Practical Machine Learning_Course Project

Luoning

6/13/2020

Background

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.

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Data loading and cleaning

Dataset

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

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

The data for this project come from http://groupware.les.inf.puc-rio.br/har.

Environment prepration

```
library(caret)
```

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

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

```
library(knitr)
library(rpart)
library(rpart.plot)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(reshape2)
```

Data loading and cleaning

The next step is loading the dataset from the URL provided above. The training dataset is then partinioned in 2 to create a Training set (70% of the data) for the modeling process and a Test set (with the remaining 30%) for the validations. The testing dataset is not changed and will only be used for testing the trained model.

```
urltrain<-'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
urlval<-'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'

data<-read.csv(url(urltrain),header = T, na.strings = c("", "NA"))
validation<-read.csv(url(urlval),header = T, na.strings = c("", "NA"))

dim(data)
```

```
## [1] 19622   160
```

```
dim(validation)
```

```
## [1]  20 160
```

The datasets have 160 variables. But many of them have lots of NAs or near zero variables.

```
#remove variables with nearly zero variance
NZVvar<-nearZeroVar(data)
fulldata<-data[,-NZVvar]
validation<-validation[,-NZVvar]
dim(fulldata)
```

```
## [1] 19622   117
```

```
#remove variables that are mostly NA
NAratio<-sapply(fulldata, function(x) mean(is.na(x)))>0.95
fulldata<-fulldata[,NAratio==FALSE]
validation<-validation[,NAratio==FALSE]
intrain<-createDataPartition(fulldata$classe,p=0.7,list = FALSE)
training<-fulldata[intrain,]
training<-training[,-(1:5)]
testing<-fulldata[-intrain,]
testing<-testing[,-(1:5)]
dim(training)
```

```
## [1] 13737    54
```

```
dim(testing)
```

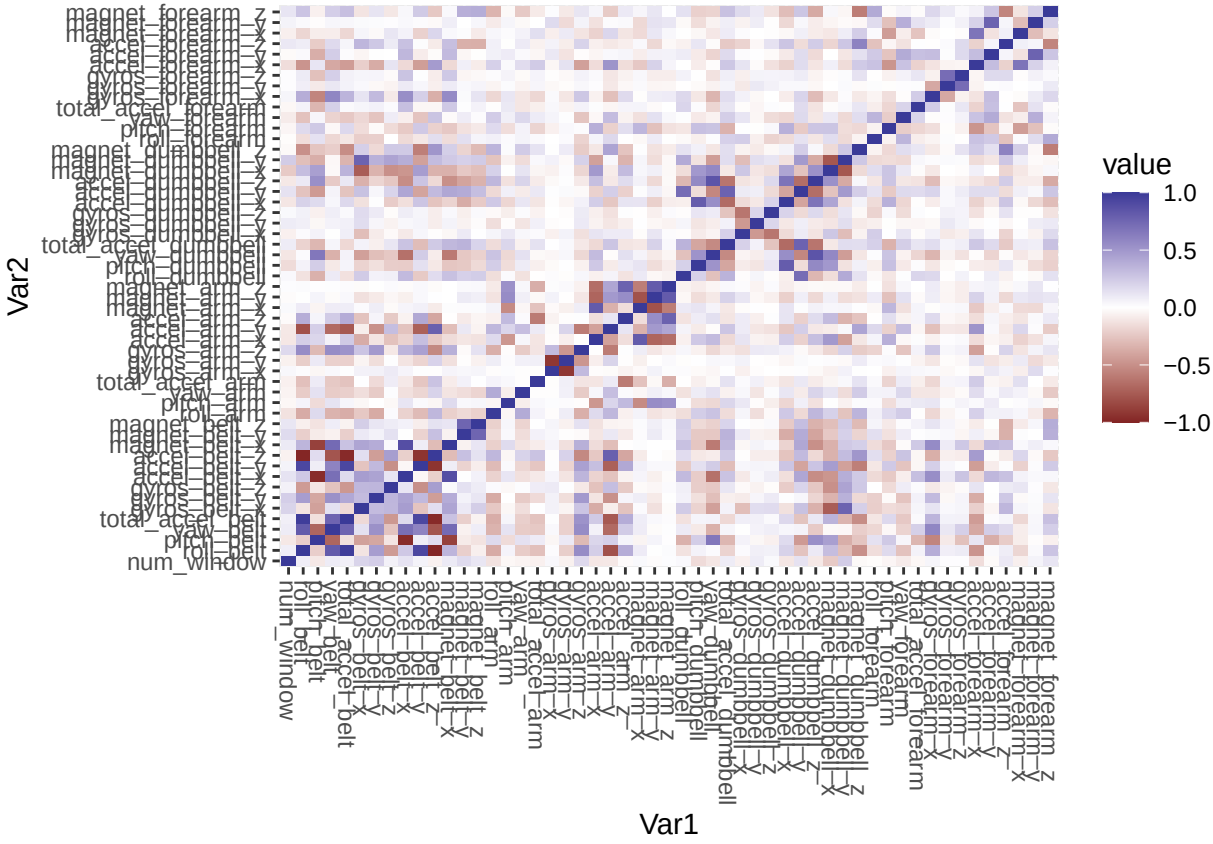
```
## [1] 5885   54
```

With the cleaning process above, the number of variables for the analysis has been reduced to 54 only.

