Untitled

# Preliminary Data Analysis

## Source

The dataset is publically available on kaggle’s website as a part of competition held five years ago on 19th Jan, 2011 under the name of “Stay Alert! The Ford Challenge”.

##### Reading the files for analysis #####  
# Train Data would be used for training the model and testing the model performance  
data <- read.csv("Data/fordTrain.csv",  
 header=TRUE, stringsAsFactors=FALSE, na.strings = c("NA", ""),  
 strip.white = TRUE, blank.lines.skip=TRUE, skip=0)  
  
# Test Data would be used as a holdout set for model validation  
validatedata <- read.csv("Data/fordTest.csv",  
 header=TRUE, stringsAsFactors=FALSE, na.strings = c("NA", ""),  
 strip.white = TRUE, blank.lines.skip=TRUE, skip=0)  
  
dim(data)

## [1] 604329 33

## Data Description

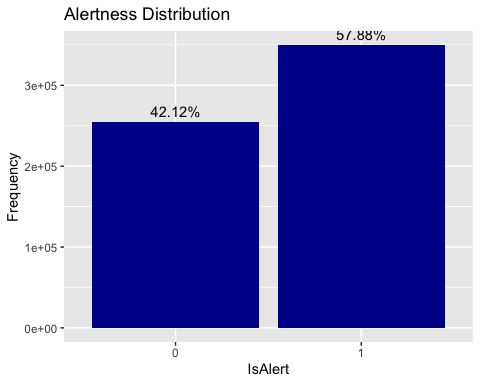
The dataset of 604329 observations consists of 100 drivers of both genders, of different ages and ethnic backgrounds, who have been sampled a total of 500 times against 3 key sets of variables. There are 33 attributes in the dataset in total.

* Physiological (8 features) defined simply as P1 to P8.
* Environmental (11 features) defined simply as E1 to E11.
* Vehicular (11 features) defined simply as V1 to V11.

Each driver trail has been recorded in a simulated driving environment for a period of 2 minutes and an observation recorded every 100 milliseconds. Each driver’s trail has been defined uniquely and labelled at TrialID and every observation within each trail is defined uniquely and labelled as ObsNum.

The objective is to design a classifier that will detect whether the driver is alert or not alert using predictors like the driver’s physiological attributes combined with vehicular and environmental attributes acquired from the simulated environment. The outcome variable is also provide in the dataset and is labelled as IsAlert and is a binary (0 or 1) outcome where ‘1’ means alert and ‘0’ means not alert.

##### Printing the distribution of Result #####  
counts <- table(data$IsAlert)  
countsframe<-as.data.frame(counts)  
  
ggplot(countsframe, aes(x = Var1, y = Freq)) +  
 geom\_bar(stat = "identity", fill = '#000099') +  
 geom\_text(aes(label = sprintf("%.2f%%", Freq/sum(Freq) \* 100)) ,   
 vjust = -.5)+  
 scale\_size\_area() +  
 ggtitle("Alertness Distribution") + xlab("IsAlert") + ylab("Frequency")



From the above plot, it is noted that 42.12% of the observations are not alert, which means the dataset is fairly balanced for modelling assuming we don’t drop any observation in data pre-processing.

## Descriptive statistics and analysis

**Dataset Assumption**: It is assumed that the dataset is collected in same simulated environment using the same sensors for all trials of different drivers.

**Descriptive statistics**:

#Descriptive Statistics  
psych::describe(data, fast = FALSE )

## vars n mean sd median trimmed mad min  
## TrialID 1 604329 250.17 145.45 250.00 249.49 185.32 0.00  
## ObsNum 2 604329 603.84 348.93 604.00 603.83 447.75 0.00  
## IsAlert 3 604329 0.58 0.49 1.00 0.60 0.00 0.00  
## P1 4 604329 35.45 7.48 34.15 34.56 4.00 -22.48  
## P2 5 604329 12.00 3.76 11.40 11.74 2.62 -45.63  
## P3 6 604329 1026.67 309.28 1000.00 1008.35 320.24 504.00  
## P4 7 604329 64.06 19.76 60.00 62.45 20.53 23.89  
## P5 8 604329 0.18 0.37 0.11 0.13 0.03 0.04  
## P6 9 604329 845.38 2505.34 800.00 794.79 177.91 128.00  
## P7 10 604329 77.89 18.58 75.00 77.20 17.35 0.26  
## P8 11 604329 0.00 0.00 0.00 0.00 0.00 0.00  
## E1 12 604329 10.51 14.05 0.00 9.06 0.00 0.00  
## E2 13 604329 102.79 127.26 0.00 85.89 0.00 0.00  
## E3 14 604329 0.29 1.01 0.00 0.00 0.00 0.00  
## E4 15 604329 -4.23 35.51 0.00 -1.26 11.86 -250.00  
## E5 16 604329 0.02 0.00 0.02 0.02 0.00 0.01  
## E6 17 604329 358.67 27.40 365.00 360.09 19.27 260.00  
## E7 18 604329 1.76 2.85 1.00 1.02 1.48 0.00  
## E8 19 604329 1.38 1.61 1.00 1.10 1.48 0.00  
## E9 20 604329 0.88 0.33 1.00 0.97 0.00 0.00  
## E10 21 604329 63.31 18.89 67.00 63.49 11.86 0.00  
## E11 22 604329 1.32 5.25 0.00 0.00 0.00 0.00  
## V1 23 604329 76.97 44.39 100.40 81.48 17.11 0.00  
## V2 24 604329 -0.04 0.40 0.00 -0.03 0.26 -4.80  
## V3 25 604329 573.79 298.41 511.00 559.53 379.55 240.00  
## V4 26 604329 19.96 63.27 3.02 5.23 2.27 0.00  
## V5 27 604329 0.18 0.38 0.00 0.10 0.00 0.00  
## V6 28 604329 1715.69 618.18 1994.00 1769.39 302.45 0.00  
## V7 29 604329 0.00 0.00 0.00 0.00 0.00 0.00  
## V8 30 604329 12.71 11.53 12.80 11.78 18.09 0.00  
## V9 31 604329 0.00 0.00 0.00 0.00 0.00 0.00  
## V10 32 604329 3.31 1.24 4.00 3.50 0.00 1.00  
## V11 33 604329 11.67 9.93 10.77 11.35 4.94 1.68  
## max range skew kurtosis se  
## TrialID 510.00 510.00 0.02 -1.17 0.19  
## ObsNum 1210.00 1210.00 0.00 -1.20 0.45  
## IsAlert 1.00 1.00 -0.32 -1.90 0.00  
## P1 101.35 123.83 2.45 17.77 0.01  
## P2 71.17 116.80 0.86 10.06 0.00  
## P3 2512.00 2008.00 0.51 -0.28 0.40  
## P4 119.05 95.16 0.64 -0.30 0.03  
## P5 27.20 27.16 20.22 937.06 0.00  
## P6 228812.00 228684.00 89.79 8164.54 3.22  
## P7 468.75 468.49 1.94 22.75 0.02  
## P8 0.00 0.00 NaN NaN 0.00  
## E1 243.99 243.99 0.77 0.15 0.02  
## E2 360.00 360.00 0.75 -1.00 0.16  
## E3 4.00 4.00 3.35 9.42 0.00  
## E4 260.00 510.00 -2.47 19.76 0.05  
## E5 0.02 0.02 0.18 3.82 0.00  
## E6 513.00 253.00 -0.47 2.95 0.04  
## E7 25.00 25.00 2.84 8.13 0.00  
## E8 9.00 9.00 2.20 6.34 0.00  
## E9 1.00 1.00 -2.29 3.26 0.00  
## E10 127.00 127.00 -0.43 1.11 0.02  
## E11 52.40 52.40 4.06 15.66 0.01  
## V1 129.70 129.70 -0.94 -0.85 0.06  
## V2 3.99 8.79 -0.76 14.69 0.00  
## V3 1023.00 783.00 0.26 -1.42 0.38  
## V4 484.49 484.49 5.14 27.41 0.08  
## V5 1.00 1.00 1.67 0.78 0.00  
## V6 4892.00 4892.00 -0.87 -0.79 0.80  
## V7 0.00 0.00 NaN NaN 0.00  
## V8 82.10 82.10 0.41 -0.70 0.01  
## V9 0.00 0.00 NaN NaN 0.00  
## V10 7.00 6.00 -1.17 -0.16 0.00  
## V11 262.53 260.86 21.17 527.15 0.01

The primary issue at play is the lack of data dictionary. The various predictor variables are merely categorised in to three types of attributes as Physiological, Environmental and Vehicular and simply labelled as P1-P8, E1-E11 and V1-V11. The descriptive stats show that

* All the variables/attributes are numerical as categorical variables are usually listed with an asterisk in the descriptive summary. However looking at low range of some of the variables like E3, E8, E9, V5, V10 it can be assumed that the variables may be categorical.
* One particular variable (P6) is highly skewed having a max value of 228812 and a mean if just 845.38 which suggest some extreme outliers. We will have look at outliers in the outliers’ analysis section.
* Variable P8, V7 and V9 are all zeroes irrespective of the response variable being 0 or 1. These three variables will be dropped from the dataset for modelling purpose.
* Looking at the skewness, many of the variables are not ‘normal’ in their distribution. This affects the choice of models that be applied to the dataset.

The response variable IsAlert is definitely a categorical variable with possible values of 1 (Alert) and 0 (Not Alert). For modelling purpose, we will convert the response variable in to a factor variable.

