Logistic Regression vs Gradient Boosting

Kasi Laxmanan

Case Study on Loans Application Dataset: Kaagle.com

1. Objective and Dataset info, Read dataset, variable subsets and Descriptive statistics.
2. Create Training Sample(80%), Test Sample(20%), Baseline Incidence Rate on Training Sample
3. **Logistic Regression Model**

- Run glm, Summary and Model Diagnostics
- Confusion Matrix,Prediction Rate(and Error Rate), AUC
- ROC Curves

4. **Gradient Boosting Model**

- Simulation for optimal shrinkage factor, Run gbm, Review Variable importance and partial dependency plots
- Confusion Matrix,Prediction Rate(and Error Rate), AUC
- ROC Curves

5. **Overall Commentary based on results**

1.1 Objective

Perform a comparitive analysis of Logistic Regression Model verus Gradient Boosted Trees.

1.2 Dataset for the exploration:

The source of the dataset for this exploration topic was obtained from kaggle.com for Loan application data. application_{train|test}.csv. This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET). One row represents one loan in our data sample. R was used to read this into a dataset. For our exploration the training dataset was used based on a subset of attributes.

1.3 Reading Data

```
#rows2read=10000
rows2read=-1
# Read a Comma Separated Value (CSV) file.
loans_train <- read.table("~/Documents/SASUniversityEdition/myfolders/application_train.csv"
                          header=TRUE,sep=",", na.strings=" ", nrows=rows2read)
```

1.4 Subsetting Variables

```
# This stores the data as one long text string.
mystring <-
```

```
"VARS
SK_ID_CURR
TARGET
NAME_CONTRACT_TYPE
FLAG_OWN_CAR
FLAG_OWN_REALTY
AMT_INCOME_TOTAL
AMT_CREDIT
AMT_ANNUITY
AMT_GOODS_PRICE
NAME_EDUCATION_TYPE
NAME_HOUSING_TYPE
OCCUPATION_TYPE"

# Read with more flexible read.table.
myvars <- read.table( textConnection(mystring),
   header=TRUE,
   na.strings=" ")

myvars
```

```
##                   VARS
## 1           SK_ID_CURR
## 2               TARGET
## 3   NAME_CONTRACT_TYPE
## 4         FLAG_OWN_CAR
## 5      FLAG_OWN_REALTY
## 6     AMT_INCOME_TOTAL
## 7           AMT_CREDIT
## 8          AMT_ANNUITY
## 9      AMT_GOODS_PRICE
## 10 NAME_EDUCATION_TYPE
## 11   NAME_HOUSING_TYPE
## 12     OCCUPATION_TYPE
```

```
# convert to a vector
myvars_v <- myvars[,1]
loans_tr <- loans_train[myvars_v]
```

1.5.1 Next section we look at some Descriptive Stats: Numeric Variables

```
## Select only numeric variables in full kaagle dataset
loans.numeric <- loans_train[,sapply(loans_train,is.numeric)]
ncol(loans.numeric)
```

```
## [1] 106
```

```
min <- vector("double",ncol(loans.numeric))
mean <- vector("double",ncol(loans.numeric))
max <- vector("double",ncol(loans.numeric))
nmiss <- vector("integer",ncol(loans.numeric))
count <- vector("integer",ncol(loans.numeric))
pct_miss <- vector("double",ncol(loans.numeric))

names(min) <- names(loans.numeric)
names(mean) <- names(loans.numeric)
names(max) <- names(loans.numeric)
```

```
names(nmiss) <- names(loans.numeric)
names(count) <- names(loans.numeric)
names(pct_miss) <- names(loans.numeric)

for (i in names(loans.numeric)) {

  min[i] <- min(loans.numeric[[i]],na.rm=TRUE)
  mean[i] <- mean(loans.numeric[[i]],na.rm=TRUE)
  max[i] <- max(loans.numeric[[i]],na.rm=TRUE)
  nmiss[i] <- sum(is.na(loans.numeric[[i]]))
  count[i] <- length(loans.numeric[[i]])
  pct_miss[i] <- round((nmiss[i]/count[i])*100,2)
}
summary.loans <- data.frame(
                      format(count,big.mark=",",scientific=FALSE),
                      format(round(min,2),big.mark=",",nsmall=2,scientific=FALSE),
                      format(round(mean,2),big.mark=",",nsmall=2,scientific=FALSE),
                      format(round(max,2),big.mark=",",nsmall=2,scientific=FALSE),
                      format(nmiss,big.mark=",",scientific=FALSE),
                      format(pct_miss,digit=2))
kable(summary.loans,col.names=c("Count","Minimum","Mean","Maximum","Missing-Count","Missing-
            "html",caption ="Loans Dataset- - Min Mean Max Missing(all numeric)", booktabs =
kable_styling(bootstrap_options = "striped","hover", full_width = F, position = "left")
```

