Prediction-Assignment

AM

2023-01-23

Course Project

Executive Summary

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.The goal of your project is to predict the manner in which they did the exercise.

Load libraries

```
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(randomForest)

## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

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

### Download the Data
train <- read.csv("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv", na.strings=c("NA","#DIV/0!", ""))
test <- read.csv('http://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv', na.strings=c("NA","#DIV/0!", ""))
```

```
### Exploratory Analysis
dim(train)

## [1] 19622   160

dim(test)

## [1]  20 160

### Clean data
train <-train[,colSums(is.na(train)) == 0]
test <-test[,colSums(is.na(test)) == 0]
train <-train[,-c(1:7)]
test <-test[,-c(1:7)]

### Slice the data
set.seed(1234)
trainpart <- createDataPartition(y=train$classe, p=0.75, list=FALSE)
Training <- train[trainpart, ]
Testing <- train[-trainpart, ]

### Model Train
rfModel <- randomForest(as.factor(classe)~ ., data = Training, importance =
TRUE, ntrees = 10)
predictRf <- predict(rfModel , Training)
u1 <- union(predictRf ,Training$classe)
t1 <- table(factor(predictRf, u1), factor(Training$classe, u1))
print(confusionMatrix(t1))

## Confusion Matrix and Statistics
##
##
##         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
```

```
### Test data prediction
predictRf <- predict(rfModel , test)
predictRf
```

```
##  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
```