**Missing Value Analysis**:

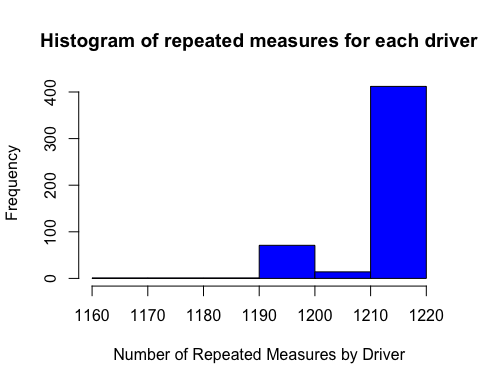
#No missing values  
summary(data)

## TrialID ObsNum IsAlert P1   
## Min. : 0.0 Min. : 0.0 Min. :0.0000 Min. :-22.48   
## 1st Qu.:125.0 1st Qu.: 302.0 1st Qu.:0.0000 1st Qu.: 31.76   
## Median :250.0 Median : 604.0 Median :1.0000 Median : 34.15   
## Mean :250.2 Mean : 603.8 Mean :0.5788 Mean : 35.45   
## 3rd Qu.:374.0 3rd Qu.: 906.0 3rd Qu.:1.0000 3rd Qu.: 37.31   
## Max. :510.0 Max. :1210.0 Max. :1.0000 Max. :101.35   
## P2 P3 P4 P5   
## Min. :-45.629 Min. : 504 Min. : 23.89 Min. : 0.03892   
## 1st Qu.: 9.904 1st Qu.: 792 1st Qu.: 49.18 1st Qu.: 0.09211   
## Median : 11.400 Median :1000 Median : 60.00 Median : 0.10508   
## Mean : 11.997 Mean :1027 Mean : 64.06 Mean : 0.17892   
## 3rd Qu.: 13.644 3rd Qu.:1220 3rd Qu.: 75.76 3rd Qu.: 0.13881   
## Max. : 71.174 Max. :2512 Max. :119.05 Max. :27.20220   
## P6 P7 P8 E1   
## Min. : 128.0 Min. : 0.2622 Min. :0 Min. : 0.00   
## 1st Qu.: 668.0 1st Qu.: 66.6667 1st Qu.:0 1st Qu.: 0.00   
## Median : 800.0 Median : 75.0000 Median :0 Median : 0.00   
## Mean : 845.4 Mean : 77.8876 Mean :0 Mean : 10.51   
## 3rd Qu.: 900.0 3rd Qu.: 89.8204 3rd Qu.:0 3rd Qu.: 28.24   
## Max. :228812.0 Max. :468.7500 Max. :0 Max. :243.99   
## E2 E3 E4 E5   
## Min. : 0.0 Min. :0.0000 Min. :-250.00 Min. :0.00800   
## 1st Qu.: 0.0 1st Qu.:0.0000 1st Qu.: -8.00 1st Qu.:0.01569   
## Median : 0.0 Median :0.0000 Median : 0.00 Median :0.01600   
## Mean :102.8 Mean :0.2906 Mean : -4.23 Mean :0.01626   
## 3rd Qu.:211.6 3rd Qu.:0.0000 3rd Qu.: 6.00 3rd Qu.:0.01669   
## Max. :360.0 Max. :4.0000 Max. : 260.00 Max. :0.02394   
## E6 E7 E8 E9   
## Min. :260.0 Min. : 0.000 Min. :0.000 Min. :0.0000   
## 1st Qu.:348.0 1st Qu.: 0.000 1st Qu.:0.000 1st Qu.:1.0000   
## Median :365.0 Median : 1.000 Median :1.000 Median :1.0000   
## Mean :358.7 Mean : 1.757 Mean :1.383 Mean :0.8768   
## 3rd Qu.:367.0 3rd Qu.: 2.000 3rd Qu.:2.000 3rd Qu.:1.0000   
## Max. :513.0 Max. :25.000 Max. :9.000 Max. :1.0000   
## E10 E11 V1 V2   
## Min. : 0.00 Min. : 0.000 Min. : 0.00 Min. :-4.79500   
## 1st Qu.: 52.00 1st Qu.: 0.000 1st Qu.: 41.93 1st Qu.:-0.17500   
## Median : 67.00 Median : 0.000 Median :100.40 Median : 0.00000   
## Mean : 63.31 Mean : 1.315 Mean : 76.97 Mean :-0.03771   
## 3rd Qu.: 73.00 3rd Qu.: 0.000 3rd Qu.:108.50 3rd Qu.: 0.07000   
## Max. :127.00 Max. :52.400 Max. :129.70 Max. : 3.99000   
## V3 V4 V5 V6   
## Min. : 240.0 Min. : 0.000 Min. :0.0000 Min. : 0   
## 1st Qu.: 255.0 1st Qu.: 1.488 1st Qu.:0.0000 1st Qu.:1259   
## Median : 511.0 Median : 3.019 Median :0.0000 Median :1994   
## Mean : 573.8 Mean : 19.961 Mean :0.1798 Mean :1716   
## 3rd Qu.: 767.0 3rd Qu.: 7.481 3rd Qu.:0.0000 3rd Qu.:2146   
## Max. :1023.0 Max. :484.488 Max. :1.0000 Max. :4892   
## V7 V8 V9 V10 V11   
## Min. :0 Min. : 0.00 Min. :0 Min. :1.000 Min. : 1.677   
## 1st Qu.:0 1st Qu.: 0.00 1st Qu.:0 1st Qu.:3.000 1st Qu.: 7.948   
## Median :0 Median :12.80 Median :0 Median :4.000 Median : 10.773   
## Mean :0 Mean :12.71 Mean :0 Mean :3.312 Mean : 11.668   
## 3rd Qu.:0 3rd Qu.:21.90 3rd Qu.:0 3rd Qu.:4.000 3rd Qu.: 15.271   
## Max. :0 Max. :82.10 Max. :0 Max. :7.000 Max. :262.534

Looking at the summary of the dataset, we can see that none of the variables have missing values otherwise the variable would have been summarized with a statistics of NA’s having a valid positive integer values.

**Repeated Measure Analysis**:

########### Repeated Measure Analysis ################  
hist\_data <- data %>% count(TrialID)  
hist(hist\_data$n, main = "Histogram of repeated measures for each driver",   
 xlab="Number of Repeated Measures by Driver", ylab="Frequency", col = "blue"  
 , breaks=seq(1160,1220,by=10))



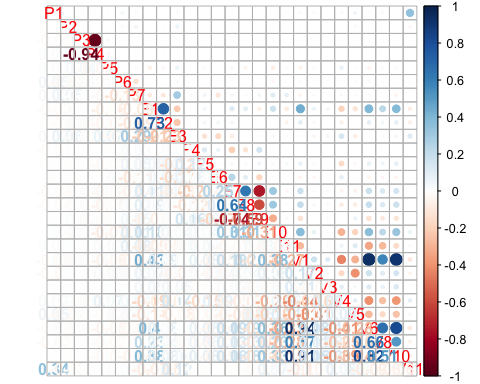
**Outlier Analysis**: Box plots and histograms of individual attributes can be found in the appendix.

All variables have some outliers. In some cases these are quite extreme. We are unable to determine if these from data quality issues or true values as we don’t have the data description of each of the attributes.

Looking at the box plot of P5, P6, E3, E7, E11 and V4 suggests that the data consists mostly of zeroes. However we can’t simply exclude them from analysis as outliers might be a key contributor in predicting the outcome variable. It would be good to look at the distribution of outcome variable for these attributes. For this we will split the dataset based on outcome variable as alert and not alert dataset. Looking at the overlaid histogram (included in the appendix) of the above attributes, we can see that these variables are almost equally distributed among the two subsets. Hence, we will include these attributes in our analysis.

**Multicollinearity Analysis**: Physiological attributes 3&4 and Vehicular attributes 1 is highly correlated with vehicular attribute 6 &10. Only 1 from each mentioned set of correlated attributes will likely be used for this analysis. The plot below shows the collinearity among various attributes. Usually a threshold of .75 for coefficient of correlation is considered to be of high strength relationship.

############ Correlation Analysis #####################  
cor\_data\_full <- data[,!(names(data) %in% c("IsAlert","TrialID","ObsNum","P8","V7","V9"))]  
  
##### Computing the correlation matrix ######  
cor\_mat\_full <- cor(cor\_data\_full, use="complete.obs")  
  
##### Computing the correlation matrix ######  
corrplot.mixed(cor\_mat\_full, lower="number", upper="circle")



#Data Preparation

## Variable Importance

We have already dropped three variables namely P8, V7 and V9 because they are all zeroes. We have already done the collinearity analysis to find out the highly correlated variables, which will help us in dimensionality reduction.

Another technique which we are going to use is Ensemble Feature Selection using fscaret package in R to find out individual variable importance. The fscaret package is closely related to caret package in R and uses the underlying caret function to get its job done. The ensemble feature selection takes in a data set and a list of models and, in return, fscaret will scale and return the importance of each variable for each model and for the ensemble of models. The tool extracts the importance of each variable by using the selected models’ VarImp or similar measuring function. For example, linear models use the absolute value of the t-statistic for each parameter and decision-tree models, total the importance of the individual trees, etc. It returns individual and combined MSEs and RMSEs:

MSE (Mean Squared Error): the variance of the estimator  
RMSE (Root Mean Squared Error): the standard deviation of the sample

For this technique to work, the data needs to be formatted in multiple in, single out (MISO) format. Also the output needs to be the last column in the data frame. Since our outcome variable IsAlert is not the last column, we will need to format the dataset.

#Formatting Dataset  
data<-data[c("TrialID", "ObsNum", "P1", "P2", "P3", "P4", "P5",   
 "P6", "P7", "P8", "E1", "E2", "E3", "E4", "E5", "E6", "E7", "E8",   
 "E9", "E10", "E11", "V1", "V2", "V3", "V4", "V5", "V6", "V7",   
 "V8", "V9", "V10", "V11", "IsAlert")]

As mention in the descriptive analysis, some of the variables in the dataset appear to be categorical in nature. We will convert those variables to factor using a custom function.

# Convert any potential factors in the data through heuristic.  
# If number of unique values in dataset is less than specified threshold  
# then treat as categorical data  
auto\_convert\_factors <- function(data, cat\_threshold=10, cols\_ignore=list()) {  
   
 for (col in names(data)) {  
 if (!is.factor(data[[col]]) &&   
 length(unique(data[[col]])) <= cat\_threshold &&   
 !is.element(col, cols\_ignore)) {  
 data[[col]] <- as.factor(data[[col]])  
 cat(col, " converted to factor\n")  
 }  
 }  
 data  
}  
  
isAlertData <- auto\_convert\_factors(data, 10, cols\_ignore = list('IsAlert'))

## P8 converted to factor  
## E3 converted to factor  
## E8 converted to factor  
## E9 converted to factor  
## V5 converted to factor  
## V7 converted to factor  
## V9 converted to factor  
## V10 converted to factor

Next step is to dummify the factor variables. To do that, we will use some of the caret functions.