Loans Dataset- - Min Mean Max Missing(all numeric)

	Count	Minimum	Mean	Maximum	Missing-Count	Missi %
SK_ID_CURR	307,511	100,002.00	278,180.52	456,255.00	0	0.00
TARGET	307,511	0.00	0.08	1.00	0	0.00
CNT_CHILDREN	307,511	0.00	0.42	19.00	0	0.00
AMT_INCOME_TOTAL	307,511	25,650.00	168,797.92	117,000,000.00	0	0.00
AMT_CREDIT	307,511	45,000.00	599,026.00	4,050,000.00	0	0.00
AMT_ANNUITY	307,511	1,615.50	27,108.57	258,025.50	12	0.00
AMT_GOODS_PRICE	307,511	40,500.00	538,396.21	4,050,000.00	278	0.09
REGION_POPULATION_RELATIVE	307,511	0.00	0.02	0.07	0	0.00
DAYS_BIRTH	307,511	-25,229.00	-16,037.00	-7,489.00	0	0.00
DAYS_EMPLOYED	307,511	-17,912.00	63,815.05	365,243.00	0	0.00
DAYS_REGISTRATION	307,511	-24,672.00	-4,986.12	0.00	0	0.00
DAYS_ID_PUBLISH	307,511	-7,197.00	-2,994.20	0.00	0	0.00
OWN_CAR_AGE	307,511	0.00	12.06	91.00	202,929	65.99
FLAG_MOBIL	307,511	0.00	1.00	1.00	0	0.00
FLAG_EMP_PHONE	307,511	0.00	0.82	1.00	0	0.00
FLAG_WORK_PHONE	307,511	0.00	0.20	1.00	0	0.00
FLAG_CONT_MOBILE	307,511	0.00	1.00	1.00	0	0.00
FLAG_PHONE	307,511	0.00	0.28	1.00	0	0.00

	Count	Minimum	Mean	Maximum	Missing-Count	Missi %
FLAG_EMAIL	307,511	0.00	0.06	1.00	0	0.00
CNT_FAM_MEMBERS	307,511	1.00	2.15	20.00	2	0.00
REGION_RATING_CLIENT	307,511	1.00	2.05	3.00	0	0.00
REGION_RATING_CLIENT_W_CITY	307,511	1.00	2.03	3.00	0	0.00
HOUR_APPR_PROCESS_START	307,511	0.00	12.06	23.00	0	0.00
REG_REGION_NOT_LIVE_REGION	307,511	0.00	0.02	1.00	0	0.00
REG_REGION_NOT_WORK_REGION	307,511	0.00	0.05	1.00	0	0.00
LIVE_REGION_NOT_WORK_REGION	307,511	0.00	0.04	1.00	0	0.00
REG_CITY_NOT_LIVE_CITY	307,511	0.00	0.08	1.00	0	0.00
REG_CITY_NOT_WORK_CITY	307,511	0.00	0.23	1.00	0	0.00
LIVE_CITY_NOT_WORK_CITY	307,511	0.00	0.18	1.00	0	0.00
EXT_SOURCE_1	307,511	0.01	0.50	0.96	173,378	56.38
EXT_SOURCE_2	307,511	0.00	0.51	0.85	660	0.21
EXT_SOURCE_3	307,511	0.00	0.51	0.90	60,965	19.83
APARTMENTS_AVG	307,511	0.00	0.12	1.00	156,061	50.75
BASEMENTAREA_AVG	307,511	0.00	0.09	1.00	179,943	58.52
YEARS_BEGINEXPLUATATION_AVG	307,511	0.00	0.98	1.00	150,007	48.78
YEARS_BUILD_AVG	307,511	0.00	0.75	1.00	204,488	66.50
COMMONAREA_AVG	307,511	0.00	0.04	1.00	214,865	69.87
ELEVATORS_AVG	307,511	0.00	0.08	1.00	163,891	53.30
ENTRANCES_AVG	307,511	0.00	0.15	1.00	154,828	50.35
FLOORSMAX_AVG	307,511	0.00	0.23	1.00	153,020	49.76
FLOORSMIN_AVG	307,511	0.00	0.23	1.00	208,642	67.85
LANDAREA_AVG	307,511	0.00	0.07	1.00	182,590	59.38
LIVINGAPARTMENTS_AVG	307,511	0.00	0.10	1.00	210,199	68.35
LIVINGAREA_AVG	307,511	0.00	0.11	1.00	154,350	50.19
NONLIVINGAPARTMENTS_AVG	307,511	0.00	0.01	1.00	213,514	69.43
NONLIVINGAREA_AVG	307,511	0.00	0.03	1.00	169,682	55.18
APARTMENTS_MODE	307,511	0.00	0.11	1.00	156,061	50.75
BASEMENTAREA_MODE	307,511	0.00	0.09	1.00	179,943	58.52
YEARS_BEGINEXPLUATATION_MODE	307,511	0.00	0.98	1.00	150,007	48.78
YEARS_BUILD_MODE	307,511	0.00	0.76	1.00	204,488	66.50