Correlation analysis

To analyze the correlation between variables, a correlation map is plotted.

```
cor.matrix<-cor(training[sapply(training, is.numeric)])
temp<-melt(cor.matrix)
qplot(x=Var1, y=Var2, data=temp, fill=value, geom="tile") +
   scale_fill_gradient2(limits=c(-1, 1)) +
    theme(axis.text.x = element_text(angle=-90, vjust=0.5, hjust=0))
```



Some of the variables appear to be highly correlated, and the correlated variables were removed using threshold (> 0.9).

```
temp<-findCorrelation(cor.matrix,cutoff = 0.9)
training<-training[,-temp]
testing<-testing[,-temp]
dim(training)
```

```
## [1] 13737    49
```

```
dim(testing)
```

```
## [1] 5885   49
```

Prediction model

Random forest

```
set.seed(21218)
modFitRF<-train(classe~.,data = training,method='rf',trControl=trainControl(method="cv", number=3, verbo
modFitRF$finalModel
```

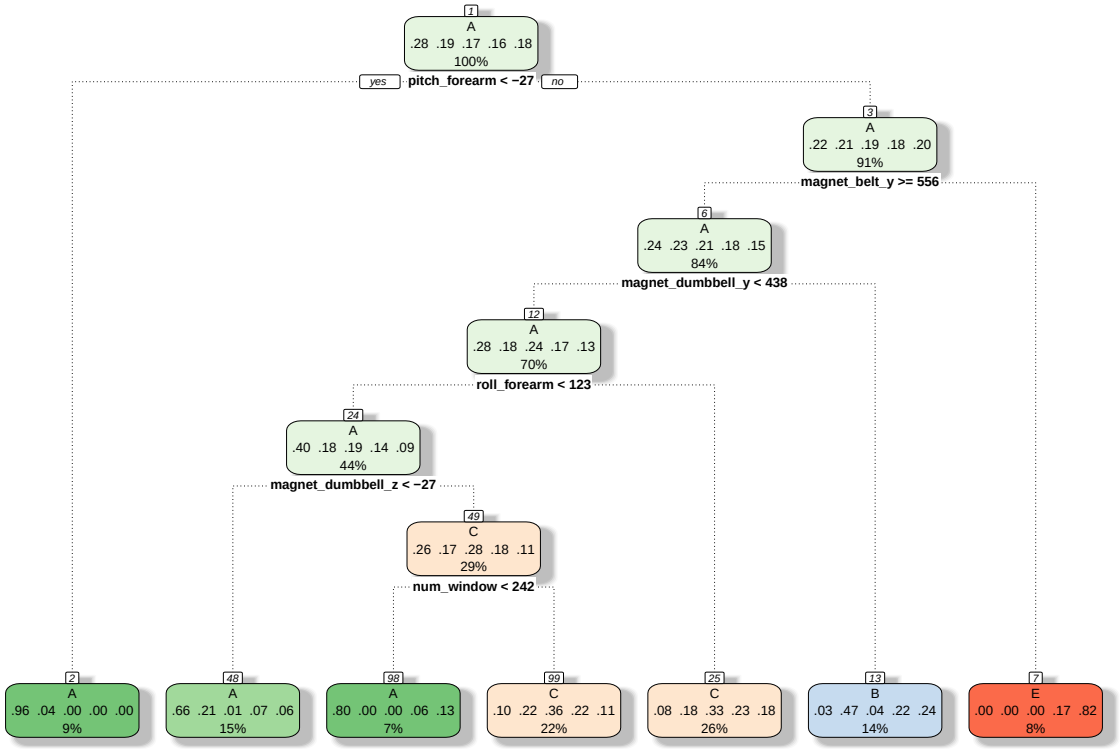
```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 25
##
##         OOB estimate of  error rate: 0.22%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3904    2    0    0    0 0.0005120328
## B    7 2649    2    0    0 0.0033860045
## C    0    5 2389    2    0 0.0029215359
## D    0    0    8 2244    0 0.0035523979
## E    0    0    0    4 2521 0.0015841584
```

```
predRF<-predict(modFitRF,newdata=testing)
confRF<-confusionMatrix(predRF,testing$classe)
confRF$overall[1]
```

```
##  Accuracy
## 0.9966015
```

Decision trees

```
set.seed(21218)
modFitDT<-train(classe~.,data = training,method='rpart')
fancyRpartPlot(modFitDT$finalModel)
```



Rattle 2020−Jun−13 23:02:05 luoni

```
predDT<-predict(modFitDT,testing)
confDT<-confusionMatrix(predDT,testing$classe)
confDT$overall[1]
```

```
##  Accuracy
## 0.5274427
```

Generalized Boosting

```
set.seed(21218)
ControlGBM<-trainControl(method = "repeatedcv", number = 5, repeats = 1)
modFitGBM<-train(classe~.,data = training,method='gbm',trControl=ControlGBM,verbose=FALSE)
modFitGBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 48 predictors of which 48 had non-zero influence.
```

```
predGBM<-predict(modFitGBM,newdata=testing)
confGBM<-confusionMatrix(predGBM,testing$classe)
confGBM$overall[1]
```

```
##  Accuracy
## 0.9913339
```

Predictions

The accuracy for the models is Random forest > Generalized boosting > Decision tree. Apply the Random forest model to predict the 20 testing data.

```
predTest<-predict(modFitRF,validation)
predTest
```

```
##  [1] B A A A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```