Course Project

JJD

12/30/2021

Overview

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. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here:

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

(see the section on the Weight Lifting Exercise Dataset).

Data Processing

First the data is loaded before it is partitioned. 70% of the data will go into training while the remaining will be for the validation portion.

```
## [1] 13737   160
```

```
## [1] 5885  160
```

The training dataset has 13,737 records and 160 columns. While the training test set has 5,885 records and 160 columns.

A lot of the 160 columns has NA so they are removed from the dataset along with the first five columns since the purpose of the assignment is to see how well each excercise is done from the different participants and accelerometer placements.

```
## [1] 13737    54
```

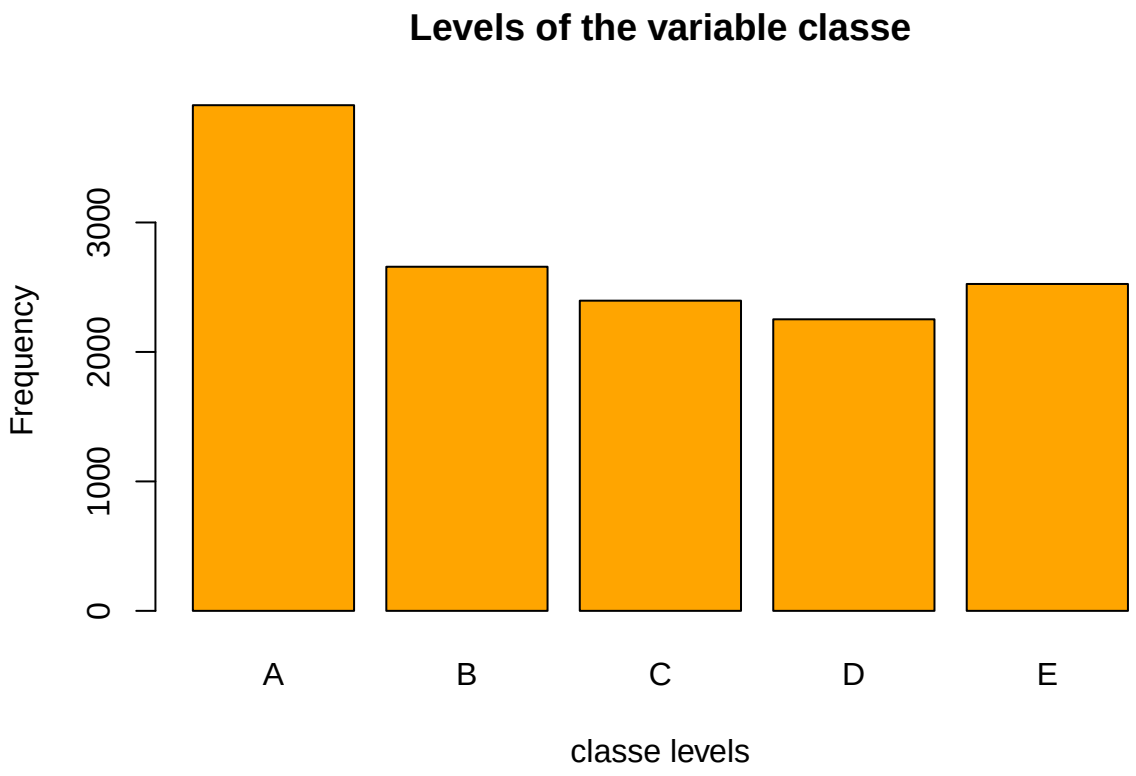
```
## [1] 5885   54
```

Now that the unnecessary columns have been removed from the dataset there are 54 columns left.

Exploratory analysis

The variable `classe` contains 5 levels. The plot of the outcome variable shows the frequency of each levels in the subTraining data.

```
plot(training$classe, col="orange", main="Levels of the variable classe", xlab="classe levels", ylab="F
```



The plot above shows that Level A is the most frequent classe. D appears to be the least frequent one.

The expected out-of-sample error will correspond to the quantity: 1-accuracy in the cross-validation data.

Accuracy is the proportion of correct classified observation over the total sample in the traintest data set.

Expected accuracy is the expected accuracy in the out-of-sample data set (i.e. original testing data set).

Thus, the expected value of the out-of-sample error will correspond to the expected number of missclassified observations/total observations in the Test data set, which is the quantity: 1-accuracy found from the cross-validation data set.

Prediction Model Selection

Three models will be used to determine which one is the most accurate in predicting.

