Praticle Machine Learning Assignment4

YunShen

2/2/2018

Praticle Machine Learning Assginment 4:HAR(Human Activity Analysis)

library loading

```
library(knitr)
library(caret)
```

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

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

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

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

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

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(e1071)
```

1 Dataloading and cleaning

First download the datasets if they are not in the working director then cleaning the predictors with near zero variance method then is none or the percentage of none value is more than 95%.

```
# reading datasets
if(!file.exists("pml-training.csv") | !file.exists("pml-testing.csv")){
    trainingurl<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
    testingurl<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
    training<-read.csv(url(trainingurl))
    testing<-read.csv(url(trainingurl))
}else{
    training<-read.csv("pml-training.csv")
    testing<-read.csv("pml-testing.csv")
}
#preproceesing trainining
NZV<-nearZeroVar(training)
training<-training[,-NZV]
```

```r
trainingNA<-sapply(training, function(x) mean(is.na(x))>0.97)
trainingnew<-training[,trainingNA == FALSE]
#remove idntification
trainingnew<-trainingnew[,-(1:6)]
dim(trainingnew)
```

```
## [1] 19622    53
```

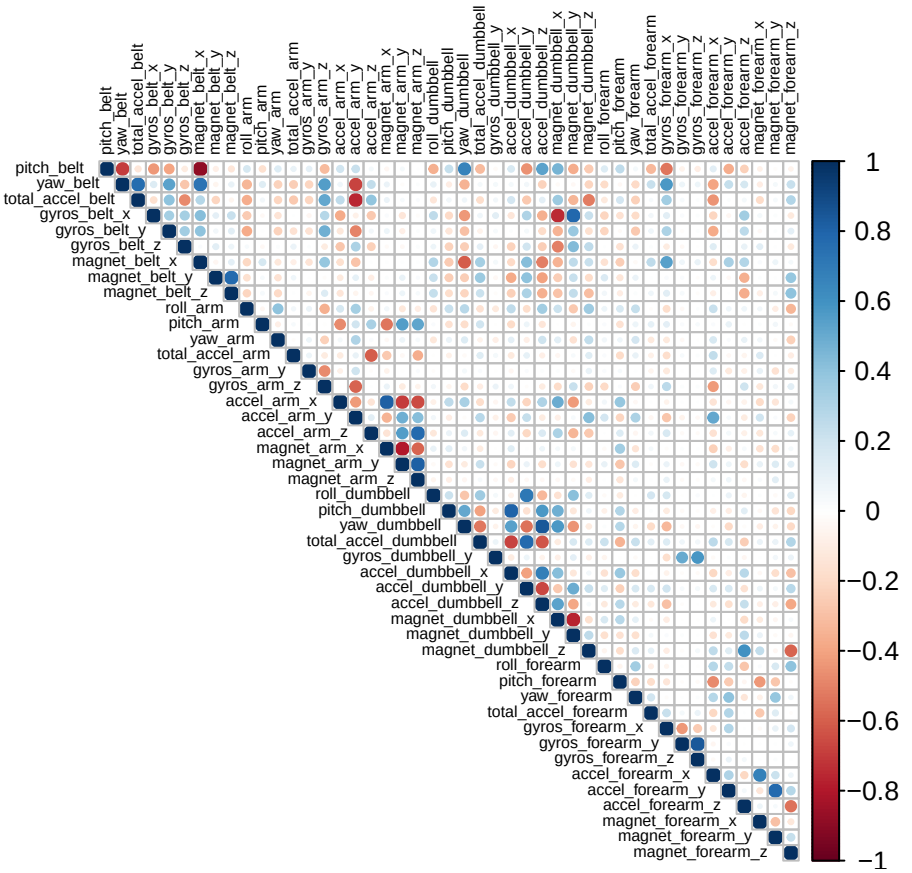
2) Feature Selection

Since reduce the number of predictors for feature selection,first preprocessing, then remove the highly correlated features ,then perform PCA onto them.

```r
removehighlycorr<-preProcess(trainingnew[,-53],method="corr")
trainingnew2<-predict(removehighlycorr,trainingnew[,-53])
testingnew2<-testing[,colnames(trainingnew2)]
zscale<-preProcess(trainingnew2)
scaledtrainingnew2<-predict(zscale,trainingnew2)
scaledtestingnew2<-predict(zscale,testingnew2)
pcatrain<-preProcess(scaledtrainingnew2,metho="pca", thresh = 0.99)
pcatrain
```

```
## Created from 19622 samples and 45 variables
##
## Pre-processing:
##   - centered (45)
##   - ignored (0)
##   - principal component signal extraction (45)
##   - scaled (45)
##
## PCA needed 35 components to capture 99 percent of the variance
```

```r
corrplot(cor(scaledtrainingnew2),type="upper",order = "original",tl.col = "black",tl.srt = 90,number.fo
```



From the correlation matrix after remove the highly correlated feature and the PCA anaysis, we need 35 components which didn't reduce much predictor from the one just removed highly coreelated predictor(45), so we will not perform PCA on the scaled traingning set

```
trainingfinal<-cbind(scaledtrainingnew2,trainingnew[,53])
# spliting the training datasets
colnames(trainingfinal)[46]<-"classe"
inTrain<-createDataPartition(trainingfinal$classe, p=0.75, list = FALSE)
trainingset<-trainingfinal[inTrain,]
validationset<-trainingfinal[-inTrain,]
```

3) Training model

a) RandomForest

```
set.seed(1000)
RFsetting <- trainControl(method="cv", number=3, verboseIter=FALSE)
FitRandForest <- train(classe ~ ., data=trainingset, method="rf",
                           trControl=RFsetting)
FitRandForest$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: 23
##
##         OOB estimate of  error rate: 0.56%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4179    3    1    0    2 0.001433692
## B   16 2826    5    0    1 0.007724719
## C    0   12 2549    6    0 0.007012076
## D    0    0   22 2387    3 0.010364842
## E    0    0    4    7 2695 0.004065041
```

```
predictRandForest <- predict(FitRandForest, newdata=validationset)
confMatRandForest <- confusionMatrix(predictRandForest, validationset$classe)
confMatRandForest$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    4    0    0    0
##          B    1  943    2    0    0
##          C    0    2  850   10    2
##          D    0    0    3  794    3
##          E    0    0    0    0  896
```

```
accuracyRF<-confMatRandForest$overall[1]
```

The acuracy of random forest for the validation set is 0.9944943

b) SVM

```
set.seed(1000)
fitsvm <- svm(classe ~ . , trainingset)
predictsvm<-predict(fitsvm,newdata = validationset)
confMatSVM <- confusionMatrix(predictsvm, validationset$classe)
accuracySVM<-confMatSVM$overall[1]
```

Both the SVM model and random forest model have high accuracy,x but RF is better compared to the SVM acuracy 0.9459625

4) Predicting testingset

```
predict(FitRandForest, newdata=scaledtestingnew2)
```

```
##  [1] 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
```