PMLProject

NJL

31/05/2021

Inroduction

The goal of this project is to predict the outcome based om the training set. This is the “classe” variable in the training set. Any of the other variables can be used to predict with. Create a report describing how the model is built, how cross validation was used, what is the expected out of sample error, and whywere the choices made. The chosen model will be used to predict 20 different test cases.

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://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Loading data

The training data is stored in the dataframe Trdat and the testing data in Tstdat

Analyse and preProcess training data

There are 160 variables and 19622 samples in the training data set and 20 samples in the testing data set.

The first study is to decrease the number of variables, derived statistical variables, variables with large numbers of NA are removed.

Descriptive and time variables are analysed and found to be not required for this prediction exercise.

Partitioning of Training Data

The training data is partitioned in the ratio of 70:30, 70 for training and 30 for out of sample error testing.

First trial - fitting a model using Decision Tree algorithm

The method used is Cross Validation with 3 K-folds. Using confusion matrix to check for accuracy, thereby the out of sample error.

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##          A 1519  473  484  451  156
##          B   28  337   45    9  125
##          C   83  117  423  131  131
##          D   40  212   74  373  181
##          E    4    0    0    0  489
##
## Overall Statistics
##
##               Accuracy : 0.5337
##                 95% CI : (0.5209, 0.5465)
##    No Information Rate : 0.2845
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.3921
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                     Class: A Class: B Class: C Class: D Class: E
## Sensitivity           0.9074  0.29587  0.41228  0.38693  0.45194
## Specificity           0.6286  0.95638  0.90492  0.89697  0.99917
## Pos Pred Value        0.4927  0.61949  0.47797  0.42386  0.99189
## Neg Pred Value        0.9447  0.84984  0.87940  0.88192  0.89002
## Prevalence            0.2845  0.19354  0.17434  0.16381  0.18386
## Detection Rate        0.2581  0.05726  0.07188  0.06338  0.08309
## Detection Prevalence  0.5239  0.09244  0.15038  0.14953  0.08377
## Balanced Accuracy     0.7680  0.62613  0.65860  0.64195  0.72555
```

Second trial - fitting a model using RandomForest algorithm

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##          A 1673    4    0    0    0
##          B    1 1132   10    0    0
##          C    0    3 1015    6    0
##          D    0    0    1  957    0
##          E    0    0    0    1 1082
##
## Overall Statistics
##
##               Accuracy : 0.9956
##                 95% CI : (0.9935, 0.9971)
##    No Information Rate : 0.2845
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.9944
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                     Class: A Class: B Class: C Class: D Class: E
## Sensitivity           0.9994   0.9939   0.9893   0.9927   1.0000
## Specificity           0.9991   0.9977   0.9981   0.9998   0.9998
## Pos Pred Value        0.9976   0.9904   0.9912   0.9990   0.9991
## Neg Pred Value        0.9998   0.9985   0.9977   0.9986   1.0000
## Prevalence            0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate        0.2843   0.1924   0.1725   0.1626   0.1839
## Detection Prevalence  0.2850   0.1942   0.1740   0.1628   0.1840
## Balanced Accuracy     0.9992   0.9958   0.9937   0.9963   0.9999
```

