Practical Machine Learning Project

Miao

August 20, 2015

1 Assignment

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).

2 Preparation of Working Environment

```
setwd("D:/R document/coursera/predmachlearn/project")
library(randomForest)
library(Hmisc)
library(caret)
library(doParallel)
library(foreach)
options(warn=-1)
set.seed(9999)
```

3 Data Processing

The values containing "#DIV/0!" are replaced with an NA value.

```
DataProcess <- function(file, url) {
  if (! file.exists(file)) {
    paste("downloading", url, "into", file, "...")
    download.file(url, file, method="wget")
  }
  read.csv(file, na.strings=c("#DIV/0!"))
}
training <- DataProcess("pml-training.csv",

"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
testing <- DataProcess("pml-testing.csv",

"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
```

```
head(training)
head(testing)
```

4 Preparation of Dataset for Analysis

After checking the dataset, there are some characters for these data. First, the first 7 columns contains user names, time stamps, and windows. They should be removed. Then, there are many columns only containing "NA". They have little contribution for the modeling. So the dataset need to be cleaned up for modeling purpose.

```
# removal of first 7 columns, and cast other columns into numeric
for(i in c(8:ncol(training)-1)) {training[,i] =
as.numeric(as.character(training[,i]))}

for(i in c(8:ncol(testing)-1)) {testing[,i] =
as.numeric(as.character(testing[,i]))}

# Dataset for modeling
ExtractData <- colnames(training[colSums(is.na(training)) == 0])[-
(1:7)]
ModelData<- training[ExtractData]
ExtractData
```

5 Create Data Partitions for Cross-Validation

```
library(caret)

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

partition <- createDataPartition(y=ModelData$classe, p=0.75, list=FALSE
)
training <- ModelData[partition,]
testing <- ModelData[-partition,]
```

6 Random Forest Model

Random Forest Model iss used to fit the data. Also parallel processing with random forests is used to speed up the process.

```
library(doParallel)

## Warning: package 'doParallel' was built under R version 3.2.2

## Loading required package: foreach
## Loading required package: iterators
## Loading required package: parallel

registerDoParallel()
x <- training[-ncol(training)]
y <- training$classe
```

```
rf_model <- foreach(ntree=rep(150, 6), .combine=randomForest::combine,
.packages='randomForest') %dopar% {
randomForest(x, y, ntree=ntree)
}
```

7 Create Error Reports

```
predict1 <- predict(rf_model, newdata=training)
confusionMatrix(predict1,training$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
```

```
predict2 <- predict(rf_model, newdata=testing)
confusionMatrix(predict2,testing$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    5    0    0    0
##          B    0  942    5    0    0
##          C    0    2  850   11    0
##          D    0    0    0  793    2
##          E    0    0    0    0  899
##
## Overall Statistics
##
##                Accuracy : 0.9949
##                  95% CI : (0.9925, 0.9967)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9936
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9926   0.9942   0.9863   0.9978
## Specificity            0.9986   0.9987   0.9968   0.9995   1.0000
## Pos Pred Value         0.9964   0.9947   0.9849   0.9975   1.0000
## Neg Pred Value         1.0000   0.9982   0.9988   0.9973   0.9995
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1921   0.1733   0.1617   0.1833
## Detection Prevalence   0.2855   0.1931   0.1760   0.1621   0.1833
## Balanced Accuracy      0.9993   0.9957   0.9955   0.9929   0.9989
```

8 Data Submission

```
ExtractData <- colnames(training)
newdata<- testing

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=FA
LSE)
  }
}

x <- testing
x <- x[ExtractData[ExtractData!='classe']]
answers <- predict(rf_model, newdata=x)

answers

pml_write_files(answers)
```

9 Conclusion

From the confusion matrix, we can know the model fit the dataset perfectly. The accuracy of the model fitting test data is around 99%. Other models were also tested, but cannot acheive the comparable accuracy as this one. The submission results are all right according to the submission part of this project.