Machine Learning11

Saket Rathi

June 29, 2016

# Machine Learning Programme

#### Loading Required Packages

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

#### Read both training and test data

test <-read.csv("C:/Users/saket/Desktop/Data Science/coursera ass2/Machine Learning/pml-testing.csv")  
train <- read.csv("C:/Users/saket/Desktop/Data Science/coursera ass2/Machine Learning/pml-training.csv")

#### Looking into the data dimension of Test and Train Data

dim(train)

## [1] 19622 160

dim(test)

## [1] 20 160

#### Values of columns

head(train)  
head(test)

#### From basic observation we can see we need to remove few variables from our analysis .Index, user\_name,rawtimestamp1 and 2, Cvtd timestamp

train<-train[,-c(1:6)]

#### still training set has 160-6 variables that are too many to make any practical inference

train<- train[ ,-nearZeroVar(train)]

#### We have got a good starting point but still we could futher reduce removing NA lets remove columnwith 90% values being NA

limit <- length(train)\*.90  
index<- !apply(train,2,function(x) sum(is.na(x))> limit || sum(x=="")> limit)  
train<-train[,index]

#### Looking into the data dimension of Test and Train Data

dim(train)

## [1] 19622 54

dim(test)

## [1] 20 160

#### Now we have reasonable number of predictotrs

#### Usually for better prediction mtry should be p/3-> 54/3= 18 .Due to computer harware limitation I have ran only for mtry =5 which gave me good estimate.With little error

modelfit <- randomForest(classe~.,data=train,mtry=5)  
modelfit

##   
## Call:  
## randomForest(formula = classe ~ ., data = train, mtry = 5)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 0.18%  
## Confusion matrix:  
## A B C D E class.error  
## A 5579 1 0 0 0 0.0001792115  
## B 3 3793 1 0 0 0.0010534633  
## C 0 8 3413 1 0 0.0026300409  
## D 0 0 16 3199 1 0.0052860697  
## E 0 0 0 4 3603 0.0011089548

#### For mtry = 5 we got.16% OOB error. Our accuracy will be 1-OOB error .Which is 100-.16= 99.84% error.

#### CV claculation if you want to do.We can optimize the K value using cross validation and making mtry as variable to be optimized but currently my system is not supporting length calculation . Although I have tried it for mtry = 2 and mtry =5 .Accuracy increases.

#### For practical Purpose to solve the given problem you dont need this as accuracy is already very high . I have done the quiz and all predictions matched.But sure Cross Validation will improve your accuracy further.

#### prediction

test1<- predict(modelfit,test)  
 test1

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 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