Human Activity Recognition

Executive Summary

The purpose of the report is to describe the methodology used to produce a model that was then applied to the training results of group of 20 individuals that resulted in 100% success of predicting the training class used by each participant. Applied machine learning techniques In R programming and gbm (generalized boosted regression model) algorithm were used.

Model accuracy and estimated error was calculated in two ways: a 20% validation data split and a k-fold cross validation where k = 10. K-fold cross validation was .9621 while the validation set was predicted at a rate of .9609.

Description of Experiment

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. This purpose of this experiment is to use data from accelerometers on the belt, forearm, arm, and dumbbells of 6 participants and determine if they are performing perform barbell lifts correctly or incorrectly.

Description of Data

Data for this experiment is sourced from the Human Activity Recognition (HAR) project at <http://groupware.les.inf.puc-rio.br/har>. Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: 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).

Data gathering

**library**(caret)

**library**(splines)

**library**(parallel)

**library**(plyr)

**require**(RCurl)

*# Fetch data from Human Activity Recognition project*

trainingdata <- getURL("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")

pml\_training  <- read.csv(text = trainingdata, header=TRUE, sep=",", na.strings=c("NA",""))

*# Remove the first column of each row*

pml\_training <- pml\_training[,-1] # Remove the first column that represents a ID Row

*# create training and validating data sets*

alltrain = createDataPartition(pml\_training$classe, p=0.80, list=FALSE)

training = pml\_training[alltrain,]

crossvalidating = pml\_training[-allTrain,]

testcsv <- getURL("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")

dftestdata <- as.data.frame(read.csv(textConnection(testcsv)))

Data preparation:

The data from the HAR project consists of a training set and a test set. The training set is made up of 160 variables and 15,699 observations while the testing set consists of 160 variables and 20 observations or prediction instances.

Review of the training and test data sets reveals multiple columns with NA’s. These columns have been removed since they would offer no value to the model and prediction.

Remove non-relevant data:

*# remove columns without data*

Keep <- c((colSums(!is.na(training[,-ncol(training)])) >= 0.6\*nrow(training)))

training <- training[,Keep]

crossvalidating <- crossvalidating[,Keep]

Boost GBM Model

Using training data, a model is created using GBM (generalized boosted regression model) package. The results are summarized below.

*# Create model*

control <- trainControl(method="cv", number=10)

model\_gbm <- train(classe~. , data = training , method="gbm" , trControl=control ,  verbose = FALSE )

model\_gbm$finalModel

## A gradient boosted model with multinomial loss function.

## 150 iterations were performed.

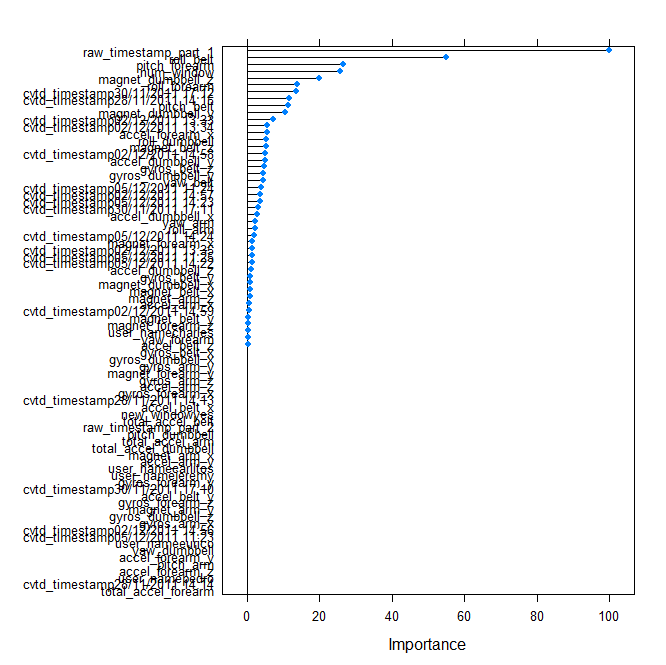
## There were 52 predictors of which 44 had non-zero influence.

Visualization of Model Variables

The model variables and their relative importance are indicated below. Based on the importance analysis, the model performance could have been further enhanced by deselecting the variables with no explanatory power. Note also that out of the 53 input variables, ~10% have an importance >20 suggesting significant prospect for further enhancement.

