Machine Learning write-up assignment

Loo Chee Wee

August 23, 2015

Installing and loading packages

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 3.1.3
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: splines
## Loading required package: Formula
```

```
## Warning: package 'Formula' was built under R version 3.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.1.2
```

```
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

```
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(ggplot2)
```

Summary

This is a write-up for the programming assignment under the Machine Learning module on Coursera. The goal is to create an algorithm that allows us to predict the classe (how well the subject is doing it) of an exercise based on measurements from a fitbit similar device.

Loading data.

We begin first by downloading the data, and cleasing the data set by changing empty spaces, and division by zero entries all into “NA”s.

```
Train <- read.csv("train.csv", header = TRUE, na.strings=c("NA","#DIV/0!",""))
Test <- read.csv("test.csv", header = TRUE, na.strings=c("NA","#DIV/0!",""))
```

Subsetting the dataset

We next remove columns where there are more than 10,000 NA entries:

```
noofna <- c(1:160)
for (i in 1:160 ) {
    noofna[i] = sum(is.na(Train[,i]))
}

y <- which(noofna > 10000, arr.ind = T)

Train2 <- Train[,-y]
```

The following columns were also removed: 1) X 2) user_name 3) raw_timestamp_part_1 4) raw_timestamp_part_2 5) cvtd_timestamp

Reasons being that we should be blind to identify of subject and time of execution in predicting the classe of movement.

```
Train4 <- Train2[,-c(1:5)]
```

So from an intial of 19622 observations of 160 variables, we are now left with 19622 observations of 55 variables.

Cross-Validation

We want to create a testing data set from the training set which will allow us to test our model. To this we will use 20% of the training set for this purpose.

```
set.seed(12345)
intrain <- createDataPartition(y=Train4$classe, p=0.8, list = FALSE)
Train5 <- Train4[intrain,]
CrossTrain <- Train4[-intrain,]
```

Model Construction

We then create a prediction model using the random forest algorithm

```
Fit2 <- train(classe~., data = Train5, method = "rf", proxy = TRUE)
```

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 3.1.2
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:Hmisc':
##
##     combine
```

We can see that the model has a high level of accuracy (approx 0.997).

Testing on cross-validation set

We then try this model on the Cross-Validation set:

```
predictionvalidation <- predict(Fit2, newdata = CrossTrain)
confusionMatrix(predictionvalidation, CrossTrain$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    2    0    0    0
##          B    0  757    4    0    0
##          C    0    0  680    1    0
##          D    0    0    0  642    2
##          E    0    0    0    0  719
## 
## Overall Statistics
## 
##                Accuracy : 0.9977
##                  95% CI : (0.9956, 0.999)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9971
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9942   0.9984   0.9972
## Specificity            0.9993   0.9987   0.9997   0.9994   1.0000
## Pos Pred Value         0.9982   0.9947   0.9985   0.9969   1.0000
## Neg Pred Value         1.0000   0.9994   0.9988   0.9997   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1930   0.1733   0.1637   0.1833
## Detection Prevalence   0.2850   0.1940   0.1736   0.1642   0.1833
## Balanced Accuracy      0.9996   0.9981   0.9969   0.9989   0.9986
```

And we see that the accuracy from the cross-validation results is approx 0.997 too. With that we can estimate the out of bag (OOB) error from this model is approx 0.003.

Preparing the test set

Finally we will clease our test set in a similar manner to the training set before we apply our model onto it:

```
noofnatest <- c(1:160)
for (i in 1:160 ) {
    noofnatest[i] = sum(is.na(Test[,i]))
}

z <- which(noofnatest > 10, arr.ind = T)

Test2 <- Test[,-z]

Test4 <- Test2[,-c(1:5)]
```

Now we apply our model onto the cleansed testing data set:

Prediction

```
predicttest <- predict(Fit2,  newdata = Test4)
```

Data Souce

All data from this project is sourced from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har), I thank them for the generousity for sharing such precious data.