Prediction assignment

# Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: <http://groupware.les.inf.puc-rio.br/har> (see the section on the Weight Lifting Exercise Dataset).

# Data

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>

# Loading data and initialization

set.seed(777)

library(caret)

library(randomForest)

mydata <- read.table("c:/Data/pml-training.csv", header=TRUE, sep=",",row.names = "X")

validation <- read.table("c:/Data/pml-testing.csv", header=TRUE, sep=",",row.names = "X")

# Data processing

Removing non predictive variables

mydata <- mydata[,-c(1:6)]

validation<-validation[,-c(1:6)]

Removing variables with 90% or more missing values

x <- sapply(mydata, function(x) mean(is.na(x))) > 0.9

mydata <- mydata[, x==FALSE]

validation <- validation[, x==FALSE]

Transform to numeric

mydata[,-ncol(mydata)] <- apply(mydata[,-ncol(mydata)], 2, function(x){as.numeric(as.character(x))})

Removing again variables with 90% or more missing values

x <- sapply(mydata, function(x) mean(is.na(x))) > 0.9

mydata <- mydata[, x==FALSE]

validation <- validation[, x==FALSE]

# Prediction

Train and test samples

inTrain <- createDataPartition(mydata$classe, p=0.7, list=FALSE)

train <- mydata[inTrain, ]

test <- mydata[-inTrain, ]

Prediction

fit1<-randomForest(train$classe ~ ., data=train, do.trace=F, ntree=50)

pred1<-predict(fit1, newdata = test)

Prediction accuracy and variable importance

confusionMatrix(pred1,test$classe)

varImpPlot(fit1)

Plot of Confusion Matrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1672 5 0 0 0

B 0 1131 7 0 0

C 2 3 1018 9 1

D 0 0 1 953 3

E 0 0 0 2 1078

Overall Statistics

Accuracy : 0.9944

95% CI : (0.9921, 0.9961)

No Information Rate : 0.2845

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9929

Mcnemar's Test P-Value : NA

Statistics by Class:

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

Sensitivity 0.9988 0.9930 0.9922 0.9886 0.9963

Specificity 0.9988 0.9985 0.9969 0.9992 0.9996

Pos Pred Value 0.9970 0.9938 0.9855 0.9958 0.9981

Neg Pred Value 0.9995 0.9983 0.9984 0.9978 0.9992

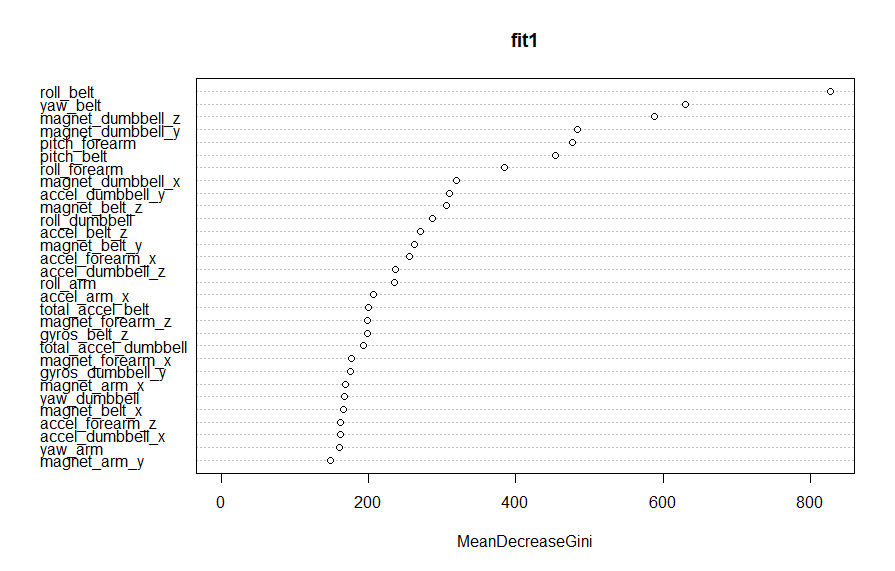
Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839

Detection Rate 0.2841 0.1922 0.1730 0.1619 0.1832

Detection Prevalence 0.2850 0.1934 0.1755 0.1626 0.1835

Balanced Accuracy 0.9988 0.9958 0.9946 0.9939 0.9979

Plot of variable importance



## Assignment results

Results <- predict(fit1, newdata=validation)

Results