#Dropping P8, V7, V9 as they are all zeroes. Dropping TrailID and ObsNum as they are unique ids.  
isAlertData<-isAlertData[c("P1", "P2", "P3", "P4", "P5",   
 "P6", "P7", "E1", "E2", "E3", "E4", "E5", "E6", "E7", "E8",   
 "E9", "E10", "E11", "V1", "V2", "V3", "V4","V5", "V6", "V8",  
 "V10", "V11", "IsAlert")]  
  
datadummy<-dummyVars("~.",data=isAlertData,fullRank = F)  
datatemp<-as.data.frame(predict(datadummy,isAlertData))

We now need a training and a test dataset, for which, again we will use a **caret** function called *createDataPartition*.

#Partitioning dataset in to training and test dataset.  
splitIndexMulti <- createDataPartition(datatemp$IsAlert, p=.01, list = FALSE, times = 2)  
  
trainDataset <- datatemp[splitIndexMulti[,1],]  
testDataset <- datatemp[splitIndexMulti[,2],]  
  
dim(datatemp)

## [1] 604329 45

dim(trainDataset)

## [1] 6044 45

dim(testDataset)

## [1] 6044 45

Finally, we need to select an ensemble of models and feed the data and list of models to the main function of the fscaret package named as its package, fscaret. Since our problem is a classification problem, we can choose either the models specific for classification or dual purpose modes (classification and regression). We have chosen seven models here.

#Variable Importance  
fsModels2 <- c("glm","gbm","treebag","ridge","lasso","rf","xgbLinear")  
myFs2<-fscaret(trainDataset, testDataset, myTimeLimit = 40, preprocessData = TRUE,  
 Used.funcRegPred = fsModels2, with.labels = TRUE,  
 supress.output=FALSE, no.cores = 2, installReqPckg = TRUE)

##   
## ----Loading required packages----

## Loading required package: R.utils

## Loading required package: R.oo

## Loading required package: R.methodsS3

## R.methodsS3 v1.7.1 (2016-02-15) successfully loaded. See ?R.methodsS3 for help.

## R.oo v1.23.0 successfully loaded. See ?R.oo for help.

##   
## Attaching package: 'R.oo'

## The following object is masked from 'package:R.methodsS3':  
##   
## throw

## The following objects are masked from 'package:methods':  
##   
## getClasses, getMethods

## The following objects are masked from 'package:base':  
##   
## attach, detach, load, save

## R.utils v2.9.0 successfully loaded. See ?R.utils for help.

##   
## Attaching package: 'R.utils'

## The following object is masked from 'package:utils':  
##   
## timestamp

## The following objects are masked from 'package:base':  
##   
## cat, commandArgs, getOption, inherits, isOpen, parse, warnings

## Installing package into '/Users/anuj/git/packtest/packrat/lib/x86\_64-apple-darwin18.2.0/3.5.3'  
## (as 'lib' is unspecified)

## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

## Loading required package: multicore

##   
## ----Packages loaded successfully----  
##   
##   
## -----Warnings have been supressed!----  
##   
## ----Processing files:----  
## [1] "28in\_default\_REGControl\_glm.RData"   
## [2] "28in\_default\_REGControl\_treebag.RData"  
## [1] ""  
## [1] "Calculating error for model:"  
## [1] "28in\_default\_REGControl\_glm.RData"  
## [1] ""  
## [1] ""  
## [1] "Calculating error for model:"  
## [1] "28in\_default\_REGControl\_treebag.RData"  
## [1] ""  
##   
## ----Processing files:----  
## [1] "28in\_default\_REGControl\_VarImp\_glm.txt"   
## [2] "28in\_default\_REGControl\_VarImp\_treebag.txt"  
##   
## matrycaVarImp.RMSE after   
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 3.2307598 0 0 0 0  
## [2,] 1.1896445 0 0 0 0  
## [3,] 0.7318956 0 0 0 0  
## [4,] 2.7431136 0 0 0 0  
## [5,] 0.1801848 0 0 0 0  
## [6,] 4.9054814 0 0 0 0  
## [7,] 0.2696821 0 0 0 0  
## [8,] 0.6390385 0 0 0 0  
## [9,] 3.3585705 0 0 0 0  
## [10,] 1.2127235 0 0 0 0  
## [11,] 0.4411352 0 0 0 0  
## [12,] 1.7962115 0 0 0 0  
## [13,] 7.0922255 0 0 0 0  
## [14,] 2.5333880 0 0 0 0  
## [15,] 0.7952113 0 0 0 0  
## [16,] 7.8502437 0 0 0 0  
## [17,] 1.7042753 0 0 0 0  
## [18,] 1.9952307 0 0 0 0  
## [19,] 11.7370864 0 0 0 0  
## [20,] 6.3703497 0 0 0 0  
## [21,] 0.4819320 0 0 0 0  
## [22,] 2.2773929 0 0 0 0  
## [23,] 1.1039857 0 0 0 0  
## [24,] 0.5039236 0 0 0 0  
## [25,] 4.6725725 0 0 0 0  
## [26,] 0.6970519 0 0 0 0  
## [27,] 9.8367760 0 0 0 0  
## [28,] 6.7947500 0 0 0 0  
##   
##   
## matrycaVarImp.RMSE after   
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 3.2307598 2.07943260 0 0 0  
## [2,] 1.1896445 0.03134224 0 0 0  
## [3,] 0.7318956 0.00000000 0 0 0  
## [4,] 2.7431136 9.08905214 0 0 0  
## [5,] 0.1801848 10.01334409 0 0 0  
## [6,] 4.9054814 9.73896291 0 0 0  
## [7,] 0.2696821 0.12833867 0 0 0  
## [8,] 0.6390385 0.00000000 0 0 0  
## [9,] 3.3585705 0.00000000 0 0 0  
## [10,] 1.2127235 0.00000000 0 0 0  
## [11,] 0.4411352 0.01723047 0 0 0  
## [12,] 1.7962115 1.07790888 0 0 0  
## [13,] 7.0922255 3.64101928 0 0 0  
## [14,] 2.5333880 7.00303953 0 0 0  
## [15,] 0.7952113 0.99436259 0 0 0  
## [16,] 7.8502437 10.06593089 0 0 0  
## [17,] 1.7042753 3.88935286 0 0 0  
## [18,] 1.9952307 0.01609853 0 0 0  
## [19,] 11.7370864 2.10004691 0 0 0  
## [20,] 6.3703497 7.38878728 0 0 0  
## [21,] 0.4819320 0.00000000 0 0 0  
## [22,] 2.2773929 0.26550449 0 0 0  
## [23,] 1.1039857 3.27622263 0 0 0  
## [24,] 0.5039236 0.43406307 0 0 0  
## [25,] 4.6725725 7.68611347 0 0 0  
## [26,] 0.6970519 4.71283170 0 0 0  
## [27,] 9.8367760 6.33277166 0 0 0  
## [28,] 6.7947500 10.01824311 0 0 0

The output of the fscaret function (myFs) holds a lot of information. One of the most interesting result set is the matrixVarImp.MSE (Mean Squared Error). This returns the top variables from the perspective of all models involved (the MSE is scaled to compare each model equally):

names(myFs2)

## [1] "ModelPred" "VarImp" "PPlabels" "PPTrainDF" "PPTestDF"

myFs2$VarImp

## $rawMSE  
## glm treebag  
## 1 0.1518937 0.1153515  
##   
## $rawRMSE  
## glm treebag  
## 1 0.3897354 0.3396343  
##   
## $matrixVarImp.RMSE  
## glm treebag SUM SUM% ImpGrad Input\_no  
## 16 7.8502437 10.06593089 17.9161746 100.000000 0.0000000 16  
## 28 6.7947500 10.01824311 16.8129931 93.842539 6.1574610 28  
## 27 9.8367760 6.33277166 16.1695477 90.251117 3.8270724 27  
## 6 4.9054814 9.73896291 14.6444444 81.738679 9.4319480 6  
## 19 11.7370864 2.10004691 13.8371333 77.232633 5.5127461 19  
## 20 6.3703497 7.38878728 13.7591370 76.797292 0.5636741 20  
## 25 4.6725725 7.68611347 12.3586860 68.980607 10.1783343 25  
## 4 2.7431136 9.08905214 11.8321657 66.041809 4.2603261 4  
## 13 7.0922255 3.64101928 10.7332448 59.908128 9.2875719 13  
## 5 0.1801848 10.01334409 10.1935289 56.895677 5.0284506 5  
## 14 2.5333880 7.00303953 9.5364275 53.228034 6.4462599 14  
## 17 1.7042753 3.88935286 5.5936281 31.221107 41.3446168 17  
## 26 0.6970519 4.71283170 5.4098836 30.195529 3.2848890 26  
## 1 3.2307598 2.07943260 5.3101924 29.639097 1.8427613 1  
## 23 1.1039857 3.27622263 4.3802083 24.448346 17.5131896 23  
## 9 3.3585705 0.00000000 3.3585705 18.746024 23.3239560 9  
## 12 1.7962115 1.07790888 2.8741204 16.042043 14.4242932 12  
## 22 2.2773929 0.26550449 2.5428974 14.193305 11.5243270 22  
## 18 1.9952307 0.01609853 2.0113292 11.226332 20.9040346 18  
## 15 0.7952113 0.99436259 1.7895739 9.988594 11.0253117 15  
## 2 1.1896445 0.03134224 1.2209868 6.814997 31.7722073 2  
## 10 1.2127235 0.00000000 1.2127235 6.768875 0.6767739 10  
## 24 0.5039236 0.43406307 0.9379867 5.235418 22.6545305 24  
## 3 0.7318956 0.00000000 0.7318956 4.085111 21.9716417 3  
## 8 0.6390385 0.00000000 0.6390385 3.566825 12.6872022 8  
## 21 0.4819320 0.00000000 0.4819320 2.689927 24.5848290 21  
## 11 0.4411352 0.01723047 0.4583657 2.558390 4.8899680 11  
## 7 0.2696821 0.12833867 0.3980207 2.221572 13.1652414 7  
##   
## $matrixVarImp.MSE  
## glm treebag SUM SUM% ImpGrad Input\_no  
## 16 6.8410820 10.06593089 16.9070129 100.000000 0.000000 16  
## 28 5.9212737 10.01824311 15.9395169 94.277546 5.722454 28  
## 27 8.5722423 6.33277166 14.9050140 88.158767 6.490177 27  
## 6 4.2748738 9.73896291 14.0138367 82.887715 5.979044 6  
## 20 5.5514308 7.38878728 12.9402181 76.537578 7.661132 20  
## 19 10.2282647 2.10004691 12.3283116 72.918331 4.728718 19  
## 25 4.0719057 7.68611347 11.7580192 69.545219 4.625876 25  
## 4 2.3904818 9.08905214 11.4795340 67.898061 2.368470 4  
## 5 0.1570217 10.01334409 10.1703658 60.154717 11.404367 5  
## 13 6.1805083 3.64101928 9.8215276 58.091442 3.429948 13  
## 14 2.2077168 7.00303953 9.2107563 54.478910 6.218699 14  
## 17 1.4851879 3.88935286 5.3745407 31.788825 41.649301 17  
## 26 0.6074448 4.71283170 5.3202765 31.467868 1.009654 26  
## 1 2.8154403 2.07943260 4.8948729 28.951731 7.995892 1  
## 23 0.9620665 3.27622263 4.2382892 25.068232 13.413704 23  
## 9 2.9268207 0.00000000 2.9268207 17.311282 30.943345 9  
## 12 1.5653056 1.07790888 2.6432145 15.633835 9.689909 12  
## 22 1.9846303 0.26550449 2.2501348 13.308884 14.871275 22  
## 18 1.7387405 0.01609853 1.7548391 10.379356 22.011825 18  
## 15 0.6929856 0.99436259 1.6873482 9.980167 3.845986 15  
## 2 1.0367138 0.03134224 1.0680560 6.317237 36.702097 2  
## 10 1.0568259 0.00000000 1.0568259 6.250814 1.051457 10  
## 24 0.4391434 0.43406307 0.8732065 5.164759 17.374615 24  
## 3 0.6378092 0.00000000 0.6378092 3.772454 26.957799 3  
## 8 0.5568891 0.00000000 0.5568891 3.293835 12.687202 8  
## 21 0.4199788 0.00000000 0.4199788 2.484051 24.584829 21  
## 11 0.3844265 0.01723047 0.4016570 2.375683 4.362559 11  
## 7 0.2350140 0.12833867 0.3633527 2.149124 9.536585 7  
##   
## $model  
## list()

