Course Project

Chee Seong

July 16, 2020

## Course Project

### Loading of Library

# load all the necessary libraries  
library(caret)

## Warning: package 'caret' was built under R version 4.0.2

## Loading required package: lattice

## Loading required package: ggplot2

library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.2

library(RColorBrewer)  
library(rattle)

## Warning: package 'rattle' was built under R version 4.0.2

## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.0.2

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':  
##   
## importance

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

library(gbm)

## Warning: package 'gbm' was built under R version 4.0.2

## Loaded gbm 2.1.5

library(randomForest)

### Cleaning Data

First, we begin to import the data and start cleaning the data.

# Load the training and testing data.  
training\_data <- read.csv("./pml-training.csv" )  
testing\_data <- read.csv("./pml-testing.csv")  
dim(training\_data)

## [1] 19622 160

dim(testing\_data)

## [1] 20 160

# Cleaning the Data  
trainData <- training\_data[,colSums(is.na(training\_data))==0]  
testing\_data <- testing\_data[,colSums(is.na(testing\_data))==0]  
dim(trainData)

## [1] 19622 93

dim(testing\_data)

## [1] 20 60

# Removing the first seven variables as they have little impact  
# on the outcome classe  
trainData <- trainData[, -c(1:7)]  
testing\_data <- testing\_data[, -c(1:7)]  
dim(trainData)

## [1] 19622 86

dim(testing\_data)

## [1] 20 53

#Preparing the datasets for prediction  
set.seed(1234)  
inTrain <- createDataPartition(trainData$classe, p=0.7,list=FALSE)  
trainData <- trainData[inTrain,]  
testData<- trainData[-inTrain,]  
dim(trainData)

## [1] 13737 86

dim(testData)

## [1] 4123 86

# Removing near-zero-variance  
NZV <- nearZeroVar(trainData)  
trainData <- trainData[, -NZV]  
testData <- testData[,-NZV]  
dim(trainData)

## [1] 13737 53

dim(testData)

## [1] 4123 53

### Modeling of Random Forest

controlRF <- trainControl(method="cv", number=3, verboseIter=FALSE)  
model\_rf <- train(classe ~ ., data=trainData, method="rf",trControl=controlRF)  
model\_rf$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: 27  
##   
## OOB estimate of error rate: 0.71%  
## Confusion matrix:  
## A B C D E class.error  
## A 3902 3 0 0 1 0.001024066  
## B 18 2633 7 0 0 0.009405568  
## C 0 16 2371 9 0 0.010434057  
## D 0 1 29 2222 0 0.013321492  
## E 0 2 5 7 2511 0.005544554

predict\_rf <- predict(model\_rf, newdata=testData)

Using the confusion matrix, we see that it has high accuracy in predicting the testing set from the training data. We proceed to test for the validation testing data set.

### Final Modeling of the Test Variables

predict\_rf\_real <- predict(model\_rf,newdata=testing\_data)  
predict\_rf\_real

## [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