Practical Machine Learning - Project

Author J

Sunday, April 26, 2015

# Introduction

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>

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

## Load Library

library(caret)

## Warning: package 'caret' was built under R version 3.1.3

## Loading required package: lattice  
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.1.2

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.1.3

## randomForest 4.6-10  
## Type rfNews() to see new features/changes/bug fixes.

## Read files and Clean up data

data <- read.csv(file= "pml-training.csv", header= TRUE, sep = ",", na.strings= c("","NA"))

Remove columns 1-7 as it is non numerical data

data <- data[,-(1:7)]

Remove NA

data <- data[,which(as.numeric(colSums(is.na(data)))==0)]

Remove near zero variance columns as it would meaningless

end <- ncol(data)  
data[,-end] <- data.frame(sapply(data[,-end], as.numeric))  
nzv <- nearZeroVar(data[, -end], saveMetrics=TRUE)  
data <- data[,!as.logical(nzv$nzv)]

## Data Processing

Partition data

set.seed(888)  
inTrain<-createDataPartition(y=data$classe, p=0.6, list=FALSE)  
train <- data[inTrain,]  
validation <- data[-inTrain,]

Fitting a model to predict classe and everything else as a predictor

rfModel <- randomForest(classe ~ ., data = train)  
  
ptrain <- predict(rfModel, train)  
print(confusionMatrix(ptrain, train$classe))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 3348 0 0 0 0  
## B 0 2279 0 0 0  
## C 0 0 2054 0 0  
## D 0 0 0 1930 0  
## E 0 0 0 0 2165  
##   
## Overall Statistics  
##   
## Accuracy : 1   
## 95% CI : (0.9997, 1)  
## No Information Rate : 0.2843   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 1.0000 1.0000 1.0000 1.0000  
## Specificity 1.0000 1.0000 1.0000 1.0000 1.0000  
## Pos Pred Value 1.0000 1.0000 1.0000 1.0000 1.0000  
## Neg Pred Value 1.0000 1.0000 1.0000 1.0000 1.0000  
## Prevalence 0.2843 0.1935 0.1744 0.1639 0.1838  
## Detection Rate 0.2843 0.1935 0.1744 0.1639 0.1838  
## Detection Prevalence 0.2843 0.1935 0.1744 0.1639 0.1838  
## Balanced Accuracy 1.0000 1.0000 1.0000 1.0000 1.0000

## Validation the 40% of the remaining data

The model is used to classify the remaining 40% of data. The results were placed in a confusion matrix along with the actual classifications in order to determine the accuracy of the model.

pvalidation <- predict(rfModel, validation)  
print(confusionMatrix(pvalidation, validation$classe))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 2232 11 0 0 0  
## B 0 1506 11 0 0  
## C 0 1 1356 17 1  
## D 0 0 1 1269 5  
## E 0 0 0 0 1436  
##   
## Overall Statistics  
##   
## Accuracy : 0.994   
## 95% CI : (0.992, 0.9956)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9924   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9921 0.9912 0.9868 0.9958  
## Specificity 0.9980 0.9983 0.9971 0.9991 1.0000  
## Pos Pred Value 0.9951 0.9927 0.9862 0.9953 1.0000  
## Neg Pred Value 1.0000 0.9981 0.9981 0.9974 0.9991  
## Prevalence 0.2845 0.1935 0.1744 0.1639 0.1838  
## Detection Rate 0.2845 0.1919 0.1728 0.1617 0.1830  
## Detection Prevalence 0.2859 0.1933 0.1752 0.1625 0.1830  
## Balanced Accuracy 0.9990 0.9952 0.9941 0.9929 0.9979

The model has an accuracy of 99.4%. The out of sample range is 0.6% which shows that the model is robust and adequate to predict the new data.

# Final Test Set

The model tested is now used to predict the 20 test data set.

predictionTest <- read.csv(file= "pml-testing.csv", header= TRUE, sep = ",",na.strings= c("","NA"))  
  
ptest <- predict(rfModel, predictionTest)  
ptest

## 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

# Conclusion

The accuracy of the model with 99.4% accuracy provided the following predictions:

B A B A A E D B A A B C B A E E A B B B