myFs2$PPlabels

## Orig Input No Labels  
## 1 1 P1  
## 2 2 P2  
## 3 4 P4  
## 4 5 P5  
## 5 6 P6  
## 6 7 P7  
## 7 8 E1  
## 8 9 E2  
## 9 10 E3.0  
## 10 12 E3.4  
## 11 13 E4  
## 12 14 E5  
## 13 15 E6  
## 14 16 E7  
## 15 17 E8.0  
## 16 18 E8.1  
## 17 19 E8.2  
## 18 20 E8.3  
## 19 28 E9.1  
## 20 29 E10  
## 21 32 V2  
## 22 33 V3  
## 23 34 V4  
## 24 36 V5.1  
## 25 37 V6  
## 26 39 V10.1  
## 27 42 V10.4  
## 28 44 V11

myFs2$VarImp$matrixVarImp.MSE

## glm treebag SUM SUM% ImpGrad Input\_no  
## 16 6.8410820 10.06593089 16.9070129 100.000000 0.000000 16  
## 28 5.9212737 10.01824311 15.9395169 94.277546 5.722454 28  
## 27 8.5722423 6.33277166 14.9050140 88.158767 6.490177 27  
## 6 4.2748738 9.73896291 14.0138367 82.887715 5.979044 6  
## 20 5.5514308 7.38878728 12.9402181 76.537578 7.661132 20  
## 19 10.2282647 2.10004691 12.3283116 72.918331 4.728718 19  
## 25 4.0719057 7.68611347 11.7580192 69.545219 4.625876 25  
## 4 2.3904818 9.08905214 11.4795340 67.898061 2.368470 4  
## 5 0.1570217 10.01334409 10.1703658 60.154717 11.404367 5  
## 13 6.1805083 3.64101928 9.8215276 58.091442 3.429948 13  
## 14 2.2077168 7.00303953 9.2107563 54.478910 6.218699 14  
## 17 1.4851879 3.88935286 5.3745407 31.788825 41.649301 17  
## 26 0.6074448 4.71283170 5.3202765 31.467868 1.009654 26  
## 1 2.8154403 2.07943260 4.8948729 28.951731 7.995892 1  
## 23 0.9620665 3.27622263 4.2382892 25.068232 13.413704 23  
## 9 2.9268207 0.00000000 2.9268207 17.311282 30.943345 9  
## 12 1.5653056 1.07790888 2.6432145 15.633835 9.689909 12  
## 22 1.9846303 0.26550449 2.2501348 13.308884 14.871275 22  
## 18 1.7387405 0.01609853 1.7548391 10.379356 22.011825 18  
## 15 0.6929856 0.99436259 1.6873482 9.980167 3.845986 15  
## 2 1.0367138 0.03134224 1.0680560 6.317237 36.702097 2  
## 10 1.0568259 0.00000000 1.0568259 6.250814 1.051457 10  
## 24 0.4391434 0.43406307 0.8732065 5.164759 17.374615 24  
## 3 0.6378092 0.00000000 0.6378092 3.772454 26.957799 3  
## 8 0.5568891 0.00000000 0.5568891 3.293835 12.687202 8  
## 21 0.4199788 0.00000000 0.4199788 2.484051 24.584829 21  
## 11 0.3844265 0.01723047 0.4016570 2.375683 4.362559 11  
## 7 0.2350140 0.12833867 0.3633527 2.149124 9.536585 7

The input\_no is actually the numeric label for each of the attributes. This can be reformatted as shown below to show the actual variables and are listed in the descending order of their importance. For example, the above output, input\_no 28 means V11 variable and the input\_no 19 is the E9.1 variable which is the dummified version of E9 variable.

results <- myFs2$VarImp$matrixVarImp.MSE  
results$Input\_no <- as.numeric(results$Input\_no)  
results <- results[c("SUM","SUM%","ImpGrad","Input\_no")]  
myFs2$PPlabels$Input\_no <- as.numeric(rownames(myFs2$PPlabels))  
results <- merge(x=results, y=myFs2$PPlabels, by="Input\_no", all.x=T)  
results <- results[c('Labels', 'SUM')]  
results <- subset(results,results$SUM !=0)  
results <- results[order(-results$SUM),]  
print(results)

## Labels SUM  
## 16 E8.1 16.9070129  
## 28 V11 15.9395169  
## 27 V10.4 14.9050140  
## 6 P7 14.0138367  
## 20 E10 12.9402181  
## 19 E9.1 12.3283116  
## 25 V6 11.7580192  
## 4 P5 11.4795340  
## 5 P6 10.1703658  
## 13 E6 9.8215276  
## 14 E7 9.2107563  
## 17 E8.2 5.3745407  
## 26 V10.1 5.3202765  
## 1 P1 4.8948729  
## 23 V4 4.2382892  
## 9 E3.0 2.9268207  
## 12 E5 2.6432145  
## 22 V3 2.2501348  
## 18 E8.3 1.7548391  
## 15 E8.0 1.6873482  
## 2 P2 1.0680560  
## 10 E3.4 1.0568259  
## 24 V5.1 0.8732065  
## 3 P4 0.6378092  
## 8 E2 0.5568891  
## 21 V2 0.4199788  
## 11 E4 0.4016570  
## 7 E1 0.3633527

The most import variable is the vehicular attribute 11 to predict the alertness of the driver. The next most important variable is the categorical variable E9 and to be precise the E9 being ‘1’. Also, from the MSE output it is noted that the different models have predicted the variable importance differently. For example, the most important variable as per the GBM, XGBoost models is V11 while E9.1 is the most important variable as per GLM, lasso and ridge. It is also noted that the above function has dropped two variables, namely P4 and V1 from its analysis which we found to be highly correlated with P3 and V6, V10 respectively. So, we can drop the two variables P4 and V1 from our dataset for modelling purposes.

# Modelling and Evaluation

## Tuning and Modelling

We will again use the caret package in R to build our models and evaluate them. The train function in the package will be used for evaluating the effect of model tuning parameters on performance using resampling, choosing the optimal model across these parameters and estimating the model performance from a training set.

The first step is choosing a model. We will use Stochastic Gradient Boosting (gbm) and Extreme Gradient Boosting (xgbLinear) for our modelling purpose. Both are dual purpose models and can be used for both classification and regression.

**Basic Parameter Tuning**  
We will use 5-fold stratified repeated cross validation using the traincontrol function to estimate model performance and generalize the model to limit over fitting.

# Uses caret library, doing Automatic grid search (possible to do manual one as well)  
  
modelTrain<-trainDataset  
modelTrain$IsAlert <- as.factor(ifelse(modelTrain$IsAlert == 1,'Y','N'))  
  
modelTest<-testDataset  
modelTest$IsAlert <- as.factor(ifelse(modelTest$IsAlert == 1,'Y','N'))  
  
#Defining training control  
control <- trainControl(  
 method = "repeatedcv",  
 number = 5,  
 repeats = 2,  
 search = "grid",  
 classProbs = TRUE,  
 summaryFunction = twoClassSummary, #ROC AUC   
 verboseIter = TRUE  
)

In the above trainControl function, we are asking to compute additional performance metric of the classification model called twoClassSummary. By default, accuracy and Kappa metrics are computed for a classification model. The twoClassSummary function will be used to compute the sensitivity, specificity and area under the ROC curve.

