machine learning

ng

April 15, 2019

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. In this project, my goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.

The training data is found here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv The test data is found here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv The data is source here: http://groupware.les.inf.puc-rio.br/har.

The goal of my project is to predict the manner in which participants did the exercise. This is the "classe" variable in the training set. The report describes how I built your model, how I used cross validation, what the expected out of sample error is, and why I made the choices I did

Executive Summary

Given the computing power of random forests - a model is developed through random forests from 60% of cases in the training dataset. The model is then applied on 40% of cases in the training dataset (Test / validation). The expected out of sample error is very small. As it turns out - the confusion matrix yields 99% accuracy. Subsequently, the model is applied on the testing dataset (20 cases) where I achieved 100 correct predictions

Loading packages

Loading packages required for this machine learning assignment

```
library(caret)

## Warning: package 'caret' was built under R version 3.5.3

## Loading required package: lattice
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.3

library(rpart)
library(RColorBrewer)
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.5.3

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

Data Download

Download files including a training dataset and a testing dataset which will be used to test the accuracy of predictions from the model

```
TrainLINK <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
TestLINK  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
TRAINING <- read.csv(url(TrainLINK))
TESTING  <- read.csv(url(TestLINK))
```

Data Transformation

Filter in only useful elements. Remove all columns containing NA and remove independent variables that are not in the testing dataset. will also remove the first 7 independent variables

```
NAMEStesting <- names(TESTING[,colSums(is.na(TESTING)) == 0])[8:59]
TRAINING <- TRAINING[,c(NAMEStesting,"classe")]
TESTING <- TESTING[,c(NAMEStesting,"problem_id")]
dim(TRAINING);

## [1] 19622    53

dim(TESTING);

## [1] 20 53
```

Partition training dataset

we will split our data into a training data set (60% of cases from training dataset) and a testing data set (40% of cases from training dataset). The purpose of the testing dataset here is for validation to estimate the out of sample error of our predictor

```
set.seed(12345)
inTRAIN <- createDataPartition(TRAINING$classe, p=0.6, list=FALSE)
INtraining <- TRAINING[inTRAIN,]
INtesting <- TRAINING[-inTRAIN,]
dim(INtraining)

## [1] 11776    53

dim(INtesting)

## [1] 7846   53
```

Random forest

Create random forest model on training dataset. The out of sample error should be small. The error will be estimated using the 40% testing sample from the training dataset.

```
set.seed(12345)
RFmodel <- randomForest(classe ~ ., data = INtraining, ntree = 1000)
predictINtesting <- predict(RFmodel, INtesting, type = "class")
confusionMatrix(predictINtesting, INtesting$classe)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2230    9    0    0    0
##          B    2 1505    7    0    0
##          C    0    4 1361   16    2
##          D    0    0    0 1268    4
##          E    0    0    0    2 1436
##
## Overall Statistics
##
##                Accuracy : 0.9941
##                  95% CI : (0.9922, 0.9957)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9926
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9991   0.9914   0.9949   0.9860   0.9958
## Specificity            0.9984   0.9986   0.9966   0.9994   0.9997
## Pos Pred Value         0.9960   0.9941   0.9841   0.9969   0.9986
## Neg Pred Value         0.9996   0.9979   0.9989   0.9973   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2842   0.1918   0.1735   0.1616   0.1830
## Detection Prevalence   0.2854   0.1930   0.1763   0.1621   0.1833
## Balanced Accuracy      0.9988   0.9950   0.9957   0.9927   0.9978
```

Apply model on Test dataset

Run random forest model on test dataset (20 cases). We find the prediction to be very accurate. This is expected because the out of sample error is very small

```
set.seed(12345)
predictTESTING <- predict(RFmodel, TESTING, type = "class")
predictTESTING
```

```
##  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
```