course\_8

abhinav palaparthy

4 November 2017

## Loading the data

library(caret)

## Warning: package 'caret' was built under R version 3.4.2

## Loading required package: lattice

## Loading required package: ggplot2

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.4.2

## 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

traindata <- read.csv("./training.csv")  
testdata<-read.csv("./testing.csv")

## Data Exploration

Used str(traindata) in order to examine types of coloumns and dimensions. Used table(training$classe) to examine number of unique classe.We observe that all non numerical variables exempt classe are irrelevent in our model.Moreover there exists incomplete data sets which will make our analysis difficult , I have remove them altogether. (not shown results to optimize space) ## Cleaning Data and preparation

valid\_traincols<-colSums(is.na(traindata))  
valid\_testcols<-colSums(is.na(testdata))  
traindata<-traindata[,valid\_traincols==0]  
testdata<-testdata[,valid\_traincols==0]  
  
idw <- which(names(traindata) %in% c("X","raw\_timestamp\_part\_1","raw\_timestamp\_part\_2","cvtd\_timestamp" ) )  
traindata <- traindata[,-idw]  
testdata<-testdata[,-idw]  
  
  
classe <- traindata$classe  
train<-traindata[,sapply(traindata,is.numeric)]  
train$classe<-classe  
test <- testdata[, sapply(testdata, is.numeric)]

## Creating cross validation set

Now we split the preprocessed training data into training set and validation set.(70,30 split)

set.seed(696569)  
x <- createDataPartition(train$classe, p=0.70, list=F)  
tset <- train[x, ]  
vset <- train[-x, ]

## prediction

tr <- trainControl(method="cv", 5)  
model\_rf <- train(classe ~ ., data=tset, method="rf", trControl=tr, ntree = 250)  
model\_rf

## Random Forest   
##   
## 13737 samples  
## 53 predictor  
## 5 classes: 'A', 'B', 'C', 'D', 'E'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 10989, 10988, 10991, 10991, 10989   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.9939577 0.9923560  
## 27 0.9968698 0.9960406  
## 53 0.9947586 0.9933696  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 27.

use confusion matrix to measure sample error and accuracy

p <- predict(model\_rf, vset)  
confusionMatrix(vset$classe,p)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1673 0 0 0 1  
## B 2 1136 1 0 0  
## C 0 1 1025 0 0  
## D 0 0 6 958 0  
## E 0 0 0 0 1082  
##   
## Overall Statistics  
##   
## Accuracy : 0.9981   
## 95% CI : (0.9967, 0.9991)  
## No Information Rate : 0.2846   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9976   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.9988 0.9991 0.9932 1.0000 0.9991  
## Specificity 0.9998 0.9994 0.9998 0.9988 1.0000  
## Pos Pred Value 0.9994 0.9974 0.9990 0.9938 1.0000  
## Neg Pred Value 0.9995 0.9998 0.9986 1.0000 0.9998  
## Prevalence 0.2846 0.1932 0.1754 0.1628 0.1840  
## Detection Rate 0.2843 0.1930 0.1742 0.1628 0.1839  
## Detection Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Balanced Accuracy 0.9993 0.9992 0.9965 0.9994 0.9995

accuracy <- postResample(p,vset$classe)  
accuracy

## Accuracy Kappa   
## 0.9981308 0.9976357

## applying model to test dataset

outcome<-predict(model\_rf,test[, -length(names(test))])  
outcome

## [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