Practical machine Learning

Sarah Shikangah

April 30, 2017

## Abstract

The main aim of this project is to predict the a behavior pattern labelled as "classe" variable from exercise activities. The data for this project come from this source: <http://groupware.les.inf.puc-rio.br/har>. The collected data from accelerometers on belt, forearm, arm, and dumbell of 6 participants will be used to perform machine learning project. Links to the datasets are; <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv>, for training data set,;<https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv>, for testing data set. ## Download required packages and data We shall download packages required and data.

#Required packages  
library(caret)

## Warning: package 'caret' was built under R version 3.2.5

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.2.5

library(ggplot2)  
library(lattice)  
library(rattle)

## Warning: package 'rattle' was built under R version 3.2.5

## Rattle: A free graphical interface for data mining with R.  
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.2.5

## Loading required package: rpart

## Warning: package 'rpart' was built under R version 3.2.5

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.2.5

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(MASS)  
set.seed(12356)  
PmlTraining <- read.table("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", header = TRUE, sep = ",", dec = ".", na.strings=c("NA","#DIV/0!",""))  
pmlTesting <- read.table("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", header = TRUE, sep = ",", dec = ".", na.strings=c("NA","#DIV/0!",""))  
#str(PmlTraining) to check dataset

## Data cleaning

Processing data for analyses, by removing variables with missing data and character observations.

#Replace characters #DIV/0!with NA  
tr <- as.data.frame(sapply(PmlTraining,gsub,pattern="#DIV/0!",replacement="NA"))  
test <- as.data.frame(sapply(pmlTesting,gsub,pattern="#DIV/0!",replacement="NA"))  
# Remove variables with missing values  
PmlTraining1 <-tr[,!sapply(tr,function(x) any(is.na(x)))]  
pmlTesting <- test[,!sapply(test,function(x) any(is.na(x)))]  
#Remove the first seven variables to avoid interferance .  
Training1 <- PmlTraining1[,-c(1:7)]  
Testing1 <- pmlTesting[, -c(1:7)]  
#set all variables as numeric class with exception of classe variable  
Training1[, 1:52] <- lapply(Training1[, 1:52], as.numeric)  
Testing1[, 1:52] <- lapply(Testing1[, 1:52], as.numeric)  
dim(Training1)

## [1] 19622 53

dim(Testing1)

## [1] 20 53

## Splitting training dataset

set.seed(12356)  
inTrain <- createDataPartition(y=Training1$classe, p=0.75, list=FALSE)  
training <- Training1[inTrain,]; validation <- Training1[-inTrain,]  
dim(training);dim(validation)

## [1] 14718 53

## [1] 4904 53

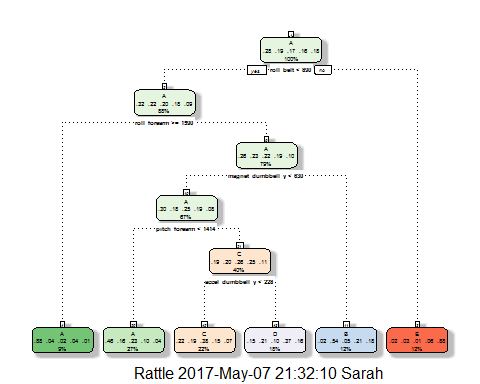
## Exploratory analysis

#Due to the space limited the plots will not be shown  
# check covariance and corrilation using (cov(training[, 1:53]);cor(training[, 1:53]))  
#featurePlot(x=Training1[, c(1:52)], y = Training1$classe, plot = "pairs")

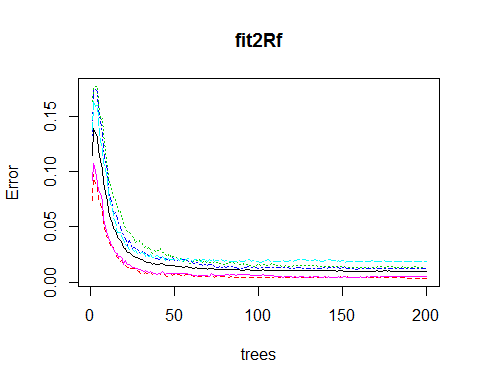
## Fitting three different models.

we shall fit three different models and compare there accuracy and choose the best model for prediction on the testing data set. ### Model 1 : Desicion tree

set.seed(12356)  
Fit <- train(classe~., method = "rpart", data = training)  
fancyRpartPlot(Fit$finalModel)

 ### Model 2 : Random Forest

fit2Rf <- randomForest(classe~.,data=training, ntree=200, importance=TRUE)  
plot(fit2Rf)

 ### Model 3 : Linear Discriminant Analysis

Fitlda <- train(classe~., method = "lda", data = training)

## Predict on the testing set

pred1 <- predict(Fit, validation)  
pred2 <- predict(fit2Rf, validation)  
pred3 <- predict(Fitlda, validation)  
predDf <- data.frame(pred1, pred2, pred3, classe = validation$classe)  
CombMod <- train(classe~., method = "rf", data = predDf)  
pred4 <- predict(CombMod, predDf)  
  
  
rbind(postResample(pred1, obs = validation$classe), postResample(pred2, obs = validation$classe), postResample(pred3, obs = validation$classe), postResample(pred4, obs = validation$classe))

## Accuracy Kappa  
## [1,] 0.5207993 0.3898592  
## [2,] 0.9942904 0.9927767  
## [3,] 0.6188825 0.5168612  
## [4,] 0.9942904 0.9927767

AccuTest <- confusionMatrix(pred4, validation$classe)  
AccuTest

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1394 7 0 2 0  
## B 1 939 2 0 0  
## C 0 3 851 7 2  
## D 0 0 2 795 2  
## E 0 0 0 0 897  
##   
## Overall Statistics  
##   
## Accuracy : 0.9943   
## 95% CI : (0.9918, 0.9962)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9928   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.9993 0.9895 0.9953 0.9888 0.9956  
## Specificity 0.9974 0.9992 0.9970 0.9990 1.0000  
## Pos Pred Value 0.9936 0.9968 0.9861 0.9950 1.0000  
## Neg Pred Value 0.9997 0.9975 0.9990 0.9978 0.9990  
## Prevalence 0.2845 0.1935 0.1743 0.1639 0.1837  
## Detection Rate 0.2843 0.1915 0.1735 0.1621 0.1829  
## Detection Prevalence 0.2861 0.1921 0.1760 0.1629 0.1829  
## Balanced Accuracy 0.9984 0.9944 0.9962 0.9939 0.9978

From the above comparison result Random Forests model provided the best result with accuracy of 99.43% which gives sample error to 0.57% and so as to the combined models. For this project we shall use Random Forests model to predict on the testing dataset. ## Predicting on the testing dataset

pred5T<- predict(fit2Rf, newdata = Testing1, type = "class")  
pred5T

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## E E E E E E E E E E E E E E E E E E E E   
## Levels: A B C D E