We will now use the train function of the caret package to train our two models. The same models are trained on the complete dataset and reduced dataset (Removing highly correlated variables). We will customise the tuning process by using pre-processing options of centring, scaling and imputation. We do not have missing values in our dataset, hence imputation won’t be required.

model1 <- train(IsAlert ~ .,   
 data = modelTrain[,!(names(modelTrain) %in% c("TrialID","ObsNum","P8","V7","V9"))],   
 method = "gbm",   
 metric = "ROC",  
 na.action = na.pass,  
 preProcess = c("center", "scale", "medianImpute"),  
 trControl = control)

## + Fold1.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3325 nan 0.1000 0.0147  
## 2 1.3100 nan 0.1000 0.0114  
## 3 1.2893 nan 0.1000 0.0098  
## 4 1.2714 nan 0.1000 0.0094  
## 5 1.2530 nan 0.1000 0.0089  
## 6 1.2363 nan 0.1000 0.0074  
## 7 1.2204 nan 0.1000 0.0072  
## 8 1.2071 nan 0.1000 0.0068  
## 9 1.1945 nan 0.1000 0.0057  
## 10 1.1808 nan 0.1000 0.0067  
## 20 1.0826 nan 0.1000 0.0042  
## 40 0.9626 nan 0.1000 0.0019  
## 60 0.8963 nan 0.1000 0.0013  
## 80 0.8527 nan 0.1000 0.0005  
## 100 0.8260 nan 0.1000 0.0003  
## 120 0.8060 nan 0.1000 0.0003  
## 140 0.7908 nan 0.1000 0.0000  
## 150 0.7832 nan 0.1000 -0.0001  
##   
## - Fold1.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold1.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3066 nan 0.1000 0.0271  
## 2 1.2615 nan 0.1000 0.0212  
## 3 1.2250 nan 0.1000 0.0177  
## 4 1.1941 nan 0.1000 0.0152  
## 5 1.1672 nan 0.1000 0.0128  
## 6 1.1431 nan 0.1000 0.0112  
## 7 1.1230 nan 0.1000 0.0100  
## 8 1.1046 nan 0.1000 0.0084  
## 9 1.0772 nan 0.1000 0.0138  
## 10 1.0552 nan 0.1000 0.0107  
## 20 0.9328 nan 0.1000 0.0046  
## 40 0.8227 nan 0.1000 0.0018  
## 60 0.7742 nan 0.1000 0.0004  
## 80 0.7334 nan 0.1000 0.0002  
## 100 0.7076 nan 0.1000 0.0001  
## 120 0.6803 nan 0.1000 0.0008  
## 140 0.6609 nan 0.1000 -0.0001  
## 150 0.6535 nan 0.1000 0.0000  
##   
## - Fold1.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold1.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2973 nan 0.1000 0.0313  
## 2 1.2449 nan 0.1000 0.0260  
## 3 1.1964 nan 0.1000 0.0235  
## 4 1.1578 nan 0.1000 0.0185  
## 5 1.1285 nan 0.1000 0.0146  
## 6 1.0924 nan 0.1000 0.0169  
## 7 1.0643 nan 0.1000 0.0137  
## 8 1.0384 nan 0.1000 0.0124  
## 9 1.0194 nan 0.1000 0.0084  
## 10 0.9975 nan 0.1000 0.0108  
## 20 0.8612 nan 0.1000 0.0037  
## 40 0.7548 nan 0.1000 0.0008  
## 60 0.6981 nan 0.1000 0.0010  
## 80 0.6579 nan 0.1000 0.0004  
## 100 0.6306 nan 0.1000 -0.0000  
## 120 0.6072 nan 0.1000 0.0005  
## 140 0.5866 nan 0.1000 -0.0000  
## 150 0.5782 nan 0.1000 -0.0000  
##   
## - Fold1.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold2.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3316 nan 0.1000 0.0147  
## 2 1.3073 nan 0.1000 0.0110  
## 3 1.2867 nan 0.1000 0.0106  
## 4 1.2683 nan 0.1000 0.0093  
## 5 1.2499 nan 0.1000 0.0093  
## 6 1.2341 nan 0.1000 0.0078  
## 7 1.2173 nan 0.1000 0.0079  
## 8 1.2036 nan 0.1000 0.0065  
## 9 1.1893 nan 0.1000 0.0066  
## 10 1.1768 nan 0.1000 0.0060  
## 20 1.0777 nan 0.1000 0.0041  
## 40 0.9597 nan 0.1000 0.0017  
## 60 0.8908 nan 0.1000 0.0008  
## 80 0.8506 nan 0.1000 0.0006  
## 100 0.8231 nan 0.1000 0.0001  
## 120 0.8050 nan 0.1000 0.0002  
## 140 0.7896 nan 0.1000 -0.0001  
## 150 0.7828 nan 0.1000 0.0003  
##   
## - Fold2.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold2.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3057 nan 0.1000 0.0283  
## 2 1.2614 nan 0.1000 0.0227  
## 3 1.2244 nan 0.1000 0.0186  
## 4 1.1916 nan 0.1000 0.0157  
## 5 1.1650 nan 0.1000 0.0125  
## 6 1.1384 nan 0.1000 0.0120  
## 7 1.1167 nan 0.1000 0.0108  
## 8 1.0968 nan 0.1000 0.0094  
## 9 1.0780 nan 0.1000 0.0083  
## 10 1.0612 nan 0.1000 0.0077  
## 20 0.9255 nan 0.1000 0.0044  
## 40 0.8214 nan 0.1000 0.0011  
## 60 0.7717 nan 0.1000 0.0008  
## 80 0.7291 nan 0.1000 0.0005  
## 100 0.7024 nan 0.1000 0.0002  
## 120 0.6788 nan 0.1000 0.0001  
## 140 0.6587 nan 0.1000 -0.0001  
## 150 0.6508 nan 0.1000 0.0002  
##   
## - Fold2.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold2.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2951 nan 0.1000 0.0328  
## 2 1.2418 nan 0.1000 0.0259  
## 3 1.1986 nan 0.1000 0.0212  
## 4 1.1550 nan 0.1000 0.0212  
## 5 1.1238 nan 0.1000 0.0151  
## 6 1.0973 nan 0.1000 0.0125  
## 7 1.0714 nan 0.1000 0.0123  
## 8 1.0466 nan 0.1000 0.0119  
## 9 1.0262 nan 0.1000 0.0094  
## 10 0.9992 nan 0.1000 0.0135  
## 20 0.8552 nan 0.1000 0.0051  
## 40 0.7471 nan 0.1000 0.0015  
## 60 0.6944 nan 0.1000 0.0006  
## 80 0.6528 nan 0.1000 0.0002  
## 100 0.6229 nan 0.1000 0.0001  
## 120 0.5993 nan 0.1000 0.0007  
## 140 0.5812 nan 0.1000 -0.0003  
## 150 0.5734 nan 0.1000 -0.0002  
##   
## - Fold2.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold3.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3328 nan 0.1000 0.0146  
## 2 1.3095 nan 0.1000 0.0111  
## 3 1.2876 nan 0.1000 0.0107  
## 4 1.2699 nan 0.1000 0.0086  
## 5 1.2510 nan 0.1000 0.0089  
## 6 1.2332 nan 0.1000 0.0084  
## 7 1.2179 nan 0.1000 0.0068  
## 8 1.2007 nan 0.1000 0.0084  
## 9 1.1891 nan 0.1000 0.0055  
## 10 1.1750 nan 0.1000 0.0072  
## 20 1.0732 nan 0.1000 0.0043  
## 40 0.9514 nan 0.1000 0.0018  
## 60 0.8841 nan 0.1000 0.0014  
## 80 0.8413 nan 0.1000 0.0005  
## 100 0.8161 nan 0.1000 0.0004  
## 120 0.7950 nan 0.1000 0.0003  
## 140 0.7818 nan 0.1000 0.0000  
## 150 0.7748 nan 0.1000 0.0001  
##   
## - Fold3.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold3.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3051 nan 0.1000 0.0286  
## 2 1.2595 nan 0.1000 0.0227  
## 3 1.2222 nan 0.1000 0.0185  
## 4 1.1934 nan 0.1000 0.0129  
## 5 1.1622 nan 0.1000 0.0145  
## 6 1.1365 nan 0.1000 0.0123  
## 7 1.1144 nan 0.1000 0.0107  
## 8 1.0957 nan 0.1000 0.0094  
## 9 1.0670 nan 0.1000 0.0140  
## 10 1.0519 nan 0.1000 0.0073  
## 20 0.9232 nan 0.1000 0.0044  
## 40 0.8156 nan 0.1000 0.0009  
## 60 0.7668 nan 0.1000 0.0007  
## 80 0.7263 nan 0.1000 0.0004  
## 100 0.6996 nan 0.1000 0.0000  
## 120 0.6770 nan 0.1000 0.0003  
## 140 0.6588 nan 0.1000 0.0001  
## 150 0.6491 nan 0.1000 -0.0000  
##   
## - Fold3.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold3.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2973 nan 0.1000 0.0311  
## 2 1.2448 nan 0.1000 0.0258  
## 3 1.2017 nan 0.1000 0.0216  
## 4 1.1660 nan 0.1000 0.0181  
## 5 1.1254 nan 0.1000 0.0196  
## 6 1.0974 nan 0.1000 0.0126  
## 7 1.0711 nan 0.1000 0.0124  
## 8 1.0493 nan 0.1000 0.0104  
## 9 1.0242 nan 0.1000 0.0120  
## 10 0.9969 nan 0.1000 0.0134  
## 20 0.8523 nan 0.1000 0.0056  
## 40 0.7441 nan 0.1000 0.0006  
## 60 0.6861 nan 0.1000 0.0009  
## 80 0.6501 nan 0.1000 0.0007  
## 100 0.6226 nan 0.1000 0.0003  
## 120 0.6037 nan 0.1000 -0.0001  
## 140 0.5817 nan 0.1000 0.0002  
## 150 0.5730 nan 0.1000 0.0001  
##   
## - Fold3.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold4.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3322 nan 0.1000 0.0146  
## 2 1.3104 nan 0.1000 0.0104  
## 3 1.2875 nan 0.1000 0.0114  
## 4 1.2680 nan 0.1000 0.0092  
## 5 1.2489 nan 0.1000 0.0094  
## 6 1.2323 nan 0.1000 0.0074  
## 7 1.2162 nan 0.1000 0.0082  
## 8 1.2019 nan 0.1000 0.0068  
## 9 1.1882 nan 0.1000 0.0064  
## 10 1.1750 nan 0.1000 0.0062  
## 20 1.0716 nan 0.1000 0.0040  
## 40 0.9504 nan 0.1000 0.0022  
## 60 0.8811 nan 0.1000 0.0011  
## 80 0.8393 nan 0.1000 0.0003  
## 100 0.8121 nan 0.1000 0.0005  
## 120 0.7910 nan 0.1000 0.0003  
## 140 0.7758 nan 0.1000 0.0001  
## 150 0.7687 nan 0.1000 0.0001  
##   
## - Fold4.