

Predicting Sales Volume for 4 Different Product Types

Jennifer Brosnahan

7/10/2020

The purpose of this project is to predict sales of four different product types and assess the impact of service reviews and customer reviews have on sales.

Target variable: 'Volume' for the product types: PC, Laptops, Netbooks, and Smartphones

Loading packages

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1
```

```
## v ggplot2 3.3.1    v purrr  0.3.4
## v tibble  3.0.1    v dplyr  1.0.0
## v tidyr   1.1.0    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.5.0
```

```
## -- Conflicts ----- tidyverse_conflic
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

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

```
##
```

```
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
## lift
```

```
library(ggplot2)
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(openxlsx)
library(h2o)
```

```
##
## -----
##
## Your next step is to start H2O:
##   > h2o.init()
##
## For H2O package documentation, ask for help:
##   > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit http://docs.h2o.ai
##
## -----

##
## Attaching package: 'h2o'

## The following objects are masked from 'package:stats':
##
##   cor, sd, var

## The following objects are masked from 'package:base':
##
##   %*%, %in%, &&, ||, apply, as.factor, as.numeric, colnames,
##   colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##   log10, log1p, log2, round, signif, trunc
```

Importing data

```
existing <- read.csv(file.path('C:/Users/jlbrow/OneDrive/C3T3', 'existing.csv'),
  stringsAsFactors = TRUE)
```

Checking structure

```
str(existing)
```

```
## 'data.frame':   80 obs. of  18 variables:
## $ ProductType      : Factor w/ 12 levels "Accessories",...: 7 7 7 5 5 1 1 1 1 1 ...
## $ ProductNum       : int   101 102 103 104 105 106 107 108 109 110 ...
## $ Price            : num   949 2250 399 410 1080 ...
## $ x5StarReviews    : int    3 2 3 49 58 83 11 33 16 10 ...
## $ x4StarReviews    : int    3 1 0 19 31 30 3 19 9 1 ...
## $ x3StarReviews    : int    2 0 0 8 11 10 0 12 2 1 ...
## $ x2StarReviews    : int    0 0 0 3 7 9 0 5 0 0 ...
## $ x1StarReviews    : int    0 0 0 9 36 40 1 9 2 0 ...
```

```
## $ PositiveServiceReview: int 2 1 1 7 7 12 3 5 2 2 ...
## $ NegativeServiceReview: int 0 0 0 8 20 5 0 3 1 0 ...
## $ Recommendproduct      : num 0.9 0.9 0.9 0.8 0.7 0.3 0.9 0.7 0.8 0.9 ...
## $ BestSellersRank       : int 1967 4806 12076 109 268 64 NA 2 NA 18 ...
## $ ShippingWeight        : num 25.8 50 17.4 5.7 7 1.6 7.3 12 1.8 0.75 ...
## $ ProductDepth          : num 23.9 35 10.5 15 12.9 ...
## $ ProductWidth          : num 6.62 31.75 8.3 9.9 0.3 ...
## $ ProductHeight         : num 16.9 19 10.2 1.3 8.9 ...
## $ ProfitMargin          : num 0.15 0.25 0.08 0.08 0.09 0.05 0.05 0.05 0.05 0.05 ...
## $ Volume                : int 12 8 12 196 232 332 44 132 64 40 ...
```

Because regression algorithms can easily misinterpret categorical variables in which there are more than 2 values, we will dummify categorical data for regression modeling to binarize the values.

```
existingDummy <- dummyVars('~ .', data = existing)
existing2 <- data.frame(predict(existingDummy, newdata = existing))
```

Check structure again

```
str(existing2)
```

```
## 'data.frame': 80 obs. of 29 variables:
## $ ProductType.Accessories : num 0 0 0 0 0 1 1 1 1 1 ...
## $ ProductType.Display : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.ExtendedWarranty: num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.GameConsole : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Laptop : num 0 0 0 1 1 0 0 0 0 0 ...
## $ ProductType.Netbook : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.PC : num 1 1 1 0 0 0 0 0 0 0 ...
## $ ProductType.Printer : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.PrinterSupplies : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Smartphone : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Software : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Tablet : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductNum : num 101 102 103 104 105 106 107 108 109 110 ...
## $ Price : num 949 2250 399 410 1080 ...
## $ x5StarReviews : num 3 2 3 49 58 83 11 33 16 10 ...
## $ x4StarReviews : num 3 1 0 19 31 30 3 19 9 1 ...
## $ x3StarReviews : num 2 0 0 8 11 10 0 12 2 1 ...
## $ x2StarReviews : num 0 0 0 3 7 9 0 5 0 0 ...
## $ x1StarReviews : num 0 0 0 9 36 40 1 9 2 0 ...
## $ PositiveServiceReview : num 2 1 1 7 7 12 3 5 2 2 ...
## $ NegativeServiceReview : num 0 0 0 8 20 5 0 3 1 0 ...
## $ Recommendproduct : num 0.9 0.9 0.9 0.8 0.7 0.3 0.9 0.7 0.8 0.9 ...
## $ BestSellersRank : num 1967 4806 12076 109 268 ...
## $ ShippingWeight : num 25.8 50 17.4 5.7 7 1.6 7.3 12 1.8 0.75 ...
## $ ProductDepth : num 23.9 35 10.5 15 12.9 ...
## $ ProductWidth : num 6.62 31.75 8.3 9.9 0.3 ...
## $ ProductHeight : num 16.9 19 10.2 1.3 8.9 ...
```

```
## $ ProfitMargin          : num  0.15 0.25 0.08 0.08 0.09 0.05 0.05 0.05 0.05 0.05 ...
## $ Volume                : num  12 8 12 196 232 332 44 132 64 40 ...
```

Check summary for descriptive and NAs

```
summary(existing2)
```

```
## ProductType.Accessories ProductType.Display ProductType.ExtendedWarranty
## Min.      :0.000          Min.      :0.0000          Min.      :0.000
## 1st Qu.:0.000          1st Qu.:0.0000          1st Qu.:0.000
## Median :0.000          Median :0.0000          Median :0.000
## Mean    :0.325          Mean    :0.0625          Mean    :0.125
## 3rd Qu.:1.000          3rd Qu.:0.0000          3rd Qu.:0.000
## Max.    :1.000          Max.    :1.0000          Max.    :1.000
##
## ProductType.GameConsole ProductType.Laptop ProductType.Netbook ProductType.PC
## Min.      :0.000          Min.      :0.0000          Min.      :0.000          Min.      :0.00
## 1st Qu.:0.000          1st Qu.:0.0000          1st Qu.:0.000          1st Qu.:0.00
## Median :0.000          Median :0.0000          Median :0.000          Median :0.00
## Mean    :0.025          Mean    :0.0375          Mean    :0.025          Mean    :0.05
## 3rd Qu.:0.000          3rd Qu.:0.0000          3rd Qu.:0.000          3rd Qu.:0.00
## Max.    :1.000          Max.    :1.0000          Max.    :1.000          Max.    :1.00
##
## ProductType.Printer ProductType.PrinterSupplies ProductType.Smartphone
## Min.      :0.00          Min.      :0.0000          Min.      :0.00
## 1st Qu.:0.00          1st Qu.:0.0000          1st Qu.:0.00
## Median :0.00          Median :0.0000          Median :0.00
## Mean    :0.15          Mean    :0.0375          Mean    :0.05
## 3rd Qu.:0.00          3rd Qu.:0.0000          3rd Qu.:0.00
## Max.    :1.00          Max.    :1.0000          Max.    :1.00
##
## ProductType.Software ProductType.Tablet ProductNum Price
## Min.      :0.000          Min.      :0.0000          Min.      :101.0          Min.      : 3.60
## 1st Qu.:0.000          1st Qu.:0.0000          1st Qu.:120.8          1st Qu.: 52.66
## Median :0.000          Median :0.0000          Median :140.5          Median :132.72
## Mean    :0.075          Mean    :0.0375          Mean    :142.6          Mean    :247.25
## 3rd Qu.:0.000          3rd Qu.:0.0000          3rd Qu.:160.2          3rd Qu.:352.49
## Max.    :1.000          Max.    :1.0000          Max.    :200.0          Max.    :2249.99
##
## x5StarReviews x4StarReviews x3StarReviews x2StarReviews
## Min.      : 0.0          Min.      : 0.00          Min.      : 0.00          Min.      : 0.00
## 1st Qu.: 10.0          1st Qu.: 2.75          1st Qu.: 2.00          1st Qu.: 1.00
## Median : 50.0          Median : 22.00          Median : 7.00          Median : 3.00
## Mean    :176.2          Mean    : 40.20          Mean    :14.79          Mean    :13.79
## 3rd Qu.:306.5          3rd Qu.: 33.00          3rd Qu.:11.25          3rd Qu.: 7.00
## Max.    :2801.0          Max.    :431.00          Max.    :162.00          Max.    :370.00
##
## x1StarReviews PositiveServiceReview NegativeServiceReview Recommendproduct
## Min.      : 0.00          Min.      : 0.00          Min.      : 0.000          Min.      :0.100
## 1st Qu.: 2.00          1st Qu.: 2.00          1st Qu.: 1.000          1st Qu.:0.700
## Median : 8.50          Median : 5.50          Median : 3.000          Median :0.800
## Mean    : 37.67          Mean    :51.75          Mean    : 6.225          Mean    :0.745
```

```
## 3rd Qu.: 15.25 3rd Qu.: 42.00 3rd Qu.: 6.250 3rd Qu.:0.900
## Max. :1654.00 Max. :536.00 Max. :112.000 Max. :1.000
##
## BestSellersRank ShippingWeight ProductDepth ProductWidth
## Min. : 1 Min. : 0.0100 Min. : 0.000 Min. : 0.000
## 1st Qu.: 7 1st Qu.: 0.5125 1st Qu.: 4.775 1st Qu.: 1.750
## Median : 27 Median : 2.1000 Median : 7.950 Median : 6.800
## Mean : 1126 Mean : 9.6681 Mean : 14.425 Mean : 7.819
## 3rd Qu.: 281 3rd Qu.:11.2050 3rd Qu.: 15.025 3rd Qu.:11.275
## Max. :17502 Max. :63.0000 Max. :300.000 Max. :31.750
## NA's :15
## ProductHeight ProfitMargin Volume
## Min. : 0.000 Min. :0.0500 Min. : 0
## 1st Qu.: 0.400 1st Qu.:0.0500 1st Qu.: 40
## Median : 3.950 Median :0.1200 Median : 200
## Mean : 6.259 Mean :0.1545 Mean : 705
## 3rd Qu.:10.300 3rd Qu.:0.2000 3rd Qu.: 1226
## Max. :25.800 Max. :0.4000 Max. :11204
##
```

Reveals 15 NA's for 'BestSellersRank'

Deleting BestSellersRank, only variable with NAs

```
existing2$BestSellersRank <- NULL
```

Correlation matrix of all variables

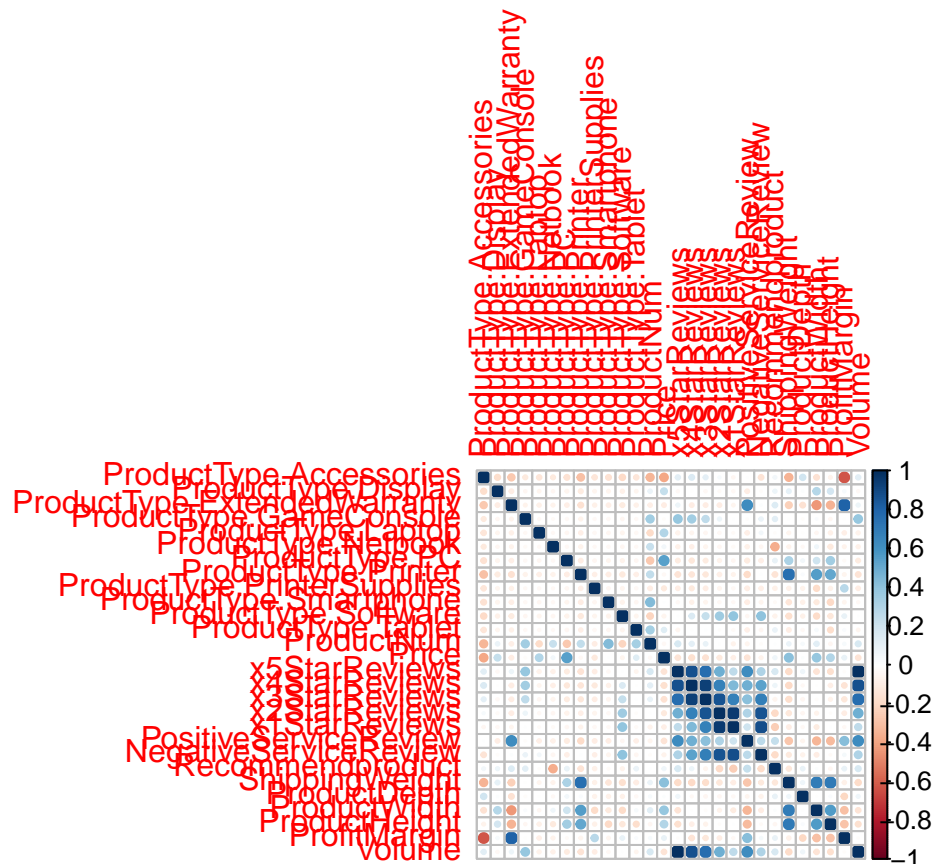
```
corrData <- cor(existing2)
```

Exporting correlation to excel

```
write.xlsx(corrData, file = "corrData.xlsx", row.names=TRUE)
write.xlsx(existing2, file = 'existing2.xlsx')
```

Viewing correlation heatmap, as you can see, it's unreadable with so many variables

```
corrplot(corrData)
```



Removing 5 Star since perfect correlation of 1 to target variable, risks overfitting

Also removing low correlated variables

```
existing3 <- subset(existing2, select = -c(1:4, 8:9, 11:12, 15, 24:27))
str(existing3)
```

```
## 'data.frame': 80 obs. of 15 variables:
## $ ProductType.Laptop : num 0 0 0 1 1 0 0 0 0 0 ...
## $ ProductType.Netbook : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.PC : num 1 1 1 0 0 0 0 0 0 0 ...
## $ ProductType.Smartphone: num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductNum : num 101 102 103 104 105 106 107 108 109 110 ...
## $ Price : num 949 2250 399 410 1080 ...
## $ x4StarReviews : num 3 1 0 19 31 30 3 19 9 1 ...
## $ x3StarReviews : num 2 0 0 8 11 10 0 12 2 1 ...
## $ x2StarReviews : num 0 0 0 3 7 9 0 5 0 0 ...
## $ x1StarReviews : num 0 0 0 9 36 40 1 9 2 0 ...
## $ PositiveServiceReview : num 2 1 1 7 7 12 3 5 2 2 ...
## $ NegativeServiceReview : num 0 0 0 8 20 5 0 3 1 0 ...
## $ Recommendproduct : num 0.9 0.9 0.9 0.8 0.7 0.3 0.9 0.7 0.8 0.9 ...
## $ ShippingWeight : num 25.8 50 17.4 5.7 7 1.6 7.3 12 1.8 0.75 ...
## $ Volume : num 12 8 12 196 232 332 44 132 64 40 ...
```

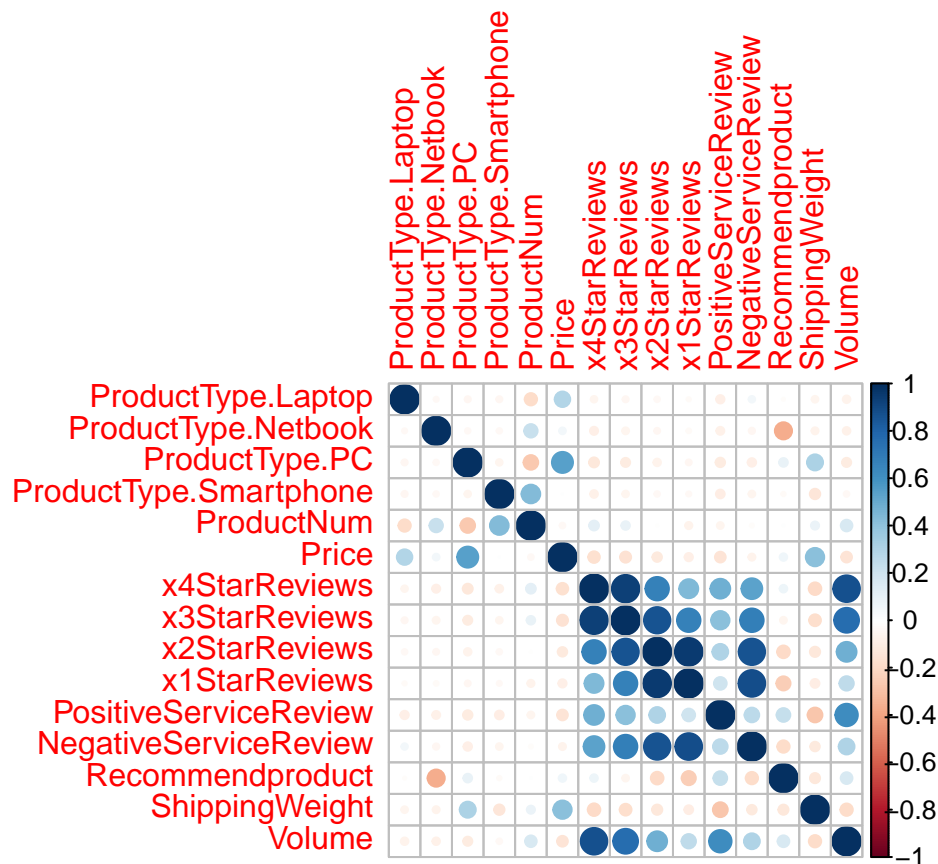
Transmute new column that is average of all Star Reviews to try as form of PCA

```
existing4 <- existing3 %>%
  rowwise() %>% mutate(AvgStarReviews = (mean(c(x4StarReviews, x3StarReviews, x2StarReviews, x1StarReviews))))
```

EDA

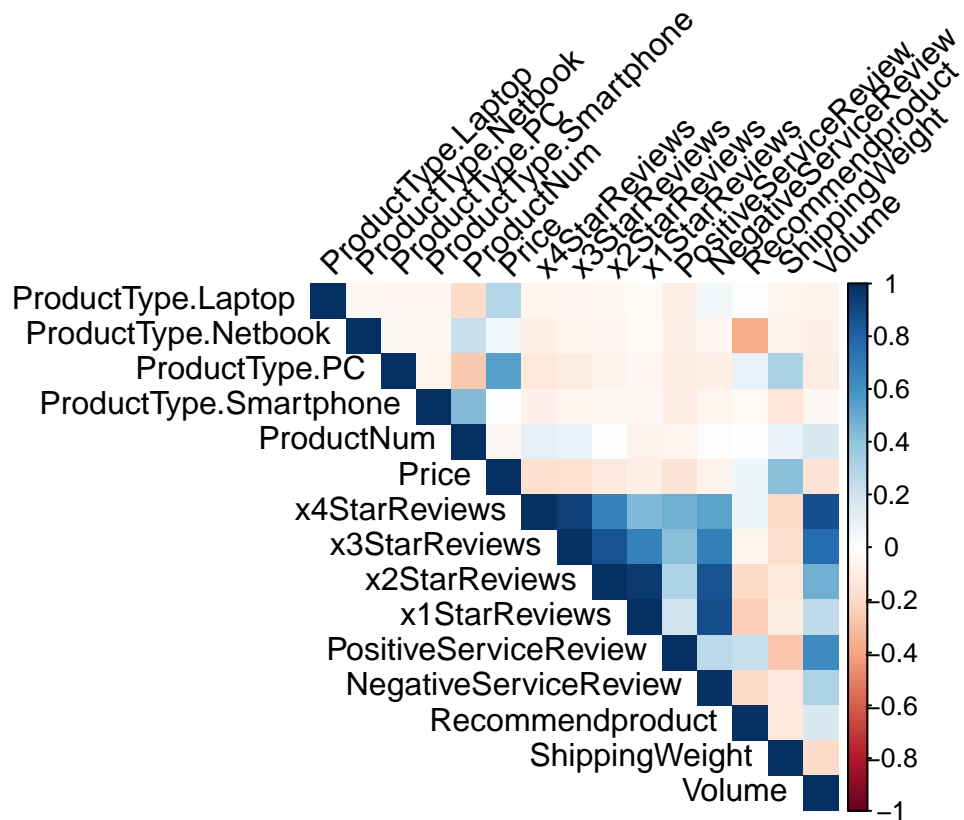
viewing correlation heatmap

```
corrData3 <- cor(existing3)
corrplot(corrData3)
```



Enhancing the correlation heatmap

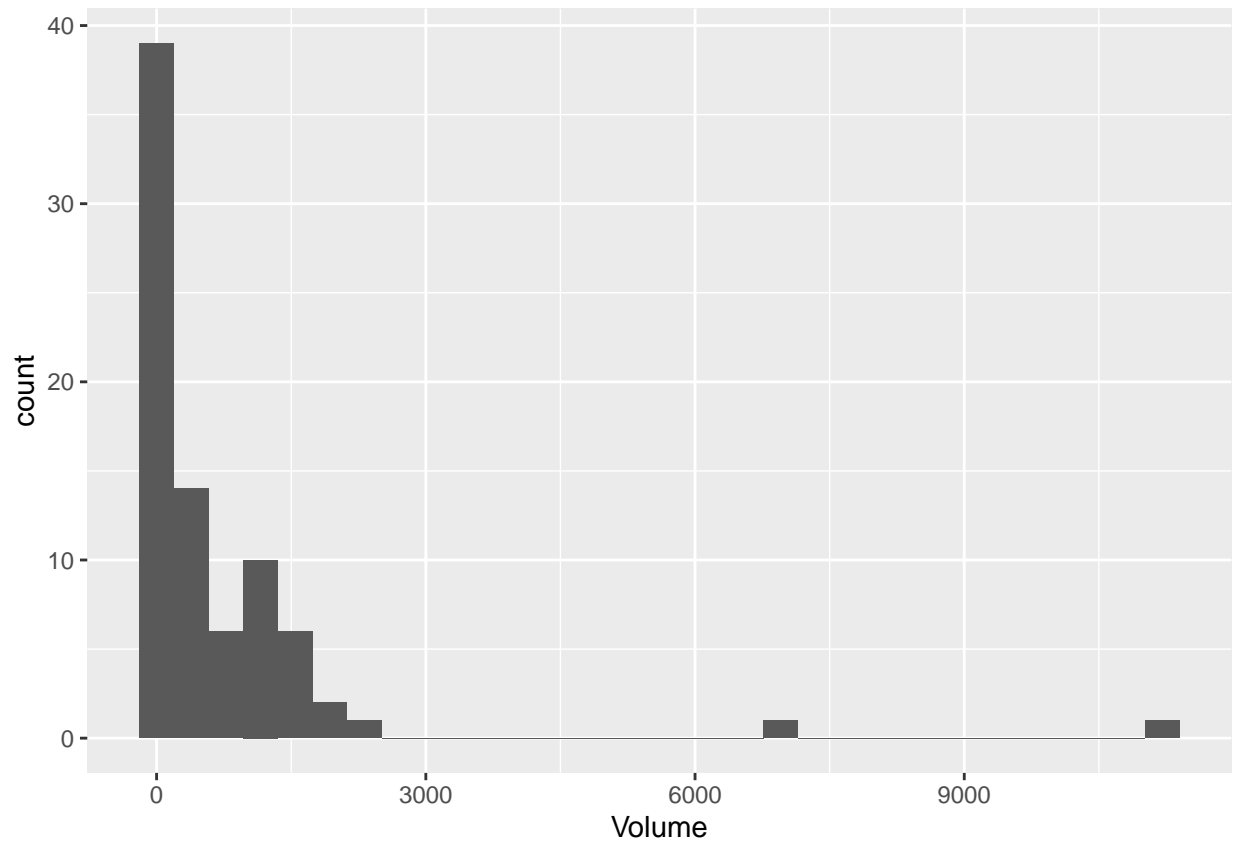
```
color <- colorRampPalette(c('#BB4444', '#EE9988', '#FFFFFF', '#77AADD', '#4477AA'))
corrplot(corrData3, method = 'shade', shade.col = NA, tl.col = 'black',
  type = 'upper', tl.srt = 45)
```



Histogram of Volume, reveals couple outliers

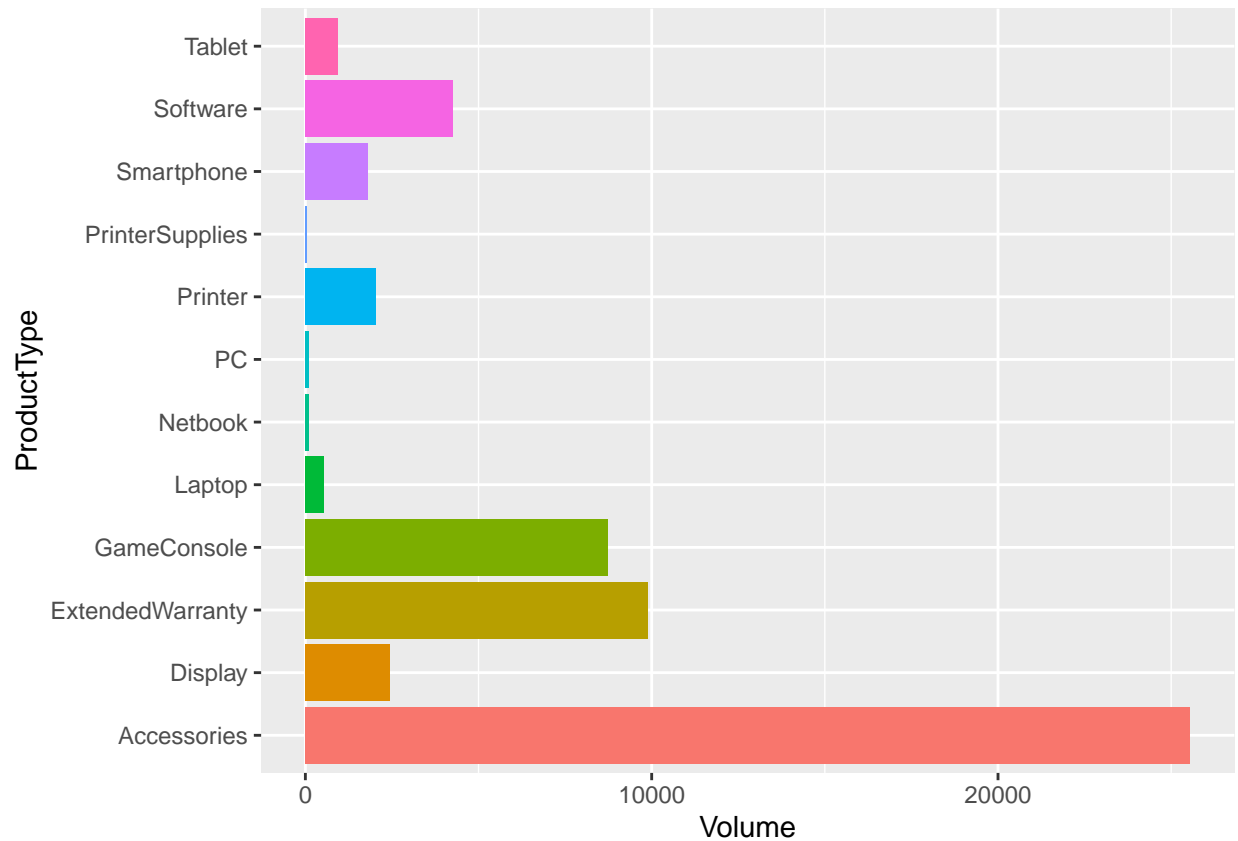
```
ggplot(data = existing3, mapping = aes(x = Volume)) +  
  geom_histogram()
```

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.



