Course Project - Machine Learning

Andrew Dieterich

2022-08-22

Disclaimer

Disclaimer Data for the source of this project comes from:

Link: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

Cited as: Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.

Notes about this project:

-the goal is to use data from accelerometers on 4 parts of 6 participants doing dumbbell curls -more info: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset)

-should predict "how" they did the exercise (classe variable at end of training set)

-need to report model, cross validation, out of sample error, and prediction model to predict test cases (20 total)

-link to a Github repo with your R markdown and compiled HTML file describing your analysis - use <2000 words, less than 5 figures

Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5 Apply your machine learning algorithm to the 20 test cases available in the test data above and submit your predictions in appropriate format to the Course Project Prediction Quiz for automated grading

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(lattice)
library(caret)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
setwd("/Users/andrewdieterich/RStudio/datasciencecoursera")

# importing training and test .csv files from Coursera downloads; removing DIV/0! values into 'NA' values:
training<-read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!"))
testing<-read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!"))
```

Examining the training data set

First I examined my training data set, in order to simplify and clean this dataset

```
# examining these 2 files, dimensions, classes of the columns, and report dimensions:
dim(testing); dim(training)
```

```
## [1]  20 160
```

```
## [1] 19622   160
```

```
# data processing

## removing the first 7 columns with irrelevant data
training <- training[,-c(1:7)]
training<-training[, apply(training, 2, function(x) !any(is.na(x)))]
sum(is.na(training))
```

```
## [1] 0
```

Information about the data-set:

ways (classe) A through E the 6 men did 10 dumbbell curl repetitions exactly according to the specification (Class A) throwing the elbows to the front (Class B) lifting the dumbbell only halfway (Class C) lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E)

Now I make my training and test partitions:

using caret package's createDataPartition function

```
inTrain <- createDataPartition(y=training$classe, p=0.7, list=FALSE)

TRAINING <- training[inTrain,]
TESTING <- training[-inTrain,]

## NOTE, that I could not get the full ~13,000 row dataset to run on my computer
## And had to reduce it to about 8,000 randomly sampled rows, using this line of code:
TRAINING2 <- sample_n(TRAINING, 5000, replace = TRUE)

# the model fit with a random forest machine learning algorithm
ModelFit <- train(classe~ ., data=TRAINING2, method = "rf",
                  trControl=trainControl(method="cv",number=2),prox=TRUE, verbose = FALSE)
```

Results of the Random Forest model

```
#reporting the results of the Random Forest model of the model fit:
ModelFit
```

```
## Random Forest
##
## 5000 samples
##   52 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 2500, 2500
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    2    0.9574    0.9462359
##   27    0.9592    0.9485239
##   52    0.9558    0.9442370
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
# seeing the result, then the final model:
ModelFit$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, proximity = TRUE,      verbose = FALSE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 2%
## Confusion matrix:
##      A   B   C   D   E class.error
## A 1350   5   3   1   0 0.006622517
## B   20 916  15   3   0 0.039832285
## C    0  12 932   5   0 0.017913593
## D    0   1  20 783   3 0.029739777
## E    0   7   1   4 919 0.012889366
```

```
####

# Evaluation results with predict for test set (from the data partition):
predict <- predict(ModelFit, newdata = TESTING)

# confusion matrix:
CONFUSION <- confusionMatrix(factor(predict), factor(TESTING$classe))
CONFUSION
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1664   36    0    0    0
##          B    3 1078   20    2    7
##          C    1   22  997   29    3
##          D    5    2    9  926    8
##          E    1    1    0    7 1064
##
## Overall Statistics
##
##                Accuracy : 0.9735
##                  95% CI : (0.9691, 0.9774)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9665
##
##  Mcnemar's Test P-Value : 7.1e-08
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9940   0.9464   0.9717   0.9606   0.9834
## Specificity            0.9915   0.9933   0.9887   0.9951   0.9981
## Pos Pred Value         0.9788   0.9712   0.9477   0.9747   0.9916
## Neg Pred Value         0.9976   0.9872   0.9940   0.9923   0.9963
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2828   0.1832   0.1694   0.1573   0.1808
## Detection Prevalence   0.2889   0.1886   0.1788   0.1614   0.1823
## Balanced Accuracy      0.9927   0.9699   0.9802   0.9779   0.9907
```

```
# My accuracy is 97.5%, when testing this model on the test data partition
## this my out of sample error is 0.025%
## This is the validation step
```

Results

```
TEST_set_prediction <- predict(ModelFit, newdata=testing)
TEST_set_prediction
```

```
##  [1] B A B A A E D D A A B C B A E E A B B B
## Levels: A B C D E
```

And that is my test outcome for the 20 rows in the test set: B A B A A E D D A A B C B A E E A B B B

I have used my random forest model to make $classe predictions based on the pml-testing.csv file which has been untouched since this project started, and is used only now, to test the model ## the end