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold4.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3037 nan 0.1000 0.0279  
## 2 1.2581 nan 0.1000 0.0231  
## 3 1.2196 nan 0.1000 0.0183  
## 4 1.1881 nan 0.1000 0.0158  
## 5 1.1618 nan 0.1000 0.0133  
## 6 1.1384 nan 0.1000 0.0116  
## 7 1.1098 nan 0.1000 0.0145  
## 8 1.0907 nan 0.1000 0.0085  
## 9 1.0706 nan 0.1000 0.0098  
## 10 1.0526 nan 0.1000 0.0086  
## 20 0.9223 nan 0.1000 0.0042  
## 40 0.8087 nan 0.1000 0.0027  
## 60 0.7540 nan 0.1000 0.0006  
## 80 0.7154 nan 0.1000 0.0008  
## 100 0.6846 nan 0.1000 0.0003  
## 120 0.6628 nan 0.1000 -0.0001  
## 140 0.6464 nan 0.1000 -0.0001  
## 150 0.6376 nan 0.1000 0.0006  
##   
## - Fold4.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold4.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2952 nan 0.1000 0.0312  
## 2 1.2420 nan 0.1000 0.0260  
## 3 1.1975 nan 0.1000 0.0218  
## 4 1.1571 nan 0.1000 0.0201  
## 5 1.1225 nan 0.1000 0.0162  
## 6 1.0876 nan 0.1000 0.0167  
## 7 1.0587 nan 0.1000 0.0150  
## 8 1.0367 nan 0.1000 0.0103  
## 9 1.0159 nan 0.1000 0.0100  
## 10 0.9963 nan 0.1000 0.0091  
## 20 0.8493 nan 0.1000 0.0038  
## 40 0.7421 nan 0.1000 0.0010  
## 60 0.6857 nan 0.1000 0.0008  
## 80 0.6458 nan 0.1000 0.0008  
## 100 0.6153 nan 0.1000 -0.0002  
## 120 0.5963 nan 0.1000 -0.0002  
## 140 0.5786 nan 0.1000 -0.0002  
## 150 0.5699 nan 0.1000 -0.0001  
##   
## - Fold4.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold5.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3318 nan 0.1000 0.0143  
## 2 1.3103 nan 0.1000 0.0108  
## 3 1.2866 nan 0.1000 0.0107  
## 4 1.2687 nan 0.1000 0.0090  
## 5 1.2501 nan 0.1000 0.0090  
## 6 1.2332 nan 0.1000 0.0080  
## 7 1.2160 nan 0.1000 0.0086  
## 8 1.2019 nan 0.1000 0.0068  
## 9 1.1876 nan 0.1000 0.0076  
## 10 1.1752 nan 0.1000 0.0062  
## 20 1.0758 nan 0.1000 0.0034  
## 40 0.9531 nan 0.1000 0.0016  
## 60 0.8838 nan 0.1000 0.0008  
## 80 0.8455 nan 0.1000 0.0006  
## 100 0.8187 nan 0.1000 0.0005  
## 120 0.7981 nan 0.1000 0.0002  
## 140 0.7833 nan 0.1000 0.0001  
## 150 0.7768 nan 0.1000 -0.0001  
##   
## - Fold5.Rep1: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold5.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3026 nan 0.1000 0.0284  
## 2 1.2557 nan 0.1000 0.0228  
## 3 1.2258 nan 0.1000 0.0145  
## 4 1.1921 nan 0.1000 0.0171  
## 5 1.1627 nan 0.1000 0.0140  
## 6 1.1379 nan 0.1000 0.0127  
## 7 1.1191 nan 0.1000 0.0096  
## 8 1.0971 nan 0.1000 0.0101  
## 9 1.0783 nan 0.1000 0.0088  
## 10 1.0610 nan 0.1000 0.0082  
## 20 0.9292 nan 0.1000 0.0051  
## 40 0.8168 nan 0.1000 0.0012  
## 60 0.7631 nan 0.1000 0.0034  
## 80 0.7225 nan 0.1000 0.0002  
## 100 0.6937 nan 0.1000 0.0004  
## 120 0.6721 nan 0.1000 0.0003  
## 140 0.6544 nan 0.1000 0.0001  
## 150 0.6472 nan 0.1000 0.0001  
##   
## - Fold5.Rep1: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold5.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2932 nan 0.1000 0.0333  
## 2 1.2391 nan 0.1000 0.0268  
## 3 1.1933 nan 0.1000 0.0213  
## 4 1.1495 nan 0.1000 0.0209  
## 5 1.1160 nan 0.1000 0.0156  
## 6 1.0805 nan 0.1000 0.0172  
## 7 1.0553 nan 0.1000 0.0120  
## 8 1.0316 nan 0.1000 0.0106  
## 9 1.0101 nan 0.1000 0.0100  
## 10 0.9879 nan 0.1000 0.0103  
## 20 0.8487 nan 0.1000 0.0035  
## 40 0.7371 nan 0.1000 0.0013  
## 60 0.6859 nan 0.1000 -0.0001  
## 80 0.6475 nan 0.1000 0.0005  
## 100 0.6202 nan 0.1000 0.0001  
## 120 0.5942 nan 0.1000 0.0006  
## 140 0.5746 nan 0.1000 0.0002  
## 150 0.5664 nan 0.1000 0.0003  
##   
## - Fold5.Rep1: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold1.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3312 nan 0.1000 0.0134  
## 2 1.3082 nan 0.1000 0.0115  
## 3 1.2875 nan 0.1000 0.0097  
## 4 1.2657 nan 0.1000 0.0105  
## 5 1.2466 nan 0.1000 0.0089  
## 6 1.2291 nan 0.1000 0.0080  
## 7 1.2135 nan 0.1000 0.0076  
## 8 1.1994 nan 0.1000 0.0067  
## 9 1.1860 nan 0.1000 0.0061  
## 10 1.1730 nan 0.1000 0.0060  
## 20 1.0733 nan 0.1000 0.0047  
## 40 0.9517 nan 0.1000 0.0019  
## 60 0.8835 nan 0.1000 0.0012  
## 80 0.8442 nan 0.1000 0.0005  
## 100 0.8180 nan 0.1000 0.0002  
## 120 0.7994 nan 0.1000 0.0003  
## 140 0.7862 nan 0.1000 -0.0001  
## 150 0.7797 nan 0.1000 0.0001  
##   
## - Fold1.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold1.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3031 nan 0.1000 0.0281  
## 2 1.2566 nan 0.1000 0.0222  
## 3 1.2187 nan 0.1000 0.0178  
## 4 1.1876 nan 0.1000 0.0146  
## 5 1.1619 nan 0.1000 0.0119  
## 6 1.1314 nan 0.1000 0.0146  
## 7 1.1109 nan 0.1000 0.0095  
## 8 1.0883 nan 0.1000 0.0111  
## 9 1.0679 nan 0.1000 0.0098  
## 10 1.0495 nan 0.1000 0.0086  
## 20 0.9259 nan 0.1000 0.0041  
## 40 0.8166 nan 0.1000 0.0016  
## 60 0.7674 nan 0.1000 0.0003  
## 80 0.7319 nan 0.1000 0.0003  
## 100 0.7047 nan 0.1000 0.0002  
## 120 0.6778 nan 0.1000 0.0001  
## 140 0.6617 nan 0.1000 0.0006  
## 150 0.6538 nan 0.1000 -0.0002  
##   
## - Fold1.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold1.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2957 nan 0.1000 0.0326  
## 2 1.2422 nan 0.1000 0.0271  
## 3 1.1926 nan 0.1000 0.0239  
## 4 1.1543 nan 0.1000 0.0184  
## 5 1.1214 nan 0.1000 0.0156  
## 6 1.0916 nan 0.1000 0.0144  
## 7 1.0639 nan 0.1000 0.0128  
## 8 1.0393 nan 0.1000 0.0111  
## 9 1.0142 nan 0.1000 0.0125  
## 10 0.9893 nan 0.1000 0.0113  
## 20 0.8543 nan 0.1000 0.0039  
## 40 0.7456 nan 0.1000 0.0019  
## 60 0.6921 nan 0.1000 -0.0002  
## 80 0.6518 nan 0.1000 -0.0001  
## 100 0.6275 nan 0.1000 0.0003  
## 120 0.6046 nan 0.1000 0.0001  
## 140 0.5811 nan 0.1000 0.0003  
## 150 0.5717 nan 0.1000 0.0001  
##   
## - Fold1.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold2.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3311 nan 0.1000 0.0149  
## 2 1.3077 nan 0.1000 0.0120  
## 3 1.2841 nan 0.1000 0.0110  
## 4 1.2656 nan 0.1000 0.0089  
## 5 1.2461 nan 0.1000 0.0085  
## 6 1.2300 nan 0.1000 0.0079  
## 7 1.2113 nan 0.1000 0.0081  
## 8 1.1978 nan 0.1000 0.0063  
## 9 1.1845 nan 0.1000 0.0068  
## 10 1.1718 nan 0.1000 0.0057  
## 20 1.0709 nan 0.1000 0.0049  
## 40 0.9510 nan 0.1000 0.0027  
## 60 0.8834 nan 0.1000 0.0016  
## 80 0.8424 nan 0.1000 0.0007  
## 100 0.8168 nan 0.1000 0.0001  
## 120 0.7964 nan 0.1000 0.0001  
## 140 0.7793 nan 0.1000 0.0003  
## 150 0.7719 nan 0.1000 0.0001  
##   
## - Fold2.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold2.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3045 nan 0.1000 0.0274  
## 2 1.2580 nan 0.1000 0.0235  
## 3 1.2187 nan 0.1000 0.0195  
## 4 1.1879 nan 0.1000 0.0146  
## 5 1.1580 nan 0.1000 0.0149  
## 6 1.1338 nan 0.1000 0.0121  
## 7 1.1035 nan 0.1000 0.0148  
## 8 1.0871 nan 0.1000 0.0073  
## 9 1.0676 nan 0.1000 0.0090  
## 10 1.0516 nan 0.1000 0.0073  
## 20 0.9222 nan 0.1000 0.0040  
## 40 0.8125 nan 0.1000 0.0014  
## 60 0.7537 nan 0.1000 0.0027  
## 80 0.7147 nan 0.1000 0.0006  
## 100 0.6794 nan 0.1000 0.0003  
## 120 0.6572 nan 0.1000 0.0002  
## 140 0.6391 nan 0.1000 0.0002  
## 150 0.6306 nan 0.1000 0.0002  
##   
## - Fold2.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold2.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2960 nan 0.1000 0.0320  
## 2 1.2433 nan 0.1000 0.0260  
## 3 1.1985 nan 0.1000 0.0220  
## 4 1.1603 nan 0.1000 0.0184  
## 5 1.1267 nan 0.1000 0.0160  
## 6 1.0990 nan 0.1000 0.0138  
## 7 1.0729 nan 0.1000 0.0126  
## 8 1.0424 nan 0.1000 0.0146  
## 9 1.0188 nan 0.1000 0.0121  
## 10 1.0019 nan 0.1000 0.0082  
## 20 0.8598 nan 0.1000 0.0044  
## 40 0.7386 nan 0.1000 0.0015  
## 60 0.6779 nan 0.1000 0.0004  
## 80 0.6426 nan 0.1000 0.0004  
## 100 0.6123 nan 0.1000 0.0007  
## 120 0.5903 nan 0.1000 0.0000  
## 140 0.5690 nan 0.1000 0.0007  
## 150 0.5608 nan 0.1000 -0.0002  
##   
## - Fold2.