Plotting Sales Volume by Product Type

```
ggplot(data = existing, aes(x = ProductType, y = Volume, fill = ProductType)) +  
  geom_bar(stat = 'identity') +  
  guides(fill=FALSE) +  
  coord_flip()
```



Plotting the impact 5 Star Reviews have on Sales Volume

```
ggplot(data=existing, aes(x=x5StarReviews, y=Volume)) +
  geom_point(aes(color=ProductType, size=2)) +
  theme_bw() +
  scale_x_continuous(trans = 'log2') +
  scale_y_continuous(trans = 'log2') +
  geom_line() +
  facet_wrap(~ProductType) +
  xlab('Number of 5 Star Reviews') +
  ylab('Sales Volume') +
  ggtitle('Effect of 5 Star Reviews on Sales Volume')
```

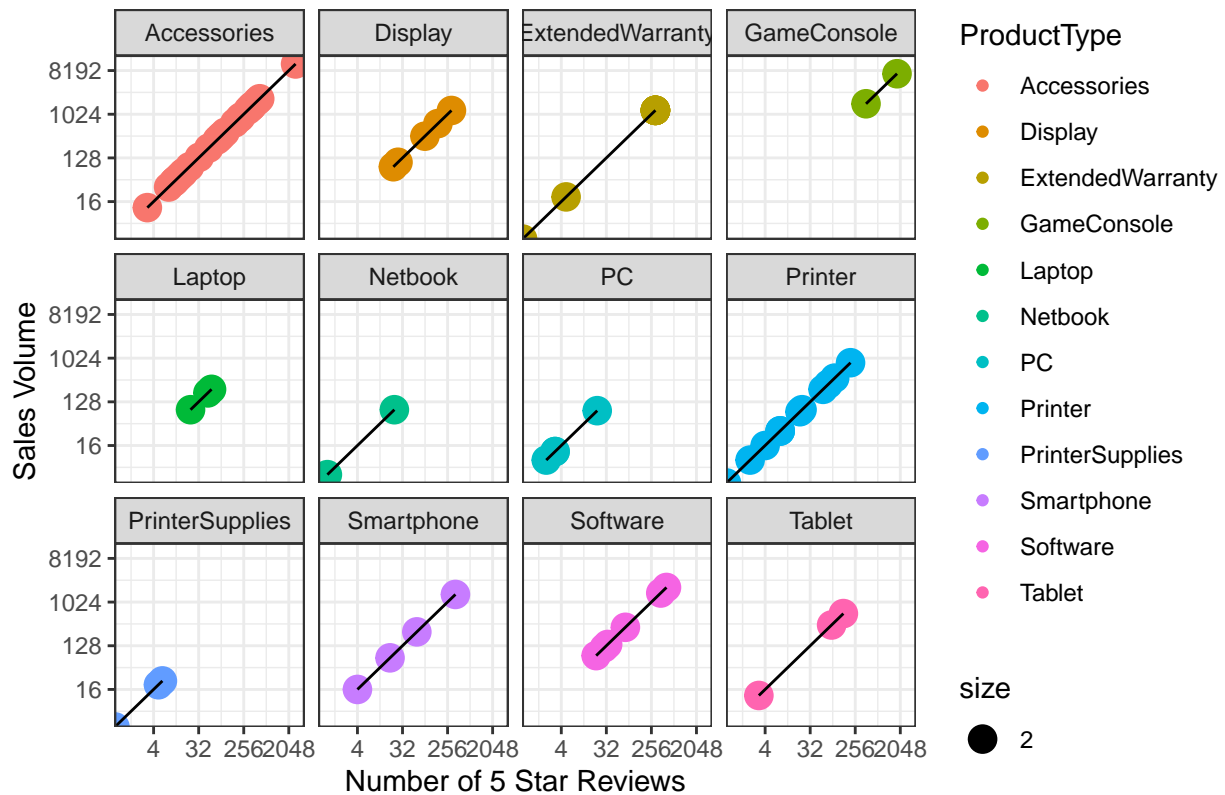
Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis

Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis

Effect of 5 Star Reviews on Sales Volume



Now plotting impact of 4 Star Reviews on Sales Volume

```
ggplot(data=existing, aes(x=x4StarReviews, y=Volume)) +
  geom_point(aes(color=ProductType, size=2)) +
  theme_bw() +
  scale_x_continuous(trans = 'log2') +
  scale_y_continuous(trans = 'log2') +
  geom_line() +
  facet_wrap(~ProductType) +
  xlab('Number of 4 Star Reviews') +
  ylab('Sales Volume') +
  ggtitle('Effect of 4 Star Reviews on Sales Volume')
```

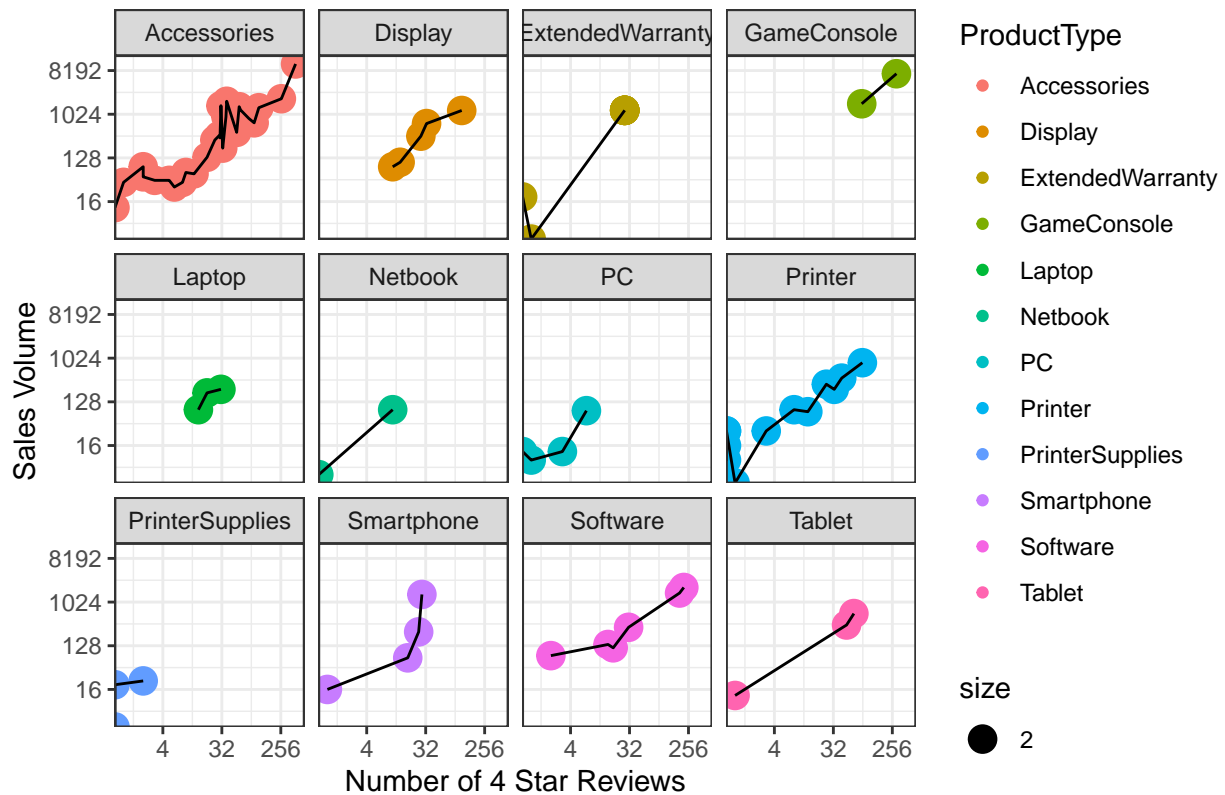
Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis

Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis

Effect of 4 Star Reviews on Sales Volume



Now plotting impact of Positive Service Reviews on Sales Volume

```
ggplot(data=existing, aes(x=PositiveServiceReview, y=Volume)) +
  geom_point(aes(color=ProductType, size=2)) +
  theme_bw() +
  scale_x_continuous(trans = 'log2') +
  scale_y_continuous(trans = 'log2') +
  geom_line() +
  facet_wrap(~ProductType) +
  xlab('Number of Positive Service Reviews') +
  ylab('Sales Volume') +
  ggtitle('Effect of Positive Service Reviews on Sales Volume')
```

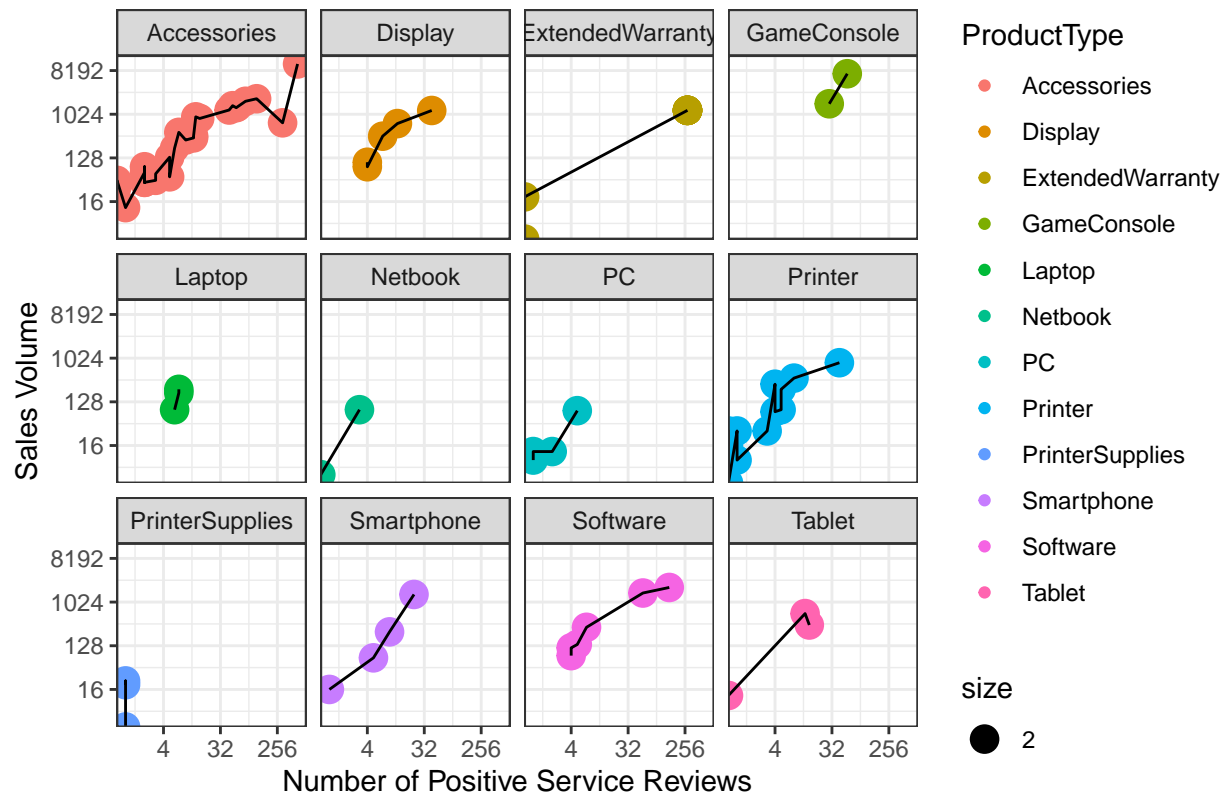
Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis

Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis

Effect of Positive Service Reviews on Sales Volume



Modeling

Creating data partition and setting cross validation

Two rows eventually were removed due to outlier volumes

```
set.seed(123)

# CreateDataPartition() 75% and 25%
index1 <- createDataPartition(existing3$Volume, p=0.75, list = FALSE)
train1 <- existing3[ index1,]
test1 <- existing3[-index1,]

# Removing 2 outlier rows #18 and #48 from test set
test1_rem_out <- test1[!rownames(test1) %in% c('18', '48'), ]

# Checking structure of train1
str(train1)

## 'data.frame': 61 obs. of 15 variables:
## $ ProductType.Laptop : num 0 0 0 1 0 0 0 0 0 0 ...
## $ ProductType.Netbook : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.PC : num 1 1 1 0 0 0 0 0 0 0 ...
```

```
## $ ProductType.Smartphone: num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductNum           : num 101 102 103 104 106 107 108 109 110 111 ...
## $ Price                 : num 949 2250 399 410 114 ...
## $ x4StarReviews         : num 3 1 0 19 30 3 19 9 1 2 ...
## $ x3StarReviews         : num 2 0 0 8 10 0 12 2 1 2 ...
## $ x2StarReviews         : num 0 0 0 3 9 0 5 0 0 4 ...
## $ x1StarReviews         : num 0 0 0 9 40 1 9 2 0 15 ...
## $ PositiveServiceReview : num 2 1 1 7 12 3 5 2 2 2 ...
## $ NegativeServiceReview : num 0 0 0 8 5 0 3 1 0 1 ...
## $ Recommendproduct      : num 0.9 0.9 0.9 0.8 0.3 0.9 0.7 0.8 0.9 0.5 ...
## $ ShippingWeight        : num 25.8 50 17.4 5.7 1.6 7.3 12 1.8 0.75 1 ...
## $ Volume                : num 12 8 12 196 332 44 132 64 40 84 ...
```

```
# Setting cross validation
control1 <- trainControl(method = 'repeatedcv',
                          number = 10,
                          repeats = 1)
```

Random forest model and tuning

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

# Creating dataframe for manual tuning
rfGrid <- expand.grid(mtry = c(2,3,4,5,6,7,8))

rf1 <- train(Volume ~ x4StarReviews + PositiveServiceReview + x2StarReviews + x3StarReviews +
             x1StarReviews + NegativeServiceReview + Recommendproduct + ShippingWeight + Price,
             data = train1,
             method = 'rf',
             trControl = control1,
             tuneGrid = rfGrid)

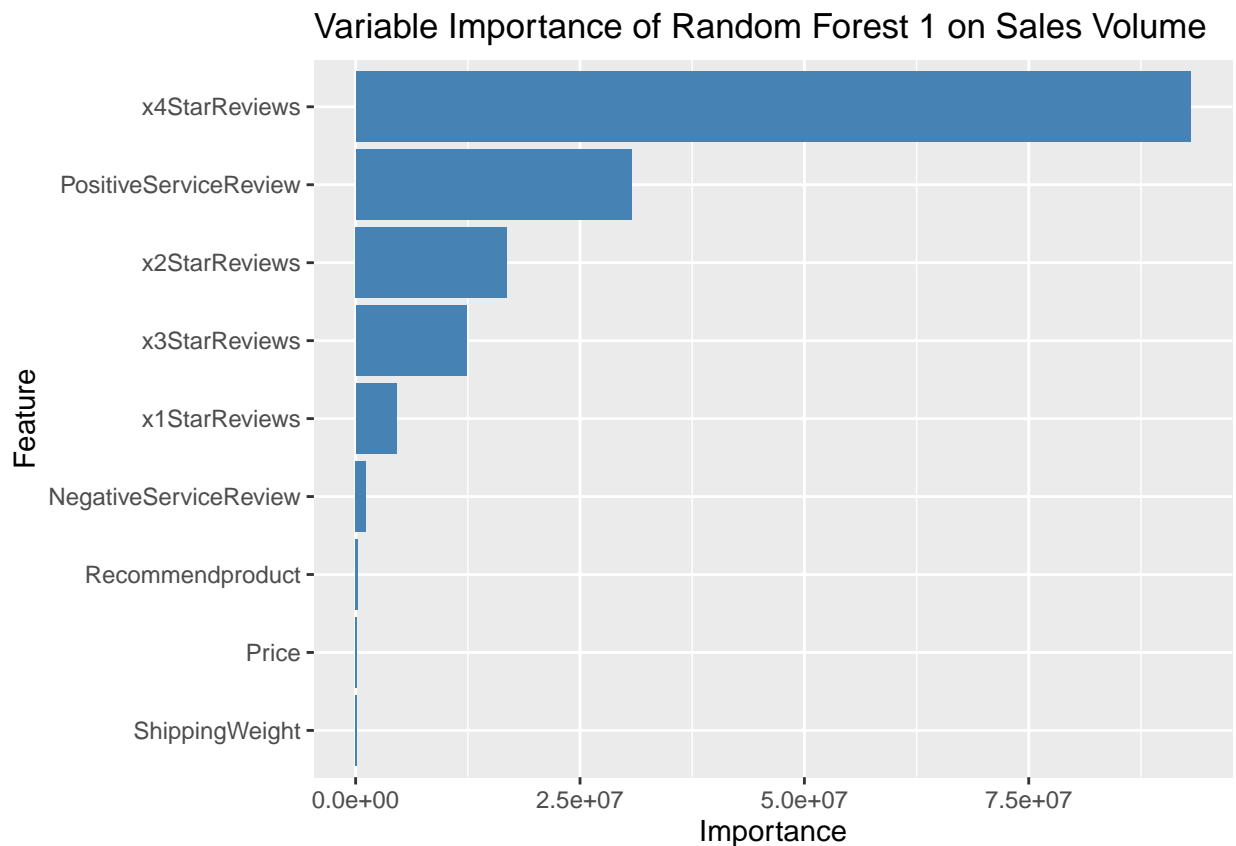
rf1
```

```
## Random Forest
##
## 61 samples
## 9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  mtry  RMSE      Rsquared  MAE
##  2      869.2921  0.8755901  416.0646
##  3      849.5229  0.8871013  400.1741
##  4      824.7775  0.8939530  386.7741
##  5      827.7373  0.8980015  384.5929
##  6      801.6069  0.9043345  372.1979
```

```
## 7      802.9288  0.9073910  372.2425
## 8      788.6419  0.9081729  365.2084
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 8.
```

Level of importance for variables in model

```
ggplot(varImp(rf1, scale=FALSE)) +
  geom_bar(stat = 'identity', fill = 'steelblue') +
  ggtitle('Variable Importance of Random Forest 1 on Sales Volume')
```

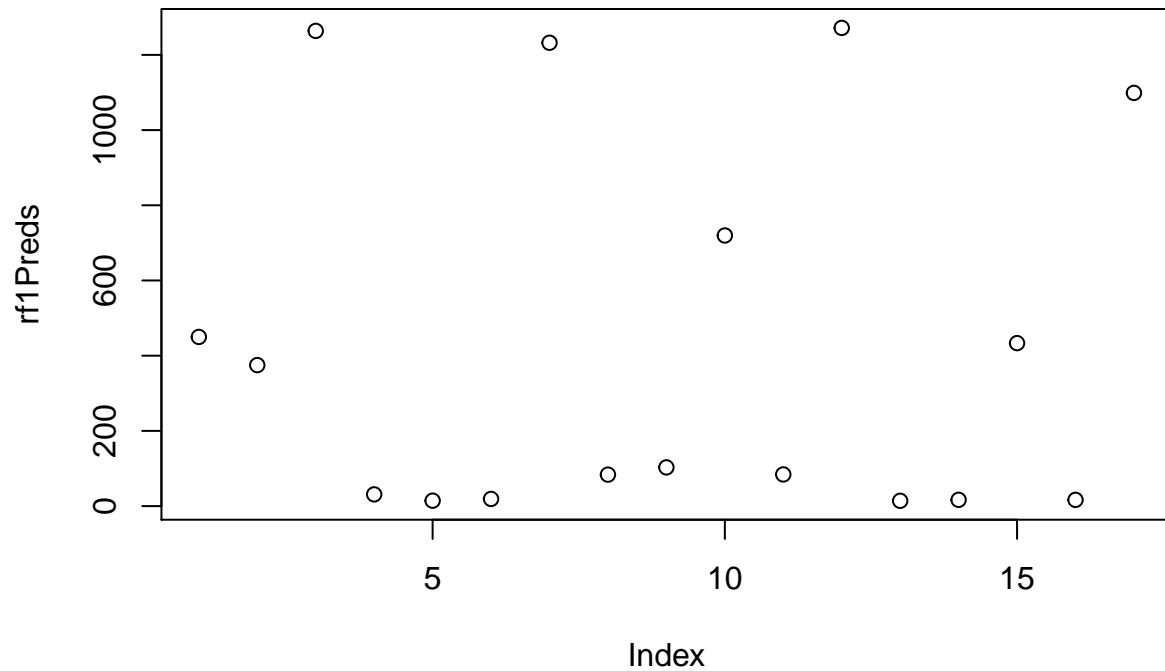


Predicting rf on test1

```
rf1Preds <- predict(rf1, newdata = test1_rem_out)
summary(rf1Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 14.11   19.05   102.97   425.15  719.71 1271.74
```

```
plot(rf1Preds)
```



A symmetrical pattern means a good residual plot!

postResample to test if it will do well on new data or if overfitting

```
postResample(rf1Preds, test1_rem_out$Volume)
```

```
##          RMSE    Rsquared      MAE
## 190.4253816  0.9452884  98.7387608
```

CV RMSE=788, R2=.908

PostResample RMSE=190, R2=.945

Random Forest using feature selection

```
set.seed(123)

rf2 <- train(Volume ~ x4StarReviews + PositiveServiceReview + x2StarReviews,
```



```
data = train1,  
method = 'rf',  
trControl = control1)
```

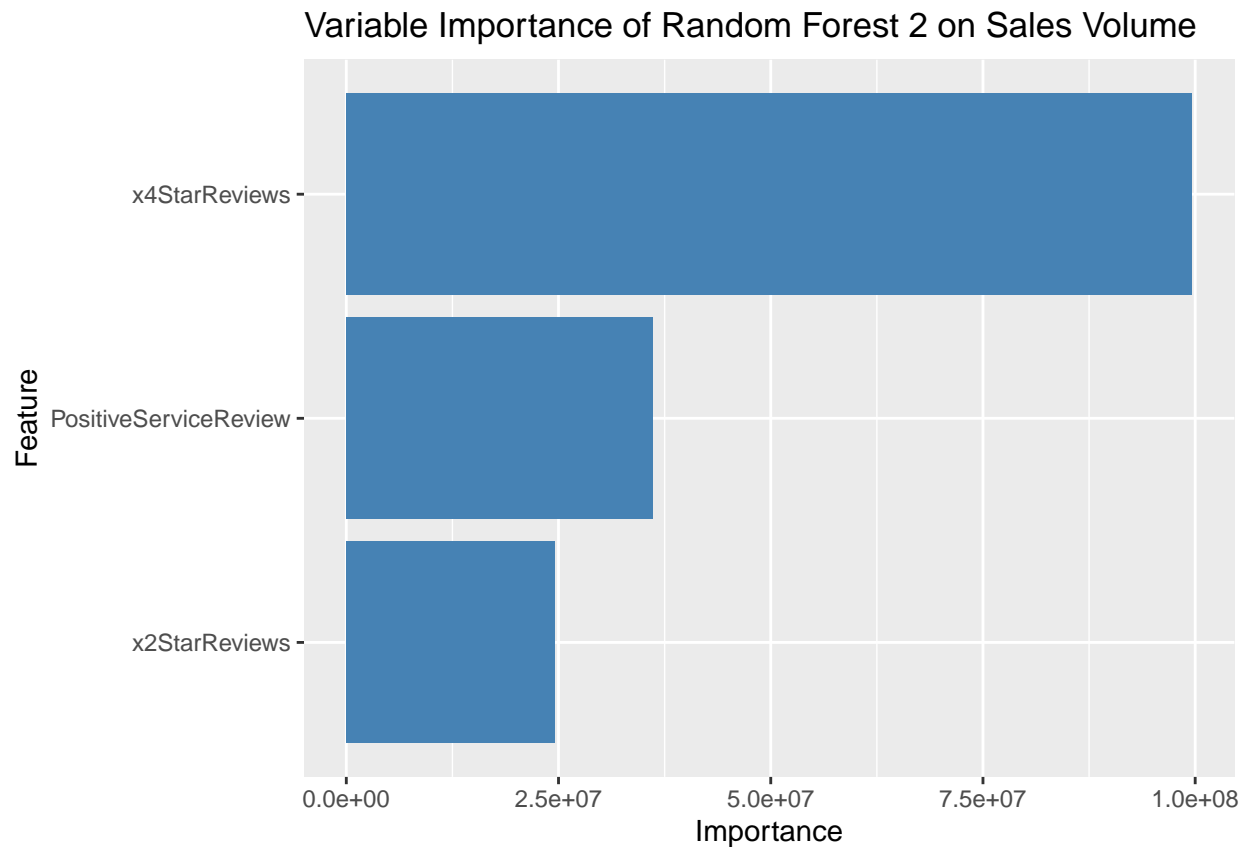
note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .

```
rf2
```

```
## Random Forest  
##  
## 61 samples  
## 3 predictor  
##  
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)  
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...  
## Resampling results across tuning parameters:  
##  
##   mtry  RMSE      Rsquared  MAE  
##    2    771.2710  0.9218973  349.2301  
##    3    745.3771  0.9284383  338.8776  
##  
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was mtry = 3.
```

Variable importance

```
ggplot(varImp(rf2, scale=FALSE)) +  
  geom_bar(stat = 'identity', fill = 'steelblue') +  
  ggtitle('Variable Importance of Random Forest 2 on Sales Volume')
```

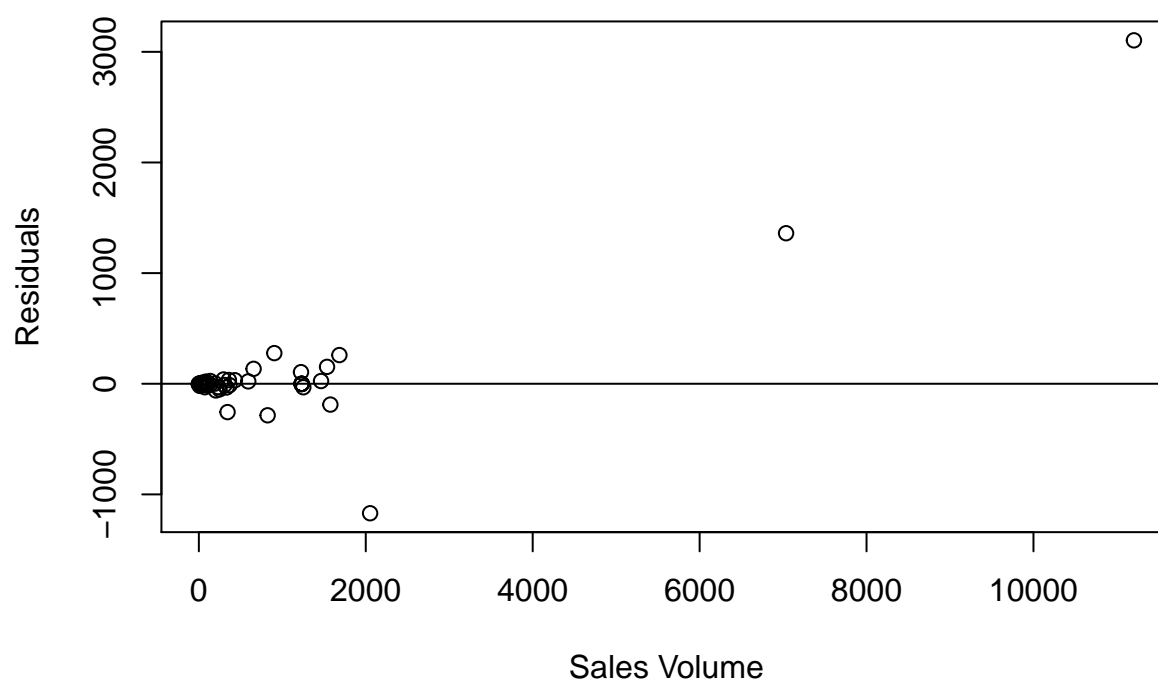


Plotting the residuals against the actual values for Volume

This graph shows volume outlier

```
resid_rf2 <- residuals(rf2)
plot(train1$Volume, resid_rf2,
     xlab = 'Sales Volume',
     ylab = 'Residuals',
     main = 'Predicted Sales Volume Residuals Plot',
     abline(0,0))
```

Predicted Sales Volume Residuals Plot

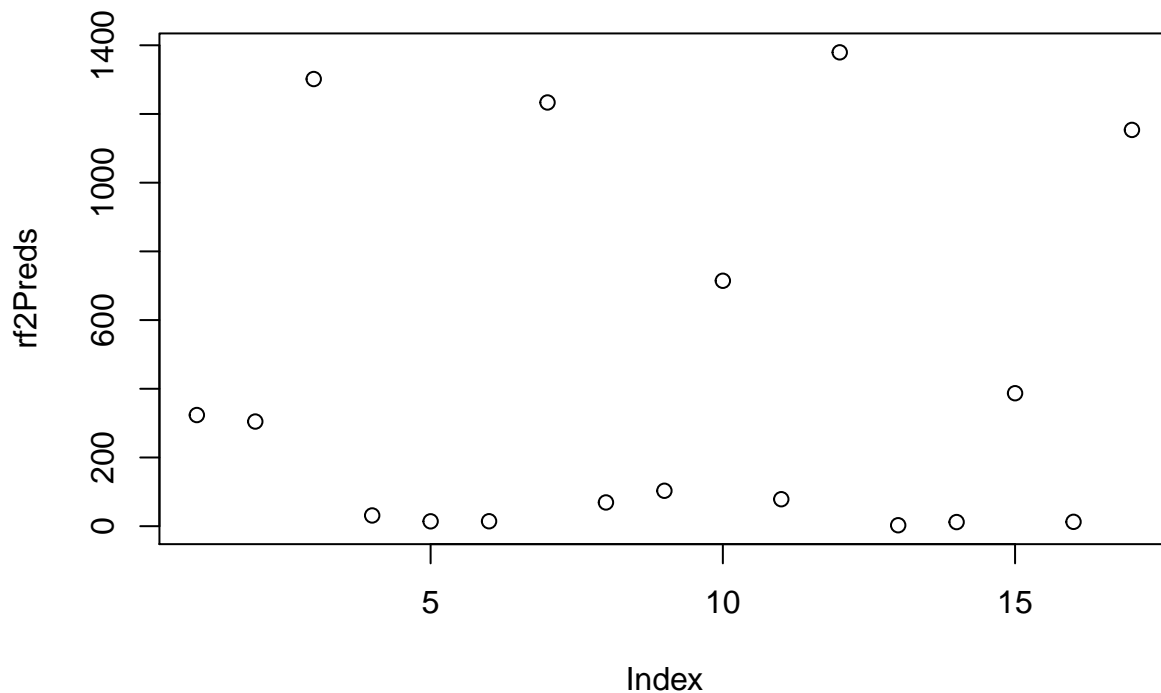


Predicting rf2 on test1

```
rf2Preds <- predict(rf2, newdata = test1_rem_out)
summary(rf2Preds)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
##    2.742   14.336   103.066   419.753   714.428  1379.420
```

```
plot(rf2Preds)
```



Another excellent residual plot, showing our predictions are consistent with regression
 postResample to test if it will do well on new data or if overfitting

```
postResample(rf2Preds, test1_rem_out$Volume)
```

```
##          RMSE      Rsquared        MAE
## 153.8183544  0.9718405  74.7555212
```

CV RMSE = 745, R2=.928

PostResample RMSE=153, R2=.972

The postResample R2 and RMSE for a regression model is excellent. This is our top model!

Random Forest using feature selection

```
set.seed(123)

rf3 <- train(Volume ~ x4StarReviews + PositiveServiceReview + x3StarReviews,
             data = train1,
```

```
method = 'rf',
trControl = control1)
```

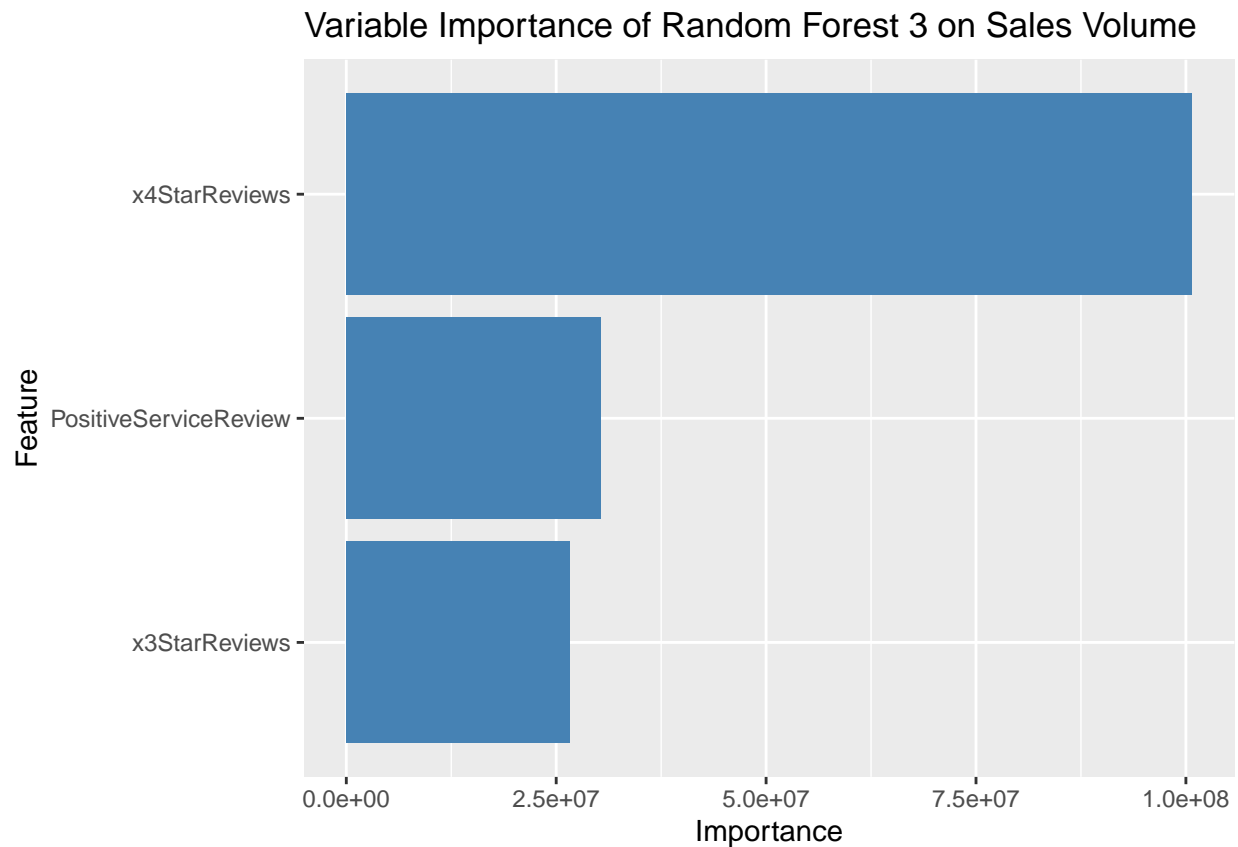
note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .

