Practical machine learning project

Xi, MIAO

11/11/2020

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.

Downloading and cleaning data

1.Download data from the links as csv 2.Set invalid data to NA 3.Remove unnecessary columns 4.Remove NA columns

```
traincsv <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
temptrain <- tempfile()
download.file(traincsv, temptrain, method = "curl")
pml_training <- read.csv(temptrain, na.strings = c("NA", "#DIV/0!", ""))

testcsv <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
temptest <- tempfile()
download.file(testcsv, temptest, method = "curl")
pml_testing <- read.csv(temptest, na.strings = c("NA", "#DIV/0!", ""))

col_len <- length(colnames(pml_training))

pml_training <- pml_training[,8:col_len]
pml_testing <- pml_testing[,8:col_len]

col_trainning <- complete.cases(t(pml_training))
col_test <- complete.cases(t(pml_testing))
col_complete <- col_trainning & col_test

pml_training <- pml_training[,col_complete]
pml_testing <- pml_testing[,col_complete]

pml_training$classe <- factor(pml_training$classe)
```

Partitioning training data and validation data

Use 70% training data for training and the rest for validation

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
set.seed(191211)
trainset <- createDataPartition(pml_training$classe, p = 0.7, list = FALSE)
pml_validation <- pml_training[-trainset, ]
pml_training <- pml_training[trainset, ]
```

Model train

Train model with random forest

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
fit <- randomForest(classe~., data = pml_training,importance=TRUE)
```

Model validation

Training set accuracy

```
pred_train <- predict(fit,pml_training)
pred_train <- factor(pred_train)
confusionMatrix(pred_train,pml_training$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 3906    0    0    0    0
```

```
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
##
## 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
```

The result is close to 1, the model performs well

Validation set accuracy

```
pred_validation <- predict(fit,pml_validation)
pred_validation <- factor(pred_validation)
confusionMatrix(pred_validation,pml_validation$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    2    0    0    0
##          B    2 1135    2    0    0
##          C    0    2 1024   14    4
##          D    0    0    0  947    0
##          E    0    0    0    3 1078
##
## Overall Statistics
##
##                Accuracy : 0.9951
##                  95% CI : (0.9929, 0.9967)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.9938
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9988   0.9965   0.9981   0.9824   0.9963
## Specificity            0.9995   0.9992   0.9959   1.0000   0.9994
## Pos Pred Value         0.9988   0.9965   0.9808   1.0000   0.9972
## Neg Pred Value         0.9995   0.9992   0.9996   0.9966   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2841   0.1929   0.1740   0.1609   0.1832
## Detection Prevalence   0.2845   0.1935   0.1774   0.1609   0.1837
## Balanced Accuracy      0.9992   0.9978   0.9970   0.9912   0.9978
```

The result is close to 1, the model performs well

Test Prediction

The prediction of test set is:

```
pred_test <- predict(fit,pml_testing)
print(pred_test)
```

```
##  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
```