Practical Machine Learning - Course Project

Petar Luketic

Saturday, October 24, 2015

Executive Summary

In this project, we use data from accelerometers on the belt, forearm, arm, and dumbell to predict the manner in which six participants did the barbell lifts. They were asked to perform exercise correctly and incorrectly in 5 different ways.

After extensive training data analysis and modeling trials, random forest algorythm proved to be the most accurate algorithm for predicting the way in which an exercise has been conducted. In sample and out of sample (20 cases) accuracy is 100%, consequently expected out of sample error is near zero.

Loading the Data

```
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

train <- read.csv(url(trainUrl),head=TRUE,sep=",", na.strings=c("NA","#DIV/0!",""))
test <- read.csv(url(testUrl),head=TRUE,sep=",", na.strings=c("NA","#DIV/0!",""))

library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

Data cleansing

Identify variables that are predominantly empty (NA)

```
naprops <- colSums(is.na(train))/nrow(train)
mostlyNAs <- names(naprops[naprops > 0.75])
mostlyNACols <- which(naprops > 0.75)
train2 <- train[,-mostlyNACols]
```

Exclude variables whoose variance is near zero

```
nzv <- nearZeroVar(train2)
train3 <- train2[,-nzv]
```

Create partitions for training and testing purpose.

```
inTrain = createDataPartition(train3$classe, p = 0.75)[[1]]
train_in_sample = train3[inTrain, ]
test_in_sample = train3[-inTrain, ]
```

Exclude row number, user_name and cvtd_timestamp columnS

```
train_in_sample <- train_in_sample[,-grep("X|user_name|cvtd_timestamp",names(train_in_sample))]
test_in_sample <- test_in_sample[,-grep("X|user_name|cvtd_timestamp",names(test_in_sample))]
```

Final structure of training data set for predictive model creation

```
modelVars <- names(train_in_sample)
modelVars1 <- modelVars[-grep("classe",modelVars)]

cleanedTrainData <- train[,modelVars]
```

Random forest model creation

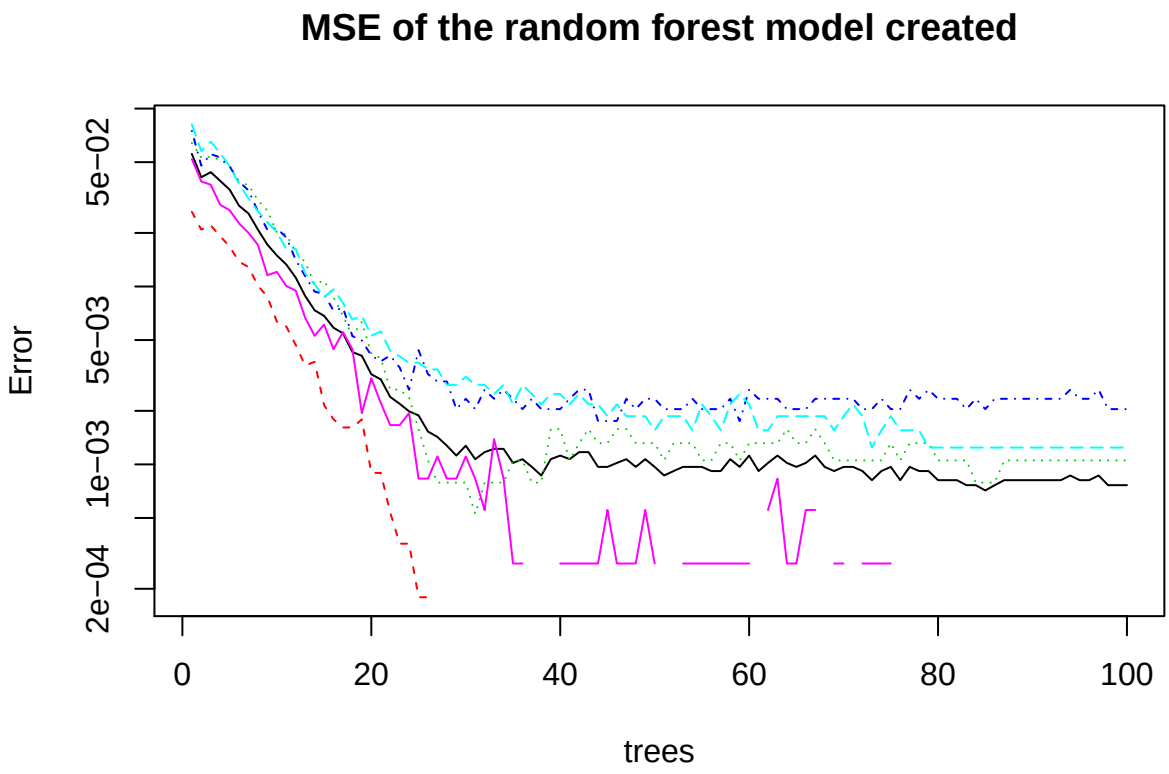
First, we build random forest model with up to 100 trees to grow.

