that very few of the test samples will be misclassified.