Practical Machine Learning - Human Activity Recognition Report

Jean Pelkey

September 10, 2017

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

Synopsis

A Random Forest Model accurately predicted the 20 quiz test cases with 100% out of sample accuracy. The model was 99.89% accurate for the validation data set. The author would like to recognize the team of Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. for their article and data set: Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements

[Read more:] (http://groupware.les.inf.puc-rio.br/har#ixzz4sNxpe8oI (http://groupware.les.inf.puc-rio.br/har#ixzz4sNxpe8oI))

Details of Analysis

Data was extracted consisted of 19622 observations and 160 columns in the training data set. The test data set included 20 observations representing the 20 cases to be predicted in the final quiz. The data was cleaned to remove columns that consisted of greater than half N/A or missing data as well as the first column which was the observation number. After cleaning, 59 columns remained in the training data set. The test data set was then coerced to have the same data types as the corresponding columns in training data se to prevent errors when trying to prediict classe with the final selected model.

The training data set was subsetted into a training and validation data set using the CreateDataPartition function in R. The partition randomly selected 60% of the training data set (data frame is sub.train) for training and the remaining 40% for model validation (data frame valid). The training dta set contained 11776 observations and the validation data set contained 7846 observations.

```
#* create training and validation partitions from the training .CSV data set using 60%
/40% training & validation, respectively.
inTrain<- createDataPartition(alltrain$classe, p = 0.60, list=FALSE)
sub.train<-alltrain[inTrain,]
valid<-alltrain[-inTrain,]
```

Analysis was completed using two methods: random forest and decision tree. The random forest performed the best on the training and validation data. First set of results published below show the results of the decision tree prediction on the validation data set. The decision tree approach looks to split the data to create “like” or homogeneous groups by minimizing the differences within the groups. The trees that result are easy to interpret but can lead to overfitting of the model and results are highly variable. Accuracy for this model is 56% with many misclassifications of the classe responses in the validation data set. Further work was done to find a better model.

```
require(caret)
fit1.rpart<-train(classe ~., data=sub.train, method="rpart")
pred.rpart<-predict(fit1.rpart,valid)
conf.rpart<-confusionMatrix(pred.rpart, valid$classe)
print(conf.rpart)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1681  379   43   71   22
##          B    4  233    1    0    0
##          C  363  350 1159  557  391
##          D  177  556  165  658  387
##          E    7    0    0    0  642
##
## Overall Statistics
##
##                Accuracy : 0.5574
##                  95% CI : (0.5463, 0.5684)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4439
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.7531  0.15349   0.8472  0.51166  0.44521
## Specificity            0.9083  0.99921   0.7436  0.80412  0.99891
## Pos Pred Value         0.7655  0.97899   0.4110  0.33865  0.98921
## Neg Pred Value         0.9025  0.83110   0.9584  0.89361  0.88884
## Prevalence             0.2845  0.19347   0.1744  0.16391  0.18379
## Detection Rate         0.2142  0.02970   0.1477  0.08386  0.08183
## Detection Prevalence   0.2799  0.03033   0.3594  0.24764  0.08272
## Balanced Accuracy      0.8307  0.57635   0.7954  0.65789  0.72206
```

A second model considered was the random forest. The random forest model is a resampling method within the set. Each of the resamples leads to a different regression tree. The results are averaged across all possible trees which leads to a higher level of accuracy in teh resulting prediction tree. For this predictio problem, the Random Foresst method gavev much better results for predicting classe response variable vs. the decision tree method attempted previously. Evaluating the output below, the prediction data set the model shows significant improvement in accuracy versus the prediction tree model with accuracy of 99.89% on out of sample error.

```
fit1.rf<-randomForest(classe~., sub.train)
pred.rf<-predict(fit1.rf, valid)
conf.rf<-confusionMatrix(pred.rf, valid$classe)
print(conf.rf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    1    0    0    0
##          B    0 1517    3    0    0
##          C    0    0 1365    4    0
##          D    0    0    0 1282    2
##          E    0    0    0    0 1440
##
## Overall Statistics
##
##                Accuracy : 0.9987
##                  95% CI : (0.9977, 0.9994)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9984
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9993   0.9978   0.9969   0.9986
## Specificity            0.9998   0.9995   0.9994   0.9997   1.0000
## Pos Pred Value         0.9996   0.9980   0.9971   0.9984   1.0000
## Neg Pred Value         1.0000   0.9998   0.9995   0.9994   0.9997
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1933   0.1740   0.1634   0.1835
## Detection Prevalence   0.2846   0.1937   0.1745   0.1637   0.1835
## Balanced Accuracy      0.9999   0.9994   0.9986   0.9983   0.9993
```

Given the accuracy of the random forest, the testing data set was used to predict the final 20 conditions. The data was submitted to the Coursera prediction quiz was 20/20 or 100% accurate. No further work was done given the out of sample accuracy observed with the random forest results.

```
pred.test<-predict(fit1.rf, testing)
final.answer<-data.frame(testing$source, pred.test)
print(final.answer)
```

```
##       testing.source pred.test
## 19623              1         B
## 19624              2         A
## 19625              3         B
## 19626              4         A
## 19627              5         A
## 19628              6         E
## 19629              7         D
## 19630              8         B
## 19631              9         A
## 19632             10         A
## 19633             11         B
## 19634             12         C
## 19635             13         B
## 19636             14         A
## 19637             15         E
## 19638             16         E
## 19639             17         A
## 19640             18         B
## 19641             19         B
## 19642             20         B
```

Summary of Results

The Random Forest model was the best prediction model of the two models attempted. The model was 99.89% accurate for the validation data set of 7846 observations and predicted 20/20 or 100% of the testing data set.