1) Boosting
2) Decision Tree
3) Random Forest

Boosting

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 53 predictors of which 43 had non-zero influence.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1667    6    0    0    0
##          B    7 1111   11    7    6
##          C    0   21 1015   15    3
##          D    0    0    0  941   20
##          E    0    1    0    1 1053
##
## Overall Statistics
##
##                Accuracy : 0.9833
##                  95% CI : (0.9797, 0.9865)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9789
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9958   0.9754   0.9893   0.9761   0.9732
## Specificity            0.9986   0.9935   0.9920   0.9959   0.9996
## Pos Pred Value         0.9964   0.9729   0.9630   0.9792   0.9981
## Neg Pred Value         0.9983   0.9941   0.9977   0.9953   0.9940
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2833   0.1888   0.1725   0.1599   0.1789
## Detection Prevalence   0.2843   0.1941   0.1791   0.1633   0.1793
## Balanced Accuracy      0.9972   0.9844   0.9906   0.9860   0.9864
```

Descision Tree

```
## n= 13737
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 13737 9831 A (0.28 0.19 0.17 0.16 0.18)
##    2) roll_belt< 129.5 12486 8627 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -33.35 1116   10 A (0.99 0.009 0 0 0) *
##      5) pitch_forearm>=-33.35 11370 8617 A (0.24 0.23 0.21 0.2 0.12)
##       10) magnet_dumbbell_y< 439.5 9620 6918 A (0.28 0.18 0.24 0.19 0.1)
##         20) roll_forearm< 123.5 5994 3573 A (0.4 0.19 0.19 0.17 0.054) *
##         21) roll_forearm>=123.5 3626 2416 C (0.077 0.17 0.33 0.23 0.18) *
##       11) magnet_dumbbell_y>=439.5 1750  843 B (0.029 0.52 0.041 0.22 0.19) *
```

```
##    3) roll_belt>=129.5 1251   47 E (0.038 0 0 0 0.96) *
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1498  467  473  427  166
##          B   30  379   36  177  137
##          C  119  293  517  360  284
##          D    0    0    0    0    0
##          E   27    0    0    0  495
##
## Overall Statistics
##
##                Accuracy : 0.4909
##                  95% CI : (0.4781, 0.5038)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3351
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8949   0.3327  0.50390   0.0000  0.45749
## Specificity            0.6360   0.9199  0.78267   1.0000  0.99438
## Pos Pred Value         0.4942   0.4993  0.32867      NaN  0.94828
## Neg Pred Value         0.9383   0.8517  0.88196   0.8362  0.89055
## Prevalence             0.2845   0.1935  0.17434   0.1638  0.18386
## Detection Rate         0.2545   0.0644  0.08785   0.0000  0.08411
## Detection Prevalence   0.5150   0.1290  0.26729   0.0000  0.08870
## Balanced Accuracy      0.7654   0.6263  0.64328   0.5000  0.72593
```

Random Forest

In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the execution. So, we proceed with the training the model (Random Forest) with the training data set.

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

```
## randomForest 4.6-14
```

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

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

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

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

```
## n= 13737
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 13737 9831 A (0.28 0.19 0.17 0.16 0.18)
##    2) roll_belt< 129.5 12486 8627 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -33.35 1116   10 A (0.99 0.009 0 0 0) *
##      5) pitch_forearm>=-33.35 11370 8617 A (0.24 0.23 0.21 0.2 0.12)
##       10) magnet_dumbbell_y< 439.5 9620 6918 A (0.28 0.18 0.24 0.19 0.1)
##         20) roll_forearm< 123.5 5994 3573 A (0.4 0.19 0.19 0.17 0.054) *
##         21) roll_forearm>=123.5 3626 2416 C (0.077 0.17 0.33 0.23 0.18) *
##       11) magnet_dumbbell_y>=439.5 1750  843 B (0.029 0.52 0.041 0.22 0.19) *
##    3) roll_belt>=129.5 1251   47 E (0.038 0 0 0 0.96) *
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    3    0    0    0
##          B    0 1134    3    0    2
##          C    0    2 1023    2    0
##          D    0    0    0  962   10
##          E    0    0    0    0 1070
##
## Overall Statistics
##
##                Accuracy : 0.9963
##                  95% CI : (0.9943, 0.9977)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9953
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9956   0.9971   0.9979   0.9889
## Specificity            0.9993   0.9989   0.9992   0.9980   1.0000
## Pos Pred Value         0.9982   0.9956   0.9961   0.9897   1.0000
## Neg Pred Value         1.0000   0.9989   0.9994   0.9996   0.9975
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1927   0.1738   0.1635   0.1818
## Detection Prevalence   0.2850   0.1935   0.1745   0.1652   0.1818
## Balanced Accuracy      0.9996   0.9973   0.9981   0.9979   0.9945
```

Conclusion

Result

The confusion matrices show, that the Boosted Model and Random Forest algorithm performes better than Decision Trees and the Boosted model. The accuracy for the Random Forest model was 0.996 (95% CI: (0.994, 0.997)) while the Boosted model was 0.983 (95% CI: (0.979, 0.987)) compared to the Decision Tree 0.491 (95% CI: (0.478, 0.504)) for Decision Tree model. The Random Forest model is slightly higher in accuracy and in the confidence interval so that was chosen over the Boosted model.

Expected out-of-sample error

The expected out-of-sample error is estimated at 0.005, or 0.5%. The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set. Our Test data set comprises 20 cases. With an accuracy above 99% on our cross-validation data, we can expect that very few, or none, of the test samples will be missclassified.