CKME136_RandomForest

Ebunoluwa Odeniyi

Load required packages

```
require(plyr)
```

```
## Loading required package: plyr
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

Load data

```
rdmf <- read.csv("C:/Users/YENN/Desktop/UST/FARS2016N/accident2016.csv", header = T, s
tringsAsFactors = F)
```

Remove TWAY_ID2 attribute, the only variable with missing values: <sum(is.na (accs$TWAY_ID2))> and TWAY_ID, not appropiate for the research project

Remove YEAR, MONTH, DAY, HOUR, MINUTE attributes - it's been merged into Timestamps 12:14

Remove WEATHER1, WEATHER2 attributes, are duplicate of the original WEATHER

Remove RAIL attribute, no relevant to the research

```
rdmf2016 <- rdmf[,-c(1:2,10:14,16:17,23:24,37:38,41)]
```

```
require(randomForest)
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
set.seed(123)
```

```
dim(rdmf2016)
```

```
## [1] 34439    38
```

Saperating Training and Test Sets

```
#training Sample with 20600 observations
train=sample(1:nrow(rdmf2016),20600)
```

We are going to use variable 'FATALS' as the Response variable, which is the number of fatalities recorded. I will fit 500 Trees. Fitting the Random Forest

Will use all the Predictor in the dataset.

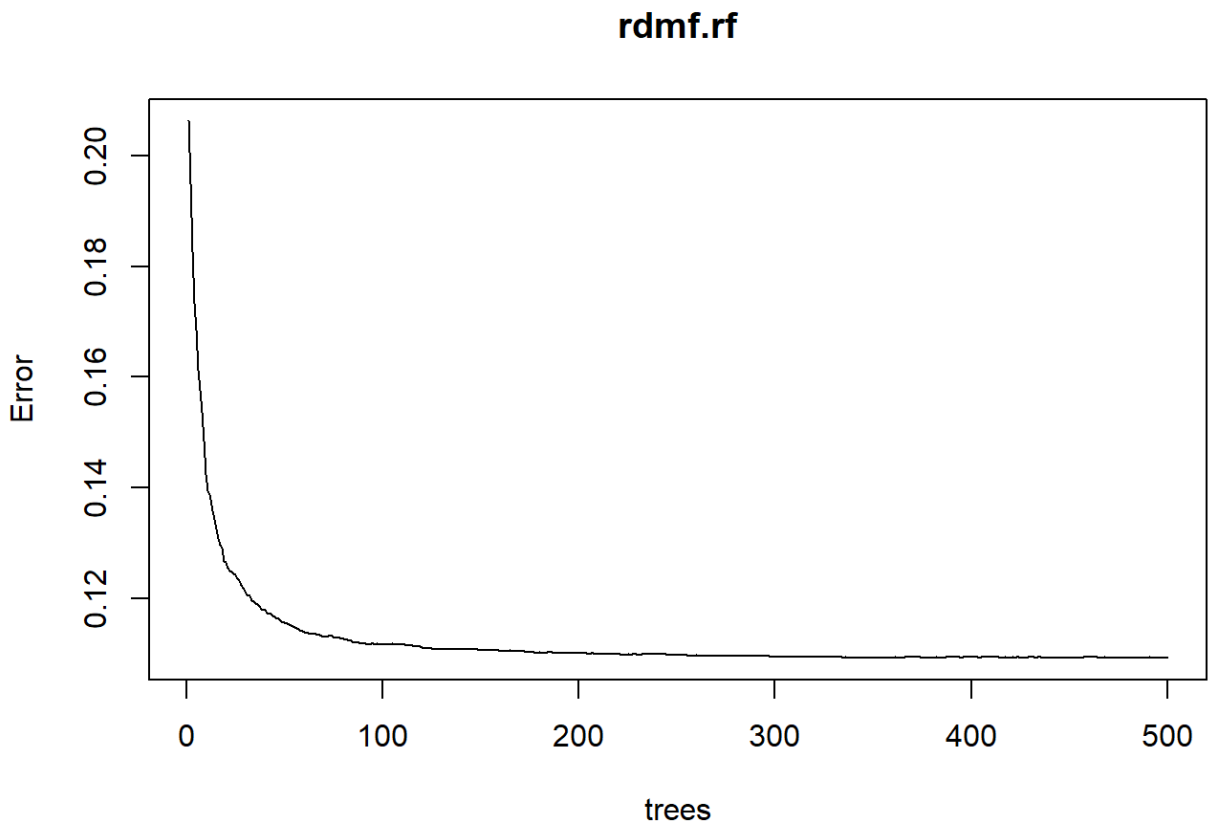
```
rdmf.rf=randomForest(FATALS ~ . , data = rdmf2016 , subset = train)
rdmf.rf
```

```
##
## Call:
##  randomForest(formula = FATALS ~ ., data = rdmf2016, subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 12
##
##           Mean of squared residuals: 0.1092732
##                     % Var explained: 15
```

The above Mean Squared Error and Variance explained are calculated using Out of Bag Error Estimation.In this $\frac{2}{3}$ of Training data is used for training and the remaining (13) is used to Validate the Trees. Also, the number of variables randomly selected at each split is 12.

Plotting the Error vs Number of Trees Graph.

