Using Algorithm Random Forest to predict personal activity

Using Algorithm Random Forest to predict personal activity

Summary: Using ramdom forest algorithm to training the model on 70% data, and use 30% data to test the result. before training the model, remove the columns with low variance, too many 'NA' (more than 50% of # of rows) and remove time/date related columns.

Backgound: 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 â€“ 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.

Data processing, set workdir and load data.

```
setwd("c:\\temp\\ml")
library(caret)
pml_data <- read.csv("pml-training.csv")
```

remove the columns with low variance

```
col_rm <- nearZeroVar(pml_data)
pml_data1 <- pml_data[, -col_rm]
```

remove column with too many 'NA' (more than 50% of # of rows)

```
pml_data2 <- pml_data1[ ,colSums(is.na(pml_data1))<nrow(pml_data1)/2]
```

remove column time/date

```
pml_data3 <- pml_data2[ ,c(-1,-3,-4,-5)]
```

Select 70% data as training data

```
randomSelection <- createDataPartition(pml_data3$classe, p = 0.7, list = FALSE)
modelTraining <- pml_data3[randomSelection, ]
modelTesting <- pml_data3[-randomSelection, ]
```

training RF model, 1.5 hours, remarked to save time, as saved previously.

```
# rfModel <- train(classe ~ ., data=modelTraining, method="rf", verbose=FALSE)
```

save / load the model to save time

```
if (!file.exists("rfModel.save")) {
        rfModel <- rfModel
        save(rfModel, file="rfModel.save")
} else {
        load("rfModel.save")
}
```

test on 30% data and show confusion matrix

```
rfPredictions <- predict(rfModel, modelTesting)
confusionMatrix(rfPredictions, modelTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1139    0    0    0
##          C    0    0 1026    0    0
##          D    0    0    0  964    1
```

```
##          E    0    0    0    0 1081
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.999, 1)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    1.000    1.000    1.000    0.999
## Specificity             1.000    1.000    1.000    1.000    1.000
## Pos Pred Value          1.000    1.000    1.000    0.999    1.000
## Neg Pred Value          1.000    1.000    1.000    1.000    1.000
## Prevalence              0.284    0.194    0.174    0.164    0.184
## Detection Rate          0.284    0.194    0.174    0.164    0.184
## Detection Prevalence    0.284    0.194    0.174    0.164    0.184
## Balanced Accuracy       1.000    1.000    1.000    1.000    1.000
```

data source: http://groupware.les.inf.puc-rio.br/har