	Count	Minimum	Mean	Maximum	Missing-Count	Missi %
COMMONAREA_MODE	307,511	0.00	0.04	1.00	214,865	69.87
ELEVATORS_MODE	307,511	0.00	0.07	1.00	163,891	53.30
ENTRANCES_MODE	307,511	0.00	0.15	1.00	154,828	50.35
FLOORSMAX_MODE	307,511	0.00	0.22	1.00	153,020	49.76
FLOORSMIN_MODE	307,511	0.00	0.23	1.00	208,642	67.85
LANDAREA_MODE	307,511	0.00	0.06	1.00	182,590	59.38
LIVINGAPARTMENTS_MODE	307,511	0.00	0.11	1.00	210,199	68.35
LIVINGAREA_MODE	307,511	0.00	0.11	1.00	154,350	50.19
NONLIVINGAPARTMENTS_MODE	307,511	0.00	0.01	1.00	213,514	69.43
NONLIVINGAREA_MODE	307,511	0.00	0.03	1.00	169,682	55.18
APARTMENTS_MEDI	307,511	0.00	0.12	1.00	156,061	50.75
BASEMENTAREA_MEDI	307,511	0.00	0.09	1.00	179,943	58.52
YEARS_BEGINEXPLUATATION_MEDI	307,511	0.00	0.98	1.00	150,007	48.78
YEARS_BUILD_MEDI	307,511	0.00	0.76	1.00	204,488	66.50
COMMONAREA_MEDI	307,511	0.00	0.04	1.00	214,865	69.87
ELEVATORS_MEDI	307,511	0.00	0.08	1.00	163,891	53.30
ENTRANCES_MEDI	307,511	0.00	0.15	1.00	154,828	50.35
FLOORSMAX_MEDI	307,511	0.00	0.23	1.00	153,020	49.76
FLOORSMIN_MEDI	307,511	0.00	0.23	1.00	208,642	67.85
LANDAREA_MEDI	307,511	0.00	0.07	1.00	182,590	59.38
LIVINGAPARTMENTS_MEDI	307,511	0.00	0.10	1.00	210,199	68.35
LIVINGAREA_MEDI	307,511	0.00	0.11	1.00	154,350	50.19
NONLIVINGAPARTMENTS_MEDI	307,511	0.00	0.01	1.00	213,514	69.43
NONLIVINGAREA_MEDI	307,511	0.00	0.03	1.00	169,682	55.18
TOTALAREA_MODE	307,511	0.00	0.10	1.00	148,431	48.27
OBS_30_CNT_SOCIAL_CIRCLE	307,511	0.00	1.42	348.00	1,021	0.33
DEF_30_CNT_SOCIAL_CIRCLE	307,511	0.00	0.14	34.00	1,021	0.33
OBS_60_CNT_SOCIAL_CIRCLE	307,511	0.00	1.41	344.00	1,021	0.33
DEF_60_CNT_SOCIAL_CIRCLE	307,511	0.00	0.10	24.00	1,021	0.33
DAYS_LAST_PHONE_CHANGE	307,511	-4,292.00	-962.86	0.00	1	0.00
FLAG_DOCUMENT_2	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_3	307,511	0.00	0.71	1.00	0	0.00