Third trial - fitting a model using Gradient Boosted Trees algorithm

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##         A 1672    9    0    0    0
##         B    1 1123   14    0    1
##         C    1    7 1009    7    4
##         D    0    0    3  955    1
##         E    0    0    0    2 1076
##
## Overall Statistics
##
##               Accuracy : 0.9915
##                 95% CI : (0.9888, 0.9937)
##    No Information Rate : 0.2845
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.9893
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                     Class: A Class: B Class: C Class: D Class: E
## Sensitivity           0.9988   0.9860   0.9834   0.9907   0.9945
## Specificity           0.9979   0.9966   0.9961   0.9992   0.9996
## Pos Pred Value        0.9946   0.9860   0.9815   0.9958   0.9981
## Neg Pred Value        0.9995   0.9966   0.9965   0.9982   0.9988
## Prevalence            0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate        0.2841   0.1908   0.1715   0.1623   0.1828
## Detection Prevalence  0.2856   0.1935   0.1747   0.1630   0.1832
## Balanced Accuracy     0.9983   0.9913   0.9898   0.9949   0.9970
```

Choice of model to predict the test data

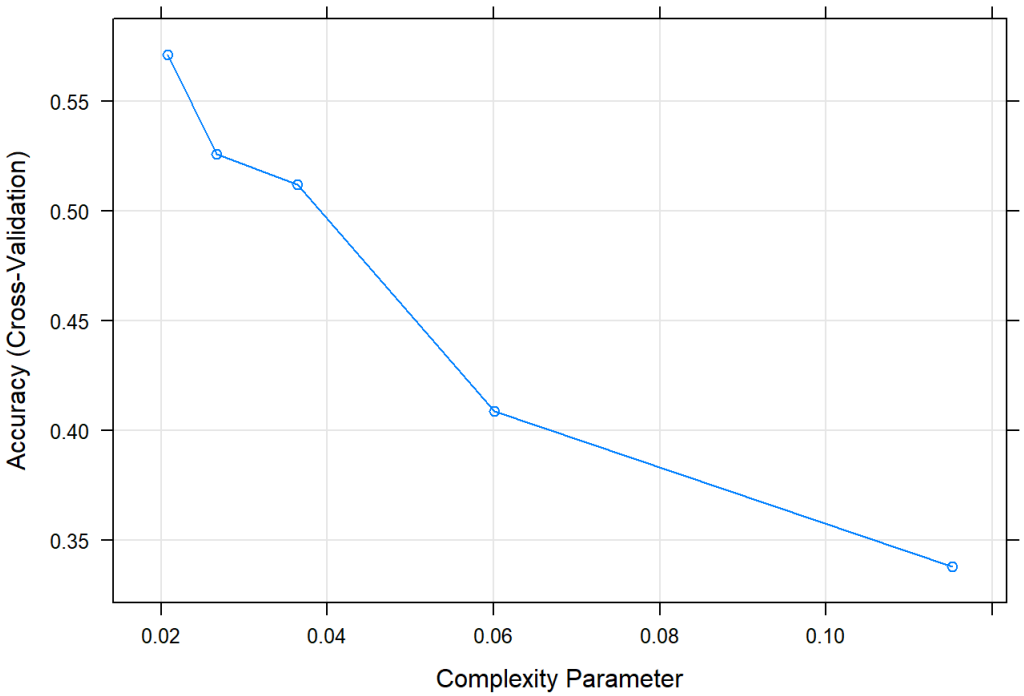
From the Confusion Matrix output, the accuracy of the RandomForest is the best, the prediction is done using the RF model “mod_rf”

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

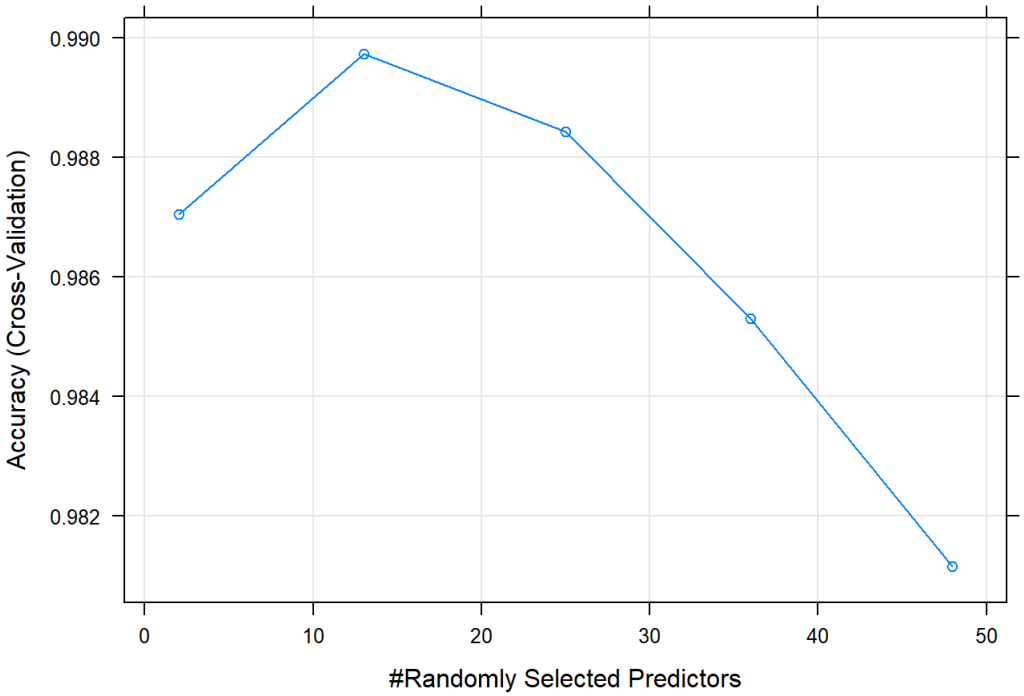
Appendix

Plots of models

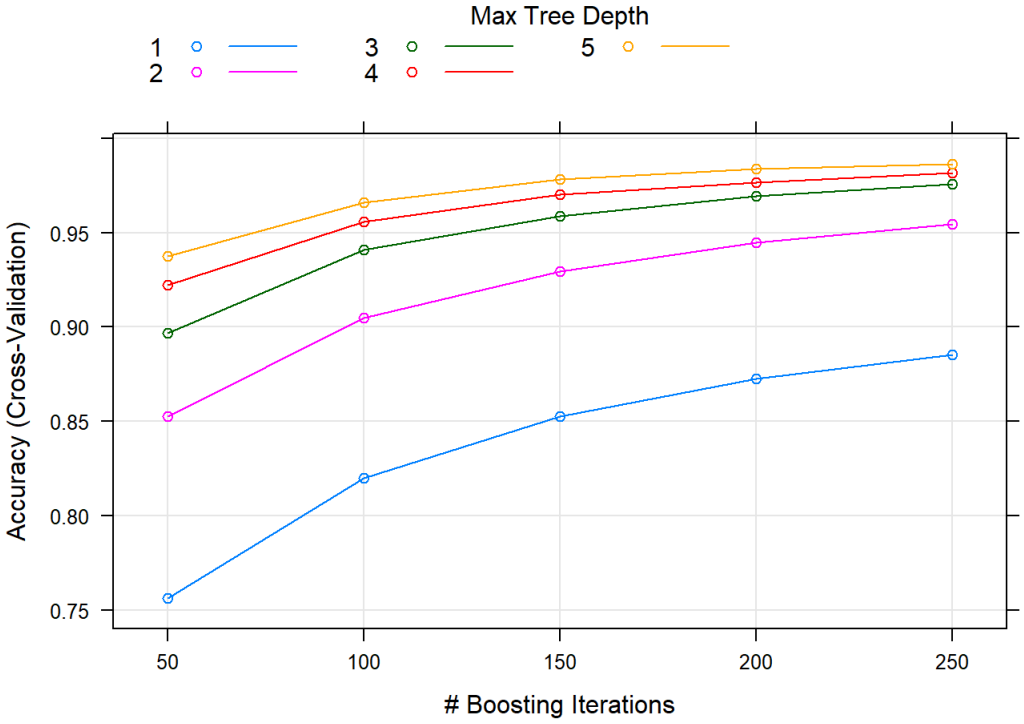
Decision Trees



RandomForest



Gradient Boosted Trees



Further analysis by applying Technical knowledge on the data

The processed data set contains raw sensor data and derived data.

The raw data variables have names ending with "_x“,”_y“,”_z".

The derived data are the Roll, Pitch and Yaw variables.

Since the derived data are less in number, we can use only the derived data to fit the model and use it for prediction.

```
x1 <- grep("_x$|_y$|_z$", names(train))
train <- train[, -x1]
dim(train)
```

```
## [1] 13737    13
```

```
mod_rf <- train(classe~., data=train, method="rf", trControl = control, tuneLength = 5)
pred <- predict(mod_rf, dat2)
print(pred)
```

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

The predictions are the same though the number of variables is reduced.