Application of Machine Learning

Navya

10/23/2020

Application of Machine Learning

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.

The goal of this project is to quantify exercises performed by six participants analyzing the data obtained from accelerometers of dumbell, forearm,arm and belt

```
library(caret)
library(rpart)
library(ggplot2)
library(corrplot)
set.seed(1000)
```

Loading Dataset:

```
tstUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trUrl  <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"

tst <- read.csv(url(tstUrl))[,-1]
tr <- read.csv(url(trUrl))[,-1]
```

Data Visualization:

```
dim(tr)
```

```
## [1] 19622   159
```

```
dim(tst)
```

```
## [1]  20 159
```

Data Exploration for Preprocessing the Data:

```r
#Cleaning the Data:
NV <- nearZeroVar(tr)
tr <- tr[, -NV]
tst <- tst[, -NV]

NV <- sapply(tr, function(x) mean(is.na(x))) > 0.9
tr <- tr[, NV == "FALSE"]
tst <- tst[, NV == "FALSE"]

# Clearing Time and ID values
tr<- tr[,-c(1:5)]
tst <- tst[,-c(1:5)]

dim(tr)
```

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

```r
dim(tst)
```

```
## [1] 20 53
```

```r
table(tr$classe)
```

```
## 
##    A    B    C    D    E 
## 5580 3797 3422 3216 3607
```

Modelling with Cross Validation:

```r
set.seed(1234)
cv3 = trainControl(method="cv",number=3,allowParallel=TRUE,verboseIter=TRUE)
modrf = train(classe~., data=tr, method="rf",trControl=cv3)
```

```
## + Fold1: mtry= 2 
## - Fold1: mtry= 2 
## + Fold1: mtry=27 
## - Fold1: mtry=27 
## + Fold1: mtry=52 
## - Fold1: mtry=52 
## + Fold2: mtry= 2 
## - Fold2: mtry= 2 
## + Fold2: mtry=27 
## - Fold2: mtry=27 
## + Fold2: mtry=52 
## - Fold2: mtry=52 
## + Fold3: mtry= 2 
## - Fold3: mtry= 2 
## + Fold3: mtry=27 
## - Fold3: mtry=27 
## + Fold3: mtry=52
## - Fold3: mtry=52
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 2 on full training set
```

```
modtree = train(classe~.,data=tr,method="rpart",trControl=cv3)
```

```
## + Fold1: cp=0.03568
## - Fold1: cp=0.03568
## + Fold2: cp=0.03568
## - Fold2: cp=0.03568
## + Fold3: cp=0.03568
## - Fold3: cp=0.03568
## Aggregating results
## Selecting tuning parameters
## Fitting cp = 0.0357 on full training set
```

```
prf=predict(modrf,tr)
ptree=predict(modtree,tr)
table(prf,tr$classe)
```

```
## 
## prf    A    B    C    D    E
##   A 5580    0    0    0    0
##   B    0 3797    0    0    0
##   C    0    0 3422    0    0
##   D    0    0    0 3216    0
##   E    0    0    0    0 3607
```

```
table(ptree,tr$classe)
```

```
## 
## ptree    A    B    C    D    E
##     A 5080 1581 1587 1449  524
##     B   81 1286  108  568  486
##     C  405  930 1727 1199  966
##     D    0    0    0    0    0
##     E   14    0    0    0 1631
```

```
prf=predict(modrf,tst)
ptree=predict(modtree,tst)
table(prf,ptree)
```

```
##    ptree
## prf A B C D E
##   A 7 0 0 0 0
##   B 3 0 5 0 0
##   C 0 0 1 0 0
##   D 0 0 1 0 0
##   E 1 0 2 0 0
```

Conclusion:

```
r=predict(modrf,tst)
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)
  }
}
r
```

```
##  [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
```

The Predicted Classes for the TestCases are as follows:

```
pml_write_files(r)
```