	Count	Minimum	Mean	Maximum	Missing-Count	Missi%
FLAG_DOCUMENT_4	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_5	307,511	0.00	0.02	1.00	0	0.00
FLAG_DOCUMENT_6	307,511	0.00	0.09	1.00	0	0.00
FLAG_DOCUMENT_7	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_8	307,511	0.00	0.08	1.00	0	0.00
FLAG_DOCUMENT_9	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_10	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_11	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_12	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_13	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_14	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_15	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_16	307,511	0.00	0.01	1.00	0	0.00
FLAG_DOCUMENT_17	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_18	307,511	0.00	0.01	1.00	0	0.00
FLAG_DOCUMENT_19	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_20	307,511	0.00	0.00	1.00	0	0.00
FLAG_DOCUMENT_21	307,511	0.00	0.00	1.00	0	0.00
AMT_REQ_CREDIT_BUREAU_HOUR	307,511	0.00	0.01	4.00	41,519	13.50
AMT_REQ_CREDIT_BUREAU_DAY	307,511	0.00	0.01	9.00	41,519	13.50
AMT_REQ_CREDIT_BUREAU_WEEK	307,511	0.00	0.03	8.00	41,519	13.50
AMT_REQ_CREDIT_BUREAU_MON	307,511	0.00	0.27	27.00	41,519	13.50
AMT_REQ_CREDIT_BUREAU_QRT	307,511	0.00	0.27	261.00	41,519	13.50
AMT_REQ_CREDIT_BUREAU_YEAR	307,511	0.00	1.90	25.00	41,519	13.50

Loans Dataset- - Min Mean Max Missing(subset numeric)

	Count	Minimum	Mean	Maximum	Missing-Count	Missing-%
AMT_GOODS_PRICE	307,511	40,500.00	538,396.21	4,050,000.00	278	0.09
CNT_CHILDREN	307,511	0.00	0.42	19.00	0	0.00
TARGET	307,511	0.00	0.08	1.00	0	0.00
SK_ID_CURR	307,511	100,002.00	278,180.52	456,255.00	0	0.00
AMT_INCOME_TOTAL	307,511	25,650.00	168,797.92	117,000,000.00	0	0.00

	Count	Minimum	Mean	Maximum	Missing-Count	Missing-%
AMT_CREDIT	307,511	45,000.00	599,026.00	4,050,000.00	0	0.00
AMT_ANNUITY	307,511	1,615.50	27,108.57	258,025.50	12	0.00

1.5.2 Factor Variables

```
summ.factor <- data.frame(summary(select(loans_tr,c("NAME_TYPE_SUITE","FLAG_OWN_CAR",
                        "FLAG_OWN_REALTY","CODE_GENDER","NAME_CONTRACT_TYPE")))) %>%
  filter(Freq!='NA') %>%
  select(Var2,Freq)
  kable(summ.factor,col.names = c("Varible Name","Frequency"),"html",caption ="Factor Varibl
  kable_styling(bootstrap_options = "striped","hover",full_width = F, position = "left")
```

