Practical Machine Learning Project

jemin lee

2015년 11월 22일

## 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>.

The data can also be downloaded using the following R scoprt:

# essential library   
library(caret)

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

library(randomForest)

## randomForest 4.6-12  
## Type rfNews() to see new features/changes/bug fixes.

downloadFiles <- function(dataURL = "", destF = "t.csv")  
{  
 if(!file.exists(destF)){  
 download.file(dataURL, destF, method="curl")  
 }else{  
 message("data already downloaded.")  
 }  
}  
trainURL<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"  
testURL <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"  
downloadFiles(trainURL, "pml-training.csv")

## data already downloaded.

downloadFiles(testURL, "pml-testing.csv")

## data already downloaded.

Load the data from csv files

train <- read.csv("./pml-training.csv")  
test <- read.csv("./pml-testing.csv")

Check the structure of data and the number of each class in training set

dim(train)

## [1] 19622 160

table(train$classe)

##   
## A B C D E   
## 5580 3797 3422 3216 3607

Split training dataset into training and validation for evaluating model

set.seed(123456)  
inTrain <- createDataPartition(train$classe, p = 3/4, list = FALSE)  
trainingSet <- train[inTrain, ]  
# create validation set for testing in sample error  
validationSet <- train[-inTrain, ]

## Fetures Slection for makeing model

check the near zero covariates(featrues)

nzvMatrix <- nearZeroVar(trainingSet, saveMetrics = TRUE)  
trainingSet\_rmovedZero <- trainingSet[,!nzvMatrix$nzv]

exclude columns with m40% ore more missing values exclude descriptive columns like name etc

cntlength <- sapply(trainingSet, function(x) {  
 sum(!(is.na(x) | x == ""))  
})  
nullcol <- names(cntlength[cntlength < 0.6 \* length(trainingSet$classe)])

discards unsueful covariates(feautres) beacuse these featrues are descriptive features we consider only numeric type of covariate from HAR sensor

descriptcol <- c("X", "user\_name", "raw\_timestamp\_part\_1", "raw\_timestamp\_part\_2",   
 "cvtd\_timestamp", "new\_window", "num\_window")  
excludecols <- c(descriptcol, nullcol)  
trainingSet <- trainingSet[, !names(trainingSet) %in% excludecols]

## Making Model using Random Forest Algorithm

rfModel <- randomForest(classe ~ ., data = trainingSet, importance = TRUE, ntrees = 10)

## Testing Constructed Model

We test model in term of in sample error and out of sample error.

# training sample  
ptraining <- predict(rfModel, trainingSet)  
print(confusionMatrix(ptraining, trainingSet$classe))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 4185 0 0 0 0  
## B 0 2848 0 0 0  
## C 0 0 2567 0 0  
## D 0 0 0 2412 0  
## E 0 0 0 0 2706  
##   
## 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 0.1839  
## Detection Rate 0.2843 0.1935 0.1744 0.1639 0.1839  
## Detection Prevalence 0.2843 0.1935 0.1744 0.1639 0.1839  
## Balanced Accuracy 1.0000 1.0000 1.0000 1.0000 1.0000

# out of sample  
pvalidation <- predict(rfModel, validationSet)  
print(confusionMatrix(pvalidation, validationSet$classe))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1395 2 0 0 0  
## B 0 945 4 0 0  
## C 0 2 851 6 4  
## D 0 0 0 798 1  
## E 0 0 0 0 896  
##   
## Overall Statistics  
##   
## Accuracy : 0.9961   
## 95% CI : (0.994, 0.9977)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9951   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9958 0.9953 0.9925 0.9945  
## Specificity 0.9994 0.9990 0.9970 0.9998 1.0000  
## Pos Pred Value 0.9986 0.9958 0.9861 0.9987 1.0000  
## Neg Pred Value 1.0000 0.9990 0.9990 0.9985 0.9988  
## Prevalence 0.2845 0.1935 0.1743 0.1639 0.1837  
## Detection Rate 0.2845 0.1927 0.1735 0.1627 0.1827  
## Detection Prevalence 0.2849 0.1935 0.1760 0.1629 0.1827  
## Balanced Accuracy 0.9997 0.9974 0.9962 0.9961 0.9972

## Predicting test set, including no class data

# test set prediction

ptest <- predict(rfModel, test)  
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

We then save the output to files according to instructions and post it to the submission page. # prediction assignment submission: instructions #answers = rep("A", 20)

answers <- as.vector(ptest)  
pml\_write\_files = function(x){  
 n = length(x)  
 for(i in 1:n){  
 filename = paste0("problem\_id\_",i,".txt")  
 write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)  
 }  
}  
pml\_write\_files(answers)