Practical Machine Learning Writeup!

Pradeep Peddineni

February 1, 2017

Goal:

We are Given with two files which were collected Using devices such as Jawbone Up, Nike FuelBand, and Fitbit,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.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.

Problem:

we are asked to develop a machine learning paradigm to predict the type of exercise performed based an a variety of measurements. The first file is the training file, used to develop our algorithm, and the second is used to make our final predictions.

Data:

The training data is: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The testing data is: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

Data Cleansing:

Once downloaded to our working directory, we read in the file to perform some basic exploratory data analysis. We notice there are many blanks and NA values (I do not show here in the interest of space), so I shall re-read the file so that all non-valid entries (blanks, DIV/0, NA), are read in as NA in R. I continue to examine and remove columns which contain NA's, as well as remove columns which I do not believe have any outcome on the class.

Read Data

```r
a=read.csv('pml-training.csv',na.strings=c('','NA'))
b=a[,!apply(a,2,function(x) any(is.na(x)) )]
c=b[,-c(1:7)]
```

This leaves us with 19622 observations and 53 predictors, also inorder to continue download and call the following packages.

```r
##install.packages('randomForest')
library('randomForest')
##install.packages('caret')
library('caret')
##install.packages('e1071')
library('e1071')
```

For cross validation sake i am splitting data in to two subgroups 60, 40.

```
subGrps=createDataPartition(y=c$classe, p=0.6, list=FALSE)
subTraining=c[subGrps,]
subTesting=c[-subGrps, ]
dim(subTraining);dim(subTesting)
```

```
## [1] 11776    53
```

```
## [1] 7846   53
```

We see there are 11776 in the subTraining group, and 7846 in the subTesting group.

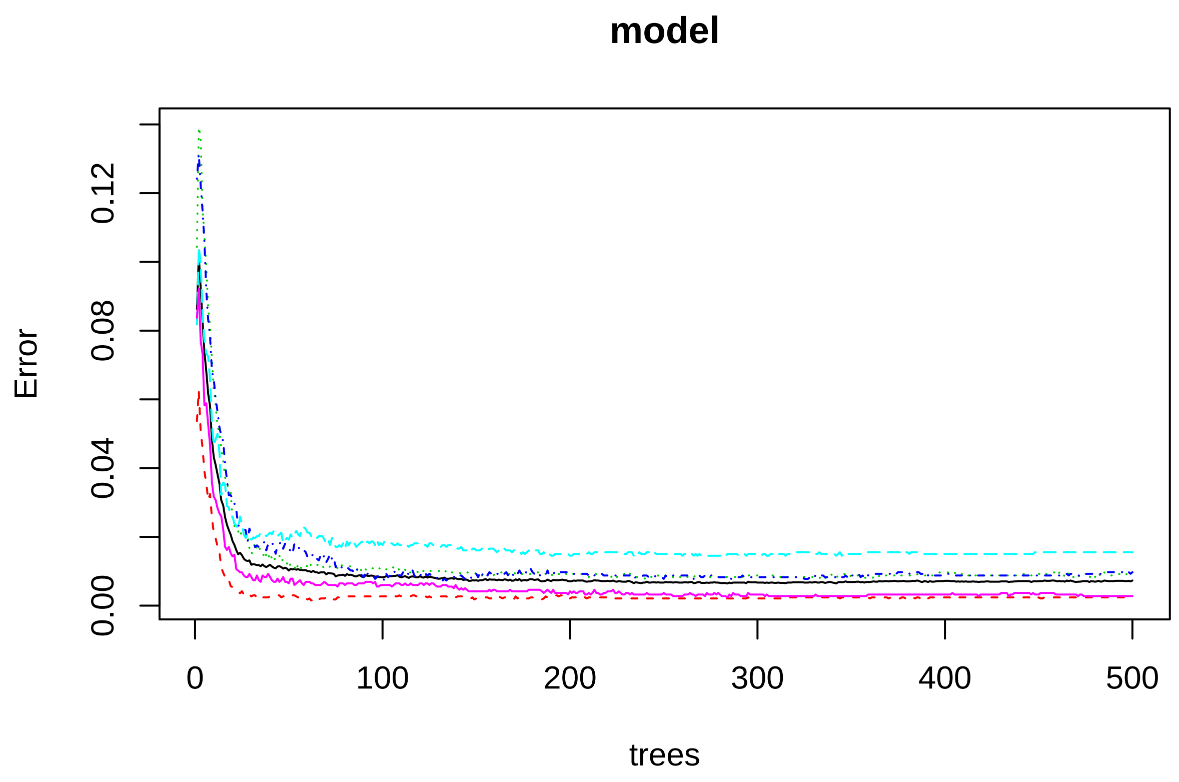
Prediction with Random Forests

I now continue to make a predictive model based on the random forest paradigm, as it is one of the best performing, using the subTraining group. Once the model is made, we predict the outcome of the other group, subTesting, and examine the confusion matrix to see how well the predictive model performed