*# Visualization*

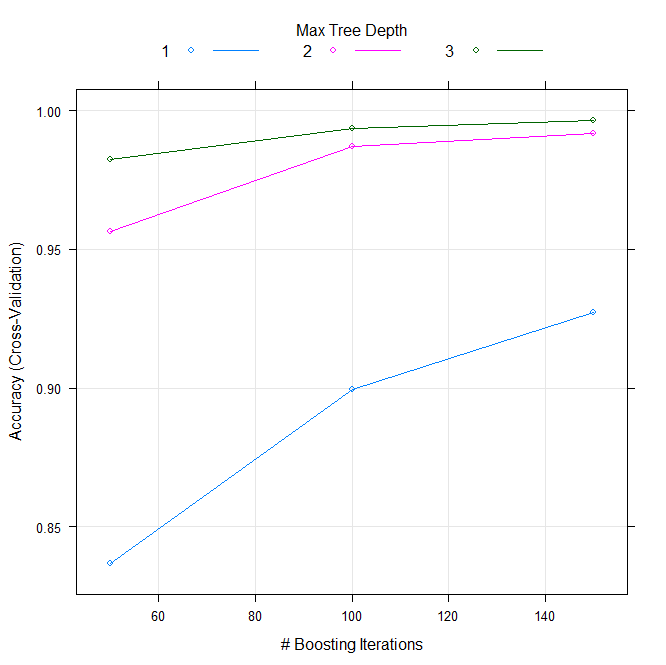
plot(varImp(model\_gbm))



Cross Validation and Predicted Error

The BOOST gbm algorithm was conducted with a 10-fold cross validation. The data and plot below shows an achieved accuracy on the 3rd iteration of 96.09% rising from an initial accuracy of 75.15% in teh first iteration.

plot(model\_gbm)



model\_gbm

Stochastic Gradient Boosting

15699 samples

58 predictors

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 14129, 14129, 14130, 14129, 14130, 14128, ...

Resampling results across tuning parameters:

interaction.depth n.trees Accuracy Kappa Accuracy SD Kappa SD

1 50 0.8411362 0.7985087 0.011015858 0.014072039

1 100 0.8992299 0.8723744 0.008630729 0.010949349

1 150 0.9282761 0.9091366 0.007617383 0.009675722

2 50 0.9579594 0.9467798 0.008094323 0.010277777

2 100 0.9871330 0.9837246 0.003969270 0.005021412

2 150 0.9920378 0.9899292 0.002868143 0.003627873

3 50 0.9836299 0.9792924 0.003315789 0.004195634

3 100 0.9933753 0.9916210 0.001856036 0.002347686

3 150 0.9969426 0.9961328 0.001401931 0.001773376

Tuning parameter 'shrinkage' was held constant at a value of 0.1

Tuning parameter 'n.minobsinnode' was held constant at a value of 10

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were n.trees = 150, interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10..

Validation Accuracy

Model accuracy and error were further estimated by randomly splitting 20% of the training data prior to model construction. The below confusion matrix shows the estimated accuracy to be 99.64%.

cross\_validation\_results\_boost <- predict(model\_gbm , newdata = crossvalidating)

confusionMatrix(df.val$classe , cross\_validation\_results\_boost)

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1116 0 0 0 0

B 0 759 0 0 0

C 0 1 677 6 0

D 0 0 5 638 0

E 0 0 0 2 719

Overall Statistics

Accuracy : 0.9964

95% CI : (0.994, 0.998)

No Information Rate : 0.2845

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9955

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 0.9987 0.9927 0.9876 1.0000

Specificity 1.0000 1.0000 0.9978 0.9985 0.9994

Pos Pred Value 1.0000 1.0000 0.9898 0.9922 0.9972

Neg Pred Value 1.0000 0.9997 0.9985 0.9976 1.0000

Prevalence 0.2845 0.1937 0.1738 0.1647 0.1833

Detection Rate 0.2845 0.1935 0.1726 0.1626 0.1833

Detection Prevalence 0.2845 0.1935 0.1744 0.1639 0.1838

Balanced Accuracy 1.0000 0.9993 0.9953 0.9930 0.9997

Model results

The resulting data produced from the boost modeling of the HAR data presented below in order from 1 to 20. On submission to the Coursera Project Page, the results were shown to be 100% correct.

predict.test.gbm <- predict(model\_gbm , newdata = dftestdata )

*# Produce result files*

n = length(predict.test.gbm)

**for**(i **in** 1:n){

    filename = paste0("problem\_id\_",i,".txt")

    write.table(predict.test.gbm[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)

    }

The resulting data produced from the boost modeling of the HAR data presented below in order from 1 to 20. On submission to the Coursera Project Page, the results were shown to be 100% correct.

predict.test.gbm

[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