```
rf3
```

```
## Random Forest
##
## 61 samples
## 3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  mtry  RMSE      Rsquared  MAE
##  2     710.5834  0.9285459  323.8009
##  3     684.5642  0.9346194  313.5800
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 3.
```

Variable importance

```
ggplot(varImp(rf3, scale=FALSE)) +
  geom_bar(stat = 'identity', fill = 'steelblue') +
  ggtitle('Variable Importance of Random Forest 3 on Sales Volume')
```



Predicting rf3 on test1

```
rf3Preds <- predict(rf3, newdata = test1_rem_out)
summary(rf3Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.789  11.959   94.221  425.461  737.133 1358.924
```

postResample to test if it will do well on new data or if overfitting

```
postResample(rf3Preds, test1_rem_out$Volume)
```

```
##      RMSE  Rsquared    MAE
## 167.7971733  0.9549701  83.4029171
```

CV RMSe=648, R2=.934

PostResample RMSE=167, R2=.954

Random Forest using feature selection

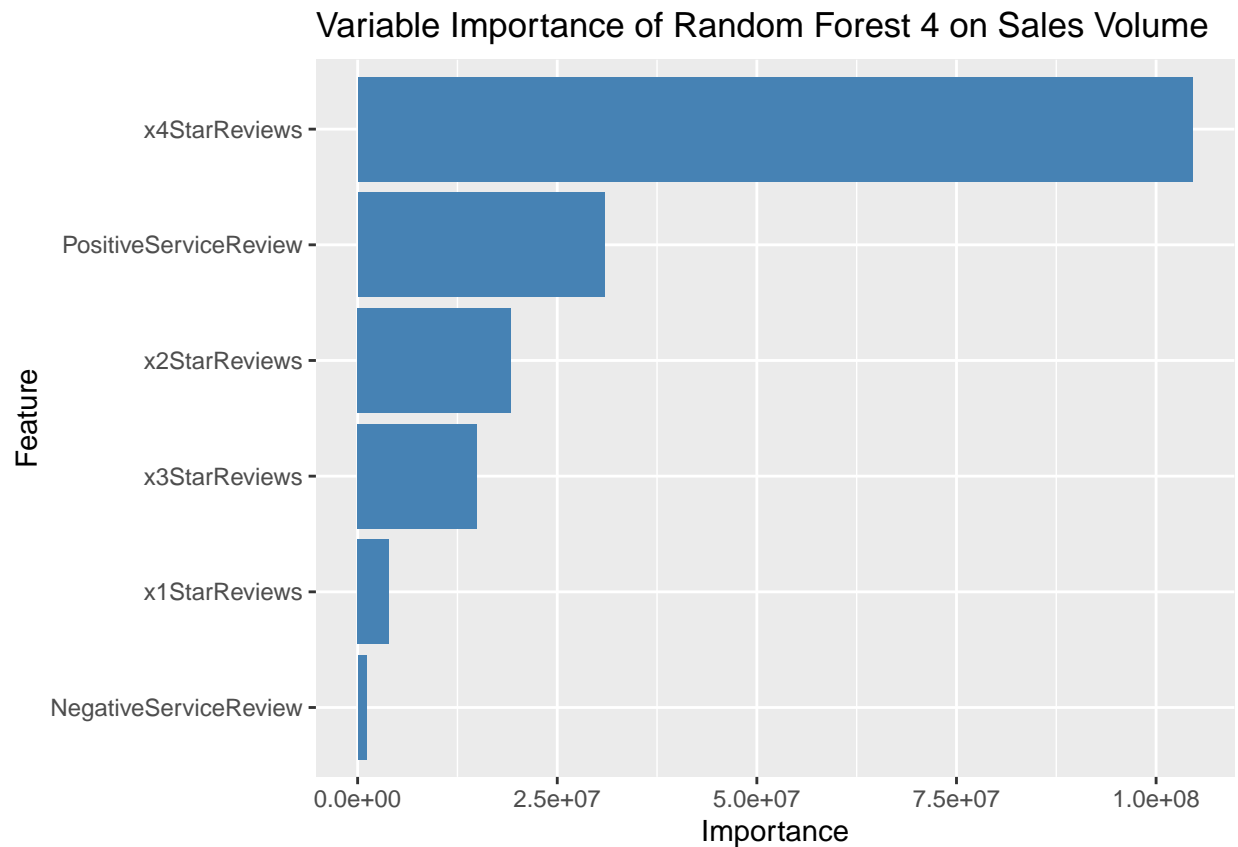
```
set.seed(123)
rf4 <- train(Volume ~ x4StarReviews + PositiveServiceReview + x3StarReviews + x2StarReviews +
             x1StarReviews + NegativeServiceReview,
             data = train1,
             method = 'rf',
             trControl = control1)
```

rf4

```
## Random Forest
##
## 61 samples
## 6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  mtry  RMSE      Rsquared  MAE
##  2      844.5227  0.8864850  396.6254
##  4      801.2553  0.9052961  370.6409
##  6      790.8997  0.9072861  362.8766
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 6.
```

Variable importance using ggplot

```
ggplot(varImp(rf4, scale=FALSE)) +
  geom_bar(stat = 'identity', fill = 'steelblue') +
  ggtitle('Variable Importance of Random Forest 4 on Sales Volume')
```

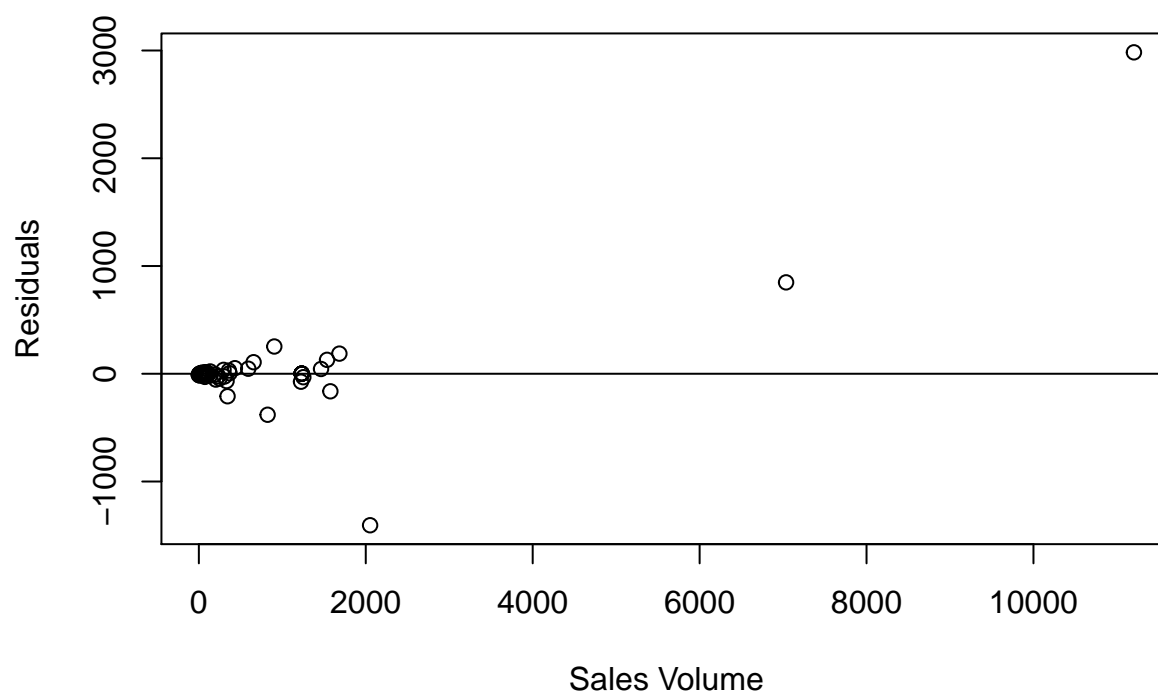


Plotting the residuals against the actual values for Volume

Graph shows outlier

```
resid_rf4 <- residuals(rf4)
plot(train1$Volume, resid_rf4, xlab = 'Sales Volume', ylab = 'Residuals',
     main='Predicted Sales Volume Residuals Plot',
     abline(0,0))
```


Predicted Sales Volume Residuals Plot



Predicting rf4 on test1

```
rf4Preds <- predict(rf4, newdata = test1_rem_out)
summary(rf4Preds)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
##    7.937   15.732   98.618   426.150   782.476  1304.662
```

postResample to test if it will do well on new data or if overfitting

```
postResample(rf4Preds, test1_rem_out$Volume)
```

```
##      RMSE   Rsquared     MAE
## 177.453911  0.952956  87.506546
```

CV RMSE=783, R2=.909

RMSE=177, R2=.952

Support Vector Machines – RBF Kernel

Set seed

```
set.seed(123)

# Creating dataframe for manual tuning
rbfGrid <- expand.grid(sigma = c(.01, .015, .2),
                      C = c(10, 100, 1000))

rbf1 <- train(Volume ~ x4StarReviews + x3StarReviews + PositiveServiceReview,
              data = train1,
              method = 'svmRadial',
              trControl = control1,
              tuneGrid = rbfGrid,
              preProc = c('center', 'scale'))

rbf1
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 61 samples
## 3 predictor
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  sigma  C      RMSE      Rsquared  MAE
##  0.010  10    944.2228  0.8594778  489.1231
##  0.010  100   930.7863  0.8149199  473.1634
##  0.010  1000  1190.2695  0.8309208  580.6230
##  0.015  10    986.8673  0.8419784  507.9508
##  0.015  100   940.5278  0.8123009  480.0726
##  0.015  1000  1236.2804  0.8613697  590.2852
##  0.200  10    913.7802  0.9182651  467.5324
##  0.200  100   879.2184  0.9194906  436.6931
##  0.200  1000   949.7834  0.9093674  462.1128
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.2 and C = 100.
```

Predicting rbf on test1

```
rbf1Preds <- predict(rbf1, newdata = test1_rem_out)
summary(rbf1Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  29.81   94.91  267.83  463.59  552.12 2146.20
```

postResample to test if it will do well on new data or if overfitting

```
postResample(rbf1Preds, test1_rem_out$Volume)
```

```
##      RMSE    Rsquared    MAE
## 264.0730623  0.8148197 177.1172248
```

CV RMSE=879, R2=.919

PostResample RMSE=264, R2=.815

Support Vector Machines – RBF Kernel feature selection

Set seed

```
set.seed(123)

# Creating dataframe for manual tuning
rbfGrid <- expand.grid(sigma = c(.01, .015, .2),
                      C = c(10, 100, 1000))

rbf2 <- train(Volume ~ x4StarReviews + PositiveServiceReview + x2StarReviews,
              data = train1,
              method = 'svmRadial',
              trControl = control1,
              tuneGrid = rbfGrid,
              preProc = c('center', 'scale'))

rbf2
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 61 samples
## 3 predictor
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  sigma  C      RMSE      Rsquared  MAE
```

```
## 0.010 10 935.7611 0.8655808 465.7621
## 0.010 100 774.4498 0.8457538 408.1163
## 0.010 1000 729.0330 0.8563199 400.4417
## 0.015 10 879.8614 0.8620022 452.0967
## 0.015 100 761.7506 0.8470788 417.3052
## 0.015 1000 791.0228 0.8280795 418.6904
## 0.200 10 838.7191 0.9322844 436.6363
## 0.200 100 783.3714 0.9512888 398.3164
## 0.200 1000 657.2931 0.9099973 356.3912
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.2 and C = 1000.
```

Predicting rbf on test1

```
rbf2Preds <- predict(rbf2, newdata = test1_rem_out)
summary(rbf2Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -253.3   71.6   256.6   512.9   343.3  2499.2
```

```
# postResample to test if it will do well on new data or if overfitting
postResample(rbf2Preds, test1_rem_out$Volume)
```

```
##           RMSE    Rsquared      MAE
## 420.0433306  0.7040338 243.6208248
```

CV RMSE=657, R2=.909

PostResample RMSE=420, R2=.704

Negatives

Support Vector Machines – Linear

Set seed

```
set.seed(123)

### Creating dataframe for manual tuning
linearGrid <- expand.grid(C = c(1, 10, 100, 1000))

linear1 <- train(Volume ~ x4StarReviews + x3StarReviews + PositiveServiceReview,
  data = train1,
  method = 'svmLinear',
  trControl = control1,
  tuneGrid = linearGrid,
  preProc = c('center', 'scale'))

linear1
```

```
## Support Vector Machines with Linear Kernel
##
## 61 samples
## 3 predictor
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##   C      RMSE      Rsquared   MAE
##   1    873.4355  0.8664637  456.9757
##   10   843.0240  0.8585048  447.9310
##   100  848.6679  0.8571624  453.5503
##  1000  847.9066  0.8571915  452.4325
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was C = 10.
```

Predicting rbf on test1

```
linearPreds <- predict(linear1, newdata = test1_rem_out)
summary(linearPreds)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -196.9 -129.0   124.7   388.2   561.6  2364.1
```

postResample to test if it will do well on new data or if overfitting

```
lin_PR <- postResample(linearPreds, test1_rem_out$Volume)
```

CV RMSE=843, R2=.858

PR RMSE=462, R2=.583

Negative predictions, move on

SVM – Linear

Changing features

Set seed

```
set.seed(123)

# Creating dataframe for manual tuning
```

```
linearGrid <- expand.grid(C = c(1, 10, 100, 1000))

linear2 <- train(Volume ~ x4StarReviews + x3StarReviews + PositiveServiceReview +
  NegativeServiceReview + Price,
  data = train1,
  method = 'svmLinear',
  trControl = control1,
  tuneGrid = linearGrid,
  preProc = c('center', 'scale'))

linear2
```

```
## Support Vector Machines with Linear Kernel
##
## 61 samples
## 5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  C      RMSE      Rsquared  MAE
##    1  553.1088  0.8300446  328.9984
##   10  538.0562  0.8151764  327.8059
##  100  538.1027  0.8148848  327.7453
## 1000  538.5593  0.8146058  327.9203
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was C = 10.
```

```
# Predicting rbf on test1
linear2Preds <- predict(linear2, newdata = test1_rem_out)
summary(linear2Preds)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -155.3142 -102.7355   0.5217  435.3928  496.4102 2631.1579
```

postResample to test if it will do well on new data or if overfitting

```
postResample(linear2Preds, test1_rem_out$Volume)
```

```
##      RMSE      Rsquared      MAE
## 501.1605728  0.5858754 334.9077977
```

RMSE=1120, R2=56.9

Negative predictions, move on

Support Vector Machines – Polynomial

Set seed

```
set.seed(123)

# Creating dataframe for manual tuning
polyGrid <- expand.grid(degree = c(2,3,4),
                       scale = c(1,2),
                       C = c(.1, 1, 10, 100))

poly1 <- train(Volume ~ x4StarReviews + x3StarReviews + PositiveServiceReview,
               data = train1,
               method = 'svmPoly',
               trControl = control1,
               tuneGrid = polyGrid,
               preProc = c('center', 'scale'))

poly1
```

```
## Support Vector Machines with Polynomial Kernel
##
## 61 samples
## 3 predictor
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  degree  scale  C      RMSE      Rsquared  MAE
##  2        1      0.1    1155.408  0.7966076  571.6401
##  2        1      1.0    4104.003  0.8301827  1770.0867
##  2        1     10.0    6987.814  0.8557033  2939.2082
##  2        1    100.0    9796.456  0.8389599  4083.3127
##  2        2      0.1    2104.421  0.8267500   966.7124
##  2        2      1.0    6491.915  0.8763071  2740.5152
##  2        2     10.0    9245.414  0.8522459  3859.6825
##  2        2    100.0   10042.438  0.8331194  4185.3819
##  3        1      0.1    1118.828  0.8898300   534.3918
##  3        1      1.0    3755.882  0.9381422  1597.5451
##  3        1     10.0    7095.492  0.9176351  2965.7635
##  3        1    100.0   39627.409  0.9024923 16252.9238
##  3        2      0.1    2667.568  0.8509251  1166.7636
##  3        2      1.0    4472.345  0.8879559  1896.1439
##  3        2     10.0   31802.988  0.9179907 13053.9787
##  3        2    100.0   86413.899  0.8848359 35365.6943
##  4        1      0.1    6877.359  0.8003001  2881.4997
```

```
## 4      1      1.0  54000.769  0.8216935  22115.1070
## 4      1     10.0  197327.453  0.9111491  80637.6759
## 4      1    100.0   25145.120  0.9146607  10346.8229
## 4      2      0.1   31923.943  0.8869538  13115.3285
## 4      2      1.0  167176.011  0.8980690  68329.9630
## 4      2     10.0   68279.157  0.9005098  27957.4632
## 4      2    100.0  157877.892  0.9005079  64537.4960
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were degree = 3, scale = 1 and C = 0.1.
```

Predicting rbf on test1

```
polyPreds <- predict(poly1, newdata = test1_rem_out)
summary(polyPreds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -31.78  -20.75   218.56   370.45   536.88  1240.00
```

postResample to test if it will do well on new data or if overfitting

```
postResample(polyPreds, test1_rem_out$Volume)
```

```
##      RMSE    Rsquared    MAE
## 334.4330248  0.7687032 174.1454650
```

RMSE=688, R2=60.2

Negative predictions, move on

SVM – Polynomial

Set seed

```
set.seed(123)

# Creating dataframe for manual tuning
polyGrid <- expand.grid(degree = c(2,3,4),
                        scale = c(1,2),
                        C = c(.1, 1, 10, 100))

poly2 <- train(Volume ~ x4StarReviews + x2StarReviews + PositiveServiceReview +
                NegativeServiceReview,
                data = train1,
                method = 'svmPoly',
                trControl = control1,
```



```
tuneGrid = polyGrid,
preProc = c('center','scale'))
```

```
poly2
```

```
## Support Vector Machines with Polynomial Kernel
##
## 61 samples
## 4 predictor
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
## degree scale C RMSE Rsquared MAE
## 2 1 0.1 876.0226 0.8081674 469.4007
## 2 1 1.0 1532.8601 0.8154286 722.1496
## 2 1 10.0 14512.8090 0.8528996 6032.6490
## 2 1 100.0 11848.1045 0.9129566 4942.3458
## 2 2 0.1 1115.2333 0.8751110 559.6023
## 2 2 1.0 5251.8265 0.8636670 2247.9786
## 2 2 10.0 11368.9766 0.8529120 4751.2822
## 2 2 100.0 11354.9053 0.9063491 4732.0598
## 3 1 0.1 18313.5053 0.8935899 7558.3279
## 3 1 1.0 24882.0057 0.8523866 10250.6328
## 3 1 10.0 61551.6914 0.9182948 25214.9606
## 3 1 100.0 36782.7066 0.8210000 15136.3743
## 3 2 0.1 37470.6589 0.8460612 15381.2988
## 3 2 1.0 34037.7172 0.8417435 13992.1891
## 3 2 10.0 62873.4712 0.8259141 25775.7687
## 3 2 100.0 100280.0776 0.8102818 41118.1724
## 4 1 0.1 96870.0477 0.8372587 39632.3275
## 4 1 1.0 9912.6724 0.8582925 4146.7389
## 4 1 10.0 174996.0882 0.8816298 71529.9157
## 4 1 100.0 334667.4580 0.7981803 136752.8839
## 4 2 0.1 233541.8543 0.8434558 95446.7978
## 4 2 1.0 404176.1594 0.8485852 165110.5747
## 4 2 10.0 704939.6023 0.8342341 287898.1043
## 4 2 100.0 790317.3179 0.8342340 322758.2980
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were degree = 2, scale = 1 and C = 0.1.
```

Predicting rbf on test1

```
poly2Preds <- predict(poly2, newdata = test1_rem_out)
summary(poly2Preds)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 132.6 154.0 256.1 469.9 472.7 1710.2
```

postResample to test if it will do well on new data or if overfitting

```
postResample(poly2Preds, test1_rem_out$Volume)
```

```
##          RMSE      Rsquared      MAE
## 402.3116913    0.5699793 256.8209379
```

RMSE=402, R2=0.57

Gradient Boosting

Set seed

```
set.seed(123)

gbm1 <- train(Volume ~ x4StarReviews + x2StarReviews + PositiveServiceReview,
              data = train1,
              method = 'gbm',
              trControl = control1,
              preProc = c('center','scale'))
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2262979.9194             nan     0.1000  50753.5120
##      2   2187640.5146             nan     0.1000  51177.8262
##      3   2140085.0369             nan     0.1000  55884.8979
##      4   2119804.1063             nan     0.1000  16353.3706
##      5   2100350.9717             nan     0.1000  18006.2880
##      6   2088652.3812             nan     0.1000  5634.3370
##      7   2071267.6596             nan     0.1000  19435.9292
##      8   2000591.0212             nan     0.1000  18321.7278
##      9   1951048.5330             nan     0.1000  13052.9121
##     10   1942952.6063             nan     0.1000   7507.2463
##     20   1852021.0090             nan     0.1000 -14547.1313
##     40   1776852.2671             nan     0.1000 -11040.5487
##     60   1752258.6295             nan     0.1000 -13082.6736
##     80   1674082.0120             nan     0.1000 -11664.6392
##    100   1638017.0030             nan     0.1000 -28525.0084
##    120   1609883.0047             nan     0.1000 -14319.8681
##    140   1629745.4477             nan     0.1000 -26778.4312
##    150   1613703.0714             nan     0.1000 -20008.9222
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2261149.4817             nan     0.1000  90433.1178
##      2   2177478.6574             nan     0.1000  49051.6495
##      3   2103729.9013             nan     0.1000  29071.1127
##      4   2083433.2241             nan     0.1000  16149.0064
##      5   2054782.7983             nan     0.1000  35585.4858
##      6   2015850.8286             nan     0.1000  51383.4386
##      7   1943725.5287             nan     0.1000  30669.8238
```

##	8	1936072.1908	nan	0.1000	4334.8826
##	9	1928693.0550	nan	0.1000	7035.9848
##	10	1886640.9086	nan	0.1000	-204.5888
##	20	1773051.6864	nan	0.1000	-5397.9315
##	40	1731097.0050	nan	0.1000	-6959.5081
##	60	1690813.3705	nan	0.1000	-21227.1997
##	80	1662740.3941	nan	0.1000	-35703.0838
##	100	1640362.2217	nan	0.1000	-13902.5905
##	120	1624560.9651	nan	0.1000	-31293.9611
##	140	1618365.8572	nan	0.1000	-11484.9595
##	150	1613451.6687	nan	0.1000	-8668.0878
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2314036.1340	nan	0.1000	75208.4269
##	2	2176140.7271	nan	0.1000	76403.7968
##	3	2130683.3430	nan	0.1000	58832.7298
##	4	2100358.4790	nan	0.1000	39280.9194
##	5	2073057.4689	nan	0.1000	38398.9470
##	6	2042955.4650	nan	0.1000	39969.2553
##	7	1975389.2904	nan	0.1000	28053.6954
##	8	1967579.3643	nan	0.1000	3215.1862
##	9	1923954.7232	nan	0.1000	19434.4113
##	10	1882373.9751	nan	0.1000	-8745.3661
##	20	1758257.2968	nan	0.1000	-2741.4803
##	40	1690773.0131	nan	0.1000	-12792.2923
##	60	1667967.1752	nan	0.1000	-6418.7500
##	80	1637313.0243	nan	0.1000	-12373.2401
##	100	1625466.8528	nan	0.1000	-15343.8628
##	120	1613313.0810	nan	0.1000	-19108.8284
##	140	1610218.6232	nan	0.1000	-23698.4308
##	150	1601612.1009	nan	0.1000	-8422.7245
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3093452.1976	nan	0.1000	139712.5650
##	2	2974878.8268	nan	0.1000	132796.8088
##	3	2880495.5763	nan	0.1000	91409.9118
##	4	2827442.8966	nan	0.1000	52961.4731
##	5	2747788.8580	nan	0.1000	91197.0110
##	6	2629086.5756	nan	0.1000	36067.8748
##	7	2568359.2561	nan	0.1000	56317.0665
##	8	2502552.1346	nan	0.1000	-23912.7942
##	9	2474903.3502	nan	0.1000	32616.0012
##	10	2426039.8502	nan	0.1000	-17078.0996
##	20	2306283.7639	nan	0.1000	-27781.9705
##	40	2239679.7335	nan	0.1000	4137.2398
##	60	2156812.2871	nan	0.1000	-31312.5282
##	80	2125065.0100	nan	0.1000	-1290.8621
##	100	2117989.5407	nan	0.1000	-44371.2537
##	120	2064989.8638	nan	0.1000	-13330.0101
##	140	2042437.4284	nan	0.1000	-72028.9161
##	150	2035886.7821	nan	0.1000	-10540.4241
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3017626.4188	nan	0.1000	72008.9933

##	2	2915242.5558	nan	0.1000	86662.5496
##	3	2881575.4922	nan	0.1000	29722.9334
##	4	2815717.8731	nan	0.1000	64966.6540
##	5	2728420.1895	nan	0.1000	-19428.4138
##	6	2614600.8128	nan	0.1000	45000.1867
##	7	2581466.7489	nan	0.1000	35064.9043
##	8	2527970.4085	nan	0.1000	44097.8229
##	9	2433162.0212	nan	0.1000	12681.4009
##	10	2436747.7394	nan	0.1000	-34155.9413
##	20	2304617.3029	nan	0.1000	14213.1106
##	40	2199806.4588	nan	0.1000	-13957.1126
##	60	2115320.6778	nan	0.1000	-19184.8060
##	80	2099014.7655	nan	0.1000	-5381.4844
##	100	2060015.8938	nan	0.1000	119.6064
##	120	2043215.3195	nan	0.1000	-1952.0719
##	140	2035822.8015	nan	0.1000	-2503.8764
##	150	2033033.8851	nan	0.1000	-35585.7718
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3064423.3658	nan	0.1000	148577.9037
##	2	2914770.1203	nan	0.1000	125762.8929
##	3	2818467.9575	nan	0.1000	80130.6102
##	4	2786313.2361	nan	0.1000	34153.7649
##	5	2701505.0399	nan	0.1000	1946.9990
##	6	2648087.3810	nan	0.1000	39998.3103
##	7	2616743.4409	nan	0.1000	33397.3514
##	8	2531220.7567	nan	0.1000	6142.8986
##	9	2469259.1791	nan	0.1000	-1176.7252
##	10	2454357.9628	nan	0.1000	-5680.7991
##	20	2282288.5888	nan	0.1000	2651.2276
##	40	2176666.0968	nan	0.1000	2522.2121
##	60	2132554.3994	nan	0.1000	-43285.2722
##	80	2114446.9660	nan	0.1000	-57332.6326
##	100	2102904.8048	nan	0.1000	-53625.7533
##	120	2092645.2164	nan	0.1000	109.5375
##	140	2088660.5144	nan	0.1000	-1994.3235
##	150	2074172.4260	nan	0.1000	-41883.2250
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2964046.0261	nan	0.1000	128495.8415
##	2	2828651.4640	nan	0.1000	141980.0581
##	3	2781031.8205	nan	0.1000	55276.7950
##	4	2742448.1778	nan	0.1000	40785.1929
##	5	2652159.5456	nan	0.1000	98352.8790
##	6	2509079.5643	nan	0.1000	25357.5265
##	7	2498537.8948	nan	0.1000	2086.6868
##	8	2454046.0625	nan	0.1000	55834.4184
##	9	2408380.0222	nan	0.1000	53430.8926
##	10	2355867.8847	nan	0.1000	42708.2777
##	20	2127852.4531	nan	0.1000	-31933.0232
##	40	1993137.0765	nan	0.1000	-35537.1085
##	60	1940590.6137	nan	0.1000	-22565.7985
##	80	1915533.1868	nan	0.1000	-15997.6686
##	100	1919460.7846	nan	0.1000	-29881.4896

```

##      120  1872765.5457          nan    0.1000 -179.0483
##      140  1840519.1192          nan    0.1000 -265.2589
##      150  1836065.9991          nan    0.1000 -2005.5944
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2960923.1984          nan    0.1000 103996.4347
##      2   2806751.3379          nan    0.1000 127874.9308
##      3   2722674.8166          nan    0.1000 91664.5027
##      4   2697573.8515          nan    0.1000 19467.7698
##      5   2588218.0563          nan    0.1000 84814.2325
##      6   2565106.1302          nan    0.1000 25387.2533
##      7   2551051.4971          nan    0.1000 8222.9826
##      8   2501341.3316          nan    0.1000 62180.3279
##      9   2464930.9088          nan    0.1000 44168.1112
##     10   2398934.1586          nan    0.1000 47926.7904
##     20   2170581.3449          nan    0.1000 9311.6193
##     40   2016947.8662          nan    0.1000 -30221.6708
##     60   1977830.9542          nan    0.1000 -2021.8299
##     80   1933714.0051          nan    0.1000 -43807.8712
##    100   1929162.3685          nan    0.1000 2977.5284
##    120   1894140.8353          nan    0.1000 -39427.7828
##    140   1831778.3475          nan    0.1000 -42132.6252
##    150   1818224.7436          nan    0.1000 -33458.5980
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2876259.6733          nan    0.1000 109117.4571
##      2   2777101.6181          nan    0.1000 104796.6916
##      3   2671290.1454          nan    0.1000 80012.5131
##      4   2622700.6754          nan    0.1000 64620.6262
##      5   2556401.5828          nan    0.1000 40893.1502
##      6   2538644.8804          nan    0.1000 11084.3636
##      7   2439612.1109          nan    0.1000 8874.7503
##      8   2405329.3311          nan    0.1000 41699.0528
##      9   2366154.7296          nan    0.1000 35217.7479
##     10   2365140.2912          nan    0.1000 -12932.7728
##     20   2186843.0150          nan    0.1000 6063.3175
##     40   2073352.0534          nan    0.1000 -13730.0570
##     60   1951937.5550          nan    0.1000 -26163.7056
##     80   1899221.0845          nan    0.1000 -19310.0739
##    100   1875792.9283          nan    0.1000 -1048.5587
##    120   1874103.3576          nan    0.1000 -48168.9582
##    140   1861519.2260          nan    0.1000 -5159.9767
##    150   1848759.5356          nan    0.1000 -9711.1705
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2963379.9423          nan    0.1000 141120.1790
##      2   2760918.8959          nan    0.1000 83605.5304
##      3   2687744.9644          nan    0.1000 86258.4624
##      4   2560992.2218          nan    0.1000 7855.3328
##      5   2490425.2302          nan    0.1000 78658.7690
##      6   2470806.9762          nan    0.1000 17014.8362
##      7   2436519.2723          nan    0.1000 40187.4054
##      8   2386496.4656          nan    0.1000 30507.9897
##      9   2335416.3162          nan    0.1000 -52072.9567

```

```

##      10  2291919.9987          nan    0.1000 -11321.2304
##      20  2144835.9068          nan    0.1000 -26409.3777
##      40  2037800.1382          nan    0.1000 -8722.5378
##      60  1966333.8169          nan    0.1000   641.6366
##      80  1961748.0299          nan    0.1000 -8799.3376
##     100  1945402.4169          nan    0.1000  2002.2537
##     120  1950222.4886          nan    0.1000 -42690.7447
##     140  1896170.7304          nan    0.1000 -36899.8643
##     150  1885198.9517          nan    0.1000  1320.2230
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   3030933.0877          nan    0.1000  68570.0080
##      2   2875292.0306          nan    0.1000  140830.2562
##      3   2800110.9008          nan    0.1000  80324.9104
##      4   2707871.4047          nan    0.1000  87131.1746
##      5   2671263.9096          nan    0.1000  44708.4612
##      6   2597957.9288          nan    0.1000  84327.4065
##      7   2536071.3039          nan    0.1000  56196.7439
##      8   2505694.6860          nan    0.1000  38746.9429
##      9   2466450.0356          nan    0.1000  38742.4664
##     10   2422447.9169          nan    0.1000  32334.3412
##     20   2174762.6753          nan    0.1000  5063.7143
##     40   2061629.6886          nan    0.1000  -944.5556
##     60   2030621.1161          nan    0.1000 -37539.2272
##     80   1972957.5489          nan    0.1000   978.7011
##    100   1946980.4708          nan    0.1000 -20765.9907
##    120   1943531.0780          nan    0.1000 -31807.3283
##    140   1898958.7899          nan    0.1000 -65084.5811
##    150   1893002.0459          nan    0.1000  4027.3433
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2949162.4384          nan    0.1000 129252.7966
##      2   2864592.4857          nan    0.1000 111588.3005
##      3   2789564.5597          nan    0.1000  90780.7552
##      4   2763187.3270          nan    0.1000  25872.5521
##      5   2730612.5029          nan    0.1000  40973.6162
##      6   2616499.0796          nan    0.1000  86770.7470
##      7   2486495.9093          nan    0.1000  16148.7231
##      8   2480732.8121          nan    0.1000 -3968.8755
##      9   2424848.4335          nan    0.1000  40329.1124
##     10   2351700.8324          nan    0.1000 -20094.4653
##     20   2183613.2281          nan    0.1000 -32611.3544
##     40   2088827.9657          nan    0.1000 -25213.5626
##     60   2058422.1464          nan    0.1000  8477.9510
##     80   2029953.7020          nan    0.1000 -29844.5876
##    100   1969542.5735          nan    0.1000 -4243.2481
##    120   1948775.9907          nan    0.1000 -1536.6292
##    140   1935485.3945          nan    0.1000 -26858.3941
##    150   1919517.4529          nan    0.1000 -7052.3764
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2892998.4125          nan    0.1000  81864.7977
##      2   2815803.3886          nan    0.1000  93813.3290
##      3   2734395.3060          nan    0.1000  86728.0827

```

##	4	2625758.9667	nan	0.1000	15871.0494
##	5	2554504.5041	nan	0.1000	-2391.2500
##	6	2466633.2086	nan	0.1000	31052.3478
##	7	2431342.5639	nan	0.1000	41447.5516
##	8	2413488.6563	nan	0.1000	20312.4830
##	9	2391294.6386	nan	0.1000	11985.1313
##	10	2345905.3577	nan	0.1000	810.2119
##	20	2235962.4208	nan	0.1000	-29357.4719
##	40	2179127.1643	nan	0.1000	-3274.7280
##	60	2090034.6824	nan	0.1000	10593.0056
##	80	2041007.4035	nan	0.1000	-9759.6622
##	100	2020114.0612	nan	0.1000	-4178.8091
##	120	1995235.1777	nan	0.1000	-38965.0057
##	140	1978349.6331	nan	0.1000	-12798.7344
##	150	1987815.7731	nan	0.1000	-25546.7150
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2941540.8773	nan	0.1000	114825.0713
##	2	2844671.2762	nan	0.1000	107431.5772
##	3	2765829.9220	nan	0.1000	88861.6210
##	4	2703337.2912	nan	0.1000	69251.6452
##	5	2684489.8364	nan	0.1000	13179.5413
##	6	2665029.9457	nan	0.1000	21579.5381
##	7	2632013.8494	nan	0.1000	48041.9193
##	8	2562183.0483	nan	0.1000	49446.5545
##	9	2521513.3756	nan	0.1000	34850.8537
##	10	2497368.1526	nan	0.1000	31691.1484
##	20	2297109.3580	nan	0.1000	12429.0545
##	40	2135988.1294	nan	0.1000	5311.4495
##	60	2099365.2326	nan	0.1000	-4066.6073
##	80	2042432.8761	nan	0.1000	-1656.5895
##	100	2052048.0403	nan	0.1000	-5301.4741
##	120	1990455.5920	nan	0.1000	-876.4276
##	140	1957699.2222	nan	0.1000	-2538.3523
##	150	1956275.1120	nan	0.1000	-923.8947
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2999722.8742	nan	0.1000	72116.9727
##	2	2890936.3532	nan	0.1000	115258.8624
##	3	2782039.9402	nan	0.1000	119690.7824
##	4	2629887.4700	nan	0.1000	51726.2668
##	5	2554695.4399	nan	0.1000	62398.8788
##	6	2515135.6975	nan	0.1000	48385.4495
##	7	2473614.7912	nan	0.1000	45127.5600
##	8	2445047.0581	nan	0.1000	31792.7833
##	9	2418246.7176	nan	0.1000	29467.3930
##	10	2420103.9499	nan	0.1000	-18748.8949
##	20	2275122.0078	nan	0.1000	831.0861
##	40	2164259.9287	nan	0.1000	-53404.1885
##	60	2109431.4163	nan	0.1000	-46211.3122
##	80	2082801.4512	nan	0.1000	223.3291
##	100	2040734.1457	nan	0.1000	-8797.6789
##	120	1999650.9739	nan	0.1000	-8694.8075
##	140	1991301.0609	nan	0.1000	-36573.0710