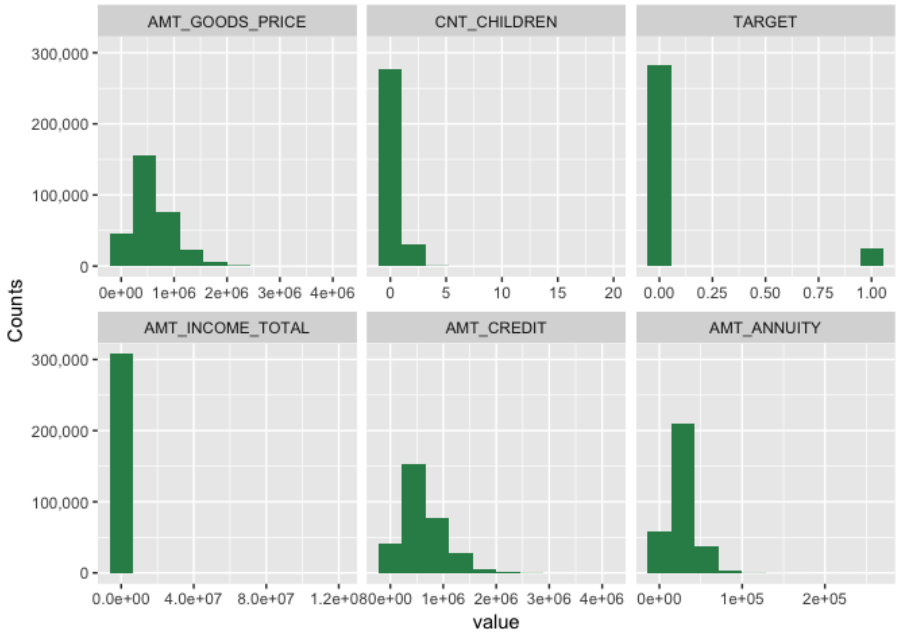
Factor Varible frequency -Loans Dataset

Varible Name	Frequency
NAME_TYPE_SUITE	Unaccompanied :248526
NAME_TYPE_SUITE	Family : 40149
NAME_TYPE_SUITE	Spouse, partner: 11370
NAME_TYPE_SUITE	Children : 3267
NAME_TYPE_SUITE	Other_B : 1770
NAME_TYPE_SUITE	: 1292
NAME_TYPE_SUITE	(Other) : 1137
FLAG_OWN_CAR	N:202924
FLAG_OWN_CAR	Y:104587
FLAG_OWN_REALTY	N: 94199
FLAG_OWN_REALTY	Y:213312
CODE_GENDER	F :202448
CODE_GENDER	M :105059
CODE_GENDER	XNA: 4
NAME_CONTRACT_TYPE	Cash loans :278232
NAME_CONTRACT_TYPE	Revolving loans: 29279

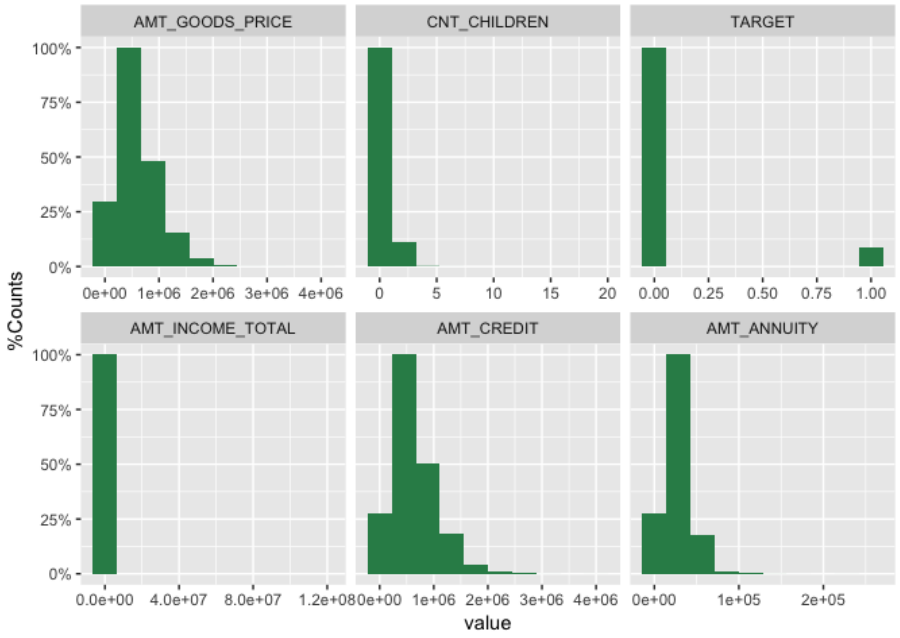
1.5.3 Here we look at some Histograms

```
# Create a Long dataset from loans_tr

loans_tr.melt <- melt(loans_tr[,sapply(loans_tr,is.numeric)],id.vars=c("SK_ID_CURR"))

## Using Melt to histogram with 10 bins
ggplot(loans_tr.melt,aes(x=value,..count..),na.rm=TRUE)+
```

```
  facet_wrap(~variable,scales='free_x') +
  geom_histogram(fill='seagreen4',bins=10) +
  scale_y_continuous(labels = comma_format(),name='Counts')
```

