MachineLearning

Jason

5/24/2020

Introduction

The assignment's purpose is to analyze the data provided on personal fitness actvity, and determine how much of a particular activity people routinely perform and how well they perform it.

Data Analysis Steps

1. Summary of dataset

Six young participants performed various fitness workout activities, and thier performance is recorded in 5 classes of data (Class A, B, C, D, E). Class A refers to the specified execution of the excercise, while the rest correspond with occurances of mistakes.

2. Data Analysis

First the requried packages are loaded.

```
library(caret)
```

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

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

```
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(foreach)
```

```
set.seed(20150125)
```

Then the required datasets are loaded into R.

```
setwd("~/R/MachineLearning")
trainingSrc   <- read.csv('pml-training.csv', na.strings=c("NA","#DIV/0!", ""))
testSrc       <- read.csv('pml-testing.csv' , na.strings=c("NA", "#DIV/0!", ""))
```

The following steps are for tidying data

```
goodVars    <- which((colSums(!is.na(trainingSrc)) >= 0.6*nrow(trainingSrc)))
trainingSrc <- trainingSrc[,goodVars]
testSrc     <- testSrc[,goodVars]
```

```
# remove problem id
testSrc <- testSrc[-ncol(testSrc)]
# fix factor levels
testSrc$new_window <- factor(testSrc$new_window, levels=c("no","yes"))
```

```
trainingSrc <- trainingSrc[,-c(1,5)]
testSrc     <- testSrc[,-c(1,5)]
```

Now splitting data into training and testing sets.

```
inTraining  <- createDataPartition(trainingSrc$classe, p = 0.6, list = FALSE)
training    <- trainingSrc[inTraining, ]
testing     <- trainingSrc[-inTraining, ]
```

For random forest modelling

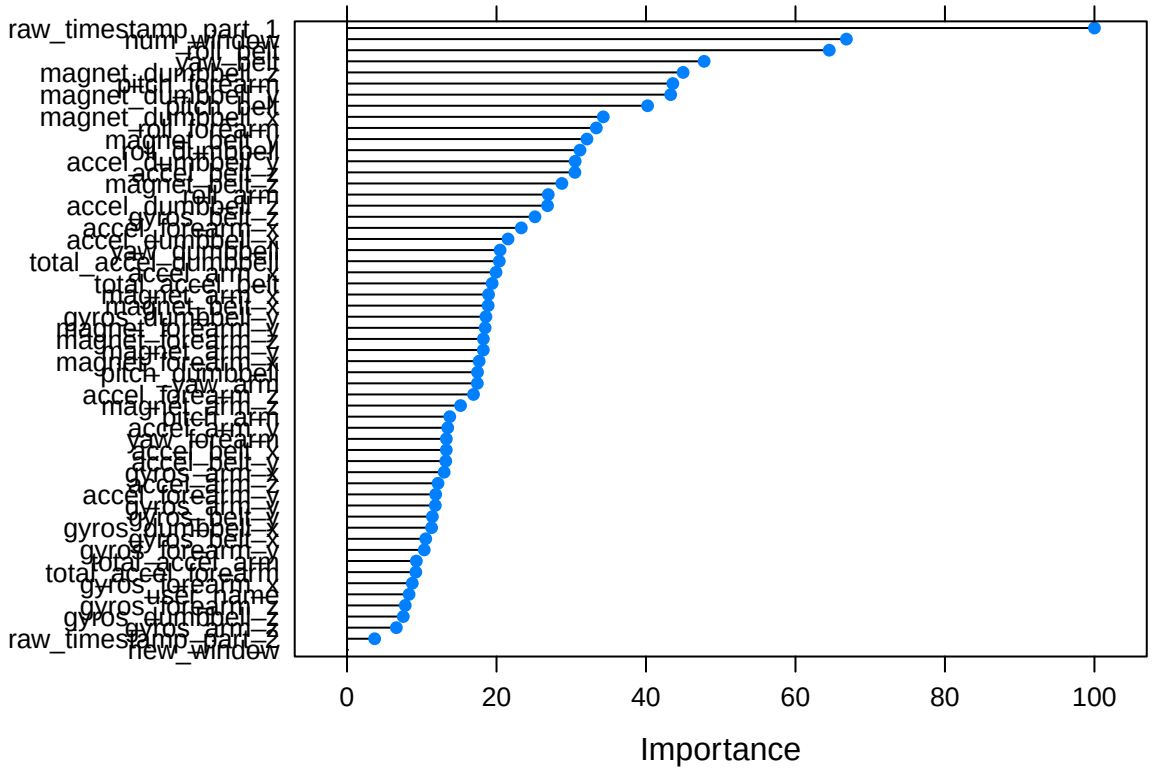
```
class <- training$classe
data  <- training[-ncol(training)]
```

```
registerDoParallel()
rf <- train(data, class, method="parRF",
    tuneGrid=data.frame(mtry=3),
    trControl=trainControl(method="none"))
```

```
rf
```

```
## Parallel Random Forest
##
## 11776 samples
##    57 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: None
```

```
plot(varImp(rf))
```



For confusion matrix testing set

```
testingPredictions <- predict(rf, newdata=testing)
confMatrix <- confusionMatrix(factor(testingPredictions),factor(testing$classe))
confMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    3    0    0    0
##          B    0 1513    2    0    0
##          C    0    2 1366    8    0
##          D    0    0    0 1278    2
##          E    0    0    0    0 1440
##
## Overall Statistics
##
##                Accuracy : 0.9978
##                  95% CI : (0.9965, 0.9987)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9973
##
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9967   0.9985   0.9938   0.9986
## Specificity            0.9995   0.9997   0.9985   0.9997   1.0000
## Pos Pred Value         0.9987   0.9987   0.9927   0.9984   1.0000
## Neg Pred Value         1.0000   0.9992   0.9997   0.9988   0.9997
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1928   0.1741   0.1629   0.1835
## Detection Prevalence   0.2849   0.1931   0.1754   0.1631   0.1835
## Balanced Accuracy      0.9997   0.9982   0.9985   0.9967   0.9993
```

The accuracy is provided in the following code:

```
confMatrix$overall[1]
```

```
##  Accuracy
## 0.9978333
```

to test the result on the test set:

```
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)
  }
}

answers <- predict(rf, testSrc)
pml_write_files(answers)
```