```

##      150  1977171.6351          nan      0.1000  -88.6500
##
## Iter    TrainDeviance  ValidDeviance  StepSize    Improve
##      1    2893009.8477          nan      0.1000  128629.7775
##      2    2770810.5606          nan      0.1000  130701.6158
##      3    2606852.3370          nan      0.1000  60785.6761
##      4    2583376.0419          nan      0.1000  22427.1497
##      5    2516737.5060          nan      0.1000  64297.5170
##      6    2489114.4295          nan      0.1000  37005.7128
##      7    2448512.7200          nan      0.1000  45202.6147
##      8    2343332.5601          nan      0.1000  8330.1315
##      9    2341528.6196          nan      0.1000 -12515.2574
##     10    2298228.6475          nan      0.1000  53184.9007
##     20    2069800.8097          nan      0.1000    -6.5835
##     40    1999976.1632          nan      0.1000 -15882.4510
##     60    1965918.9734          nan      0.1000  3774.1931
##     80    1872066.8920          nan      0.1000 -28458.8167
##    100    1860647.0335          nan      0.1000 -3235.3553
##    120    1856317.1905          nan      0.1000 -41077.0408
##    140    1842813.2663          nan      0.1000 -18819.3022
##    150    1828137.0478          nan      0.1000 -12085.2300
##
## Iter    TrainDeviance  ValidDeviance  StepSize    Improve
##      1    2813812.8572          nan      0.1000  93270.1817
##      2    2762328.5667          nan      0.1000  55597.2092
##      3    2714736.8272          nan      0.1000  53688.9122
##      4    2600268.5466          nan      0.1000  95434.7173
##      5    2485703.1372          nan      0.1000  20220.8908
##      6    2412323.3893          nan      0.1000  15795.3197
##      7    2399542.5405          nan      0.1000  7182.3577
##      8    2330826.1732          nan      0.1000   502.3398
##      9    2269563.0306          nan      0.1000 -2153.4356
##     10    2192336.3331          nan      0.1000  17383.9194
##     20    2075783.3665          nan      0.1000  20047.6946
##     40    1942293.8583          nan      0.1000 -6199.8140
##     60    1871777.6953          nan      0.1000 -71787.6049
##     80    1827937.9663          nan      0.1000 -2268.3595
##    100    1817062.9094          nan      0.1000   443.7220
##    120    1796090.0584          nan      0.1000 -39187.6281
##    140    1795351.9576          nan      0.1000 -15817.1609
##    150    1789946.6953          nan      0.1000 -5148.2871
##
## Iter    TrainDeviance  ValidDeviance  StepSize    Improve
##      1    2838428.3585          nan      0.1000  84567.7069
##      2    2646033.6284          nan      0.1000  63164.8311
##      3    2585009.4319          nan      0.1000  61187.4993
##      4    2491723.7670          nan      0.1000  82971.7000
##      5    2430122.5765          nan      0.1000  52645.9711
##      6    2409165.6398          nan      0.1000  20535.6869
##      7    2323150.4957          nan      0.1000  4263.7366
##      8    2317782.4684          nan      0.1000 -6243.6781
##      9    2317073.6287          nan      0.1000 -15538.7996
##     10    2276832.8440          nan      0.1000  34334.6587
##     20    2031592.6804          nan      0.1000  8718.6121

```



```

##      40 1963292.6677      nan    0.1000 -41609.1117
##      60 1903512.1646      nan    0.1000 -13838.3788
##      80 1814473.3203      nan    0.1000 -5820.3902
##     100 1803319.7328      nan    0.1000 -481.6812
##     120 1797032.2782      nan    0.1000 1270.2110
##     140 1782316.2975      nan    0.1000 -11013.8280
##     150 1781384.8565      nan    0.1000 -3488.7943
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1  1029682.7753      nan    0.1000 49600.8153
##      2   940611.9002      nan    0.1000 56812.0701
##      3   904411.9231      nan    0.1000 45167.5774
##      4   841532.7492      nan    0.1000 48091.0908
##      5   822176.0781      nan    0.1000 23195.1345
##      6   795873.0196      nan    0.1000 31961.4334
##      7   779085.7523      nan    0.1000 19005.3120
##      8   745893.3159      nan    0.1000 22110.3731
##      9   721335.5883      nan    0.1000 6666.2477
##     10   703912.7736      nan    0.1000 11352.4893
##     20   609298.7782      nan    0.1000 2405.2606
##     40   577485.8104      nan    0.1000 -2285.9493
##     60   560064.5844      nan    0.1000 -7194.7952
##     80   555011.7748      nan    0.1000 -658.3092
##    100   550992.6644      nan    0.1000 -5152.5835
##    120   543425.9828      nan    0.1000 -4419.9719
##    140   530157.6079      nan    0.1000 -2253.0268
##    150   527607.6948      nan    0.1000 -2638.4524
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1  1028341.0165      nan    0.1000 69489.0779
##      2   935340.4867      nan    0.1000 59719.0708
##      3   886568.1203      nan    0.1000 35267.4232
##      4   824815.7036      nan    0.1000 43019.4285
##      5   783084.3813      nan    0.1000 28230.8542
##      6   754502.2317      nan    0.1000 33980.8022
##      7   743078.1504      nan    0.1000 13399.2335
##      8   726154.1302      nan    0.1000 20650.4740
##      9   716410.5689      nan    0.1000 10905.5714
##     10   686460.6478      nan    0.1000 11180.5172
##     20   596191.3489      nan    0.1000 -1950.6238
##     40   557836.0433      nan    0.1000 -2500.3810
##     60   548734.7363      nan    0.1000 -2525.9858
##     80   545867.3061      nan    0.1000 -9738.4193
##    100   541119.1099      nan    0.1000 -8105.4613
##    120   539045.9097      nan    0.1000 -3861.4786
##    140   533913.0445      nan    0.1000 -4971.4825
##    150   532206.9183      nan    0.1000 -4146.7950
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1  1025336.9168      nan    0.1000 73081.6556
##      2   944991.8552      nan    0.1000 57672.6837
##      3   913193.0485      nan    0.1000 40962.7211
##      4   868331.1746      nan    0.1000 50440.3914
##      5   810784.9100      nan    0.1000 34553.5263

```

##	6	773282.8798	nan	0.1000	19594.2559
##	7	761583.2209	nan	0.1000	13669.7956
##	8	740307.9201	nan	0.1000	15071.5922
##	9	732215.0281	nan	0.1000	8757.3229
##	10	717311.4706	nan	0.1000	3895.9963
##	20	632755.1816	nan	0.1000	-8528.0391
##	40	610538.2020	nan	0.1000	-3688.9605
##	60	589126.3715	nan	0.1000	-6053.5343
##	80	575052.6890	nan	0.1000	-5568.2188
##	100	571210.6280	nan	0.1000	-1418.2441
##	120	565234.2199	nan	0.1000	-7249.8576
##	140	559560.5471	nan	0.1000	-5377.0800
##	150	559091.4202	nan	0.1000	-2268.0644
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2881468.1474	nan	0.1000	80013.1146
##	2	2812590.3545	nan	0.1000	85635.5688
##	3	2777417.4392	nan	0.1000	28189.8568
##	4	2694044.3785	nan	0.1000	83693.9609
##	5	2543199.3174	nan	0.1000	33268.9190
##	6	2504210.5524	nan	0.1000	40561.4521
##	7	2456858.3951	nan	0.1000	35179.9781
##	8	2400234.3151	nan	0.1000	6066.3034
##	9	2398539.3839	nan	0.1000	-16469.2506
##	10	2373833.3949	nan	0.1000	15301.9402
##	20	2167086.7782	nan	0.1000	3071.3594
##	40	2098241.4839	nan	0.1000	-19798.4215
##	60	2021570.6947	nan	0.1000	-50200.5862
##	80	2002345.6647	nan	0.1000	-611.8821
##	100	1977137.0888	nan	0.1000	-30084.6609
##	120	1930012.2126	nan	0.1000	-53060.7212
##	140	1895141.8892	nan	0.1000	-18559.6491
##	150	1880450.4664	nan	0.1000	-7908.3212
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2969840.0641	nan	0.1000	141298.0195
##	2	2794102.0210	nan	0.1000	54324.4936
##	3	2703482.7374	nan	0.1000	66232.5995
##	4	2642628.7952	nan	0.1000	67215.3037
##	5	2615394.8058	nan	0.1000	28256.1849
##	6	2544192.0277	nan	0.1000	48389.3158
##	7	2490145.8694	nan	0.1000	34064.4384
##	8	2483169.3134	nan	0.1000	-5333.2274
##	9	2448562.2156	nan	0.1000	35223.3139
##	10	2402575.4606	nan	0.1000	-12417.7093
##	20	2220916.6705	nan	0.1000	-44811.8831
##	40	2079629.8898	nan	0.1000	-18586.2744
##	60	2016900.7216	nan	0.1000	-34200.8606
##	80	1991037.4192	nan	0.1000	-43923.8915
##	100	1993424.3061	nan	0.1000	-50558.5890
##	120	1947515.7572	nan	0.1000	-68636.5667
##	140	1913565.4518	nan	0.1000	-65237.4467
##	150	1898679.5590	nan	0.1000	-31048.3489
##					

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2935432.1157	nan	0.1000	108097.3202
##	2	2819491.1126	nan	0.1000	132235.3635
##	3	2782286.9488	nan	0.1000	36713.5422
##	4	2642380.3348	nan	0.1000	48782.7365
##	5	2565195.3409	nan	0.1000	60807.0496
##	6	2503583.4889	nan	0.1000	50802.2061
##	7	2444713.7938	nan	0.1000	16045.4644
##	8	2448184.0875	nan	0.1000	-29607.5392
##	9	2443189.5658	nan	0.1000	-6714.6151
##	10	2404166.5410	nan	0.1000	49706.1273
##	20	2350990.6839	nan	0.1000	22336.7609
##	40	2100037.6334	nan	0.1000	-22466.9778
##	60	2036802.9119	nan	0.1000	-32557.8208
##	80	2020750.6147	nan	0.1000	-21722.9700
##	100	1999005.6478	nan	0.1000	-3478.9048
##	120	1996711.4084	nan	0.1000	-4297.9883
##	140	1978649.6732	nan	0.1000	294.9916
##	150	1967240.3131	nan	0.1000	-62187.7345
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2909422.8789	nan	0.1000	151329.7142
##	2	2835789.1764	nan	0.1000	91199.4752
##	3	2786884.8359	nan	0.1000	56286.7607
##	4	2659437.2842	nan	0.1000	97634.0406
##	5	2628708.5010	nan	0.1000	33913.7755
##	6	2612174.9303	nan	0.1000	7092.9918
##	7	2556021.4069	nan	0.1000	66377.6452
##	8	2547697.7391	nan	0.1000	1638.4490
##	9	2483541.0904	nan	0.1000	48114.0894
##	10	2423759.5606	nan	0.1000	37344.0464
##	20	2136837.2794	nan	0.1000	10681.3551
##	40	1979722.4541	nan	0.1000	-11584.3038
##	60	1921072.4858	nan	0.1000	-5932.4095
##	80	1915247.0202	nan	0.1000	-44088.1423
##	100	1869022.2086	nan	0.1000	-4919.6585
##	120	1862662.0139	nan	0.1000	-8262.3171
##	140	1833209.0618	nan	0.1000	-5589.8490
##	150	1809570.7400	nan	0.1000	78.3065
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3032807.3919	nan	0.1000	51905.7565
##	2	2913114.2890	nan	0.1000	99135.2908
##	3	2829150.9320	nan	0.1000	96616.1696
##	4	2700477.5539	nan	0.1000	105529.1990
##	5	2643011.6440	nan	0.1000	65281.1316
##	6	2553096.6310	nan	0.1000	73006.4714
##	7	2502301.4488	nan	0.1000	34824.6302
##	8	2492646.9563	nan	0.1000	5419.0778
##	9	2433435.7052	nan	0.1000	47651.2486
##	10	2395615.2199	nan	0.1000	25880.5312
##	20	2225567.8837	nan	0.1000	3570.8681
##	40	2180477.5972	nan	0.1000	-2678.3940
##	60	2083715.2070	nan	0.1000	-27891.2658

##	80	2054651.3390	nan	0.1000	3113.7108
##	100	1996696.5409	nan	0.1000	-31296.9486
##	120	1936382.7313	nan	0.1000	743.3873
##	140	1940526.2217	nan	0.1000	-2262.8388
##	150	1940101.5937	nan	0.1000	-69940.7493
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3014706.3808	nan	0.1000	95378.2473
##	2	2906957.6775	nan	0.1000	124913.3467
##	3	2857859.8820	nan	0.1000	58073.6979
##	4	2762981.3839	nan	0.1000	75253.9012
##	5	2652210.1584	nan	0.1000	14947.6797
##	6	2565430.7142	nan	0.1000	98351.1605
##	7	2466895.0681	nan	0.1000	79242.4496
##	8	2421912.5849	nan	0.1000	29885.3299
##	9	2385158.3658	nan	0.1000	20677.6077
##	10	2303952.5424	nan	0.1000	18347.4709
##	20	2134363.9842	nan	0.1000	6213.1896
##	40	1998980.9333	nan	0.1000	-32131.7984
##	60	1942108.9562	nan	0.1000	-9559.9167
##	80	1886612.6290	nan	0.1000	-1082.3157
##	100	1843693.2757	nan	0.1000	-3295.7247
##	120	1805399.0921	nan	0.1000	-31847.9547
##	140	1783370.3704	nan	0.1000	-530.8753
##	150	1768580.7101	nan	0.1000	-1348.6766
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3038854.5256	nan	0.1000	61370.4236
##	2	2975890.3344	nan	0.1000	78893.0998
##	3	2906849.1319	nan	0.1000	89087.8230
##	4	2742505.3715	nan	0.1000	42656.2945
##	5	2683822.9258	nan	0.1000	71008.1848
##	6	2617524.1257	nan	0.1000	39669.6659
##	7	2485222.2862	nan	0.1000	29207.1978
##	8	2429153.6500	nan	0.1000	63515.8467
##	9	2392951.9360	nan	0.1000	43382.0358
##	10	2357062.3855	nan	0.1000	24748.9242
##	20	2203447.6062	nan	0.1000	11469.5106
##	40	2075462.9299	nan	0.1000	-42664.8013
##	60	2041582.1414	nan	0.1000	-18582.4903
##	80	2049591.9119	nan	0.1000	2731.5266
##	100	1984294.5849	nan	0.1000	-19201.4475
##	120	1957039.4612	nan	0.1000	-18123.3670
##	140	1924417.1953	nan	0.1000	992.1798
##	150	1909840.6833	nan	0.1000	-25836.1898
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3019898.5869	nan	0.1000	92216.5845
##	2	2840767.0957	nan	0.1000	76307.2961
##	3	2779309.9033	nan	0.1000	67750.5622
##	4	2711016.8057	nan	0.1000	65458.5903
##	5	2690149.5946	nan	0.1000	13519.8531
##	6	2552303.6400	nan	0.1000	30412.7545
##	7	2494888.7796	nan	0.1000	68834.8587

```

##      8 2484102.6750      nan    0.1000 7711.7301
##      9 2486086.2675      nan    0.1000 -19767.9267
##     10 2453852.5234      nan    0.1000 33154.5813
##     20 2138117.4024      nan    0.1000 -17653.1799
##     40 2061015.2283      nan    0.1000 -23205.8164
##     60 1991804.7829      nan    0.1000 3775.2173
##     80 1926265.5827      nan    0.1000 -64509.5815
##    100 1915157.7307      nan    0.1000 -4086.0511
##    120 1884418.0274      nan    0.1000 -3217.2552
##    140 1847531.1697      nan    0.1000 -1438.5898
##    150 1848287.3989      nan    0.1000 -29475.9671
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2950011.3167      nan    0.1000 145190.5254
##      2 2760888.6938      nan    0.1000 80810.3120
##      3 2673676.3000      nan    0.1000 93014.0236
##      4 2619242.5000      nan    0.1000 56945.8305
##      5 2556263.0647      nan    0.1000 42074.3568
##      6 2510321.7729      nan    0.1000 53806.4154
##      7 2432340.4355      nan    0.1000 10722.5986
##      8 2430045.2070      nan    0.1000 -14492.1858
##      9 2372132.1346      nan    0.1000 -7366.5993
##     10 2375775.3475      nan    0.1000 -23724.7425
##     20 2215833.9838      nan    0.1000 -33406.3443
##     40 2162766.0145      nan    0.1000 -34351.7269
##     60 2078526.0673      nan    0.1000 -23503.1332
##     80 1992508.1563      nan    0.1000 -1093.0625
##    100 1977094.2301      nan    0.1000 -3941.6160
##    120 1952210.0964      nan    0.1000 -1256.5428
##    140 1905559.7834      nan    0.1000 -10535.3124
##    150 1891163.0335      nan    0.1000 -5688.2580
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2730530.2713      nan    0.1000 121613.5842
##      2 2580704.5269      nan    0.1000 129697.3224
##      3 2430381.1387      nan    0.1000 45643.2206
##      4 2358294.1189      nan    0.1000 79691.6565
##      5 2340176.6254      nan    0.1000 15605.0852
##      6 2278381.1437      nan    0.1000 70515.4095
##      7 2189240.5697      nan    0.1000 25102.0909
##      8 2185103.7885      nan    0.1000 -5394.1528
##      9 2121754.9693      nan    0.1000 -12246.5444
##     10 2095177.4182      nan    0.1000 31575.7257
##     20 1917330.1848      nan    0.1000 -25630.5757
##     40 1846093.0564      nan    0.1000 -17090.7622
##     50 1849396.0543      nan    0.1000 494.4342

```

gbm1

```

## Stochastic Gradient Boosting
##
## 61 samples
## 3 predictor
##

```

```
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE      Rsquared  MAE
##   1                  50      1010.966  0.8249911  571.2535
##   1                  100      1054.100  0.8371555  585.4725
##   1                  150      1024.901  0.8667286  552.3807
##   2                   50      1010.350  0.8575585  557.7797
##   2                  100      1046.985  0.8593534  568.5074
##   2                  150      1053.486  0.8588568  578.0264
##   3                   50      1010.362  0.8472539  564.3698
##   3                  100      1038.869  0.8615773  564.2031
##   3                  150      1055.289  0.8588360  567.2579
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 50, interaction.depth =
## 2, shrinkage = 0.1 and n.minobsinnode = 10.
```

Predicting gbm on test1

```
gbmPreds <- predict(gbm1, newdata = test1_rem_out)
summary(gbmPreds)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
##  9.966    35.707    40.684    587.565  1340.268  2091.828
```

postResample to test if it will do well on new data or if overfitting

```
postResample(gbmPreds, test1_rem_out$Volume)
```

```
##      RMSE      Rsquared      MAE
## 266.4904990  0.9105057 172.6952417
```

awesome step! provides comparison of predictions to actual within same DF!

```
compare_gbm1 <- data.frame(test1_rem_out, gbmPreds)
```

CV RMSE=1010, R2=.858

PostResample RMSE=266, R2=.911

Gradient Boosting Manual Tuning

Set seed

```
set.seed(123)

# Creating dataframe for manual tuning
gbmGrid <- expand.grid(n.trees = c(10,50,100),
                      interaction.depth = c(2,3),
                      shrinkage = c(.1, .15, .2),
                      n.minobsinnode = c(5,10,15))

gbm2 <- train(Volume ~ x4StarReviews + x2StarReviews + PositiveServiceReview,
              data = train1,
              method = 'gbm',
              trControl = control1,
              tuneGrid = gbmGrid,
              preProc = c('center','scale'))
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2180127.9842         nan      0.1000  76902.4013
##      2   2034912.4109         nan      0.1000  56433.8236
##      3   1982344.6911         nan      0.1000  67251.2191
##      4   1942620.6077         nan      0.1000  49945.7343
##      5   1894182.4928         nan      0.1000  64209.4036
##      6   1868292.1778         nan      0.1000  33163.8444
##      7   1847220.4830         nan      0.1000  26700.8313
##      8   1741692.7922         nan      0.1000  16274.2389
##      9   1668386.1650         nan      0.1000  3723.7348
##     10   1664177.6778         nan      0.1000   644.8657
##     20   1450131.5560         nan      0.1000 -25317.1220
##     40   1178538.4698         nan      0.1000  5030.6048
##     60    972341.2749         nan      0.1000 -4889.4390
##     80    905709.8446         nan      0.1000 -3830.8056
##    100    826742.6740         nan      0.1000 -20030.9123
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2261149.4817         nan      0.1000  90433.1178
##      2   2177478.6574         nan      0.1000  49051.6495
##      3   2103729.9013         nan      0.1000  29071.1127
##      4   2083433.2241         nan      0.1000  16149.0064
##      5   2054782.7983         nan      0.1000  35585.4858
##      6   2015850.8286         nan      0.1000  51383.4386
##      7   1943725.5287         nan      0.1000  30669.8238
##      8   1936072.1908         nan      0.1000  4334.8826
##      9   1928693.0550         nan      0.1000  7035.9848
##     10   1886640.9086         nan      0.1000 -204.5888
##     20   1773051.6864         nan      0.1000 -5397.9315
```

##	40	1731097.0050	nan	0.1000	-6959.5081
##	60	1690813.3705	nan	0.1000	-21227.1997
##	80	1662740.3941	nan	0.1000	-35703.0838
##	100	1640362.2217	nan	0.1000	-13902.5905

Warning: model fit failed for Fold01.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2282467.8602	nan	0.1000	120227.5384
##	2	2086591.5412	nan	0.1000	117559.2005
##	3	2045076.6315	nan	0.1000	54174.3854
##	4	1927166.0931	nan	0.1000	23675.1457
##	5	1825798.3263	nan	0.1000	30091.2068
##	6	1740744.4407	nan	0.1000	19986.0914
##	7	1666634.2593	nan	0.1000	10022.5984
##	8	1594820.3149	nan	0.1000	787.8302
##	9	1585076.4321	nan	0.1000	7467.1579
##	10	1531606.2544	nan	0.1000	-516.7593
##	20	1354313.4053	nan	0.1000	-27916.0864
##	40	1249552.9256	nan	0.1000	-23367.2109
##	60	1170866.9353	nan	0.1000	-22670.6896
##	80	1054110.5791	nan	0.1000	-26932.5390
##	100	997758.6612	nan	0.1000	-41396.0208

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2235689.3567	nan	0.1000	86367.1801
##	2	2136714.4073	nan	0.1000	60733.5424
##	3	2053310.7134	nan	0.1000	44999.9054
##	4	1966111.6725	nan	0.1000	39878.7685
##	5	1914760.5689	nan	0.1000	20025.4302
##	6	1903120.5341	nan	0.1000	8745.3593
##	7	1856587.1622	nan	0.1000	17176.0449
##	8	1830402.1461	nan	0.1000	2099.5355
##	9	1826575.2734	nan	0.1000	-2948.7164
##	10	1797406.2588	nan	0.1000	-8940.5789
##	20	1737983.9543	nan	0.1000	460.7451
##	40	1688243.9004	nan	0.1000	-21905.4218
##	60	1646020.6396	nan	0.1000	-12329.0470
##	80	1631187.5325	nan	0.1000	-18839.0761
##	100	1628347.7467	nan	0.1000	-29018.9320

Warning: model fit failed for Fold01.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2258806.7265	nan	0.1500	138357.1829
##	2	2054498.8980	nan	0.1500	54926.3522
##	3	1978534.1986	nan	0.1500	96897.6654
##	4	1796383.8864	nan	0.1500	26214.0404
##	5	1768937.0296	nan	0.1500	35917.5581
##	6	1732048.3604	nan	0.1500	50964.9661
##	7	1709632.4364	nan	0.1500	32617.0273


```
##      8 1687171.7191      nan    0.1500 34046.0393
##      9 1582609.7744      nan    0.1500 17216.0017
##     10 1491724.6209      nan    0.1500 -16523.3813
##     20 1310171.1586      nan    0.1500 -26408.6800
##     40 1111832.5901      nan    0.1500 -82605.2671
##     60  963841.8336      nan    0.1500 -41546.3294
##     80  766152.0515      nan    0.1500 -33865.8510
##    100  724746.9078      nan    0.1500 -43642.9373
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2258045.4892      nan    0.1500 139371.6678
##      2 2068979.2863      nan    0.1500 90383.8812
##      3 1969456.0325      nan    0.1500 54343.4767
##      4 1937565.5848      nan    0.1500 42218.8189
##      5 1922609.3014      nan    0.1500 16536.2678
##      6 1850228.9934      nan    0.1500 -13952.7050
##      7 1811737.5060      nan    0.1500 -11069.3555
##      8 1813589.1561      nan    0.1500 -19186.4835
##      9 1820646.6074      nan    0.1500 -26685.2958
##     10 1771410.0682      nan    0.1500 -13089.1126
##     20 1735292.8040      nan    0.1500 -27178.5466
##     40 1674529.0421      nan    0.1500 -24760.0106
##     60 1647180.2412      nan    0.1500 -39889.3217
##     80 1610368.5332      nan    0.1500 -37546.9575
##    100 1609242.5193      nan    0.1500 -24073.9877
```

```
## Warning: model fit failed for Fold01.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2237987.2294      nan    0.1500 174786.7199
##      2 2037152.9849      nan    0.1500 53372.8210
##      3 1965538.2947      nan    0.1500 98209.4617
##      4 1790140.3157      nan    0.1500 16381.3992
##      5 1755043.0743      nan    0.1500 44626.6898
##      6 1716923.3018      nan    0.1500 53228.9139
##      7 1702074.2203      nan    0.1500 19658.2546
##      8 1611043.3471      nan    0.1500 1374.9116
##      9 1512424.3990      nan    0.1500 -2463.3861
##     10 1454324.7212      nan    0.1500 -26585.0643
##     20 1289408.9731      nan    0.1500 -32626.1698
##     40 1107989.9298      nan    0.1500 -42500.8545
##     60  915505.0102      nan    0.1500 -50087.9903
##     80  779799.1204      nan    0.1500 -20946.1355
##    100  668153.2179      nan    0.1500  386.3101
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2269868.1296      nan    0.1500 127445.9599
##      2 2106391.8579      nan    0.1500 54547.4295
##      3 2070443.4245      nan    0.1500 41102.0116
##      4 1985456.2065      nan    0.1500 4001.2345
##      5 1967399.4479      nan    0.1500 18090.4345
##      6 1952352.3492      nan    0.1500 15288.4073
##      7 1888088.8185      nan    0.1500 -2819.6285
```

##	8	1891401.8635	nan	0.1500	-18064.6530
##	9	1854346.2043	nan	0.1500	3444.0486
##	10	1829381.2127	nan	0.1500	-11405.7597
##	20	1769039.7110	nan	0.1500	-61440.1105
##	40	1699909.1131	nan	0.1500	-22220.2317
##	60	1655015.4387	nan	0.1500	-44986.9403
##	80	1641512.8609	nan	0.1500	-33869.2882
##	100	1618103.7943	nan	0.1500	-24397.7984

Warning: model fit failed for Fold01.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2212503.1939	nan	0.2000	212519.8889
##	2	1905090.8333	nan	0.2000	118615.3251
##	3	1697661.4963	nan	0.2000	48238.9335
##	4	1570732.7830	nan	0.2000	11030.6268
##	5	1476991.5655	nan	0.2000	-12006.7231
##	6	1478364.7514	nan	0.2000	-28874.8731
##	7	1426201.0455	nan	0.2000	-79467.7326
##	8	1343198.4661	nan	0.2000	19299.6858
##	9	1335030.6021	nan	0.2000	-103660.2740
##	10	1323749.1486	nan	0.2000	-54115.2290
##	20	1150950.8947	nan	0.2000	-85679.1428
##	40	935839.0597	nan	0.2000	-54208.7630
##	60	822021.8634	nan	0.2000	-28705.0969
##	80	646171.9120	nan	0.2000	-18320.6220
##	100	597260.2117	nan	0.2000	-19269.3696

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2218969.2787	nan	0.2000	73106.0643
##	2	2043442.3179	nan	0.2000	113604.5466
##	3	1913533.3604	nan	0.2000	6041.4507
##	4	1862092.0604	nan	0.2000	65057.3791
##	5	1867456.3152	nan	0.2000	-28656.2570
##	6	1855630.5123	nan	0.2000	14275.2810
##	7	1802461.6282	nan	0.2000	-81202.7680
##	8	1789613.2336	nan	0.2000	-29897.2969
##	9	1790474.9390	nan	0.2000	-35408.8228
##	10	1793624.4978	nan	0.2000	-14854.6043
##	20	1772476.6731	nan	0.2000	-31822.5385
##	40	1734726.2114	nan	0.2000	-46583.7533
##	60	1701439.6249	nan	0.2000	-35718.5167
##	80	1675261.4827	nan	0.2000	-33662.6766
##	100	1637323.6302	nan	0.2000	-29445.8009

Warning: model fit failed for Fold01.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2216742.7634	nan	0.2000	201234.1473
##	2	2107633.3016	nan	0.2000	137564.4446
##	3	2024094.3483	nan	0.2000	116229.2062

```
##      4 1794512.2466      nan    0.2000 33542.0564
##      5 1639841.5664      nan    0.2000 9088.7953
##      6 1546891.3287      nan    0.2000 -60690.7744
##      7 1540179.1083      nan    0.2000 3652.7636
##      8 1463490.3489      nan    0.2000 -69726.9077
##      9 1444346.7715      nan    0.2000 -94332.4408
##     10 1441748.5345      nan    0.2000 -38903.8117
##     20 1265224.4109      nan    0.2000 -28808.7291
##     40 1020423.4463      nan    0.2000 -9879.0677
##     60  993770.3008      nan    0.2000 -40001.3488
##     80  775345.2614      nan    0.2000 -53402.3369
##    100  655461.9111      nan    0.2000 -30115.3610
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2099358.4596      nan    0.2000 147465.9283
##      2 2020394.2864      nan    0.2000 99021.4702
##      3 1916157.6729      nan    0.2000 18551.6382
##      4 1841393.0529      nan    0.2000 -372.2466
##      5 1834785.6111      nan    0.2000 -11971.4570
##      6 1822833.1735      nan    0.2000 1431.3257
##      7 1787553.7827      nan    0.2000 -23090.4004
##      8 1761179.7403      nan    0.2000 -10083.3556
##      9 1765239.7958      nan    0.2000 -28695.2222
##     10 1768460.6781      nan    0.2000 -15774.3912
##     20 1752104.7724      nan    0.2000 -29834.5478
##     40 1695824.9671      nan    0.2000 -34922.8363
##     60 1686147.6760      nan    0.2000 -30495.0517
##     80 1657528.9422      nan    0.2000 -11411.3061
##    100 1647909.5570      nan    0.2000 -11028.7693
```