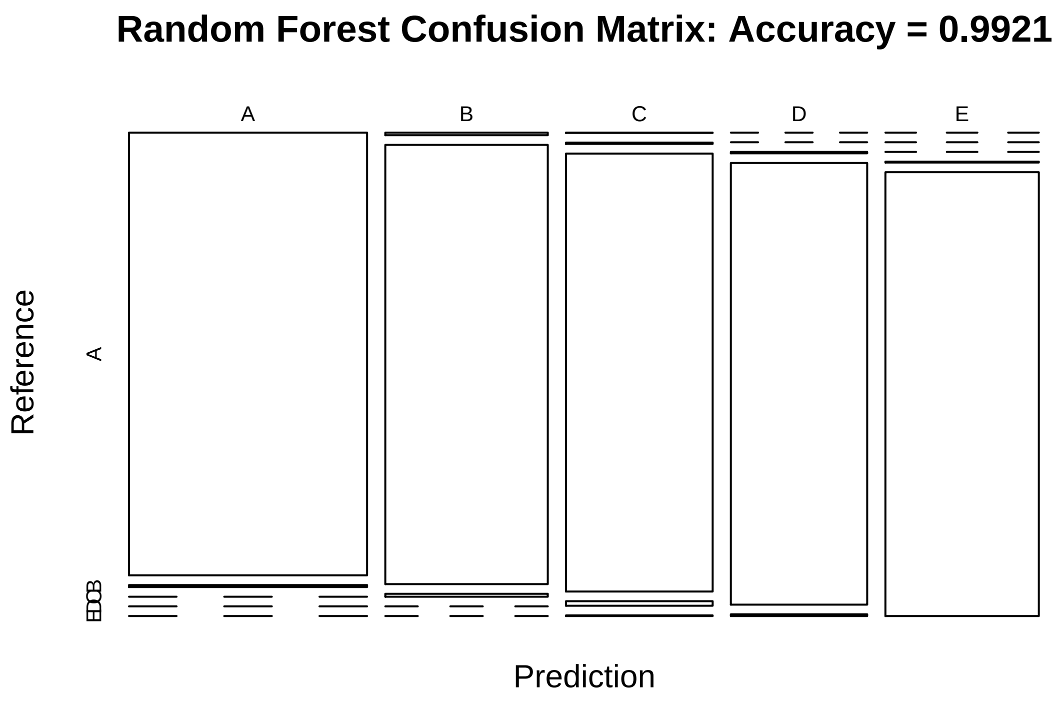
```
model=randomForest(classe~., data=subTraining, method='class')
pred=predict(model,subTesting, type='class')
z<-confusionMatrix(pred,subTesting$classe)
z
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2222   10    0    0    0
##          B    9 1504   10    0    0
##          C    1    4 1354   14    2
##          D    0    0    4 1269    5
##          E    0    0    0    3 1435
##
## Overall Statistics
##
##                Accuracy : 0.9921
##                  95% CI : (0.9899, 0.9939)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.99
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9955   0.9908   0.9898   0.9868   0.9951
## Specificity            0.9982   0.9970   0.9968   0.9986   0.9995
## Pos Pred Value         0.9955   0.9875   0.9847   0.9930   0.9979
## Neg Pred Value         0.9982   0.9978   0.9978   0.9974   0.9989
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2832   0.1917   0.1726   0.1617   0.1829
## Detection Prevalence   0.2845   0.1941   0.1752   0.1629   0.1833
## Balanced Accuracy      0.9969   0.9939   0.9933   0.9927   0.9973
```

```r
plot(model)
```



```r
plot(z$table, col = z$byClass, main = paste("Random Forest Confusion Matrix: Accuracy =", round(z$overal
```



Based on this, The accuracy is 99.5%. The out of sample error, that is the error rate on a new (subTesting) data set, here is going to be 0.69%, with a 95% confidence interval of 0.9932% to .9965%.

Predicting Results on the Test Data

Random Forests gave an Accuracy in the dataset of 99.5%, which was more accurate than what i expected. The expected out-of-sample error is 100-99.5 = 0.5%. ###Read Testing Data:

```
d=read.csv('pml-testing.csv',na.strings=c('','NA'))
e=d[,!apply(d,2,function(x) any(is.na(x)) )]
f=e[,-c(1:7)]
```

Once the dataset it processed, I continue to analyse it using the model developed above

```
predicted=predict(model,f,type='class')
save(predicted,file='predicted.RData')
load('predicted.RData')
predicted
```

```
##  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
```

Write the results to a text file for prediction: So we are creating a function that can write:

```
pml_write_files = function(x){
    n = length(x)
    for(i in 1:n){
```

```r
        filename = paste0("problem_id_",i,".txt")
        write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
    }
}
```

```r
pml_write_files(predicted)
```