```
rfFit <- randomForest(classe ~., data=cleanedTrainData, type="class", ntree = 100)
```

By ploting the model, one can observe that above 80 trees MSE becomes insignificant so we confirm provided ntree argument.

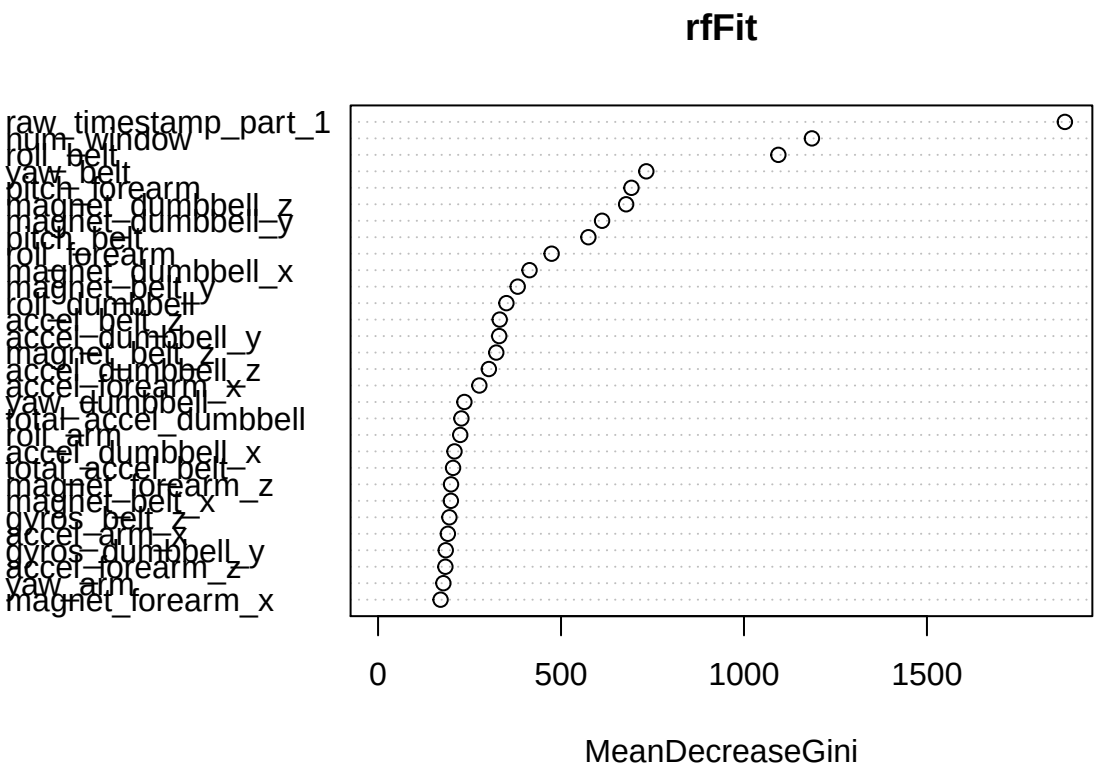
```
plot(rfFit, log="y",main="MSE of the random forest model created")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log = log): 106 y values <= 0
## omitted from logarithmic plot
```



Variable importance plot highlights relative predicor's importances.

```
varImpPlot(rfFit)
```



Model performance on the in sample training data

Get the values predicted by the model

```
RFpredTrain <- predict(rfFit,newdata=train_in_sample)
```

Confusion matrix of the model performance on the in sample training data demonstrates perfect model performance.

```
confusionMatrix(RFpredTrain,train_in_sample$classe)$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 4185    0    0    0    0
##          B    0 2848    0    0    0
##          C    0    0 2567    0    0
##          D    0    0    0 2412    0
##          E    0    0    0    0 2706
```

Model performance on the in sample test data

Exclude classe column from the test data in sample, and apply random forest model rfFit onto the test in sample data

```
classe_col <- grep("classe",names(test_in_sample))
RFpred_in_sample <- predict(rfFit, newdata = test_in_sample[,-classe_col], type="class")
```

Use a confusion matrix to get the in sample test error

```
confusionMatrix(RFpred_in_sample,test_in_sample$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    0  949    0    0    0
##          C    0    0  855    0    0
##          D    0    0    0  804    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9992, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Model predictions on out of sample data and expected error

```
RFpred_out_sample <- predict(rfFit, newdata = test, type="class")
RFpred_out_sample
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

Considering perfect match on the training and on the testing data indicates that we have a perfect model which would perform in the same manner on out of sample cases. Nevertheless, some error still should be expected on the out of sample data.

References

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.

Read more: http://groupware.les.inf.puc-rio.br/har#wle_paper_section#ixzz3pc5SknLR