```
## Warning: model fit failed for Fold01.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 3019339.0363      nan    0.1000 172069.7669
##      2 2795970.3530      nan    0.1000 163392.2923
##      3 2717142.9869      nan    0.1000 99925.2931
##      4 2573275.8860      nan    0.1000 168832.4164
##      5 2528831.7231      nan    0.1000 54000.6042
##      6 2496168.6109      nan    0.1000 37131.0910
##      7 2305962.8090      nan    0.1000 21910.1438
##      8 2151263.8564      nan    0.1000 11398.9289
##      9 2137525.4178      nan    0.1000 12590.6770
##     10 2078233.0689      nan    0.1000 71730.3351
##     20 1644216.1764      nan    0.1000 16928.9856
##     40 1419111.2806      nan    0.1000 6847.0948
##     60 1225338.2168      nan    0.1000 -58733.0395
##     80 1107457.3164      nan    0.1000  921.3864
##    100 1026397.8013      nan    0.1000 -37831.5622
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 3023828.1577      nan    0.1000 151105.2785
##      2 2910555.2452      nan    0.1000 87557.5404
##      3 2794282.5835      nan    0.1000 101040.7165
```

##	4	2766751.8809	nan	0.1000	23202.0740
##	5	2707791.4609	nan	0.1000	58286.1871
##	6	2618405.0639	nan	0.1000	73937.2243
##	7	2576622.8819	nan	0.1000	50347.6663
##	8	2565591.2458	nan	0.1000	6221.5201
##	9	2498433.6786	nan	0.1000	-11183.5583
##	10	2505763.4513	nan	0.1000	-38481.2563
##	20	2328646.5392	nan	0.1000	-28006.2162
##	40	2263998.5412	nan	0.1000	-54545.1529
##	60	2188551.5924	nan	0.1000	-9629.8032
##	80	2163574.6322	nan	0.1000	-2898.2320
##	100	2158714.7947	nan	0.1000	-13991.9298

Warning: model fit failed for Fold02.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3091633.0362	nan	0.1000	144543.2516
##	2	2801075.0998	nan	0.1000	54922.1539
##	3	2735370.1845	nan	0.1000	82929.0197
##	4	2626532.8604	nan	0.1000	133200.7961
##	5	2421945.6673	nan	0.1000	8398.6444
##	6	2277532.4517	nan	0.1000	105777.3041
##	7	2223975.5625	nan	0.1000	56282.4768
##	8	2161272.0692	nan	0.1000	73650.9944
##	9	2043748.1319	nan	0.1000	-9152.7025
##	10	1947743.4474	nan	0.1000	-12764.7310
##	20	1622103.2062	nan	0.1000	-80765.3920
##	40	1412501.8240	nan	0.1000	-85510.4284
##	60	1294037.6568	nan	0.1000	-50812.2398
##	80	1242294.3482	nan	0.1000	-55046.8976
##	100	1150057.6245	nan	0.1000	-3437.2287

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3121398.3849	nan	0.1000	94060.0797
##	2	3057491.2465	nan	0.1000	70766.1112
##	3	3012005.7678	nan	0.1000	49045.2399
##	4	2822286.2297	nan	0.1000	46967.9076
##	5	2755880.1337	nan	0.1000	82079.8124
##	6	2688209.3134	nan	0.1000	49120.1028
##	7	2626739.5425	nan	0.1000	36453.4239
##	8	2598451.5321	nan	0.1000	31498.4192
##	9	2588228.5083	nan	0.1000	6777.3450
##	10	2479975.7105	nan	0.1000	13387.3402
##	20	2301962.4020	nan	0.1000	-35299.4039
##	40	2175070.0831	nan	0.1000	-55713.7465
##	60	2110600.5849	nan	0.1000	-5106.8257
##	80	2085032.5287	nan	0.1000	-42255.8112
##	100	2089185.5441	nan	0.1000	-2945.8921

Warning: model fit failed for Fold02.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
----	------	---------------	---------------	----------	---------

```
##      1 2953545.0107      nan    0.1500 278289.1430
##      2 2724546.8333      nan    0.1500 255373.6327
##      3 2639537.0371      nan    0.1500 109405.7324
##      4 2577838.7099      nan    0.1500 87004.6484
##      5 2529272.7111      nan    0.1500 65006.8089
##      6 2256847.2257      nan    0.1500 -19280.6447
##      7 2148093.1678      nan    0.1500 129972.8570
##      8 2016518.2368      nan    0.1500 80710.5946
##      9 1929452.4289      nan    0.1500 28147.6472
##     10 1858283.9930      nan    0.1500 -144337.0691
##     20 1625222.3881      nan    0.1500 -96715.1918
##     40 1312219.1244      nan    0.1500 -152590.4812
##     60 1304651.7909      nan    0.1500 11413.3723
##     80 1147202.1319      nan    0.1500 -744.4846
##    100 1075441.4118      nan    0.1500 -2740.0106
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2976534.4292      nan    0.1500 230194.4120
##      2 2922646.7620      nan    0.1500 50808.3512
##      3 2739042.4182      nan    0.1500 63606.6770
##      4 2621023.8071      nan    0.1500 -24549.3409
##      5 2565085.6109      nan    0.1500 34081.2836
##      6 2523058.0318      nan    0.1500 47351.2296
##      7 2413996.4281      nan    0.1500 14675.1313
##      8 2365973.5346      nan    0.1500 -89620.5003
##      9 2346747.9696      nan    0.1500 6405.1098
##     10 2330655.4427      nan    0.1500 -56184.2190
##     20 2289685.9665      nan    0.1500 -12309.4018
##     40 2259562.5029      nan    0.1500 -99838.9125
##     60 2143713.0382      nan    0.1500 -226.5344
##     80 2102163.0406      nan    0.1500 -10531.7881
##    100 2118549.4062      nan    0.1500 -5195.1262
```

```
## Warning: model fit failed for Fold02.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2833875.2402      nan    0.1500 295116.2151
##      2 2739780.1223      nan    0.1500 109622.7462
##      3 2392700.9537      nan    0.1500 42864.5280
##      4 2115094.9855      nan    0.1500 -30224.9116
##      5 1901045.6828      nan    0.1500 29609.1749
##      6 1746456.4415      nan    0.1500 -65121.3509
##      7 1629364.5306      nan    0.1500 -50884.0490
##      8 1600550.8595      nan    0.1500 31012.6500
##      9 1535500.7832      nan    0.1500 -12304.1448
##     10 1477006.7774      nan    0.1500 -70585.4532
##     20 1335455.4520      nan    0.1500 -53717.5130
##     40 1134531.1839      nan    0.1500 2008.2668
##     60 1081956.9206      nan    0.1500 -48582.6277
##     80 1068386.3029      nan    0.1500 -55122.4414
##    100  923025.1825      nan    0.1500 -2198.4778
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
```

##	1	3084666.0597	nan	0.1500	130630.1763
##	2	2999843.1114	nan	0.1500	101101.7602
##	3	2830651.2287	nan	0.1500	187324.0032
##	4	2791352.4858	nan	0.1500	38838.6356
##	5	2666848.5525	nan	0.1500	78530.8517
##	6	2549989.9153	nan	0.1500	-1180.3540
##	7	2457951.3813	nan	0.1500	82955.6480
##	8	2469952.9217	nan	0.1500	-51142.0723
##	9	2396572.4201	nan	0.1500	41481.7374
##	10	2325670.7962	nan	0.1500	-46910.1452
##	20	2258613.9294	nan	0.1500	-71150.4141
##	40	2172047.9802	nan	0.1500	-4818.4162
##	60	2081816.8108	nan	0.1500	2954.8451
##	80	2045119.8115	nan	0.1500	-113108.2595
##	100	2029262.9580	nan	0.1500	-9168.8782

Warning: model fit failed for Fold02.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2600271.8174	nan	0.2000	148937.6132
##	2	2353246.6468	nan	0.2000	84211.4297
##	3	2164584.1751	nan	0.2000	188379.3946
##	4	2071912.4374	nan	0.2000	-13807.4837
##	5	1931476.1418	nan	0.2000	124689.9706
##	6	1752272.3649	nan	0.2000	-121572.7667
##	7	1796139.3326	nan	0.2000	-187135.7122
##	8	1727845.2520	nan	0.2000	60505.1713
##	9	1632248.4755	nan	0.2000	-117536.5223
##	10	1644000.7346	nan	0.2000	-72018.8798
##	20	1517232.4803	nan	0.2000	-21434.0307
##	40	1257437.6019	nan	0.2000	-76766.6474
##	60	1083192.9205	nan	0.2000	-23663.1012
##	80	884862.5895	nan	0.2000	7349.4300
##	100	819700.0239	nan	0.2000	2017.6547

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3106301.6335	nan	0.2000	70983.8850
##	2	2816365.3428	nan	0.2000	19777.8265
##	3	2591509.5083	nan	0.2000	60092.6492
##	4	2470369.7703	nan	0.2000	-2480.1135
##	5	2413352.2632	nan	0.2000	63190.6065
##	6	2380500.3231	nan	0.2000	23584.3993
##	7	2367220.0406	nan	0.2000	12050.7925
##	8	2355589.6169	nan	0.2000	14094.1267
##	9	2369314.3398	nan	0.2000	-88630.4246
##	10	2334654.0569	nan	0.2000	1849.1564
##	20	2286576.1329	nan	0.2000	-42909.1610
##	40	2218463.2042	nan	0.2000	1387.8225
##	60	2143186.0584	nan	0.2000	-107300.3180
##	80	2132182.8636	nan	0.2000	-40593.1567
##	100	2042535.8590	nan	0.2000	-147037.3050

Warning: model fit failed for Fold02.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n

```
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2840012.5711	nan	0.2000	394270.5229
##	2	2508608.8318	nan	0.2000	233855.3443
##	3	2143577.0936	nan	0.2000	4340.6801
##	4	2041899.5361	nan	0.2000	124890.9340
##	5	2055785.8694	nan	0.2000	-61192.9957
##	6	2087294.8198	nan	0.2000	-100991.4789
##	7	2015170.8613	nan	0.2000	91792.0337
##	8	1909207.8257	nan	0.2000	10463.3474
##	9	1822075.1381	nan	0.2000	94256.2817
##	10	1792762.3564	nan	0.2000	30723.2031
##	20	1501809.1850	nan	0.2000	2708.5297
##	40	1119564.3657	nan	0.2000	16598.4183
##	60	938532.2650	nan	0.2000	7849.5609
##	80	852287.0519	nan	0.2000	3365.4191
##	100	706373.5629	nan	0.2000	-5145.3700

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3039567.4531	nan	0.2000	197300.5632
##	2	2768373.6345	nan	0.2000	79871.5064
##	3	2714092.0628	nan	0.2000	54799.0937
##	4	2672863.2835	nan	0.2000	39335.3208
##	5	2622373.7659	nan	0.2000	61508.0160
##	6	2574026.1070	nan	0.2000	60115.1501
##	7	2495540.7466	nan	0.2000	65278.0374
##	8	2426404.8190	nan	0.2000	33276.9995
##	9	2378018.6442	nan	0.2000	45956.7754
##	10	2379488.8619	nan	0.2000	-5437.2695
##	20	2301942.2697	nan	0.2000	-26305.0844
##	40	2187871.1445	nan	0.2000	38974.5459
##	60	2114990.1123	nan	0.2000	-104475.6309
##	80	2110341.7012	nan	0.2000	-117723.9031
##	100	2109162.2409	nan	0.2000	-104489.5815

```
## Warning: model fit failed for Fold02.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
```

```
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2848789.6173	nan	0.1000	209254.1070
##	2	2588491.3783	nan	0.1000	81767.8050
##	3	2546258.0510	nan	0.1000	48879.1654
##	4	2340070.2916	nan	0.1000	-19632.1951
##	5	2315222.3787	nan	0.1000	18990.7048
##	6	2201238.9223	nan	0.1000	78092.3502
##	7	2051497.8237	nan	0.1000	24069.4170
##	8	2048695.3095	nan	0.1000	-11513.2774
##	9	1965020.8819	nan	0.1000	47671.5381
##	10	1890789.2231	nan	0.1000	55970.5841
##	20	1667849.7594	nan	0.1000	9952.5813
##	40	1381919.7097	nan	0.1000	-18459.6197
##	60	1204429.7330	nan	0.1000	2006.4813

```
##      80      969551.2492          nan      0.1000 -26765.4520
##      100      816335.0424          nan      0.1000 -43501.0698
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      2960504.1127          nan      0.1000 134171.8394
##      2      2829467.7449          nan      0.1000 140932.7992
##      3      2791834.4923          nan      0.1000 30044.4099
##      4      2689130.3708          nan      0.1000 111502.7063
##      5      2606645.9030          nan      0.1000 74668.8251
##      6      2520058.6321          nan      0.1000 73156.7314
##      7      2455421.8646          nan      0.1000 44397.3983
##      8      2408292.5065          nan      0.1000 34194.5486
##      9      2404907.8970          nan      0.1000 -6066.2228
##     10      2370002.9440          nan      0.1000 24106.5916
##     20      2089174.7754          nan      0.1000 -131.1338
##     40      2015844.9416          nan      0.1000   251.7140
##     60      1989123.4814          nan      0.1000 -44407.5873
##     80      1980821.3084          nan      0.1000 -1445.8321
##    100      1953252.1098          nan      0.1000 -26073.3351
```

```
## Warning: model fit failed for Fold03.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      2862504.3704          nan      0.1000 168079.5178
##      2      2711987.8986          nan      0.1000 62619.3940
##      3      2651491.9442          nan      0.1000 74824.9731
##      4      2608820.1078          nan      0.1000 48193.7384
##      5      2494319.5207          nan      0.1000 135164.8978
##      6      2354573.5884          nan      0.1000 95409.9313
##      7      2329492.7033          nan      0.1000 30387.3212
##      8      2246606.0253          nan      0.1000 99801.0128
##      9      2180293.5631          nan      0.1000 74567.0845
##     10      2124270.2576          nan      0.1000 65833.2631
##     20      1733607.9077          nan      0.1000 -67948.4558
##     40      1401576.3623          nan      0.1000 10476.5852
##     60      1132907.4829          nan      0.1000 -43104.8081
##     80      997087.1519          nan      0.1000 -13764.9802
##    100      920430.0967          nan      0.1000 -6173.2467
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      2934245.3105          nan      0.1000 77250.7800
##      2      2870778.5849          nan      0.1000 75174.5891
##      3      2741177.1640          nan      0.1000 66234.7546
##      4      2626097.3060          nan      0.1000 17011.5816
##      5      2557727.1679          nan      0.1000 -43470.2494
##      6      2452286.7621          nan      0.1000 23568.6348
##      7      2431977.0708          nan      0.1000 12989.9424
##      8      2397447.9777          nan      0.1000 -50300.6637
##      9      2390584.8206          nan      0.1000 -16763.0768
##     10      2373086.1992          nan      0.1000 19953.8663
##     20      2188746.1503          nan      0.1000 -30557.1691
##     40      2075336.7018          nan      0.1000 -1323.3996
##     60      2015302.7673          nan      0.1000 -6664.4766
```



```
##      80 1965757.1504          nan    0.1000 -11121.2049
##     100 1940720.9570          nan    0.1000 -42845.3998
```

```
## Warning: model fit failed for Fold03.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2855296.6554          nan    0.1500 238363.0559
##      2 2559452.2093          nan    0.1500 239752.0508
##      3 2259794.8224          nan    0.1500 -29592.2451
##      4 2124645.7203          nan    0.1500 26275.9866
##      5 2093537.3923          nan    0.1500 25652.6608
##      6 2017887.4886          nan    0.1500 82842.0238
##      7 2032452.8963          nan    0.1500 -50759.4410
##      8 2030890.4293          nan    0.1500 -12764.7576
##      9 1930883.5612          nan    0.1500 55454.4760
##     10 1947542.8896          nan    0.1500 -51189.5363
##     20 1429782.4920          nan    0.1500 27499.2584
##     40 1273584.9444          nan    0.1500 -52158.1118
##     60 1064450.6635          nan    0.1500 -62970.0069
##     80  908167.0831          nan    0.1500 -28563.9742
##    100  711929.9206          nan    0.1500   628.2012
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2993161.9535          nan    0.1500 85114.2854
##      2 2897435.3176          nan    0.1500 120818.9360
##      3 2684718.9527          nan    0.1500 77460.5774
##      4 2547336.3253          nan    0.1500 6894.5064
##      5 2500255.3490          nan    0.1500 54694.9721
##      6 2419369.5353          nan    0.1500 -49128.6974
##      7 2422309.1698          nan    0.1500 -33116.3805
##      8 2356629.7186          nan    0.1500 44985.7768
##      9 2346726.2131          nan    0.1500 11901.5963
##     10 2329742.0865          nan    0.1500 4846.1132
##     20 2222764.8150          nan    0.1500 -25692.9371
##     40 2119880.1543          nan    0.1500 -57886.8891
##     60 1926028.5194          nan    0.1500 -28175.6773
##     80 1858377.3442          nan    0.1500 -2602.2171
##    100 1815959.5255          nan    0.1500 -18596.5156
```

```
## Warning: model fit failed for Fold03.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2933521.6942          nan    0.1500 185820.3989
##      2 2563060.5883          nan    0.1500 1653.7399
##      3 2297982.8774          nan    0.1500 43706.6784
##      4 2231902.7007          nan    0.1500 79133.9613
##      5 2095139.0232          nan    0.1500 34817.0234
##      6 1883395.3255          nan    0.1500 -11577.0108
##      7 1890526.5618          nan    0.1500 -46405.9516
##      8 1897918.1688          nan    0.1500 -35444.0561
##      9 1859293.2534          nan    0.1500 -52596.1868
```

```

##      10  1813653.2632          nan    0.1500  54662.5523
##      20  1582588.9367          nan    0.1500 -45602.3694
##      40  1138933.0874          nan    0.1500 -135962.5144
##      60   890883.2555          nan    0.1500 -21684.6417
##      80   740157.6574          nan    0.1500  6832.3452
##     100   654560.5953          nan    0.1500 -9806.0803
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2871642.9289          nan    0.1500  228734.0744
##      2   2792297.3170          nan    0.1500  101353.4709
##      3   2649048.5247          nan    0.1500  125425.4328
##      4   2494109.5497          nan    0.1500 -8112.3276
##      5   2398681.0277          nan    0.1500 -28996.2899
##      6   2342726.7621          nan    0.1500 -66035.6522
##      7   2346819.9880          nan    0.1500 -50854.1196
##      8   2295678.6153          nan    0.1500 -29205.9068
##      9   2286358.6985          nan    0.1500  9343.2466
##     10   2189991.7096          nan    0.1500  5763.7933
##     20   2144407.7603          nan    0.1500 -42642.1244
##     40   2043443.9245          nan    0.1500 -39922.4065
##     60   1944310.0289          nan    0.1500  5353.3465
##     80   1904172.4731          nan    0.1500 -95196.0368
##    100   1900609.6433          nan    0.1500 -51680.3481

```

```

## Warning: model fit failed for Fold03.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

```

```

## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2913961.8815          nan    0.2000  205006.8631
##      2   2627607.8271          nan    0.2000  325572.3575
##      3   2410409.7990          nan    0.2000  276287.1698
##      4   2350552.8043          nan    0.2000  80142.9915
##      5   2033149.2446          nan    0.2000 -26850.7382
##      6   1792193.2588          nan    0.2000 -51995.1065
##      7   1754785.3832          nan    0.2000  48728.9434
##      8   1670521.8541          nan    0.2000 -6335.4589
##      9   1570675.7729          nan    0.2000  62695.9944
##     10   1531227.9269          nan    0.2000  8788.1331
##     20   1348089.5220          nan    0.2000 -37090.1980
##     40   1005966.0568          nan    0.2000  1576.2221
##     60    732121.0056          nan    0.2000 -66085.9669
##     80    561973.1984          nan    0.2000 -18909.3327
##    100    475402.0195          nan    0.2000  9680.0063
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2676505.8337          nan    0.2000  156692.9887
##      2   2415188.8630          nan    0.2000  50582.8056
##      3   2299043.3429          nan    0.2000 -52283.2824
##      4   2268968.7269          nan    0.2000 -314.0650
##      5   2234687.9118          nan    0.2000  38457.0416
##      6   2236823.0603          nan    0.2000 -36958.3320
##      7   2210825.0709          nan    0.2000  34432.3282
##      8   2067888.9242          nan    0.2000  13401.1892
##      9   2061416.0875          nan    0.2000  7253.0777

```

##	10	2054533.8946	nan	0.2000	10538.8449
##	20	1994763.8913	nan	0.2000	5511.0114
##	40	1843570.3134	nan	0.2000	-1479.0454
##	60	1815298.0270	nan	0.2000	-56066.9171
##	80	1810722.9103	nan	0.2000	-4945.6702
##	100	1768828.0340	nan	0.2000	-19379.8464

Warning: model fit failed for Fold03.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2893744.3545	nan	0.2000	238991.9032
##	2	2436894.4544	nan	0.2000	45556.3450
##	3	2156204.7654	nan	0.2000	214976.1502
##	4	2120570.6600	nan	0.2000	40181.5039
##	5	2135699.7785	nan	0.2000	-61869.8574
##	6	2036164.8722	nan	0.2000	-25428.3016
##	7	1961648.0774	nan	0.2000	75929.6704
##	8	1877415.8473	nan	0.2000	26012.5494
##	9	1891725.0736	nan	0.2000	-79671.9380
##	10	1726722.7287	nan	0.2000	-81985.6904
##	20	1627157.2358	nan	0.2000	-70968.5045
##	40	1158683.5498	nan	0.2000	-40560.5567
##	60	894764.1074	nan	0.2000	-84008.2448
##	80	685102.7772	nan	0.2000	-22124.2123
##	100	511509.4575	nan	0.2000	-23773.4267

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2785249.6407	nan	0.2000	243396.4281
##	2	2667842.1242	nan	0.2000	117989.6563
##	3	2406350.1547	nan	0.2000	47540.8638
##	4	2362370.2027	nan	0.2000	39397.6179
##	5	2303618.8382	nan	0.2000	53543.0059
##	6	2275921.3830	nan	0.2000	29499.9009
##	7	2218095.0150	nan	0.2000	-77080.8450
##	8	2203904.7121	nan	0.2000	7495.8028
##	9	2191945.3676	nan	0.2000	3823.5079
##	10	2175037.3650	nan	0.2000	17043.4847
##	20	2066891.3143	nan	0.2000	-11516.9200
##	40	1949680.3330	nan	0.2000	-66013.3610
##	60	1869835.1667	nan	0.2000	-39342.6427
##	80	1855899.1418	nan	0.2000	-8012.1382
##	100	1845362.9551	nan	0.2000	-68937.6781

Warning: model fit failed for Fold03.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2915131.2302	nan	0.1000	85335.4601
##	2	2832495.8262	nan	0.1000	104175.8206
##	3	2702101.5780	nan	0.1000	148582.3555
##	4	2656289.0885	nan	0.1000	58386.8617
##	5	2474171.8026	nan	0.1000	134774.6167

```
##      6 2435518.8798      nan    0.1000 51854.3688
##      7 2241160.8327      nan    0.1000 21921.0975
##      8 2129059.0262      nan    0.1000 83306.0669
##      9 2003680.2048      nan    0.1000 -49848.7517
##     10 1960795.0652      nan    0.1000 53902.8852
##     20 1653786.0677      nan    0.1000 -116337.4049
##     40 1316111.0200      nan    0.1000 -7758.4780
##     60 1178946.3295      nan    0.1000 5529.5768
##     80 1140166.0834      nan    0.1000 -58489.8701
##    100 1003421.1270      nan    0.1000 -68759.4030
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 3024408.4243      nan    0.1000 82636.0547
##      2 2913397.7946      nan    0.1000 88988.1931
##      3 2813310.6016      nan    0.1000 107714.1075
##      4 2671539.6385      nan    0.1000 24253.0945
##      5 2609134.0992      nan    0.1000 34319.8380
##      6 2591147.0593      nan    0.1000 14685.7506
##      7 2544549.4192      nan    0.1000 30376.9550
##      8 2479698.5597      nan    0.1000 -16630.9155
##      9 2461163.9079      nan    0.1000 17149.5802
##     10 2440762.9311      nan    0.1000 23559.3408
##     20 2267355.1708      nan    0.1000 -48608.3305
##     40 2191077.1668      nan    0.1000 -13491.7977
##     60 2071507.3096      nan    0.1000 -9799.4550
##     80 2014222.2091      nan    0.1000 -7938.8869
##    100 1969374.5533      nan    0.1000 -22665.6747
```

```
## Warning: model fit failed for Fold04.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2779066.3128      nan    0.1000 103093.7070
##      2 2523912.0288      nan    0.1000 64902.6447
##      3 2390791.4416      nan    0.1000 163217.7135
##      4 2280924.9771      nan    0.1000 34092.1998
##      5 2200263.3448      nan    0.1000 90207.0634
##      6 2090913.0566      nan    0.1000 79382.5685
##      7 1994323.9900      nan    0.1000 70951.8300
##      8 1879148.1787      nan    0.1000 -21693.6239
##      9 1836217.0785      nan    0.1000 53801.6358
##     10 1788088.9909      nan    0.1000 10341.4330
##     20 1551050.7376      nan    0.1000 15237.0326
##     40 1344555.7326      nan    0.1000 3418.5906
##     60 1241013.7566      nan    0.1000 -33277.7593
##     80 1107858.3256      nan    0.1000 -7514.6531
##    100 998814.9146      nan    0.1000 -3141.6952
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2962287.1622      nan    0.1000 93036.8159
##      2 2887208.0584      nan    0.1000 74973.0136
##      3 2800834.0759      nan    0.1000 98929.8806
##      4 2744250.0609      nan    0.1000 58711.6179
##      5 2724629.8991      nan    0.1000 9707.1749
```

##	6	2644328.5535	nan	0.1000	18221.7606
##	7	2500725.1237	nan	0.1000	48063.3734
##	8	2447132.2254	nan	0.1000	64872.2823
##	9	2423201.5892	nan	0.1000	26969.4702
##	10	2362049.7281	nan	0.1000	-6231.9667
##	20	2233546.8667	nan	0.1000	-33068.7361
##	40	2100538.1579	nan	0.1000	-31261.8762
##	60	2034343.0505	nan	0.1000	-16798.2382
##	80	2003270.9505	nan	0.1000	-5559.0222
##	100	1973087.0189	nan	0.1000	-9351.5449

Warning: model fit failed for Fold04.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2830423.0225	nan	0.1500	100873.4333
##	2	2440954.9671	nan	0.1500	93407.4287
##	3	2290737.0791	nan	0.1500	165694.0624
##	4	2169564.5590	nan	0.1500	140993.4235
##	5	2092706.7113	nan	0.1500	87635.4708
##	6	2019744.8981	nan	0.1500	80910.5194
##	7	1850224.7100	nan	0.1500	-38377.6330
##	8	1820413.8133	nan	0.1500	34644.9586
##	9	1839703.4750	nan	0.1500	-68195.7818
##	10	1723884.9595	nan	0.1500	-50175.9141
##	20	1458502.1796	nan	0.1500	-35277.3393
##	40	1212021.5389	nan	0.1500	-1875.2249
##	60	982270.5592	nan	0.1500	-4660.5439
##	80	929421.7435	nan	0.1500	6308.5469
##	100	814739.7007	nan	0.1500	-9073.9073

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2881492.8337	nan	0.1500	221653.5650
##	2	2811408.8480	nan	0.1500	78148.3932
##	3	2639194.3042	nan	0.1500	36815.7175
##	4	2428493.2239	nan	0.1500	44990.9344
##	5	2367178.2504	nan	0.1500	76872.1135
##	6	2318492.7434	nan	0.1500	60984.0216
##	7	2285831.1009	nan	0.1500	39954.0512
##	8	2257652.9388	nan	0.1500	-1540.9308
##	9	2264370.2271	nan	0.1500	-44638.4692
##	10	2272715.4886	nan	0.1500	-33382.8185
##	20	2077638.9517	nan	0.1500	-62954.5983
##	40	1940885.6162	nan	0.1500	-26200.2599
##	60	1873739.6763	nan	0.1500	-38405.2364
##	80	1805240.5311	nan	0.1500	-6675.9049
##	100	1831379.6938	nan	0.1500	-40127.2855

Warning: model fit failed for Fold04.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2625089.4083	nan	0.1500	201825.7985

```
##      2  2442181.2527      nan    0.1500 194808.8933
##      3  2301082.4489      nan    0.1500 152649.7969
##      4  2058125.0286      nan    0.1500 40161.6075
##      5  1908936.8630      nan    0.1500 -81879.3992
##      6  1835413.6178      nan    0.1500 14077.5445
##      7  1840943.6164      nan    0.1500 -65408.7112
##      8  1800300.1532      nan    0.1500 48948.6749
##      9  1711214.7558      nan    0.1500 55995.2492
##     10  1644557.1866      nan    0.1500 -124426.7345
##     20  1544021.0723      nan    0.1500 -59594.9173
##     40  1475582.0941      nan    0.1500 -128648.1053
##     60  1056841.2187      nan    0.1500 -40807.6845
##     80   960528.8967      nan    0.1500 -139667.2505
##    100   780671.4346      nan    0.1500 -35611.5993
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1  2983419.4664      nan    0.1500 143590.6386
##      2  2720440.7362      nan    0.1500 72563.2340
##      3  2580373.5347      nan    0.1500 153633.1465
##      4  2490750.5550      nan    0.1500 73601.1247
##      5  2414367.5832      nan    0.1500 57729.2140
##      6  2354952.1336      nan    0.1500 28167.0924
##      7  2314096.6487      nan    0.1500 28221.7664
##      8  2246317.2393      nan    0.1500 65774.2893
##      9  2237869.3533      nan    0.1500 9460.2628
##     10  2243131.9411      nan    0.1500 -27759.6287
##     20  2078050.1534      nan    0.1500 1804.5842
##     40  2010132.1126      nan    0.1500 436.9751
##     60  1922892.5447      nan    0.1500 -87044.2574
##     80  1864466.9448      nan    0.1500 -8016.3854
##    100  1846072.7771      nan    0.1500 -5079.4513
```

