PML - Course Project - Predicting Exercise

Richard Palmer

April 6, 2018

Predicting if a weight lifting exercise was done correctly

This report describes how I built my model, used cross validation, what I think the expected out of sample error is, and why I made the choices I did.

Acknowledgments

Data for this model came from: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). Here is the source paper: 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.

Building my model

I first read in my data and looked at it to determine if there were any variables that I could readily discard. As part of the data read-in process I split the data into a training set (70%) and testing test (30%) for cross validation.

```
## Loading required package: lattice
```

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

```
##
## 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
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

After examining the data I noted that there were several date/time and identifier (eg name of participant) variables that I would exclude. After further testing, I also found that there were variables that had mostly NA values. I excluded these variables from my model as well.

Which method should I use?

I used 4 different methods to see which one would predict with the highest accuracy. Here are the accuracy rates for each:

Decision Tree: 69.4% Random Forest: 95.9% *SVM: 93.5%* Linear: 74.4%

Based on this, I chose to use the Random Forest method for my final model.

Here are the summary results running the RF model on my training data.

```
##
## Call:
##  randomForest(formula = classe ~ ., data = dataset[, c(input,      target)], ntree = 500, mtr
y = 10, sampsize = c(1000), importance = TRUE,      replace = FALSE, na.action = randomForest::n
a.roughfix)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 4.12%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 5525   30   14    6    5 0.009856631
## B  185 3515   94    3    0 0.074269160
## C    0  138 3265   19    0 0.045879603
## D   12    2  189 3002   11 0.066542289
## E    3   15   44   38 3507 0.027723870
```

Running my model on the test set for cross validation

I then ran the model on my test set and found the following results:

```
##       Predicted
## Actual    A    B   C   D    E Error
##      A 1674    4   2   0    2   0.5
##      B   33 1080  19   0    0   4.6
##      C    0   27 976   5    0   3.2
##      D    3    0  49 904    3   5.7
##      E    0    2   5   7 1092   1.3
```

```
##       Predicted
## Actual    A    B    C    D    E Error
##      A 28.4  0.1  0.0  0.0  0.0   0.5
##      B  0.6 18.3  0.3  0.0  0.0   4.6
##      C  0.0  0.5 16.6  0.1  0.0   3.2
##      D  0.1  0.0  0.8 15.4  0.1   5.7
##      E  0.0  0.0  0.1  0.1 18.5   1.3
```

```
## 2.8
```

Out of sample error rate

The number reported above is the overall error percentage. This shows that my accuracy actually increased. This would mean that my out of sample error rate is 97.2%

Why I made the choices I did

In selecting variables I tried to use common sense in selecting variables. I would have used the average/std deviaion/max/min/var variables but there were so many missing values I decided to remove these variables.

I selected my model based on the accuracy of it.