```
plot(rdmf.rf)
```



This plot shows the Error and the Number of Trees.We can easily notice that how the Error is dropping as we keep on adding more and more trees and average them. Now we can compare the Out of Bag Sample Errors and Error on Test set

The above Random Forest model chose Randomly 12 variables to be considered at each split. We could now try all possible 13 variabless which can be found at each split.

```
oob.err=double(13)
test.err=double(13)

#mtry is no of Variables randomly chosen at each split
for(mtry in 1:13)
{
  rf=randomForest(FATALS ~ . , data = rdmf2016 , subset = train,mtry=mtry,ntree=400)
  oob.err[mtry] = rf$mse[400] #Error of all Trees fitted

  pred<-predict(rf,rdmf2016[-train,]) #Predictions on Test Set for each Tree
  test.err[mtry]= with(rdmf2016[-train,], mean( (FATALS - pred)^2)) #Mean Squared Tes
t Error

  cat(mtry," ") #printing the output to the console

}
```

```
## 1  2  3  4  5  6  7  8  9  10  11  12  13
```

Test Error

```
test.err
```

```
##  [1] 0.10858384 0.10198499 0.10022441 0.09945675 0.09933363 0.09923231
##  [7] 0.09935664 0.09927539 0.09980057 0.09980802 0.09980319 0.10034371
## [13] 0.10076333
```

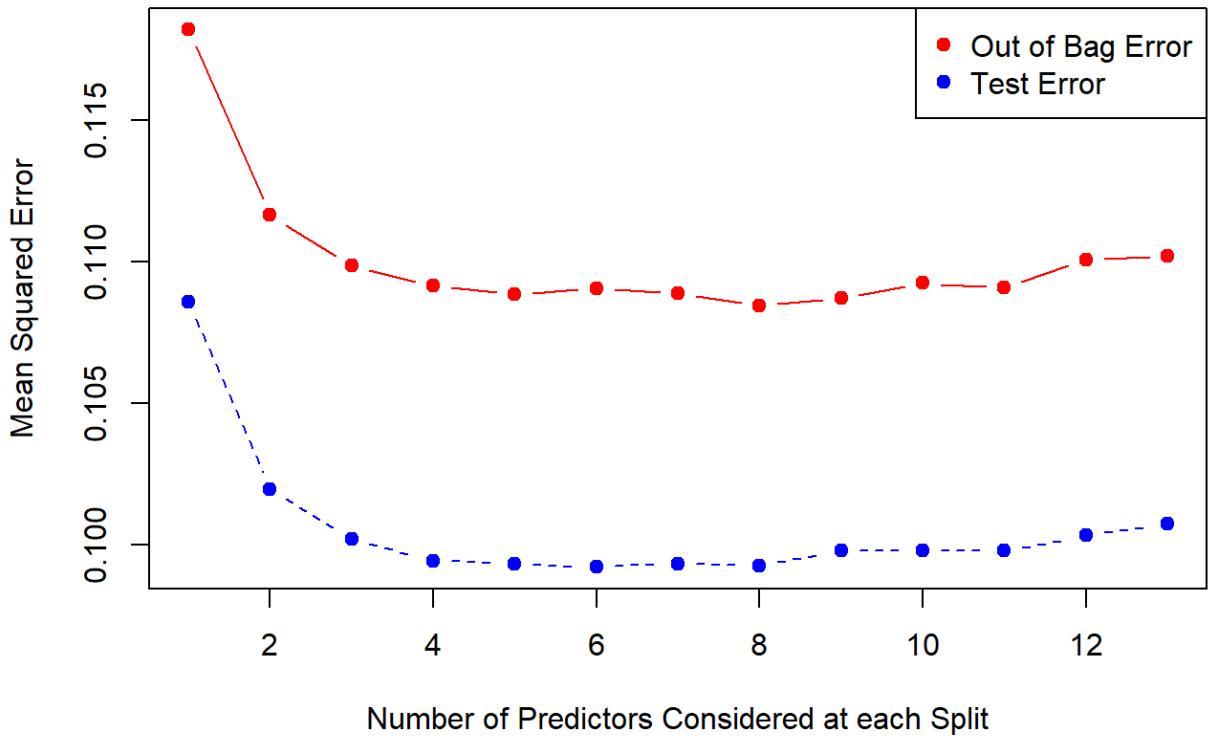
Out of Bag Error Estimation

```
oob.err
```

```
##  [1] 0.1181804 0.1116352 0.1098616 0.1091501 0.1088589 0.1090502 0.1088952
##  [8] 0.1084589 0.1087164 0.1092395 0.1090863 0.1100608 0.1101918
```

What happens is that we are growing 400 trees for 13 times i.e for all 13 predictors. Plotting both Test Error and Out of Bag Error

```
matplot(1:mtry , cbind(oob.err,test.err), pch=19 , col=c("red","blue"),type="b",ylab
="Mean Squared Error",xlab="Number of Predictors Considered at each Split")
legend("topright",legend=c("Out of Bag Error","Test Error"),pch=19, col=c("red","blu
e"))
```


Out of Bag Error
Test Error
Mean Squared Error
0.100
0.105
0.110
0.115
2
4
6
8
10
12
Number of Predictors Considered at each Split