```
## Warning: model fit failed for Fold04.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1  2744923.0933      nan    0.2000 374413.2088
##      2  2470890.2402      nan    0.2000 110958.1408
##      3  2381681.7024      nan    0.2000 106870.2082
##      4  2330411.2082      nan    0.2000 60372.5605
##      5  2286082.3933      nan    0.2000 46760.6459
##      6  2264503.8757      nan    0.2000 14271.4289
##      7  1984880.7972      nan    0.2000 -53704.4472
##      8  1999940.6618      nan    0.2000 -42999.8739
##      9  1762761.5302      nan    0.2000 -3895.0375
##     10  1665139.9154      nan    0.2000 -164235.8896
##     20  1515607.9635      nan    0.2000 3804.6241
##     40  1102769.2604      nan    0.2000 -755.6577
##     60   936298.2831      nan    0.2000 8097.7520
##     80   791099.4300      nan    0.2000 15032.4164
##    100   770526.9732      nan    0.2000 -48669.7496
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1  2906826.6475      nan    0.2000 209408.3467
```

##	2	2548539.7445	nan	0.2000	127321.4234
##	3	2522897.0854	nan	0.2000	1935.8555
##	4	2494255.4744	nan	0.2000	28260.2238
##	5	2474985.8675	nan	0.2000	18625.6475
##	6	2395577.9313	nan	0.2000	91986.7565
##	7	2393071.3555	nan	0.2000	-8798.3265
##	8	2312616.8797	nan	0.2000	51943.3948
##	9	2329234.5178	nan	0.2000	-48739.1592
##	10	2270417.9665	nan	0.2000	70423.4899
##	20	2121946.8981	nan	0.2000	-7937.1596
##	40	2007204.2904	nan	0.2000	13326.2618
##	60	1955643.6931	nan	0.2000	-20199.9151
##	80	1888777.3482	nan	0.2000	-37701.4965
##	100	1867022.0903	nan	0.2000	-29050.7870

Warning: model fit failed for Fold04.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2901483.5412	nan	0.2000	255532.9077
##	2	2357397.9842	nan	0.2000	89698.3503
##	3	2274347.1110	nan	0.2000	114696.6925
##	4	1943953.0075	nan	0.2000	17314.8202
##	5	1821208.9842	nan	0.2000	-7284.7284
##	6	1746243.5164	nan	0.2000	85971.7236
##	7	1609146.1069	nan	0.2000	-73427.7760
##	8	1536627.7189	nan	0.2000	83013.5808
##	9	1502646.1365	nan	0.2000	38710.8499
##	10	1458503.7270	nan	0.2000	13590.9687
##	20	1257571.0327	nan	0.2000	19094.6093
##	40	1083817.5930	nan	0.2000	-36939.2977
##	60	909446.0867	nan	0.2000	-38241.7155
##	80	772962.1013	nan	0.2000	-23001.2094
##	100	669078.2395	nan	0.2000	-6940.1149

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2870701.0014	nan	0.2000	262075.6724
##	2	2789473.6413	nan	0.2000	92459.8398
##	3	2681977.5986	nan	0.2000	124137.9566
##	4	2470837.8570	nan	0.2000	156313.9797
##	5	2469014.0724	nan	0.2000	-17709.6054
##	6	2337007.1287	nan	0.2000	99776.6261
##	7	2340859.2486	nan	0.2000	-19888.5562
##	8	2249478.3160	nan	0.2000	48114.3196
##	9	2165673.3421	nan	0.2000	-29594.5438
##	10	2156328.0522	nan	0.2000	12340.4470
##	20	2084687.1461	nan	0.2000	-49444.4843
##	40	2096895.1853	nan	0.2000	-149679.0377
##	60	1976688.3573	nan	0.2000	-91113.9142
##	80	1917655.4626	nan	0.2000	-8862.9691
##	100	1905829.6209	nan	0.2000	17277.6194

Warning: model fit failed for Fold04.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2746948.8893	nan	0.1000	99507.6100
##	2	2486913.6775	nan	0.1000	35570.5735
##	3	2334831.4589	nan	0.1000	113129.8067
##	4	2308718.3203	nan	0.1000	22215.7269
##	5	2286599.9982	nan	0.1000	15103.5608
##	6	2129393.2367	nan	0.1000	2231.2456
##	7	2062228.5853	nan	0.1000	80716.5806
##	8	2051138.2251	nan	0.1000	7071.4414
##	9	1995778.4286	nan	0.1000	65982.0156
##	10	1992610.3285	nan	0.1000	-4022.2045
##	20	1623942.2512	nan	0.1000	6802.6620
##	40	1296928.2004	nan	0.1000	-3199.7328
##	60	1142440.7100	nan	0.1000	-76127.3929
##	80	1002892.9230	nan	0.1000	-5234.8265
##	100	960234.0919	nan	0.1000	-31784.2106

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2964211.1121	nan	0.1000	114205.7838
##	2	2776974.1623	nan	0.1000	80851.0891
##	3	2679295.5358	nan	0.1000	83800.4291
##	4	2602444.8408	nan	0.1000	50407.0801
##	5	2586344.2498	nan	0.1000	8504.1837
##	6	2527145.5834	nan	0.1000	48586.9117
##	7	2516306.8976	nan	0.1000	2682.0327
##	8	2482135.6821	nan	0.1000	38815.7915
##	9	2440917.3174	nan	0.1000	31573.5266
##	10	2421963.1813	nan	0.1000	22992.7364
##	20	2225068.5834	nan	0.1000	-9847.6298
##	40	2144390.4735	nan	0.1000	-55844.6958
##	60	2120857.5564	nan	0.1000	-40972.3898
##	80	1999903.5540	nan	0.1000	-10961.6081
##	100	1974220.4000	nan	0.1000	-4856.6188

Warning: model fit failed for Fold05.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2848425.5997	nan	0.1000	158456.1285
##	2	2597113.7246	nan	0.1000	37455.0154
##	3	2355962.3599	nan	0.1000	53225.2854
##	4	2224674.8181	nan	0.1000	98768.5246
##	5	2052573.9880	nan	0.1000	15686.0217
##	6	1929883.9033	nan	0.1000	-380.1644
##	7	1877678.6184	nan	0.1000	63054.3163
##	8	1798630.9236	nan	0.1000	-39128.9230
##	9	1720037.8675	nan	0.1000	48301.0957
##	10	1662217.3077	nan	0.1000	-76281.9355
##	20	1368403.9143	nan	0.1000	-46063.8272
##	40	1250764.1955	nan	0.1000	-7034.5283
##	60	1078442.9652	nan	0.1000	-25519.9406
##	80	892053.0386	nan	0.1000	-52434.5293
##	100	802755.6381	nan	0.1000	-2781.6605

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2879205.9910	nan	0.1000	88415.0876
##	2	2738013.3663	nan	0.1000	120385.8944
##	3	2651916.1067	nan	0.1000	69574.4775
##	4	2596691.7075	nan	0.1000	63074.0901
##	5	2529806.0092	nan	0.1000	42593.6791
##	6	2432178.3455	nan	0.1000	21038.1498
##	7	2401022.6332	nan	0.1000	34617.1245
##	8	2358911.5798	nan	0.1000	38713.4178
##	9	2364209.2504	nan	0.1000	-32899.3763
##	10	2347065.5891	nan	0.1000	20119.7697
##	20	2218298.2027	nan	0.1000	5719.9285
##	40	2127009.6587	nan	0.1000	-41189.2251
##	60	2104672.8226	nan	0.1000	-3502.0658
##	80	2074449.0823	nan	0.1000	-50602.1333
##	100	2030913.8474	nan	0.1000	-28880.2365

Warning: model fit failed for Fold05.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2910332.6897	nan	0.1500	203464.6244
##	2	2806893.7681	nan	0.1500	141494.5270
##	3	2526458.6793	nan	0.1500	213867.9299
##	4	2262923.8927	nan	0.1500	-32600.0192
##	5	2171580.3473	nan	0.1500	109918.7686
##	6	2174364.5613	nan	0.1500	-28813.9885
##	7	2009869.0443	nan	0.1500	-59414.1132
##	8	2018713.6197	nan	0.1500	-38148.5051
##	9	1974743.1069	nan	0.1500	61587.5132
##	10	1917704.2209	nan	0.1500	61100.9383
##	20	1614060.6611	nan	0.1500	-51909.0587
##	40	1301531.8213	nan	0.1500	-71307.2096
##	60	1218443.7520	nan	0.1500	-72334.3118
##	80	893888.1351	nan	0.1500	-16221.1653
##	100	749422.1676	nan	0.1500	4755.2225

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2919136.4557	nan	0.1500	144756.0270
##	2	2799084.4527	nan	0.1500	138092.4678
##	3	2655780.5230	nan	0.1500	105781.9395
##	4	2551828.3089	nan	0.1500	70008.2859
##	5	2461054.9024	nan	0.1500	-21854.3141
##	6	2437444.1006	nan	0.1500	23234.5298
##	7	2407922.4680	nan	0.1500	11454.2631
##	8	2386258.8346	nan	0.1500	2894.3437
##	9	2367115.4286	nan	0.1500	-107863.0435
##	10	2339527.1501	nan	0.1500	14392.9522
##	20	2326228.3038	nan	0.1500	3571.8649
##	40	2164345.7110	nan	0.1500	2497.7327
##	60	2098656.4703	nan	0.1500	-89648.3309
##	80	2087692.4592	nan	0.1500	-53095.9530
##	100	1951324.9499	nan	0.1500	209.7782

```
## Warning: model fit failed for Fold05.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	2803385.1994	nan	0.1500	299908.8748
## 2	2520446.0814	nan	0.1500	182847.4879
## 3	2458675.0084	nan	0.1500	75684.9541
## 4	2194182.3913	nan	0.1500	-29338.7034
## 5	2070101.1431	nan	0.1500	145357.7511
## 6	1887199.4359	nan	0.1500	5879.1354
## 7	1741719.8465	nan	0.1500	9752.4007
## 8	1747742.0583	nan	0.1500	-27195.4761
## 9	1774908.9711	nan	0.1500	-82447.7668
## 10	1771938.6992	nan	0.1500	-309.5455
## 20	1379411.4709	nan	0.1500	26154.9470
## 40	1072493.4919	nan	0.1500	-25590.7524
## 60	922536.4302	nan	0.1500	1736.7189
## 80	763279.7579	nan	0.1500	-526.4461
## 100	650095.0028	nan	0.1500	-76504.6283

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	2858589.3313	nan	0.1500	227081.1667
## 2	2671391.6637	nan	0.1500	86234.5947
## 3	2642697.5925	nan	0.1500	9253.6646
## 4	2632822.5538	nan	0.1500	-24380.5088
## 5	2495683.1907	nan	0.1500	29855.9186
## 6	2492234.3798	nan	0.1500	-18924.7141
## 7	2444605.3792	nan	0.1500	31944.3354
## 8	2349397.2777	nan	0.1500	-11511.0581
## 9	2312274.6717	nan	0.1500	21183.7681
## 10	2321821.3687	nan	0.1500	-44494.8263
## 20	2167364.7110	nan	0.1500	-9004.4434
## 40	2042119.5623	nan	0.1500	-1993.8722
## 60	2035427.3645	nan	0.1500	-1910.0311
## 80	2019980.2389	nan	0.1500	-3284.3923
## 100	1960572.6679	nan	0.1500	-21870.8817

```
## Warning: model fit failed for Fold05.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	2514740.6099	nan	0.2000	87696.2489
## 2	2262171.3783	nan	0.2000	278686.3982
## 3	1992549.9626	nan	0.2000	906.7818
## 4	1831624.1449	nan	0.2000	-94430.6086
## 5	1745970.7717	nan	0.2000	88203.6439
## 6	1684695.2060	nan	0.2000	54917.9216
## 7	1553955.2697	nan	0.2000	-44764.2471
## 8	1509615.3092	nan	0.2000	-117449.2711
## 9	1511541.8745	nan	0.2000	-38251.5558
## 10	1471000.7965	nan	0.2000	8299.0235
## 20	1365344.9130	nan	0.2000	4981.8917
## 40	1221777.5738	nan	0.2000	-70069.6855

```

##      60 1026189.1606      nan    0.2000 11002.9557
##      80  927846.9122      nan    0.2000 -53544.8215
##     100  607774.6626      nan    0.2000 -52569.9769
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2811371.2388      nan    0.2000 286264.1015
##      2 2617217.8769      nan    0.2000 137072.8009
##      3 2445755.9825      nan    0.2000 146148.7133
##      4 2289029.0181      nan    0.2000 24418.5986
##      5 2176383.3377      nan    0.2000 -1901.3746
##      6 2153157.9636      nan    0.2000 25444.8566
##      7 2113331.9549      nan    0.2000 28010.8877
##      8 2091325.8942      nan    0.2000 -9443.8930
##      9 2072676.2997      nan    0.2000 8142.9479
##     10 2070537.3298      nan    0.2000 -3020.6514
##     20 2025404.6580      nan    0.2000 -80158.3443
##     40 1925736.1152      nan    0.2000 -1201.1519
##     60 1900192.8038      nan    0.2000 -60482.8089
##     80 1851750.4736      nan    0.2000 -44549.8402
##    100 1811793.8651      nan    0.2000 -49606.9255

```

```

## Warning: model fit failed for Fold05.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

```

```

## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2520614.0384      nan    0.2000 61460.8045
##      2 2307601.2541      nan    0.2000 231586.5325
##      3 2141899.1457      nan    0.2000 187748.0336
##      4 1980922.4529      nan    0.2000 -132799.1583
##      5 1885475.5061      nan    0.2000 116877.1579
##      6 1701645.5403      nan    0.2000 5033.6134
##      7 1714261.9836      nan    0.2000 -61327.9165
##      8 1677197.1842      nan    0.2000 43487.6282
##      9 1691794.6760      nan    0.2000 -46414.5349
##     10 1638615.5014      nan    0.2000 -32865.7406
##     20 1243273.5503      nan    0.2000 -11315.6556
##     40  837120.8424      nan    0.2000 -66972.3170
##     60  649687.5026      nan    0.2000 -100936.0073
##     80  536426.8006      nan    0.2000 -33811.0473
##    100  414778.9706      nan    0.2000 -3085.2352
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2759064.4408      nan    0.2000 264662.4314
##      2 2623287.9819      nan    0.2000 161601.8013
##      3 2587848.9052      nan    0.2000 16415.2937
##      4 2449861.4424      nan    0.2000 97066.0660
##      5 2387381.5693      nan    0.2000 14973.9923
##      6 2354491.1359      nan    0.2000 28345.0759
##      7 2329073.1175      nan    0.2000 32428.1458
##      8 2343723.7294      nan    0.2000 -61425.0060
##      9 2356765.4966      nan    0.2000 -40579.7698
##     10 2299731.3744      nan    0.2000 49017.1635
##     20 2064590.7689      nan    0.2000 -15703.8275
##     40 2033525.8496      nan    0.2000 -43791.5249

```

```
##      60 1979743.8037      nan    0.2000 -3012.3267
##      80 1940601.1028      nan    0.2000 -8322.5428
##     100 1884849.8142      nan    0.2000  2762.3124
```

```
## Warning: model fit failed for Fold05.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1  2803510.9404      nan    0.1000 189433.7451
##      2  2672115.2515      nan    0.1000  57277.3868
##      3  2544637.2621      nan    0.1000  32051.6686
##      4  2381048.1875      nan    0.1000 114385.5243
##      5  2172561.3948      nan    0.1000  21340.0394
##      6  2145715.4597      nan    0.1000  28940.4825
##      7  2068684.8971      nan    0.1000  92141.5573
##      8  2010506.1627      nan    0.1000  67455.3432
##      9  1929745.1311      nan    0.1000  55187.5945
##     10  1862756.1872      nan    0.1000  41240.7879
##     20  1574954.0429      nan    0.1000  16957.7702
##     40  1259976.4689      nan    0.1000  5636.3790
##     60  1132567.7408      nan    0.1000 -4977.5050
##     80   992035.8811      nan    0.1000 -28128.0812
##    100   875854.0055      nan    0.1000 -14374.5003
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1  2958921.2458      nan    0.1000  98185.7586
##      2  2818991.1788      nan    0.1000 122703.6093
##      3  2679762.4941      nan    0.1000 121164.8231
##      4  2627925.3587      nan    0.1000  67296.3070
##      5  2576086.5048      nan    0.1000  57740.1408
##      6  2568360.4213      nan    0.1000 -9527.9685
##      7  2527757.7712      nan    0.1000  46280.4821
##      8  2434240.5195      nan    0.1000  72179.0807
##      9  2421792.4028      nan    0.1000 14006.5325
##     10  2370081.4132      nan    0.1000  35099.4345
##     20  2079202.0269      nan    0.1000 -46570.3004
##     40  2018330.9277      nan    0.1000 -10078.3995
##     60  1909011.5487      nan    0.1000 -4114.5783
##     80  1856635.2443      nan    0.1000 -1070.9186
##    100  1826658.8999      nan    0.1000 -20362.0647
```

```
## Warning: model fit failed for Fold06.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1  2815878.0221      nan    0.1000 188454.6563
##      2  2691215.9425      nan    0.1000 144336.7397
##      3  2633099.0534      nan    0.1000  75718.2048
##      4  2499008.1331      nan    0.1000 147973.0077
##      5  2458680.6596      nan    0.1000  53669.2300
##      6  2253087.8027      nan    0.1000  35336.6017
##      7  2230144.3551      nan    0.1000  26307.0676
##      8  2073265.5664      nan    0.1000  9185.8928
```

```

##      9 2008073.3146          nan    0.1000 -5599.7302
##     10 2011864.6839          nan    0.1000 -17926.1781
##     20 1551940.0878          nan    0.1000  7368.6694
##     40 1267476.1263          nan    0.1000 -12299.8692
##     60 1112605.8725          nan    0.1000 -35580.5490
##     80  999221.5372          nan    0.1000 -23488.5659
##    100  901796.5025          nan    0.1000 -16522.7659
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2951012.4929          nan    0.1000 93027.0540
##      2 2782026.7389          nan    0.1000 72538.1364
##      3 2707080.5861          nan    0.1000 83266.4354
##      4 2651216.6413          nan    0.1000 57590.8688
##      5 2611349.5162          nan    0.1000 45480.6480
##      6 2542861.5004          nan    0.1000 57079.1689
##      7 2464468.5336          nan    0.1000 -3668.1112
##      8 2387314.8204          nan    0.1000 66049.1081
##      9 2334167.7957          nan    0.1000 -6486.4184
##     10 2281962.9101          nan    0.1000 -5353.8658
##     20 2103180.8996          nan    0.1000 -34825.0252
##     40 2015628.9175          nan    0.1000 3406.0650
##     60 1953464.4275          nan    0.1000 -5937.4754
##     80 1908075.8846          nan    0.1000 2152.7959
##    100 1852201.2807          nan    0.1000 -17690.3946

```

```

## Warning: model fit failed for Fold06.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

```

```

## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2808764.0690          nan    0.1500 96044.1716
##      2 2426050.4456          nan    0.1500 124753.3406
##      3 2165783.4175          nan    0.1500 25322.3561
##      4 2066507.6451          nan    0.1500 117919.5728
##      5 1911584.6812          nan    0.1500 -33429.7618
##      6 1822354.0378          nan    0.1500 -92075.4618
##      7 1727992.8179          nan    0.1500 78136.7156
##      8 1671013.8570          nan    0.1500 63316.7678
##      9 1693832.3992          nan    0.1500 -113928.3138
##     10 1656660.9422          nan    0.1500 39460.2436
##     20 1420841.3419          nan    0.1500 -9977.4260
##     40 1148035.2983          nan    0.1500 -17684.1036
##     60  971910.1425          nan    0.1500 -12712.4693
##     80  848160.3044          nan    0.1500 -8383.9338
##    100  722129.8196          nan    0.1500 -16564.5915
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2904456.9536          nan    0.1500 183086.0865
##      2 2840361.7599          nan    0.1500 56261.7042
##      3 2748214.0115          nan    0.1500 122855.7507
##      4 2499033.9970          nan    0.1500 72048.1967
##      5 2349927.2380          nan    0.1500 -7567.1197
##      6 2348333.3180          nan    0.1500 -20382.1543
##      7 2300253.4725          nan    0.1500 58137.8889
##      8 2250863.3168          nan    0.1500 57191.3117

```

##	9	2254843.5787	nan	0.1500	-19644.4813
##	10	2180614.1452	nan	0.1500	-41592.4190
##	20	2109464.1392	nan	0.1500	1448.3095
##	40	1953928.2095	nan	0.1500	-34487.8751
##	60	1863589.4848	nan	0.1500	-9755.9396
##	80	1792242.2907	nan	0.1500	4282.9993
##	100	1784946.7989	nan	0.1500	3525.8229

Warning: model fit failed for Fold06.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2700259.4272	nan	0.1500	299674.6908
##	2	2619396.7006	nan	0.1500	94712.2187
##	3	2371947.2974	nan	0.1500	192222.5303
##	4	2195110.7592	nan	0.1500	115445.0927
##	5	2170436.6618	nan	0.1500	22051.5706
##	6	2024963.3408	nan	0.1500	104458.7066
##	7	2014649.4773	nan	0.1500	4698.4140
##	8	1900508.0859	nan	0.1500	72275.1176
##	9	1812110.4311	nan	0.1500	44837.3804
##	10	1762641.0592	nan	0.1500	-22313.5350
##	20	1516035.0357	nan	0.1500	-39409.5753
##	40	1159910.6341	nan	0.1500	-128126.2792
##	60	975566.5303	nan	0.1500	-26537.4526
##	80	828743.9846	nan	0.1500	2935.4287
##	100	697250.3441	nan	0.1500	4055.3159

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2836215.3526	nan	0.1500	191678.9174
##	2	2772753.6156	nan	0.1500	64360.2346
##	3	2661179.2395	nan	0.1500	130607.8947
##	4	2642242.1829	nan	0.1500	2615.8739
##	5	2491507.3777	nan	0.1500	24194.8165
##	6	2421001.7240	nan	0.1500	85387.1058
##	7	2422222.3782	nan	0.1500	-20477.6256
##	8	2377221.6396	nan	0.1500	28694.6219
##	9	2279626.3195	nan	0.1500	-8038.7905
##	10	2244491.1017	nan	0.1500	30526.1277
##	20	2072503.8783	nan	0.1500	-2085.5249
##	40	1965958.5396	nan	0.1500	7472.2024
##	60	1905148.0023	nan	0.1500	-39857.8658
##	80	1826178.4182	nan	0.1500	-9570.3003
##	100	1816679.1508	nan	0.1500	-3261.4763

Warning: model fit failed for Fold06.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2819936.7910	nan	0.2000	284269.7605
##	2	2515962.8596	nan	0.2000	260951.6290
##	3	2448815.2721	nan	0.2000	93344.6189
##	4	2118070.8646	nan	0.2000	-27078.6795

```
##      5 2069251.7745      nan 0.2000 61289.7891
##      6 1955636.1313      nan 0.2000 142224.4731
##      7 1873389.6154      nan 0.2000 105882.3661
##      8 1811618.9449      nan 0.2000 79494.3603
##      9 1763353.2207      nan 0.2000 61062.3065
##     10 1710492.4987      nan 0.2000 -46289.7641
##     20 1362745.4569      nan 0.2000 -91301.9604
##     40  980286.4185      nan 0.2000 -111313.5044
##     60  722049.5297      nan 0.2000 -64030.2535
##     80  556975.4194      nan 0.2000 -1968.4894
##    100  458429.4922      nan 0.2000 -29031.0743
```

```
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2811378.5330      nan 0.2000 261453.2638
##      2 2556797.6932      nan 0.2000 93863.1490
##      3 2493292.3144      nan 0.2000 76620.4893
##      4 2412563.5293      nan 0.2000 42630.8699
##      5 2260026.8016      nan 0.2000 20889.1993
##      6 2226477.3192      nan 0.2000 -8548.0289
##      7 2159450.6152      nan 0.2000 -29358.4071
##      8 2150708.6854      nan 0.2000 6443.7508
##      9 2153992.3101      nan 0.2000 -62053.0725
##     10 2119400.0504      nan 0.2000 29997.5471
##     20 2103854.6866      nan 0.2000 -23039.8437
##     40 2006466.7078      nan 0.2000 -80900.2742
##     60 1895941.5715      nan 0.2000 8793.3948
##     80 1800093.2267      nan 0.2000 -12381.0839
##    100 1794388.0636      nan 0.2000 -73496.2727
```

```
## Warning: model fit failed for Fold06.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'
```

```
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2703977.8595      nan 0.2000 397345.2284
##      2 2447418.0724      nan 0.2000 300408.1304
##      3 2394255.1602      nan 0.2000 61202.6544
##      4 2149629.7184      nan 0.2000 160491.2917
##      5 2127272.8451      nan 0.2000 14576.5244
##      6 1862220.4666      nan 0.2000 -82476.3341
##      7 1708630.0184      nan 0.2000 -21929.0428
##      8 1671619.5840      nan 0.2000 -61664.3346
##      9 1687634.8327      nan 0.2000 -57418.5606
##     10 1659548.8644      nan 0.2000 -60715.5755
##     20 1325336.3442      nan 0.2000 -22773.6507
##     40  990651.5292      nan 0.2000 -38335.9164
##     60  772106.9642      nan 0.2000  854.9266
##     80  585517.0383      nan 0.2000 -36096.5148
##    100  543992.7091      nan 0.2000 -42636.3552
```

```
##
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2778157.5781      nan 0.2000 299708.7210
##      2 2606704.5853      nan 0.2000 125884.0703
##      3 2528857.2019      nan 0.2000 105592.0188
##      4 2308623.7140      nan 0.2000 48588.9682
```

##	5	2310703.7279	nan	0.2000	-36007.9347
##	6	2259383.5880	nan	0.2000	65671.7388
##	7	2129470.3555	nan	0.2000	5784.0165
##	8	2106184.2081	nan	0.2000	22023.9088
##	9	2121086.0132	nan	0.2000	-64067.5660
##	10	2097043.8653	nan	0.2000	28498.2521
##	20	1981242.1699	nan	0.2000	-7389.1241
##	40	1937117.6043	nan	0.2000	-11353.5867
##	60	1833505.1727	nan	0.2000	-4111.0980
##	80	1740968.8090	nan	0.2000	-37185.7479
##	100	1715240.2964	nan	0.2000	-3458.6794

Warning: model fit failed for Fold06.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	987697.6505	nan	0.1000	76387.1661
##	2	922774.7468	nan	0.1000	81748.0397
##	3	840243.8938	nan	0.1000	31823.3545
##	4	803805.6021	nan	0.1000	44005.7425
##	5	773603.8617	nan	0.1000	37761.3337
##	6	743000.0542	nan	0.1000	37856.6490
##	7	719566.5686	nan	0.1000	31735.6405
##	8	665237.4201	nan	0.1000	37979.6862
##	9	652404.1978	nan	0.1000	15164.1965
##	10	617863.8217	nan	0.1000	10404.6472
##	20	530173.9082	nan	0.1000	1869.3876
##	40	475699.9350	nan	0.1000	-14045.5067
##	60	427629.1886	nan	0.1000	-726.8511
##	80	380507.2912	nan	0.1000	-2728.9136
##	100	367975.2658	nan	0.1000	-7706.6742

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1038224.0781	nan	0.1000	60818.6546
##	2	969285.2596	nan	0.1000	61427.2272
##	3	935672.0003	nan	0.1000	41779.8908
##	4	900557.8398	nan	0.1000	41933.6281
##	5	868018.0971	nan	0.1000	39138.6324
##	6	807102.0954	nan	0.1000	32460.1159
##	7	778631.2650	nan	0.1000	17997.9693
##	8	753816.2469	nan	0.1000	9546.7891
##	9	740915.1819	nan	0.1000	15180.4228
##	10	730498.9265	nan	0.1000	11410.3190
##	20	641242.5360	nan	0.1000	4282.0777
##	40	594489.1582	nan	0.1000	-2119.0753
##	60	592050.3670	nan	0.1000	-8381.6841
##	80	582958.6294	nan	0.1000	-3104.7031
##	100	563706.2547	nan	0.1000	-10222.4431

Warning: model fit failed for Fold07.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
----	------	---------------	---------------	----------	---------


```
##      1 1008981.2588      nan    0.1000 98126.4352
##      2  953460.5153      nan    0.1000 68814.1768
##      3  859265.5605      nan    0.1000 50432.9667
##      4  793744.8027      nan    0.1000 17793.5464
##      5  720907.0521      nan    0.1000 24441.8931
##      6  672938.6756      nan    0.1000 23157.5958
##      7  633434.7712      nan    0.1000 16090.3964
##      8  620780.2415      nan    0.1000 16567.6617
##      9  586915.2285      nan    0.1000 12164.3242
##     10  579600.5861      nan    0.1000 8351.6722
##     20  467750.6136      nan    0.1000 -9580.1128
##     40  398773.8170      nan    0.1000 -8613.7031
##     60  343173.3936      nan    0.1000 -7091.6989
##     80  339046.2823      nan    0.1000 -1316.5207
##    100  333450.1423      nan    0.1000 -4896.1483
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   997898.6047      nan    0.1000 66154.0477
##      2   941265.1548      nan    0.1000 42519.9280
##      3   893203.4348      nan    0.1000 31651.4423
##      4   868971.7328      nan    0.1000 25222.2965
##      5   813988.7167      nan    0.1000 41213.8995
##      6   786559.4642      nan    0.1000 8508.0044
##      7   736531.0034      nan    0.1000 19961.0369
##      8   716241.2123      nan    0.1000 13963.6052
##      9   703790.6836      nan    0.1000 16453.0053
##     10   700206.3171      nan    0.1000   862.9446
##     20   620646.4307      nan    0.1000 -1595.3875
##     40   581999.9845      nan    0.1000 -3430.1294
##     60   567105.1710      nan    0.1000 -7259.4894
##     80   560114.8945      nan    0.1000 -5099.1777
##    100   557439.3286      nan    0.1000 -16701.7320
```

```
## Warning: model fit failed for Fold07.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   929129.9358      nan    0.1500 95423.7605
##      2   867259.5997      nan    0.1500 70889.0639
##      3   817867.7674      nan    0.1500 60017.2246
##      4   772404.5989      nan    0.1500 59791.2488
##      5   684406.6972      nan    0.1500 38971.8157
##      6   659366.0942      nan    0.1500 31612.9284
##      7   602934.8285      nan    0.1500 9299.6597
##      8   563943.7414      nan    0.1500 -455.4168
##      9   553193.2716      nan    0.1500 13177.2567
##     10   545444.7167      nan    0.1500 10364.2391
##     20   441190.7789      nan    0.1500 -10615.9922
##     40   375734.5724      nan    0.1500 -23384.9434
##     60   320446.1762      nan    0.1500 -6665.0639
##     80   267563.7674      nan    0.1500 -6651.7445
##    100   225201.4051      nan    0.1500 -733.5903
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
```

##	1	994434.8220	nan	0.1500	108169.5084
##	2	925445.7571	nan	0.1500	80641.6048
##	3	831354.5942	nan	0.1500	71875.0372
##	4	806813.9945	nan	0.1500	27685.4919
##	5	765602.2323	nan	0.1500	14774.1252
##	6	750511.9903	nan	0.1500	16437.3938
##	7	698049.6470	nan	0.1500	24802.5203
##	8	671537.8255	nan	0.1500	4306.0441
##	9	668306.6468	nan	0.1500	749.4719
##	10	665186.5618	nan	0.1500	3009.2798
##	20	601585.7349	nan	0.1500	-6659.0731
##	40	577312.8763	nan	0.1500	-4468.6581
##	60	557752.8856	nan	0.1500	-8689.0880
##	80	567231.8343	nan	0.1500	1119.4488
##	100	557751.7313	nan	0.1500	-4401.5207

Warning: model fit failed for Fold07.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	916757.6753	nan	0.1500	94695.2458
##	2	819428.7188	nan	0.1500	55764.6651
##	3	766074.6931	nan	0.1500	67720.1755
##	4	734006.4269	nan	0.1500	44174.6404
##	5	656500.0791	nan	0.1500	22342.0156
##	6	635070.7617	nan	0.1500	27491.9457
##	7	595104.5040	nan	0.1500	8935.4793
##	8	556286.4026	nan	0.1500	11408.8846
##	9	546723.1213	nan	0.1500	12592.4068
##	10	540401.2794	nan	0.1500	7718.7145
##	20	439544.7635	nan	0.1500	-13463.2318
##	40	402899.8898	nan	0.1500	-18434.4287
##	60	358135.1433	nan	0.1500	-10848.0350
##	80	306328.8580	nan	0.1500	-11078.6453
##	100	257719.8438	nan	0.1500	392.7385

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	979041.5901	nan	0.1500	73157.1989
##	2	872247.1939	nan	0.1500	88691.0167
##	3	800661.6691	nan	0.1500	50840.3886
##	4	757741.1220	nan	0.1500	19707.4342
##	5	710564.2040	nan	0.1500	32788.3955
##	6	700800.5535	nan	0.1500	9377.3715
##	7	682675.9967	nan	0.1500	21230.2814
##	8	675799.3581	nan	0.1500	7507.8592
##	9	642983.9888	nan	0.1500	11089.7901
##	10	641099.6927	nan	0.1500	308.4835
##	20	612181.5119	nan	0.1500	-15304.6106
##	40	576181.2512	nan	0.1500	-12978.5107
##	60	559186.0979	nan	0.1500	-1919.7995
##	80	558528.2146	nan	0.1500	-23826.5395
##	100	560489.2635	nan	0.1500	-3229.1845

Warning: model fit failed for Fold07.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n