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold3.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3319 nan 0.1000 0.0142  
## 2 1.3090 nan 0.1000 0.0107  
## 3 1.2863 nan 0.1000 0.0102  
## 4 1.2670 nan 0.1000 0.0095  
## 5 1.2482 nan 0.1000 0.0093  
## 6 1.2317 nan 0.1000 0.0079  
## 7 1.2154 nan 0.1000 0.0078  
## 8 1.2019 nan 0.1000 0.0065  
## 9 1.1876 nan 0.1000 0.0069  
## 10 1.1751 nan 0.1000 0.0057  
## 20 1.0716 nan 0.1000 0.0041  
## 40 0.9472 nan 0.1000 0.0022  
## 60 0.8778 nan 0.1000 0.0013  
## 80 0.8356 nan 0.1000 0.0008  
## 100 0.8083 nan 0.1000 0.0004  
## 120 0.7897 nan 0.1000 0.0001  
## 140 0.7755 nan 0.1000 0.0002  
## 150 0.7679 nan 0.1000 0.0007  
##   
## - Fold3.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold3.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3046 nan 0.1000 0.0281  
## 2 1.2590 nan 0.1000 0.0230  
## 3 1.2209 nan 0.1000 0.0184  
## 4 1.1886 nan 0.1000 0.0164  
## 5 1.1609 nan 0.1000 0.0133  
## 6 1.1349 nan 0.1000 0.0120  
## 7 1.1117 nan 0.1000 0.0112  
## 8 1.0811 nan 0.1000 0.0153  
## 9 1.0624 nan 0.1000 0.0087  
## 10 1.0464 nan 0.1000 0.0073  
## 20 0.9170 nan 0.1000 0.0041  
## 40 0.8054 nan 0.1000 0.0015  
## 60 0.7529 nan 0.1000 0.0001  
## 80 0.7211 nan 0.1000 0.0004  
## 100 0.6933 nan 0.1000 0.0003  
## 120 0.6656 nan 0.1000 0.0012  
## 140 0.6467 nan 0.1000 0.0008  
## 150 0.6379 nan 0.1000 0.0001  
##   
## - Fold3.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold3.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2961 nan 0.1000 0.0323  
## 2 1.2435 nan 0.1000 0.0262  
## 3 1.2020 nan 0.1000 0.0202  
## 4 1.1574 nan 0.1000 0.0211  
## 5 1.1222 nan 0.1000 0.0175  
## 6 1.0935 nan 0.1000 0.0147  
## 7 1.0608 nan 0.1000 0.0151  
## 8 1.0325 nan 0.1000 0.0141  
## 9 1.0111 nan 0.1000 0.0096  
## 10 0.9920 nan 0.1000 0.0092  
## 20 0.8521 nan 0.1000 0.0067  
## 40 0.7363 nan 0.1000 0.0008  
## 60 0.6818 nan 0.1000 0.0004  
## 80 0.6458 nan 0.1000 0.0002  
## 100 0.6186 nan 0.1000 -0.0000  
## 120 0.5959 nan 0.1000 -0.0000  
## 140 0.5752 nan 0.1000 0.0002  
## 150 0.5663 nan 0.1000 0.0002  
##   
## - Fold3.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold4.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3309 nan 0.1000 0.0147  
## 2 1.3081 nan 0.1000 0.0116  
## 3 1.2864 nan 0.1000 0.0107  
## 4 1.2677 nan 0.1000 0.0086  
## 5 1.2486 nan 0.1000 0.0098  
## 6 1.2331 nan 0.1000 0.0075  
## 7 1.2158 nan 0.1000 0.0085  
## 8 1.2007 nan 0.1000 0.0068  
## 9 1.1858 nan 0.1000 0.0072  
## 10 1.1727 nan 0.1000 0.0060  
## 20 1.0723 nan 0.1000 0.0044  
## 40 0.9549 nan 0.1000 0.0019  
## 60 0.8869 nan 0.1000 0.0011  
## 80 0.8453 nan 0.1000 0.0007  
## 100 0.8178 nan 0.1000 0.0004  
## 120 0.7975 nan 0.1000 0.0002  
## 140 0.7819 nan 0.1000 0.0002  
## 150 0.7752 nan 0.1000 -0.0001  
##   
## - Fold4.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold4.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3038 nan 0.1000 0.0270  
## 2 1.2573 nan 0.1000 0.0230  
## 3 1.2200 nan 0.1000 0.0183  
## 4 1.1902 nan 0.1000 0.0149  
## 5 1.1622 nan 0.1000 0.0128  
## 6 1.1359 nan 0.1000 0.0128  
## 7 1.1063 nan 0.1000 0.0143  
## 8 1.0844 nan 0.1000 0.0109  
## 9 1.0682 nan 0.1000 0.0077  
## 10 1.0505 nan 0.1000 0.0083  
## 20 0.9259 nan 0.1000 0.0048  
## 40 0.8163 nan 0.1000 0.0013  
## 60 0.7670 nan 0.1000 0.0006  
## 80 0.7217 nan 0.1000 0.0001  
## 100 0.6919 nan 0.1000 0.0004  
## 120 0.6712 nan 0.1000 0.0001  
## 140 0.6539 nan 0.1000 -0.0004  
## 150 0.6446 nan 0.1000 0.0001  
##   
## - Fold4.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold4.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2976 nan 0.1000 0.0322  
## 2 1.2455 nan 0.1000 0.0254  
## 3 1.1957 nan 0.1000 0.0243  
## 4 1.1529 nan 0.1000 0.0205  
## 5 1.1189 nan 0.1000 0.0165  
## 6 1.0899 nan 0.1000 0.0139  
## 7 1.0632 nan 0.1000 0.0126  
## 8 1.0408 nan 0.1000 0.0104  
## 9 1.0141 nan 0.1000 0.0124  
## 10 0.9935 nan 0.1000 0.0099  
## 20 0.8537 nan 0.1000 0.0042  
## 40 0.7500 nan 0.1000 0.0011  
## 60 0.6908 nan 0.1000 0.0001  
## 80 0.6557 nan 0.1000 0.0002  
## 100 0.6266 nan 0.1000 0.0004  
## 120 0.6026 nan 0.1000 -0.0001  
## 140 0.5830 nan 0.1000 0.0005  
## 150 0.5740 nan 0.1000 -0.0003  
##   
## - Fold4.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## + Fold5.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3336 nan 0.1000 0.0147  
## 2 1.3114 nan 0.1000 0.0116  
## 3 1.2903 nan 0.1000 0.0102  
## 4 1.2716 nan 0.1000 0.0087  
## 5 1.2555 nan 0.1000 0.0073  
## 6 1.2380 nan 0.1000 0.0088  
## 7 1.2235 nan 0.1000 0.0069  
## 8 1.2092 nan 0.1000 0.0069  
## 9 1.1961 nan 0.1000 0.0062  
## 10 1.1846 nan 0.1000 0.0053  
## 20 1.0853 nan 0.1000 0.0042  
## 40 0.9645 nan 0.1000 0.0012  
## 60 0.8952 nan 0.1000 0.0008  
## 80 0.8548 nan 0.1000 0.0006  
## 100 0.8283 nan 0.1000 0.0007  
## 120 0.8075 nan 0.1000 0.0002  
## 140 0.7909 nan 0.1000 0.0001  
## 150 0.7848 nan 0.1000 0.0001  
##   
## - Fold5.Rep2: shrinkage=0.1, interaction.depth=1, n.minobsinnode=10, n.trees=150   
## + Fold5.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.3071 nan 0.1000 0.0276  
## 2 1.2639 nan 0.1000 0.0214  
## 3 1.2313 nan 0.1000 0.0160  
## 4 1.1997 nan 0.1000 0.0156  
## 5 1.1724 nan 0.1000 0.0137  
## 6 1.1492 nan 0.1000 0.0111  
## 7 1.1275 nan 0.1000 0.0102  
## 8 1.1062 nan 0.1000 0.0107  
## 9 1.0890 nan 0.1000 0.0082  
## 10 1.0742 nan 0.1000 0.0066  
## 20 0.9330 nan 0.1000 0.0055  
## 40 0.8227 nan 0.1000 0.0017  
## 60 0.7718 nan 0.1000 0.0009  
## 80 0.7365 nan 0.1000 -0.0000  
## 100 0.7079 nan 0.1000 -0.0001  
## 120 0.6799 nan 0.1000 0.0005  
## 140 0.6606 nan 0.1000 0.0002  
## 150 0.6521 nan 0.1000 -0.0001  
##   
## - Fold5.Rep2: shrinkage=0.1, interaction.depth=2, n.minobsinnode=10, n.trees=150   
## + Fold5.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2956 nan 0.1000 0.0306  
## 2 1.2441 nan 0.1000 0.0253  
## 3 1.1970 nan 0.1000 0.0236  
## 4 1.1595 nan 0.1000 0.0178  
## 5 1.1298 nan 0.1000 0.0141  
## 6 1.0975 nan 0.1000 0.0155  
## 7 1.0691 nan 0.1000 0.0133  
## 8 1.0398 nan 0.1000 0.0148  
## 9 1.0121 nan 0.1000 0.0139  
## 10 0.9928 nan 0.1000 0.0086  
## 20 0.8566 nan 0.1000 0.0034  
## 40 0.7519 nan 0.1000 0.0005  
## 60 0.6907 nan 0.1000 0.0008  
## 80 0.6519 nan 0.1000 0.0005  
## 100 0.6281 nan 0.1000 0.0001  
## 120 0.6069 nan 0.1000 -0.0000  
## 140 0.5880 nan 0.1000 -0.0001  
## 150 0.5771 nan 0.1000 -0.0001  
##   
## - Fold5.Rep2: shrinkage=0.1, interaction.depth=3, n.minobsinnode=10, n.trees=150   
## Aggregating results  
## Selecting tuning parameters  
## Fitting n.trees = 150, interaction.depth = 3, shrinkage = 0.1, n.minobsinnode = 10 on full training set  
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2983 nan 0.1000 0.0319  
## 2 1.2460 nan 0.1000 0.0261  
## 3 1.2028 nan 0.1000 0.0209  
## 4 1.1579 nan 0.1000 0.0215  
## 5 1.1236 nan 0.1000 0.0161  
## 6 1.0947 nan 0.1000 0.0134  
## 7 1.0704 nan 0.1000 0.0119  
## 8 1.0437 nan 0.1000 0.0128  
## 9 1.0201 nan 0.1000 0.0118  
## 10 0.9943 nan 0.1000 0.0121  
## 20 0.8547 nan 0.1000 0.0033  
## 40 0.7478 nan 0.1000 0.0010  
## 60 0.6966 nan 0.1000 0.0006  
## 80 0.6549 nan 0.1000 0.0004  
## 100 0.6277 nan 0.1000 0.0007  
## 120 0.6055 nan 0.1000 -0.0001  
## 140 0.5860 nan 0.1000 0.0001  
## 150 0.5782 nan 0.1000 0.0002

