Finalproject.R

jy298

Sun Jul 26 16:45:11 2015

```
library(ggplot2)
library(caret)
```

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

```
library(e1071)
#enable parallel computing
library(doParallel)
```

```
## Loading required package: foreach
## Loading required package: iterators
## Loading required package: parallel
```

```
registerDoParallel(cores=2)
#load training set
setwd("~/Documents/Coursera/Machine Learning")
dataT<-read.csv("pml-training.csv")
# remove unnecessary data
dataT<-dataT[,-grep("avg",names(dataT))]
dataT<-dataT[,-grep("var",names(dataT))]
dataT<-dataT[,-grep("stddev",names(dataT))]
dataT<-dataT[,-grep("amplitude",names(dataT))]
dataT<-dataT[,-grep("min",names(dataT))]
dataT<-dataT[,-grep("max",names(dataT))]
dataT<-dataT[,-grep("kurtosis",names(dataT))]
dataT<-dataT[,-grep("skewness",names(dataT))]
dataT<-dataT[,-grep("times",names(dataT))]
dataT<-dataT[,-grep("window",names(dataT))]
dataT<-dataT[,-grep("user_name",names(dataT))]
dataT<-dataT[,-grep("X",names(dataT))]

set.seed(666)

inTrain<-createDataPartition(y=dataT$classe,p=0.6,list=FALSE)
training<-dataT[inTrain,]
testing<-dataT[-inTrain,]
modFit<-train(classe~.,data=training,method="parRF",ntree=25,prox=TRUE)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
#crossvalidate  test on testing set
Crossvalidate <- predict(modFit, testing)
confusionMatrix(testing$classe, Crossvalidate)$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper
AccuracyNull
##      0.9899312      0.9872613      0.9874668      0.9920205
0.2862605
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

```
# predict on test data set
dataS<-read.csv("pml-testing.csv")
dataS<-dataS[,-grep("avg",names(dataS))]
dataS<-dataS[,-grep("var",names(dataS))]
dataS<-dataS[,-grep("stddev",names(dataS))]
dataS<-dataS[,-grep("amplitude",names(dataS))]
dataS<-dataS[,-grep("min",names(dataS))]
dataS<-dataS[,-grep("max",names(dataS))]
dataS<-dataS[,-grep("kurtosis",names(dataS))]
dataS<-dataS[,-grep("skewness",names(dataS))]
dataS<-dataS[,-grep("times",names(dataS))]
dataS<-dataS[,-grep("window",names(dataS))]
dataS<-dataS[,-grep("user_name",names(dataS))]
dataS<-dataS[,-grep("X",names(dataS))]

Result<- predict(modFit, dataS)
Result
```

```
##  [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
```