```
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	938589.4691	nan	0.2000	184211.2985
##	2	834556.4632	nan	0.2000	123116.7710
##	3	709025.6066	nan	0.2000	62759.0681
##	4	628334.5665	nan	0.2000	56586.6067
##	5	574833.9581	nan	0.2000	17878.2440
##	6	562794.0882	nan	0.2000	16804.2154
##	7	553433.3863	nan	0.2000	8702.4835
##	8	547238.2049	nan	0.2000	7269.1153
##	9	525295.5371	nan	0.2000	-20968.9082
##	10	528344.6182	nan	0.2000	-21617.7957
##	20	462953.0632	nan	0.2000	-23163.0919
##	40	351205.9455	nan	0.2000	-21200.9926
##	60	322285.9176	nan	0.2000	-21878.4307
##	80	307975.2323	nan	0.2000	-18037.8057
##	100	276206.4991	nan	0.2000	-7457.6419

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1025332.4216	nan	0.2000	68399.2726
##	2	855478.9995	nan	0.2000	117887.7606
##	3	743179.6813	nan	0.2000	59386.0660
##	4	716420.5975	nan	0.2000	33666.8765
##	5	665778.3273	nan	0.2000	27582.0354
##	6	632528.6921	nan	0.2000	15151.5021
##	7	603407.2958	nan	0.2000	-2153.4316
##	8	582765.9559	nan	0.2000	-336.4716
##	9	581481.8161	nan	0.2000	-3344.0546
##	10	585233.1042	nan	0.2000	-15822.6975
##	20	556377.3713	nan	0.2000	-6610.2991
##	40	540155.2075	nan	0.2000	-13193.9094
##	60	539484.4083	nan	0.2000	-22247.9770
##	80	532300.5484	nan	0.2000	-14510.4185
##	100	529681.6065	nan	0.2000	-11118.0332

```
## Warning: model fit failed for Fold07.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
```

```
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	946519.6110	nan	0.2000	174345.5517
##	2	761469.5264	nan	0.2000	116272.7931
##	3	723293.2105	nan	0.2000	46408.1862
##	4	615674.0683	nan	0.2000	6392.0234
##	5	597746.1322	nan	0.2000	21578.5987
##	6	543361.2523	nan	0.2000	13826.8868
##	7	542161.3337	nan	0.2000	-4460.3908
##	8	504413.0441	nan	0.2000	1721.4300
##	9	475028.6041	nan	0.2000	-7623.1716
##	10	453173.3007	nan	0.2000	-6639.9585
##	20	383773.6041	nan	0.2000	7185.1343
##	40	319589.3895	nan	0.2000	-6573.3494
##	60	288820.4835	nan	0.2000	-6200.5873

```
##      80      269784.8065          nan      0.2000 -15782.7693
##      100      236921.4585          nan      0.2000 -18788.6447
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      942665.4010          nan      0.2000 99051.2778
##      2      834526.6531          nan      0.2000 73923.3347
##      3      772350.2932          nan      0.2000 43037.5360
##      4      709959.3276          nan      0.2000 36612.0689
##      5      670306.7814          nan      0.2000 18907.0500
##      6      659536.7295          nan      0.2000 -10876.0265
##      7      657119.7338          nan      0.2000 -7851.7162
##      8      633417.6056          nan      0.2000 12500.5211
##      9      633023.9584          nan      0.2000 -3229.7508
##     10      616802.8564          nan      0.2000 2132.3100
##     20      599526.4919          nan      0.2000   369.1275
##     40      577339.1619          nan      0.2000 -27671.2664
##     60      546177.2148          nan      0.2000 -15179.8141
##     80      543575.8461          nan      0.2000 -22041.8500
##    100      551683.1996          nan      0.2000 -1873.7160
```

```
## Warning: model fit failed for Fold07.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      2858287.7324          nan      0.1000 188807.0565
##      2      2632778.9199          nan      0.1000 185873.2152
##      3      2407703.3179          nan      0.1000 45771.9595
##      4      2373231.7810          nan      0.1000 35866.9390
##      5      2196045.3158          nan      0.1000 31847.9538
##      6      2084214.1878          nan      0.1000 81651.3711
##      7      2018124.7938          nan      0.1000 6546.0757
##      8      2014632.0003          nan      0.1000 -11094.6481
##      9      2005546.8683          nan      0.1000 6636.0386
##     10      1890809.8132          nan      0.1000 -20769.9659
##     20      1588263.9098          nan      0.1000 -32860.0119
##     40      1321742.4940          nan      0.1000 -7613.7051
##     60      1172279.3812          nan      0.1000 -62971.1171
##     80      1158494.0372          nan      0.1000 -55916.5215
##    100      983097.2955          nan      0.1000 -18943.6401
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      2961027.6958          nan      0.1000 144089.0959
##      2      2906943.7491          nan      0.1000 56081.1180
##      3      2738406.7559          nan      0.1000 50719.5819
##      4      2711980.3754          nan      0.1000 22729.4070
##      5      2668205.8098          nan      0.1000 52665.4621
##      6      2659755.9123          nan      0.1000 -6318.6298
##      7      2641213.4843          nan      0.1000 22302.3560
##      8      2629644.7973          nan      0.1000 9458.5783
##      9      2556933.9505          nan      0.1000 54518.2848
##     10      2559238.2752          nan      0.1000 -16077.6630
##     20      2355554.6710          nan      0.1000 -13316.0790
##     40      2207390.0994          nan      0.1000 -31700.0810
##     60      2103087.2430          nan      0.1000 -43553.8946
```

```
##      80 2038717.3145          nan    0.1000 -42789.2231
##     100 2003431.9449          nan    0.1000 -57888.7469
```

```
## Warning: model fit failed for Fold08.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2806406.4835          nan    0.1000 123320.2778
##      2 2668132.3559          nan    0.1000 157264.2835
##      3 2480442.9517          nan    0.1000 118646.5451
##      4 2346965.8474          nan    0.1000 93575.2096
##      5 2317193.0612          nan    0.1000 36597.3586
##      6 2296889.8691          nan    0.1000 18073.6783
##      7 2212493.2046          nan    0.1000 10978.1276
##      8 2113036.1378          nan    0.1000 72097.0445
##      9 2109686.4693          nan    0.1000 -6402.5859
##     10 2073597.9938          nan    0.1000 44195.2758
##     20 1657923.8123          nan    0.1000 15144.4347
##     40 1379092.9370          nan    0.1000 -92391.5376
##     60 1092820.8272          nan    0.1000 9773.5894
##     80  980371.1689          nan    0.1000 -4069.0758
##    100  856293.7492          nan    0.1000 -20399.5949
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2877999.3375          nan    0.1000 111657.1801
##      2 2817058.2091          nan    0.1000 82427.6504
##      3 2727243.6590          nan    0.1000 82992.9325
##      4 2617119.6107          nan    0.1000 17747.7407
##      5 2543950.2821          nan    0.1000 87103.0638
##      6 2492493.8156          nan    0.1000 42502.6425
##      7 2420216.1977          nan    0.1000 43435.0947
##      8 2413001.4992          nan    0.1000 -4493.3395
##      9 2409912.0901          nan    0.1000 -9924.5953
##     10 2359270.8950          nan    0.1000 27656.0242
##     20 2180240.6657          nan    0.1000 -8247.2911
##     40 2095400.3208          nan    0.1000 -1677.3523
##     60 2037641.4795          nan    0.1000 1155.2448
##     80 1997497.4659          nan    0.1000 -19371.9422
##    100 1961262.9350          nan    0.1000 -112.5868
```

```
## Warning: model fit failed for Fold08.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2644326.7281          nan    0.1500 100238.9853
##      2 2371051.4558          nan    0.1500 -22511.8308
##      3 2197900.4007          nan    0.1500 42995.8062
##      4 2071844.3404          nan    0.1500 141887.1380
##      5 2047411.1448          nan    0.1500 8844.4541
##      6 1867257.4557          nan    0.1500 14038.8820
##      7 1731638.8278          nan    0.1500 -7251.5802
##      8 1639164.3462          nan    0.1500 -72362.2613
##      9 1621101.6899          nan    0.1500 15193.1805
```

```

##      10  1556264.2860          nan    0.1500  18737.9623
##      20  1619262.8402          nan    0.1500 -42119.9740
##      40  1203495.3925          nan    0.1500 -62439.1192
##      60  1034190.8496          nan    0.1500 -19827.0053
##      80   900697.4721          nan    0.1500 -42805.3375
##     100   815288.7262          nan    0.1500 -36917.3037
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2866095.3707          nan    0.1500  195204.2876
##      2   2731530.7912          nan    0.1500  149686.1574
##      3   2636128.4833          nan    0.1500  108272.7711
##      4   2523584.1989          nan    0.1500  90993.0030
##      5   2425944.1114          nan    0.1500 -19347.5372
##      6   2349085.8486          nan    0.1500 -61357.7782
##      7   2323264.7080          nan    0.1500  7945.7407
##      8   2307657.4403          nan    0.1500  17793.2299
##      9   2287906.9672          nan    0.1500  20874.5445
##     10   2270692.5940          nan    0.1500 -75392.2221
##     20   2140109.6576          nan    0.1500 -40307.0252
##     40   2002265.3238          nan    0.1500 -5712.1459
##     60   1985743.7505          nan    0.1500 -810.7387
##     80   1963861.0705          nan    0.1500 -2735.0372
##    100   1940866.5340          nan    0.1500 -16128.3595

```

```

## Warning: model fit failed for Fold08.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

```

```

## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2662351.1510          nan    0.1500  131016.5233
##      2   2361648.5653          nan    0.1500  99635.4495
##      3   2184132.8776          nan    0.1500  139725.7294
##      4   2092881.0844          nan    0.1500  12464.2728
##      5   2003438.4172          nan    0.1500 -1564.9144
##      6   1897347.9222          nan    0.1500  66824.3120
##      7   1735258.1242          nan    0.1500 -7747.9356
##      8   1625305.4814          nan    0.1500 -35539.3538
##      9   1565888.1439          nan    0.1500  66454.4657
##     10   1552239.5874          nan    0.1500  15727.4330
##     20   1391125.2272          nan    0.1500  9868.6923
##     40   1010900.6476          nan    0.1500 -7222.7641
##     60    841947.4035          nan    0.1500 -40114.2232
##     80    684882.2728          nan    0.1500 -2309.5252
##    100    592725.5913          nan    0.1500 -29492.8196
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1   2807181.5949          nan    0.1500  119925.6216
##      2   2690135.9925          nan    0.1500  131568.0074
##      3   2631424.4152          nan    0.1500  58076.2718
##      4   2602945.4974          nan    0.1500  23843.1230
##      5   2474778.1621          nan    0.1500 -18056.7129
##      6   2429832.9437          nan    0.1500  20555.8641
##      7   2372234.3844          nan    0.1500 -49351.8680
##      8   2347942.6906          nan    0.1500  17692.8882
##      9   2354226.8759          nan    0.1500 -63560.0540

```

##	10	2362520.4283	nan	0.1500	-41386.5718
##	20	2272232.5918	nan	0.1500	12915.8076
##	40	2205489.0491	nan	0.1500	-2554.4110
##	60	2062607.0255	nan	0.1500	-9341.5490
##	80	2025145.0322	nan	0.1500	-4731.8256
##	100	1988638.7123	nan	0.1500	-32028.4268

Warning: model fit failed for Fold08.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2640527.6155	nan	0.2000	371622.1434
##	2	2549384.8760	nan	0.2000	103408.2102
##	3	2243715.6047	nan	0.2000	221824.6913
##	4	2031163.9576	nan	0.2000	172161.5673
##	5	1815935.6491	nan	0.2000	-12239.5945
##	6	1736391.3500	nan	0.2000	18118.9230
##	7	1683987.8382	nan	0.2000	-53141.7773
##	8	1637506.0871	nan	0.2000	-18781.0115
##	9	1562343.8540	nan	0.2000	-125848.1568
##	10	1468181.4208	nan	0.2000	-20707.1339
##	20	1269089.5277	nan	0.2000	45672.6863
##	40	994996.7271	nan	0.2000	1537.7885
##	60	790648.9380	nan	0.2000	-33993.5847
##	80	658567.6970	nan	0.2000	-30418.5417
##	100	591501.4370	nan	0.2000	-77997.2751

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2813209.6990	nan	0.2000	253945.0550
##	2	2617968.4170	nan	0.2000	-52567.0844
##	3	2498123.5892	nan	0.2000	70808.9581
##	4	2416131.9002	nan	0.2000	-86096.3218
##	5	2382173.0158	nan	0.2000	19016.4463
##	6	2369576.3223	nan	0.2000	-48101.7139
##	7	2370965.7340	nan	0.2000	-40598.6755
##	8	2352075.8100	nan	0.2000	20614.8214
##	9	2344371.7526	nan	0.2000	8418.5056
##	10	2317096.9800	nan	0.2000	16652.0941
##	20	2260529.5259	nan	0.2000	-100486.1682
##	40	2138245.2784	nan	0.2000	-83588.5319
##	60	2058350.0350	nan	0.2000	489.6362
##	80	2046347.1732	nan	0.2000	-44517.8661
##	100	2001562.1097	nan	0.2000	-55187.5941

Warning: model fit failed for Fold08.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2728866.2259	nan	0.2000	405062.1329
##	2	2375034.8717	nan	0.2000	273206.6680
##	3	2196553.1826	nan	0.2000	213375.2954
##	4	2063772.4793	nan	0.2000	156566.6916
##	5	1856704.5120	nan	0.2000	-43650.8423

```
##      6 1767067.3484      nan    0.2000 -192570.5852
##      7 1718992.4023      nan    0.2000 -86198.9569
##      8 1682474.8949      nan    0.2000 30165.4121
##      9 1577446.0391      nan    0.2000 -33086.3203
##     10 1579546.7153      nan    0.2000 -44160.1742
##     20 1327725.3769      nan    0.2000 -24444.3484
##     40  994455.9875      nan    0.2000 -128312.0406
##     60  832995.8017      nan    0.2000 -18666.4717
##     80  656171.8052      nan    0.2000 -19626.3380
##    100  504663.6893      nan    0.2000 -76198.5549
##
```

```
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 2820325.5933      nan    0.2000 306483.2045
##      2 2730026.0108      nan    0.2000 102782.7726
##      3 2610033.9650      nan    0.2000 141443.1421
##      4 2498929.7339      nan    0.2000 79213.8920
##      5 2471924.4834      nan    0.2000 26293.6919
##      6 2464812.7619      nan    0.2000 -8045.9485
##      7 2366498.0652      nan    0.2000 88833.1830
##      8 2281888.4878      nan    0.2000 52759.5511
##      9 2295991.3423      nan    0.2000 -42504.0943
##     10 2147316.8406      nan    0.2000 -25318.3878
##     20 2008149.8363      nan    0.2000 9255.9611
##     40 1945527.2629      nan    0.2000 -637.7431
##     60 1885415.2148      nan    0.2000 -9853.0990
##     80 1805214.9508      nan    0.2000 -114029.3795
##    100 1781197.4168      nan    0.2000 -5453.8778
```

```
## Warning: model fit failed for Fold08.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 3017706.1054      nan    0.1000 80602.0823
##      2 2724461.3758      nan    0.1000 73265.1667
##      3 2652073.4012      nan    0.1000 95395.2604
##      4 2515563.8638      nan    0.1000 53476.1668
##      5 2413464.6518      nan    0.1000 116640.4588
##      6 2328190.7960      nan    0.1000 100623.1401
##      7 2300402.5540      nan    0.1000 35534.2381
##      8 2138289.7372      nan    0.1000 14775.8949
##      9 2117718.3411      nan    0.1000 27588.1474
##     10 2097458.6906      nan    0.1000 26553.2527
##     20 1689933.1879      nan    0.1000 -36729.3773
##     40 1386362.7551      nan    0.1000 -14426.3968
##     60 1187209.0921      nan    0.1000 -22033.2469
##     80  951824.2308      nan    0.1000 -19508.3046
##    100  823716.2412      nan    0.1000 2027.7977
##
```

```
## Iter  TrainDeviance  ValidDeviance  StepSize  Improve
##      1 3013084.1535      nan    0.1000 95149.7259
##      2 2951123.7822      nan    0.1000 70406.2098
##      3 2798345.4008      nan    0.1000 124493.5196
##      4 2723935.0924      nan    0.1000 88395.1458
##      5 2618069.5979      nan    0.1000 10570.3851
```


##	6	2490128.8544	nan	0.1000	36048.9487
##	7	2460242.0493	nan	0.1000	31030.9685
##	8	2368628.4918	nan	0.1000	8507.0996
##	9	2351964.5286	nan	0.1000	18979.3957
##	10	2322589.9653	nan	0.1000	9181.4928
##	20	2142235.0199	nan	0.1000	-3214.6183
##	40	2008366.4119	nan	0.1000	-4422.9189
##	60	1940503.8753	nan	0.1000	-28925.6844
##	80	1896766.9215	nan	0.1000	-20853.1474
##	100	1876360.2327	nan	0.1000	-8413.1316

Warning: model fit failed for Fold09.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2900740.7424	nan	0.1000	79011.8283
##	2	2796243.3140	nan	0.1000	37776.4192
##	3	2609874.8032	nan	0.1000	156555.1225
##	4	2390799.6901	nan	0.1000	23423.6469
##	5	2293236.0963	nan	0.1000	103018.7111
##	6	2218921.6903	nan	0.1000	91572.4600
##	7	2146176.5222	nan	0.1000	49658.8354
##	8	2037259.8014	nan	0.1000	-18070.0497
##	9	1977432.5837	nan	0.1000	18866.4664
##	10	1993861.3852	nan	0.1000	-71249.3186
##	20	1761175.2242	nan	0.1000	-4599.2925
##	40	1571484.0935	nan	0.1000	10832.5620
##	60	1437511.2852	nan	0.1000	-36650.2968
##	80	1273202.9429	nan	0.1000	-11614.1238
##	100	1188387.2840	nan	0.1000	-73236.8968

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2883328.4721	nan	0.1000	109351.0932
##	2	2768700.2956	nan	0.1000	105870.4985
##	3	2658907.3037	nan	0.1000	13681.2765
##	4	2538509.6739	nan	0.1000	39755.8860
##	5	2502824.0091	nan	0.1000	36182.3702
##	6	2488287.0931	nan	0.1000	5515.4866
##	7	2455264.3048	nan	0.1000	36662.1924
##	8	2434589.1795	nan	0.1000	22009.9366
##	9	2362225.0132	nan	0.1000	-10870.9193
##	10	2319618.5595	nan	0.1000	39049.5107
##	20	2235408.7715	nan	0.1000	-21238.4171
##	40	2082370.7985	nan	0.1000	-35668.5664
##	60	2009618.4116	nan	0.1000	-5777.0948
##	80	1983427.8194	nan	0.1000	-52155.7770
##	100	1933469.7981	nan	0.1000	-50759.2910

Warning: model fit failed for Fold09.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2706372.4454	nan	0.1500	323501.6602

```
##      2 2464154.0398      nan    0.1500 160144.2538
##      3 2246999.3259      nan    0.1500 -90159.7442
##      4 2134325.8031      nan    0.1500 121992.8642
##      5 2035891.2854      nan    0.1500 110241.2983
##      6 1896143.2157      nan    0.1500 -88007.8353
##      7 1854878.2538      nan    0.1500 40112.6117
##      8 1830254.5336      nan    0.1500 -63862.6089
##      9 1787852.1958      nan    0.1500 -30297.1985
##     10 1743570.6321      nan    0.1500 25369.5373
##     20 1449335.9395      nan    0.1500 -33500.5460
##     40 1033012.1513      nan    0.1500 -87463.6868
##     60  839875.9124      nan    0.1500 -37741.5791
##     80  660480.9842      nan    0.1500 -35480.7533
##    100  571793.1319      nan    0.1500 -30246.5289
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2886972.9979      nan    0.1500 209092.9895
##      2 2826253.2055      nan    0.1500 59111.8423
##      3 2752238.4061      nan    0.1500 96016.9411
##      4 2676438.9935      nan    0.1500 90516.9365
##      5 2650610.4243      nan    0.1500 29753.3236
##      6 2559133.8469      nan    0.1500 73992.8518
##      7 2493049.5974      nan    0.1500 56934.6110
##      8 2376472.1767      nan    0.1500 -3588.5746
##      9 2302024.6175      nan    0.1500 -43343.3198
##     10 2290418.9848      nan    0.1500 12794.2858
##     20 2099198.0723      nan    0.1500 -20740.0811
##     40 2017881.9204      nan    0.1500 -35288.3864
##     60 1896015.2481      nan    0.1500 -9148.7347
##     80 1867088.0237      nan    0.1500 7252.9734
##    100 1845369.1518      nan    0.1500 -1858.6234
```

```
## Warning: model fit failed for Fold09.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2794077.9824      nan    0.1500 145131.4345
##      2 2421629.8091      nan    0.1500 84608.0549
##      3 2280722.8486      nan    0.1500 40279.1852
##      4 2245505.1446      nan    0.1500 35544.6468
##      5 2203044.6825      nan    0.1500 52484.8133
##      6 2117962.0781      nan    0.1500 94433.3676
##      7 2109197.4279      nan    0.1500 -3110.7932
##      8 1973794.1501      nan    0.1500 107387.2123
##      9 1962738.8432      nan    0.1500 10826.8055
##     10 1859287.7989      nan    0.1500 56966.3865
##     20 1409599.8633      nan    0.1500 -7545.6436
##     40 1179803.8966      nan    0.1500 6963.8391
##     60  870714.9452      nan    0.1500 -47840.6446
##     80  672859.1551      nan    0.1500 -13523.9224
##    100  594097.4744      nan    0.1500 1286.5153
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2988809.1800      nan    0.1500 115559.4653
```

##	2	2804012.1680	nan	0.1500	172468.5613
##	3	2740151.8727	nan	0.1500	73053.6097
##	4	2620939.2987	nan	0.1500	135454.7066
##	5	2449535.3973	nan	0.1500	35768.4418
##	6	2412900.7751	nan	0.1500	44043.3348
##	7	2299163.0854	nan	0.1500	-745.9416
##	8	2255649.4394	nan	0.1500	-65659.3344
##	9	2224458.4453	nan	0.1500	37734.1641
##	10	2233473.2369	nan	0.1500	-52289.5868
##	20	2054166.9959	nan	0.1500	29671.7920
##	40	2005384.0395	nan	0.1500	2978.0726
##	60	1973198.1093	nan	0.1500	7709.2058
##	80	1934115.6185	nan	0.1500	-6022.9436
##	100	1947743.0448	nan	0.1500	-1860.6436

Warning: model fit failed for Fold09.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2785187.9256	nan	0.2000	326002.6226
##	2	2495072.5323	nan	0.2000	325390.6828
##	3	2445214.4280	nan	0.2000	58481.4843
##	4	2123706.3996	nan	0.2000	-44374.9982
##	5	2018720.6065	nan	0.2000	124504.6008
##	6	2036281.6136	nan	0.2000	-70099.2089
##	7	1948666.8046	nan	0.2000	121111.6148
##	8	1797915.2371	nan	0.2000	-133798.3828
##	9	1737956.3725	nan	0.2000	16273.9608
##	10	1762854.6964	nan	0.2000	-103719.5974
##	20	1553502.9291	nan	0.2000	-66596.8964
##	40	1136599.2522	nan	0.2000	-87449.1472
##	60	963655.9173	nan	0.2000	-18445.2668
##	80	935445.7563	nan	0.2000	-12222.3253
##	100	667208.6390	nan	0.2000	7888.3095

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2891570.5883	nan	0.2000	195557.8352
##	2	2747047.7665	nan	0.2000	151646.3082
##	3	2476079.4219	nan	0.2000	54063.2706
##	4	2332935.1240	nan	0.2000	127189.4631
##	5	2339694.0203	nan	0.2000	-49712.0769
##	6	2274180.0873	nan	0.2000	47853.1503
##	7	2224666.7923	nan	0.2000	30039.7806
##	8	2233955.8967	nan	0.2000	-43015.1063
##	9	2245556.6293	nan	0.2000	-35531.5483
##	10	2220714.2869	nan	0.2000	29231.6598
##	20	2144939.1716	nan	0.2000	-7189.5339
##	40	2026966.7402	nan	0.2000	-57973.5857
##	60	1864034.1891	nan	0.2000	-57047.9262
##	80	1799446.6473	nan	0.2000	-3007.9841
##	100	1752165.4900	nan	0.2000	228.0331

Warning: model fit failed for Fold09.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2599822.6882	nan	0.2000	391468.5856
##	2	2400998.8216	nan	0.2000	228763.4408
##	3	2152346.7806	nan	0.2000	187599.2931
##	4	1996792.7237	nan	0.2000	86880.6644
##	5	1919376.0113	nan	0.2000	-12686.4468
##	6	1860670.6527	nan	0.2000	65655.8767
##	7	1880984.2015	nan	0.2000	-87801.7758
##	8	1896620.6928	nan	0.2000	-44205.0920
##	9	1826979.8107	nan	0.2000	68093.5273
##	10	1863268.9144	nan	0.2000	-104359.2451
##	20	1535947.7484	nan	0.2000	-33491.3393
##	40	1031544.8778	nan	0.2000	-188450.8154
##	60	812722.0175	nan	0.2000	-14441.2666
##	80	687385.6111	nan	0.2000	-4578.6646
##	100	547650.1746	nan	0.2000	2244.5995

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2824096.2621	nan	0.2000	256131.8039
##	2	2770541.6035	nan	0.2000	33423.9522
##	3	2677098.0207	nan	0.2000	117178.0765
##	4	2563288.2724	nan	0.2000	84190.2693
##	5	2427500.7268	nan	0.2000	9412.3343
##	6	2362048.4855	nan	0.2000	41981.0929
##	7	2312228.6012	nan	0.2000	58552.6748
##	8	2294817.0790	nan	0.2000	22315.0663
##	9	2254257.3006	nan	0.2000	-85326.6769
##	10	2227110.9816	nan	0.2000	35847.5782
##	20	2008403.6418	nan	0.2000	-47270.9879
##	40	1882789.8121	nan	0.2000	-40065.4324
##	60	1865487.2913	nan	0.2000	-77128.0584
##	80	1844657.8467	nan	0.2000	-8282.1619
##	100	1806762.4148	nan	0.2000	644.2175

Warning: model fit failed for Fold09.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2991520.0891	nan	0.1000	129284.7134
##	2	2818459.9122	nan	0.1000	183110.9775
##	3	2587354.7717	nan	0.1000	68265.8744
##	4	2475321.7201	nan	0.1000	130288.1286
##	5	2448916.0038	nan	0.1000	27725.6425
##	6	2422301.3746	nan	0.1000	33854.7051
##	7	2290417.2084	nan	0.1000	84032.6305
##	8	2134798.8091	nan	0.1000	-1579.3759
##	9	2014092.8559	nan	0.1000	-17630.8390
##	10	1916576.2244	nan	0.1000	-22977.5233
##	20	1683255.3672	nan	0.1000	20688.1139
##	40	1439172.4066	nan	0.1000	3392.7630
##	60	1351498.4673	nan	0.1000	-59773.4388
##	80	1206010.7997	nan	0.1000	-34522.0671
##	100	1087598.2607	nan	0.1000	-9583.8326

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3018462.7907	nan	0.1000	90759.1284
##	2	2962337.1081	nan	0.1000	59885.0738
##	3	2761902.7425	nan	0.1000	67409.0946
##	4	2656818.6729	nan	0.1000	76982.9024
##	5	2560653.1038	nan	0.1000	-15156.4612
##	6	2524277.4430	nan	0.1000	36381.0362
##	7	2493727.1579	nan	0.1000	30059.3334
##	8	2491454.1184	nan	0.1000	-16172.1003
##	9	2417470.5682	nan	0.1000	-1887.7153
##	10	2327352.2530	nan	0.1000	18189.6741
##	20	2161248.3107	nan	0.1000	-43265.8907
##	40	2028310.2402	nan	0.1000	-37009.5014
##	60	1995432.1002	nan	0.1000	-29912.6393
##	80	1959782.3631	nan	0.1000	-38545.8706
##	100	1942483.4940	nan	0.1000	-64683.8698

Warning: model fit failed for Fold10.Rep1: shrinkage=0.10, interaction.depth=2, n.minobsinnode=15, n
 ## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	2988266.3445	nan	0.1000	137850.2663
##	2	2899052.4462	nan	0.1000	113376.9748
##	3	2836054.0036	nan	0.1000	83930.0499
##	4	2704484.3059	nan	0.1000	147368.1984
##	5	2592864.7224	nan	0.1000	133525.6810
##	6	2431981.4442	nan	0.1000	132684.8135
##	7	2240982.9892	nan	0.1000	22415.8333
##	8	2087788.1819	nan	0.1000	-3184.6306
##	9	1994867.5800	nan	0.1000	66726.7648
##	10	1998256.5390	nan	0.1000	-21484.9665
##	20	1787828.5248	nan	0.1000	-30136.5539
##	40	1431671.8117	nan	0.1000	-32795.5983
##	60	1225852.7004	nan	0.1000	-35986.3817
##	80	1182563.7436	nan	0.1000	-16116.1734
##	100	1029065.1895	nan	0.1000	-6780.3487

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	3008655.4787	nan	0.1000	109419.8300
##	2	2940977.9734	nan	0.1000	83842.1921
##	3	2754822.6663	nan	0.1000	76632.4250
##	4	2724362.7500	nan	0.1000	28609.9236
##	5	2710219.4274	nan	0.1000	3428.1377
##	6	2567627.8119	nan	0.1000	28948.4302
##	7	2522876.7608	nan	0.1000	49614.1003
##	8	2473485.4735	nan	0.1000	54721.8261
##	9	2414287.7693	nan	0.1000	-8013.7348
##	10	2375047.2589	nan	0.1000	25915.7652
##	20	2232593.0465	nan	0.1000	8914.3611
##	40	2075346.9572	nan	0.1000	-31806.1275
##	60	2004844.9391	nan	0.1000	-866.6273
##	80	1945174.0206	nan	0.1000	-36568.6242
##	100	1917722.1425	nan	0.1000	5296.2623

```
## Warning: model fit failed for Fold10.Rep1: shrinkage=0.10, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	2750759.6561	nan	0.1500	252933.8439
## 2	2537913.5124	nan	0.1500	235931.8284
## 3	2457224.3399	nan	0.1500	105194.3109
## 4	2423498.9253	nan	0.1500	27356.9957
## 5	2219963.5924	nan	0.1500	126907.3667
## 6	1988796.6103	nan	0.1500	21286.4695
## 7	1997790.2658	nan	0.1500	-34031.2995
## 8	1934073.5253	nan	0.1500	67486.1201
## 9	1943507.6638	nan	0.1500	-29444.7786
## 10	1833536.2551	nan	0.1500	-133799.1965
## 20	1652352.3111	nan	0.1500	30800.2612
## 40	1202449.9445	nan	0.1500	10522.8561
## 60	1065250.8696	nan	0.1500	-45227.1075
## 80	849303.7453	nan	0.1500	-16266.2413
## 100	800463.7455	nan	0.1500	-22484.6615

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	2880809.9304	nan	0.1500	202132.2196
## 2	2786430.0412	nan	0.1500	110344.0638
## 3	2643041.7186	nan	0.1500	122222.6735
## 4	2585465.2585	nan	0.1500	65522.9956
## 5	2522783.4919	nan	0.1500	68895.5927
## 6	2466083.2958	nan	0.1500	53119.1985
## 7	2437004.3329	nan	0.1500	35178.8424
## 8	2399895.9528	nan	0.1500	36744.5765
## 9	2340664.1544	nan	0.1500	-42711.9938
## 10	2331949.9269	nan	0.1500	10404.9557
## 20	2255105.7272	nan	0.1500	-33594.1911
## 40	2119882.7394	nan	0.1500	-54399.6661
## 60	2030500.1782	nan	0.1500	-55616.0848
## 80	1997021.6890	nan	0.1500	-735.2720
## 100	1979968.7203	nan	0.1500	-3891.3130

```
## Warning: model fit failed for Fold10.Rep1: shrinkage=0.15, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

## Iter	TrainDeviance	ValidDeviance	StepSize	Improve
## 1	2853068.0102	nan	0.1500	98880.8541
## 2	2645634.4859	nan	0.1500	207280.4483
## 3	2502654.2405	nan	0.1500	16229.7445
## 4	2436486.0428	nan	0.1500	90920.6938
## 5	2218857.3332	nan	0.1500	175316.1989
## 6	2004733.2065	nan	0.1500	14766.7388
## 7	1833171.0432	nan	0.1500	-53416.0474
## 8	1769621.2870	nan	0.1500	33094.4676
## 9	1754376.4514	nan	0.1500	11827.7654
## 10	1711873.7007	nan	0.1500	-34404.8169
## 20	1411742.8907	nan	0.1500	-18451.9714
## 40	1153903.8832	nan	0.1500	-40145.3754