model2 <- train(IsAlert ~ .,   
 data = modelTrain[,!(names(modelTrain) %in% c("TrialID","ObsNum","P8","V7","V9"))],   
 method = "xgbLinear",   
 metric = "ROC",  
 na.action = na.pass,  
 preProcess = c("center", "scale", "medianImpute"),  
 trControl = control)

## + Fold1.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold1.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold1.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold2.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold2.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold3.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold3.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold4.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold4.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold5.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold5.Rep1: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold1.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold1.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold2.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold2.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold3.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold3.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold4.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold4.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=0e+00, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=1e-01, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=1e-04, nrounds= 50, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=0e+00, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=1e-01, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=1e-04, nrounds=100, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=0e+00, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=1e-01, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=0e+00, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=1e-01, alpha=1e-04, nrounds=150, eta=0.3   
## + Fold5.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## - Fold5.Rep2: lambda=1e-04, alpha=1e-04, nrounds=150, eta=0.3   
## Aggregating results  
## Selecting tuning parameters  
## Fitting nrounds = 150, lambda = 1e-04, alpha = 1e-04, eta = 0.3 on full training set

# Check out the hyperparameters   
print(model1)

## Stochastic Gradient Boosting   
##   
## 6044 samples  
## 44 predictor  
## 2 classes: 'N', 'Y'   
##   
## Pre-processing: centered (44), scaled (44), median imputation (44)   
## Resampling: Cross-Validated (5 fold, repeated 2 times)   
## Summary of sample sizes: 4835, 4835, 4834, 4836, 4836, 4836, ...   
## Resampling results across tuning parameters:  
##   
## interaction.depth n.trees ROC Sens Spec   
## 1 50 0.8760738 0.6605665 0.9273497  
## 1 100 0.8872697 0.7402565 0.9167870  
## 1 150 0.8947422 0.7498998 0.9217807  
## 2 50 0.8960801 0.7475396 0.9197856  
## 2 100 0.9147293 0.7640676 0.9323450  
## 2 150 0.9223501 0.7805977 0.9360571  
## 3 50 0.9149274 0.7497034 0.9346281  
## 3 100 0.9278576 0.7861118 0.9393401  
## 3 150 0.9330859 0.8014599 0.9409105  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## ROC 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.

This is the model summary for GBM. The train function automatically tunes the hyperparameters based on the largest value of ROC.  
For a gradient boosting machine (gbm) model, the main tuning parameters are:

* number of iterations, i.e. trees, (called n.trees in the gbm function)
* complexity of the tree, called interaction.depth
* learning rate: how quickly the algorithm adapts, called shrinkage
* the minimum number of training set samples in a node to commence splitting (n.minobsinnode)

# Check out the hyperparameters   
print(model2)

## eXtreme Gradient Boosting   
##   
## 6044 samples  
## 44 predictor  
## 2 classes: 'N', 'Y'   
##   
## Pre-processing: centered (44), scaled (44), median imputation (44)   
## Resampling: Cross-Validated (5 fold, repeated 2 times)   
## Summary of sample sizes: 4836, 4835, 4835, 4835, 4835, 4836, ...   
## Resampling results across tuning parameters:  
##   
## lambda alpha nrounds ROC Sens Spec   
## 0e+00 0e+00 50 0.9526953 0.8541965 0.9451922  
## 0e+00 0e+00 100 0.9532580 0.8610832 0.9446203  
## 0e+00 0e+00 150 0.9539030 0.8632474 0.9440495  
## 0e+00 1e-04 50 0.9532925 0.8541969 0.9457620  
## 0e+00 1e-04 100 0.9543453 0.8593131 0.9464752  
## 0e+00 1e-04 150 0.9547411 0.8638379 0.9453334  
## 0e+00 1e-01 50 0.9519555 0.8536091 0.9444781  
## 0e+00 1e-01 100 0.9535976 0.8634438 0.9489018  
## 0e+00 1e-01 150 0.9538955 0.8632474 0.9483322  
## 1e-04 0e+00 50 0.9529493 0.8536067 0.9460485  
## 1e-04 0e+00 100 0.9537181 0.8600989 0.9464752  
## 1e-04 0e+00 150 0.9542427 0.8628517 0.9457612  
## 1e-04 1e-04 50 0.9531566 0.8541965 0.9454767  
## 1e-04 1e-04 100 0.9546539 0.8583273 0.9459032  
## 1e-04 1e-04 150 0.9551522 0.8632435 0.9450477  
## 1e-04 1e-01 50 0.9519327 0.8536091 0.9454781  
## 1e-04 1e-01 100 0.9536588 0.8634450 0.9493299  
## 1e-04 1e-01 150 0.9541307 0.8608871 0.9484744  
## 1e-01 0e+00 50 0.9533846 0.8504625 0.9481912  
## 1e-01 0e+00 100 0.9544416 0.8587256 0.9496163  
## 1e-01 0e+00 150 0.9545839 0.8601001 0.9511863  
## 1e-01 1e-04 50 0.9534721 0.8508562 0.9486177  
## 1e-01 1e-04 100 0.9538827 0.8589221 0.9493301  
## 1e-01 1e-04 150 0.9539742 0.8595080 0.9511842  
## 1e-01 1e-01 50 0.9535594 0.8508504 0.9466207  
## 1e-01 1e-01 100 0.9546643 0.8606879 0.9474754  
## 1e-01 1e-01 150 0.9548454 0.8661970 0.9476185  
##   
## Tuning parameter 'eta' was held constant at a value of 0.3  
## ROC was used to select the optimal model using the largest value.  
## The final values used for the model were nrounds = 150, lambda =  
## 1e-04, alpha = 1e-04 and eta = 0.3.

This is the model summary for XGBoost. The train function automatically tunes the hyperparameters based on the largest value of ROC.

For an extreme gradient boosting (xbmLinear) model, the main tuning parameters are:

* the max number of iterations: nrounds
* L2 regularization term on weights: lambda
* L1 regularization term on weights: alpha
* step size of each boosting step: eta

## Results and Interpretation

**ROC as Evaluation metric**

Area under the ROC curve (AUC) is used for flexibility in deciding between minimizing the false positive rate & maximizing the true positive rate. ROC is also robust against class label imbalance (43:57 for this dataset). It is a commonly used evaluation method for binary outcome problems that involve classifying an instance as either positive or negative. Its main advantages over other evaluation methods, such as the simpler misclassification error, are:

* It is insensitive to unbalanced datasets.
* For other evaluation methods, a user has to choose a cut-off point above which the target variable is part of the positive class (e.g. a logistic regression model returns any real number between 0 and 1 - the modeller might decide that predictions greater than 0.5 mean a positive class prediction while a prediction of less than 0.5 mean a negative class prediction). AUC evaluates entries at all cut-off points, giving better insight into how well the classifier is able to separate the two classes.

The key metric used in the interpretation of results is the accuracy computed in the confusion matrix and area under the ROC curves computed using the twoClassSummary function. See appendix for complete ROC curves.

We have got a separate dataset which we have used to validate our trained models. The validation dataset is completely new dataset for the models as it was never used in training the models. The training and test dataset were derived from the original dataset using createDataPartition function of the caret package and were used to train the models. The first set of results are from the test dataset derived from original dataset and second set of results are the real tests on the validation dataset which consists of 120840 observation having the same number and type of attributes as the original dataset.

The second set of results show that we have over fitted our models since the accuracy and the AUC was significantly dropped when the model is tested on an entirely new dataset. However, based on the results of validation dataset, we can conclude that the extreme gradient boost model performed better than the gradient boost machine.

# Recommended Classifier

Gradient Boosting machines are ensemble models with the goal to build a series of under fitted (unlike random forest’s over-fitted) models, each reducing the errors of previous model where cumulative prediction is used to make the final prediction (Mayr et al, 2014).

A specific, open-source Extreme Gradient Boosting Model that is fast, scalable and produces state-of-the-art results on a wide range of problems (Chen & Guestrin, 2016) is the recommend classifier for the driver alertness problem. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting(also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment(Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

The recommendation is based on the key metric of Area under the ROC curve where XGBoost with complete set of variables as predictors stands out as the best model. Although the GBM performed at par with XGBoost if we look at the accuracy, yet the XBBoost’s area under the curve is better than GBM.

It should be noted that the recommendation made is based on the two models selected for the current experimentation and other classification model were not tried and tested. Also, due to lack high end resources the training dataset was chosen to be very small as choosing a larger training dataset was slowing down the personal laptop and consuming a considerable amount of time.