```
##      60      996629.8819          nan      0.1500      819.3258
##      80      841122.6362          nan      0.1500     -7759.6408
##     100      785976.7614          nan      0.1500    -42286.5246
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1    3010089.2365          nan      0.1500   83469.3963
##      2    2855804.6401          nan      0.1500  176405.9545
##      3    2740931.1174          nan      0.1500  136609.2384
##      4    2735168.1117          nan      0.1500  -27396.4251
##      5    2542121.3301          nan      0.1500 -12556.5987
##      6    2401499.6503          nan      0.1500  19754.5179
##      7    2325415.2041          nan      0.1500  1610.9077
##      8    2297764.4524          nan      0.1500  34569.7358
##      9    2259757.4217          nan      0.1500  49551.4508
##     10    2265678.3578          nan      0.1500 -21677.4110
##     20    2200180.9341          nan      0.1500   7652.6934
##     40    2042061.6377          nan      0.1500   6403.8075
##     60    1981310.2164          nan      0.1500  -82160.4333
##     80    1893480.2705          nan      0.1500  -6827.8704
##    100    1899593.0017          nan      0.1500 -111262.4088
```

```
## Warning: model fit failed for Fold10.Rep1: shrinkage=0.15, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs'
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1    2654658.0457          nan      0.2000  360782.7861
##      2    2403220.1796          nan      0.2000  276227.8407
##      3    2339126.8209          nan      0.2000  81757.3799
##      4    2287051.4184          nan      0.2000  64158.4165
##      5    2151634.9816          nan      0.2000  172748.8857
##      6    2139675.5958          nan      0.2000  12618.8904
##      7    1890832.4664          nan      0.2000 -16695.5315
##      8    1891054.6303          nan      0.2000 -5773.4634
##      9    1808860.1971          nan      0.2000  101160.2116
##     10    1711960.8907          nan      0.2000   51817.1715
##     20    1425704.2124          nan      0.2000 -49792.2107
##     40    1036302.2188          nan      0.2000 -52258.8840
##     60     807438.1261          nan      0.2000 -98737.0142
##     80     733977.4294          nan      0.2000 -48499.4078
##    100     640047.8754          nan      0.2000 -8776.2784
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1    2697471.3538          nan      0.2000  191647.8614
##      2    2522076.6517          nan      0.2000  147273.1334
##      3    2500147.6248          nan      0.2000  -3268.0459
##      4    2391999.7223          nan      0.2000  63880.6222
##      5    2287545.5433          nan      0.2000 -22933.0775
##      6    2245049.8606          nan      0.2000 -4443.9061
##      7    2233901.0532          nan      0.2000   9213.0518
##      8    2178342.2078          nan      0.2000 -59247.5503
##      9    2159565.8435          nan      0.2000  -614.2939
##     10    2157112.6603          nan      0.2000   2339.8182
##     20    2166058.6052          nan      0.2000 -53775.0313
##     40    2045945.8930          nan      0.2000 -2977.1764
```

```
##      60 1985654.5472          nan    0.2000 -52672.4154
##      80 1983709.5079          nan    0.2000 -27792.6015
##     100 1988289.7447          nan    0.2000 -92000.9135
```

```
## Warning: model fit failed for Fold10.Rep1: shrinkage=0.20, interaction.depth=2, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2783594.0971          nan    0.2000 336613.8336
##      2 2287669.2156          nan    0.2000 83943.8015
##      3 2245789.2065          nan    0.2000 44354.4856
##      4 2129589.8297          nan    0.2000 140955.6849
##      5 2107692.9345          nan    0.2000 24082.2224
##      6 2093111.2968          nan    0.2000 17380.2547
##      7 1830164.7816          nan    0.2000 -20371.0987
##      8 1747244.8656          nan    0.2000 -18263.6057
##      9 1760273.4252          nan    0.2000 -33060.5729
##     10 1691088.2560          nan    0.2000 60148.5747
##     20 1281074.3133          nan    0.2000 2057.7678
##     40 1162777.2931          nan    0.2000 -126160.6841
##     60  798724.5926          nan    0.2000 -1724.1257
##     80  748917.8317          nan    0.2000 -50125.9851
##    100  691513.2875          nan    0.2000 -25765.7222
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2852587.2400          nan    0.2000 274424.2477
##      2 2692344.0875          nan    0.2000 146481.9833
##      3 2440566.3531          nan    0.2000 37023.6027
##      4 2386201.5878          nan    0.2000 64057.8008
##      5 2380782.3920          nan    0.2000 -19169.4648
##      6 2393163.9315          nan    0.2000 -49785.2124
##      7 2324015.7966          nan    0.2000 50945.1614
##      8 2236919.7189          nan    0.2000 -10978.7208
##      9 2221964.4132          nan    0.2000 19926.3368
##     10 2207474.6638          nan    0.2000 6598.9662
##     20 2169250.1421          nan    0.2000 22484.9550
##     40 2097591.9410          nan    0.2000 -53541.2912
##     60 2026718.3551          nan    0.2000 -12174.5899
##     80 1997519.1324          nan    0.2000 -34225.5833
##    100 1975547.7637          nan    0.2000 -2133.1993
```

```
## Warning: model fit failed for Fold10.Rep1: shrinkage=0.20, interaction.depth=3, n.minobsinnode=15, n
## The data set is too small or the subsampling rate is too large: 'nTrain * bag.fraction <= n.minobs
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1 2621843.3896          nan    0.2000 251944.6702
```


##	2	2500278.6377	nan	0.2000	158136.0896
##	3	2182922.9481	nan	0.2000	256844.4882
##	4	2017849.6706	nan	0.2000	198057.6213
##	5	1831352.7782	nan	0.2000	128815.1636
##	6	1760879.4527	nan	0.2000	73623.7405
##	7	1775619.9493	nan	0.2000	-46389.5438
##	8	1701963.6903	nan	0.2000	87355.1940
##	9	1657366.6598	nan	0.2000	-51452.3036
##	10	1597727.5155	nan	0.2000	76483.0988

gbm2

Stochastic Gradient Boosting

##

61 samples

3 predictor

##

Pre-processing: centered (3), scaled (3)

Resampling: Cross-Validated (10 fold, repeated 1 times)

Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...

Resampling results across tuning parameters:

##

##	shrinkage	interaction.depth	n.minobsinnode	n.trees	RMSE	Rsquared
##	0.10	2	5	10	891.0882	0.9118653
##	0.10	2	5	50	1002.0715	0.9052148
##	0.10	2	5	100	1071.0478	0.8856237
##	0.10	2	10	10	921.8382	0.8211780
##	0.10	2	10	50	1039.5201	0.8285063
##	0.10	2	10	100	1078.3580	0.8376732
##	0.10	2	15	10	NaN	NaN
##	0.10	2	15	50	NaN	NaN
##	0.10	2	15	100	NaN	NaN
##	0.10	3	5	10	880.3443	0.8848206
##	0.10	3	5	50	1001.0834	0.8573247
##	0.10	3	5	100	1110.0836	0.7984490
##	0.10	3	10	10	942.9323	0.8202718
##	0.10	3	10	50	1021.7383	0.8416989
##	0.10	3	10	100	1054.5092	0.8446230
##	0.10	3	15	10	NaN	NaN
##	0.10	3	15	50	NaN	NaN
##	0.10	3	15	100	NaN	NaN
##	0.15	2	5	10	937.7011	0.8504806
##	0.15	2	5	50	1078.8380	0.8335356
##	0.15	2	5	100	1118.1606	0.8548391
##	0.15	2	10	10	986.9916	0.7906947
##	0.15	2	10	50	1017.9749	0.8368233
##	0.15	2	10	100	1016.2261	0.8590212
##	0.15	2	15	10	NaN	NaN
##	0.15	2	15	50	NaN	NaN
##	0.15	2	15	100	NaN	NaN
##	0.15	3	5	10	958.4450	0.8689360
##	0.15	3	5	50	1060.1234	0.8740476
##	0.15	3	5	100	1171.7182	0.7919667
##	0.15	3	10	10	930.2459	0.8157897

##	0.15	3	10	50	1027.8066	0.8168332
##	0.15	3	10	100	1042.3186	0.8578018
##	0.15	3	15	10	NaN	NaN
##	0.15	3	15	50	NaN	NaN
##	0.15	3	15	100	NaN	NaN
##	0.20	2	5	10	983.8076	0.8654607
##	0.20	2	5	50	1198.9075	0.7244216
##	0.20	2	5	100	1371.3810	0.6798423
##	0.20	2	10	10	949.9361	0.8239179
##	0.20	2	10	50	1071.8018	0.8105117
##	0.20	2	10	100	1135.2928	0.8127442
##	0.20	2	15	10	NaN	NaN
##	0.20	2	15	50	NaN	NaN
##	0.20	2	15	100	NaN	NaN
##	0.20	3	5	10	813.9248	0.9627133
##	0.20	3	5	50	1108.0154	0.8350850
##	0.20	3	5	100	1178.1583	0.7578330
##	0.20	3	10	10	932.6875	0.8342557
##	0.20	3	10	50	1003.2810	0.8494569
##	0.20	3	10	100	1066.4599	0.8404815
##	0.20	3	15	10	NaN	NaN
##	0.20	3	15	50	NaN	NaN
##	0.20	3	15	100	NaN	NaN
##	MAE					
##	517.1165					
##	528.2833					
##	562.4945					
##	530.3265					
##	574.5000					
##	595.5081					
##	NaN					
##	NaN					
##	NaN					
##	518.1077					
##	517.5187					
##	578.9786					
##	559.5578					
##	569.3625					
##	574.0151					
##	NaN					
##	NaN					
##	NaN					
##	529.6006					
##	570.3690					
##	569.1896					
##	549.8750					
##	569.7088					
##	551.9935					
##	NaN					
##	NaN					
##	NaN					
##	551.1686					
##	547.5088					
##	589.0885					

```
## 511.6806
## 551.6563
## 553.3406
##      NaN
##      NaN
##      NaN
## 531.4022
## 614.7804
## 678.3728
## 510.4978
## 580.8428
## 623.3629
##      NaN
##      NaN
##      NaN
## 430.5785
## 572.0150
## 586.1419
## 494.8249
## 543.8161
## 574.5581
##      NaN
##      NaN
##      NaN
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 10, interaction.depth =
## 3, shrinkage = 0.2 and n.minobsinnode = 5.
```

Predicting gbm on test1

```
gbm2Preds <- predict(gbm2, newdata = test1_rem_out)
summary(gbm2Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 100.7   100.7   185.2   589.6   519.0  2437.8
```

postResample to test if it will do well on new data or if overfitting

```
postResample(gbm2Preds, test1_rem_out$Volume)
```

```
##      RMSE    Rsquared    MAE
## 415.0999818 0.7057675 244.4735938
```

CV RMSE=813, R2=.962

PostResample RMSE=415, R2=.706

Bayesian Ridge Regression, L1

Set seed

```
set.seed(123)

bay1 <- train(Volume ~ x4StarReviews + PositiveServiceReview + x2StarReviews,
              data = train1,
              method = 'blassoAveraged',
              trControl = control1,
              preProc = c('center', 'scale'))
```

```
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
```

```
## t=100, m=2
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=2
## t=800, m=2
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
```

```
## t=100, m=3
## t=200, m=3
## t=300, m=3
## t=400, m=3
## t=500, m=3
## t=600, m=3
## t=700, m=3
## t=800, m=3
## t=900, m=3
```

```
bay1
```

```
## Bayesian Ridge Regression (Model Averaged)
##
## 61 samples
## 3 predictor
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results:
##
##      RMSE      Rsquared    MAE
## 742.8701 0.8689584 421.7571
```

Predicting gbm on test1

```
bay1Preds <- predict(bay1, newdata = test1_rem_out)
summary(bay1Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -211.99 -187.44   79.51  397.59  667.00 2491.79
```

postResample to test if it will do well on new data or if overfitting

```
postResample(bay1Preds, test1$Volume)
```

```
## Warning in pred - obs: longer object length is not a multiple of shorter object
## length
```

```
## Warning in pred - obs: longer object length is not a multiple of shorter object
## length
```

```
##      RMSE Rsquared    MAE
## 925.6151      NA 716.4564
```

Negative predictions regardless of feature selection, high RMSE, doesn't work with this task

CV RMSE=1148, R2=.753

After deleting problem outlier rows in test set - 17 observations

```
Actual_vs_Predicted_NoOutlier <- data.frame(test1_rem_out %>% select(ProductNum, Volume),
rf1Preds, rf2Preds, rf3Preds, rf4Preds, rbf1Preds, rbf2Preds, linearPreds, linear2Preds, polyPreds,
poly2Preds, gbmPreds, gbm2Preds)
```

exporting to excel

```
write.xlsx(Actual_vs_Predicted_NoOutlier, file = "Actual_vs_Predicted_NoOutlier.xlsx", row.names=TRUE)
```

Now modeling with our transmuted column, AvgStarReviews to see if it helps with predictions

Set seed

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

```
# CreateDataPartition() 75% and 25%
```

```
index2 <- createDataPartition(existing4$Volume, p=0.75, list = FALSE)
```

```
train2 <- existing4[ index2,]
```

```
## Warning: The 'i' argument of '['()' can't be a matrix as of tibble 3.0.0.
```

```
## Convert to a vector.
```

```
## This warning is displayed once every 8 hours.
```

```
## Call 'lifecycle::last_warnings()' to see where this warning was generated.
```

```
test2 <- existing4[-index2,]
```

```
# Checking structure of train1
```

```
str(train2)
```

```
## tibble [61 x 16] (S3: rowwise_df/tbl_df/tbl/data.frame)
```

```
## $ ProductType.Laptop : num [1:61] 0 0 0 1 0 0 0 0 0 0 ...
```

```
## $ ProductType.Netbook : num [1:61] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ ProductType.PC : num [1:61] 1 1 1 0 0 0 0 0 0 0 ...
```

```
## $ ProductType.Smartphone: num [1:61] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ ProductNum : num [1:61] 101 102 103 104 106 107 108 109 110 111 ...
```

```
## $ Price : num [1:61] 949 2250 399 410 114 ...
```

```
## $ x4StarReviews : num [1:61] 3 1 0 19 30 3 19 9 1 2 ...
```

```
## $ x3StarReviews : num [1:61] 2 0 0 8 10 0 12 2 1 2 ...
```

```
## $ x2StarReviews : num [1:61] 0 0 0 3 9 0 5 0 0 4 ...
```

```
## $ x1StarReviews : num [1:61] 0 0 0 9 40 1 9 2 0 15 ...
```

```
## $ PositiveServiceReview : num [1:61] 2 1 1 7 12 3 5 2 2 2 ...
```

```
## $ NegativeServiceReview : num [1:61] 0 0 0 8 5 0 3 1 0 1 ...
```

```
## $ Recommendproduct : num [1:61] 0.9 0.9 0.9 0.8 0.3 0.9 0.7 0.8 0.9 0.5 ...
```

```
## $ ShippingWeight : num [1:61] 25.8 50 17.4 5.7 1.6 7.3 12 1.8 0.75 1 ...
```

```
## $ Volume : num [1:61] 12 8 12 196 332 44 132 64 40 84 ...
```

```

## $ AvgStarReviews      : num [1:61] 1.25 0.25 0 9.75 22.25 ...
## - attr(*, "groups")= tibble [61 x 1] (S3: tbl_df/tbl/data.frame)
## ..$ .rows: list<int> [1:61]
## .. ..$ : int 1
## .. ..$ : int 2
## .. ..$ : int 3
## .. ..$ : int 4
## .. ..$ : int 5
## .. ..$ : int 6
## .. ..$ : int 7
## .. ..$ : int 8
## .. ..$ : int 9
## .. ..$ : int 10
## .. ..$ : int 11
## .. ..$ : int 12
## .. ..$ : int 13
## .. ..$ : int 14
## .. ..$ : int 15
## .. ..$ : int 16
## .. ..$ : int 17
## .. ..$ : int 18
## .. ..$ : int 19
## .. ..$ : int 20
## .. ..$ : int 21
## .. ..$ : int 22
## .. ..$ : int 23
## .. ..$ : int 24
## .. ..$ : int 25
## .. ..$ : int 26
## .. ..$ : int 27
## .. ..$ : int 28
## .. ..$ : int 29
## .. ..$ : int 30
## .. ..$ : int 31
## .. ..$ : int 32
## .. ..$ : int 33
## .. ..$ : int 34
## .. ..$ : int 35
## .. ..$ : int 36
## .. ..$ : int 37
## .. ..$ : int 38
## .. ..$ : int 39
## .. ..$ : int 40
## .. ..$ : int 41
## .. ..$ : int 42
## .. ..$ : int 43
## .. ..$ : int 44
## .. ..$ : int 45
## .. ..$ : int 46
## .. ..$ : int 47
## .. ..$ : int 48
## .. ..$ : int 49
## .. ..$ : int 50
## .. ..$ : int 51

```



```
## .. ..$ : int 52
## .. ..$ : int 53
## .. ..$ : int 54
## .. ..$ : int 55
## .. ..$ : int 56
## .. ..$ : int 57
## .. ..$ : int 58
## .. ..$ : int 59
## .. ..$ : int 60
## .. ..$ : int 61
## .. ..@ ptype: int(0)
```

```
# Setting cross validation
control1 <- trainControl(method = 'repeatedcv',
                          number = 10,
                          repeats = 1)
```

set seed

```
set.seed(123)

# Creating dataframe for manual tuning
rfGrid <- expand.grid(mtry = c(2,3,4,5,6,7,8))

rf_1 <- train(Volume ~ .,
              data = train2,
              method = 'rf',
              trControl = control1,
              tuneGrid = rfGrid)

rf_1
```

```
## Random Forest
##
## 61 samples
## 15 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 55, 53, 55, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##  mtry  RMSE      Rsquared  MAE
##  2      890.2517  0.8684611  434.8184
##  3      860.1573  0.8812491  403.8704
##  4      869.6858  0.8853209  404.3859
##  5      877.7167  0.8894399  405.0399
##  6      833.4313  0.9025322  382.2740
##  7      858.2201  0.8943709  391.7704
##  8      841.2111  0.9013432  384.3876
##
```

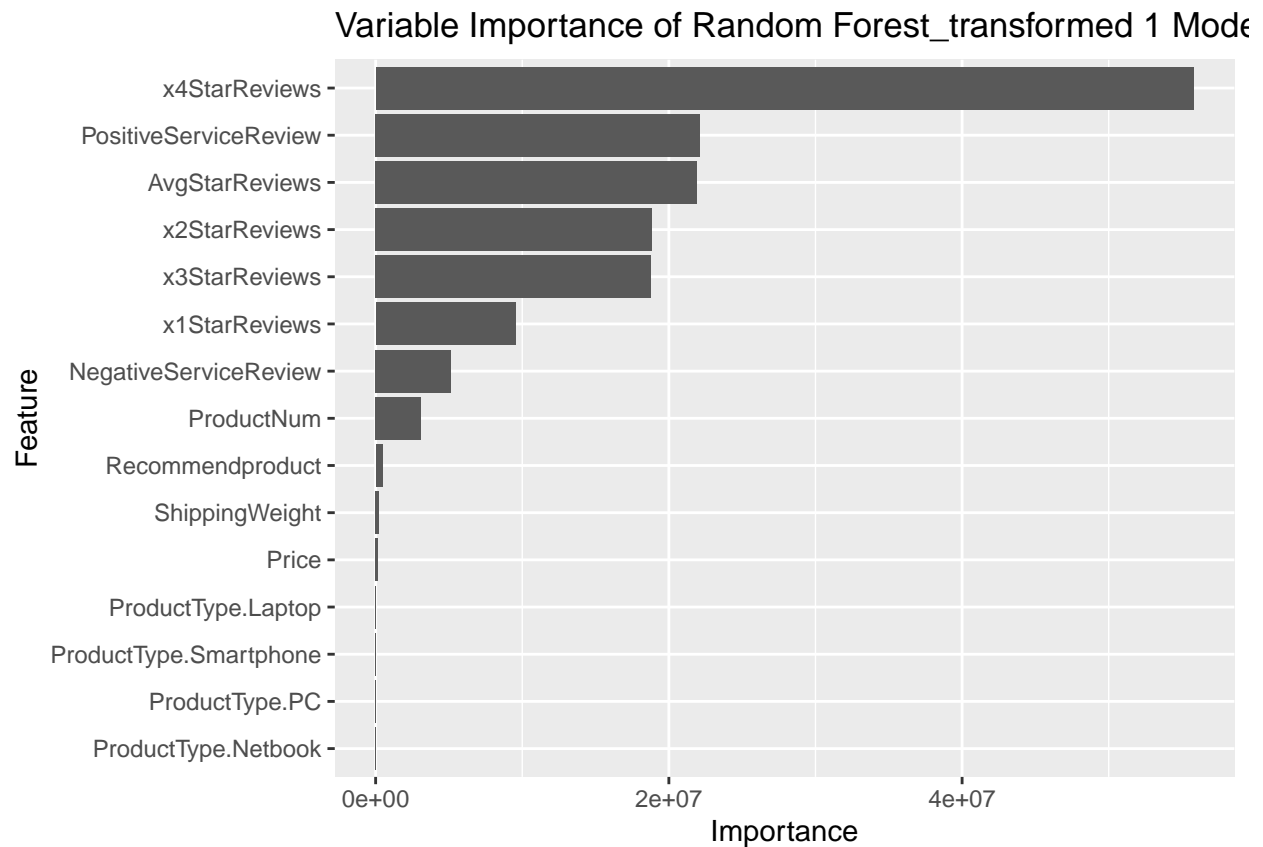
```
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was mtry = 6.
```

```
summary(rf_1)
```

##	Length	Class	Mode
## call	4	-none-	call
## type	1	-none-	character
## predicted	61	-none-	numeric
## mse	500	-none-	numeric
## rsq	500	-none-	numeric
## oob.times	61	-none-	numeric
## importance	15	-none-	numeric
## importanceSD	0	-none-	NULL
## localImportance	0	-none-	NULL
## proximity	0	-none-	NULL
## ntree	1	-none-	numeric
## mtry	1	-none-	numeric
## forest	11	-none-	list
## coefs	0	-none-	NULL
## y	61	-none-	numeric
## test	0	-none-	NULL
## inbag	0	-none-	NULL
## xNames	15	-none-	character
## problemType	1	-none-	character
## tuneValue	1	data.frame	list
## obsLevels	1	-none-	logical
## param	0	-none-	list

Variable importance using ggplot

```
ggplot(varImp(rf_1, scale=FALSE)) +  
  ggtitle('Variable Importance of Random Forest_transformed 1 Model')
```



Predicting rf on test2

```
rf_1Preds <- predict(rf_1, newdata = test2)
summary(rf_1Preds)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 12.40   28.42   258.18   758.58 1062.08 5708.03
```

postResample to test if it will do well on new data or if overfitting

```
postResample(rf_1Preds, test2$Volume)
```

```
##      RMSE    Rsquared    MAE
## 867.1132444 0.6025625 341.3013774
```

RMSE=1545 and R2=.217, poor, it does not

Using Top Model rf2 algorithm to make predictions on new product data

Target variable: 'Volume' for PC, Laptops, Netbooks, and Smartphones product types

importing data

```
new <- read.csv(file.path('C:/Users/jlbpro/OneDrive/C3T3', 'new.csv'), stringsAsFactors = TRUE)
```

checking structure

```
str(new)
```

```
## 'data.frame': 24 obs. of 18 variables:
## $ ProductType : Factor w/ 12 levels "Accessories",...: 7 7 5 5 5 6 6 6 6 12 ...
## $ ProductNum : int 171 172 173 175 176 178 180 181 183 186 ...
## $ Price : num 699 860 1199 1199 1999 ...
## $ x5StarReviews : int 96 51 74 7 1 19 312 23 3 296 ...
## $ x4StarReviews : int 26 11 10 2 1 8 112 18 4 66 ...
## $ x3StarReviews : int 14 10 3 1 1 4 28 7 0 30 ...
## $ x2StarReviews : int 14 10 3 1 3 1 31 22 1 21 ...
## $ x1StarReviews : int 25 21 11 1 0 10 47 18 0 36 ...
## $ PositiveServiceReview: int 12 7 11 2 0 2 28 5 1 28 ...
## $ NegativeServiceReview: int 3 5 5 1 1 4 16 16 0 9 ...
## $ Recommendproduct : num 0.7 0.6 0.8 0.6 0.3 0.6 0.7 0.4 0.7 0.8 ...
## $ BestSellersRank : int 2498 490 111 4446 2820 4140 2699 1704 5128 34 ...
## $ ShippingWeight : num 19.9 27 6.6 13 11.6 5.8 4.6 4.8 4.3 3 ...
## $ ProductDepth : num 20.63 21.89 8.94 16.3 16.81 ...
## $ ProductWidth : num 19.2 27 12.8 10.8 10.9 ...
## $ ProductHeight : num 8.39 9.13 0.68 1.4 0.88 1.2 0.95 1.5 0.97 0.37 ...
## $ ProfitMargin : num 0.25 0.2 0.1 0.15 0.23 0.08 0.09 0.11 0.09 0.1 ...
## $ Volume : int 0 0 0 0 0 0 0 0 0 0 ...
```

Making new dataframe same column wise as trained dataframes

```
newDummy <- dummyVars('~ .', data = new)
```

```
new2 <- data.frame(predict(newDummy, newdata = new))
```

check structure again

```
str(new2)
```

```
## 'data.frame': 24 obs. of 29 variables:
## $ ProductType.Accessories : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Display : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.ExtendedWarranty: num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.GameConsole : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Laptop : num 0 0 1 1 1 0 0 0 0 0 ...
## $ ProductType.Netbook : num 0 0 0 0 0 1 1 1 1 0 ...
## $ ProductType.PC : num 1 1 0 0 0 0 0 0 0 0 ...
## $ ProductType.Printer : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.PrinterSupplies : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Smartphone : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Software : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Tablet : num 0 0 0 0 0 0 0 0 0 1 ...
## $ ProductNum : num 171 172 173 175 176 178 180 181 183 186 ...
## $ Price : num 699 860 1199 1199 1999 ...
## $ x5StarReviews : num 96 51 74 7 1 19 312 23 3 296 ...
## $ x4StarReviews : num 26 11 10 2 1 8 112 18 4 66 ...
## $ x3StarReviews : num 14 10 3 1 1 4 28 7 0 30 ...
## $ x2StarReviews : num 14 10 3 1 3 1 31 22 1 21 ...
## $ x1StarReviews : num 25 21 11 1 0 10 47 18 0 36 ...
## $ PositiveServiceReview : num 12 7 11 2 0 2 28 5 1 28 ...
## $ NegativeServiceReview : num 3 5 5 1 1 4 16 16 0 9 ...
## $ Recommendproduct : num 0.7 0.6 0.8 0.6 0.3 0.6 0.7 0.4 0.7 0.8 ...
## $ BestSellersRank : num 2498 490 111 4446 2820 ...
## $ ShippingWeight : num 19.9 27 6.6 13 11.6 5.8 4.6 4.8 4.3 3 ...
## $ ProductDepth : num 20.63 21.89 8.94 16.3 16.81 ...
## $ ProductWidth : num 19.2 27 12.8 10.8 10.9 ...
## $ ProductHeight : num 8.39 9.13 0.68 1.4 0.88 1.2 0.95 1.5 0.97 0.37 ...
## $ ProfitMargin : num 0.25 0.2 0.1 0.15 0.23 0.08 0.09 0.11 0.09 0.1 ...
## $ Volume : num 0 0 0 0 0 0 0 0 0 0 ...
```

Removing 'BestSellersRank' since not in modeling dataset

```
new2$BestSellersRank <- NULL
str(new2)
```

```
## 'data.frame': 24 obs. of 28 variables:
## $ ProductType.Accessories : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Display : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.ExtendedWarranty: num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.GameConsole : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Laptop : num 0 0 1 1 1 0 0 0 0 0 ...
## $ ProductType.Netbook : num 0 0 0 0 0 1 1 1 1 0 ...
## $ ProductType.PC : num 1 1 0 0 0 0 0 0 0 0 ...
## $ ProductType.Printer : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.PrinterSupplies : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Smartphone : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Software : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ProductType.Tablet : num 0 0 0 0 0 0 0 0 0 1 ...
## $ ProductNum : num 171 172 173 175 176 178 180 181 183 186 ...
## $ Price : num 699 860 1199 1199 1999 ...
```

```
## $ x5StarReviews      : num  96 51 74 7 1 19 312 23 3 296 ...
## $ x4StarReviews      : num  26 11 10 2 1 8 112 18 4 66 ...
## $ x3StarReviews      : num  14 10 3 1 1 4 28 7 0 30 ...
## $ x2StarReviews      : num  14 10 3 1 3 1 31 22 1 21 ...
## $ x1StarReviews      : num  25 21 11 1 0 10 47 18 0 36 ...
## $ PositiveServiceReview : num  12 7 11 2 0 2 28 5 1 28 ...
## $ NegativeServiceReview : num  3 5 5 1 1 4 16 16 0 9 ...
## $ Recommendproduct    : num  0.7 0.6 0.8 0.6 0.3 0.6 0.7 0.4 0.7 0.8 ...
## $ ShippingWeight      : num  19.9 27 6.6 13 11.6 5.8 4.6 4.8 4.3 3 ...
## $ ProductDepth        : num  20.63 21.89 8.94 16.3 16.81 ...
## $ ProductWidth        : num  19.2 27 12.8 10.8 10.9 ...
## $ ProductHeight       : num  8.39 9.13 0.68 1.4 0.88 1.2 0.95 1.5 0.97 0.37 ...
## $ ProfitMargin        : num  0.25 0.2 0.1 0.15 0.23 0.08 0.09 0.11 0.09 0.1 ...
## $ Volume              : num  0 0 0 0 0 0 0 0 0 0 ...
```

Removing same columns as training datasets

```
new3 <- subset(new2, select = -c(1:4, 8:9, 11:12, 15, 24:27))

str(new3)
```

```
## 'data.frame': 24 obs. of 15 variables:
## $ ProductType.Laptop : num  0 0 1 1 1 0 0 0 0 0 ...
## $ ProductType.Netbook : num  0 0 0 0 0 1 1 1 1 0 ...
## $ ProductType.PC : num  1 1 0 0 0 0 0 0 0 0 ...
## $ ProductType.Smartphone: num  0 0 0 0 0 0 0 0 0 0 ...
## $ ProductNum : num  171 172 173 175 176 178 180 181 183 186 ...
## $ Price : num  699 860 1199 1199 1999 ...
## $ x4StarReviews : num  26 11 10 2 1 8 112 18 4 66 ...
## $ x3StarReviews : num  14 10 3 1 1 4 28 7 0 30 ...
## $ x2StarReviews : num  14 10 3 1 3 1 31 22 1 21 ...
## $ x1StarReviews : num  25 21 11 1 0 10 47 18 0 36 ...
## $ PositiveServiceReview : num  12 7 11 2 0 2 28 5 1 28 ...
## $ NegativeServiceReview : num  3 5 5 1 1 4 16 16 0 9 ...
## $ Recommendproduct : num  0.7 0.6 0.8 0.6 0.3 0.6 0.7 0.4 0.7 0.8 ...
## $ ShippingWeight : num  19.9 27 6.6 13 11.6 5.8 4.6 4.8 4.3 3 ...
## $ Volume : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
set.seed(123)

# Predicting rbf1 on 'new3' product data
Predicted_Volume <- predict(rf2, newdata = new3)
```

Adding our predictions to the 'new' product dataframe

```
Preds_rf2_df <- data.frame(new3 %>% select(ProductType.Laptop, ProductType.Netbook, ProductType.PC, ProductType.Smartphone, ProductNum, Price, x4StarReviews, x3StarReviews, x2StarReviews, x1StarReviews, PositiveServiceReview, NegativeServiceReview, Recommendproduct, ShippingWeight, Volume, Predicted_Volume))
```

exporting to excel

```
write.xlsx(Preds_rf2_df, "TopModel_rf2_Preds.xlsx")
```