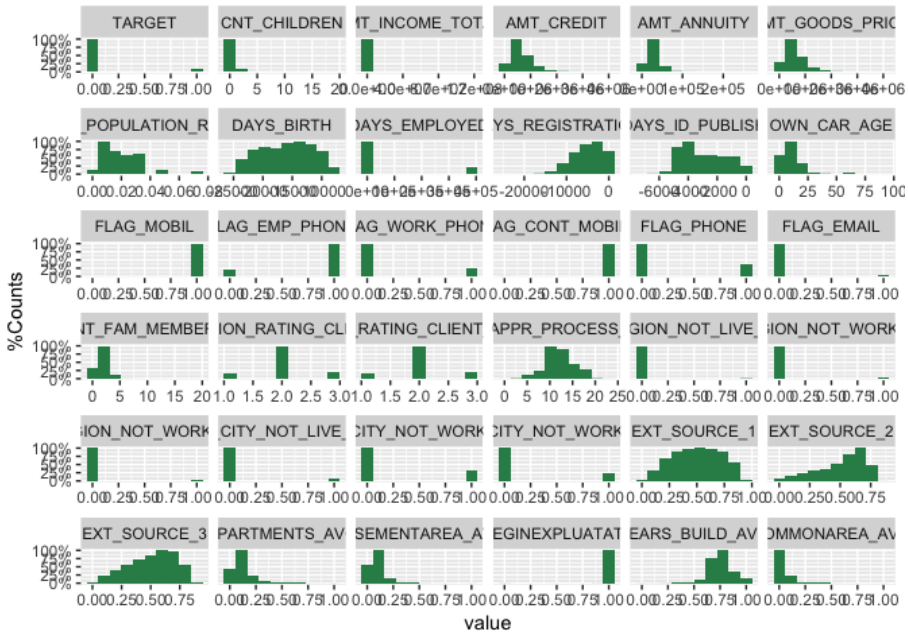



```
ggplot(loans_tr.melt,aes(x=value,..ncount..),na.rm=TRUE)+
facet_wrap(~variable,scales='free_x') +
  geom_histogram(fill='seagreen4',bins=10)   +
  scale_y_continuous(labels = percent_format(),name='%Counts')
```

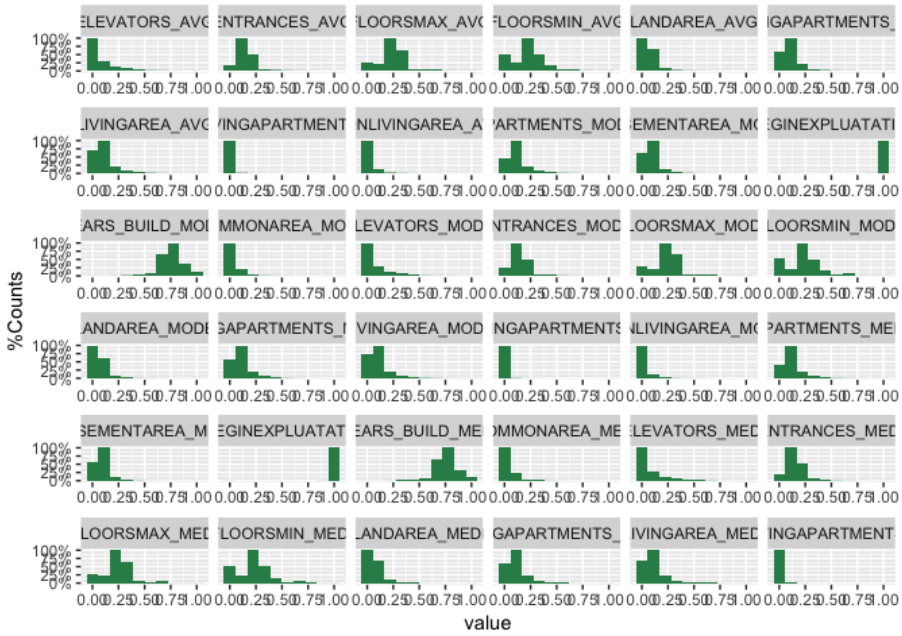



```
loans_train.melt <- melt(loans_train[,sapply(loans_train,is.numeric)],id.vars=c("SK_ID_CURR"

## Using Melt to histogram with 10 bins

ggplot(loans_train.melt,aes(x=value,..ncount..),na.rm=TRUE)+
#facet_wrap(~variable,scales='free_x') +
  facet_wrap_paginate(~variable,scales='free_x',ncol = 6, nrow = 6, page = 1) +
```

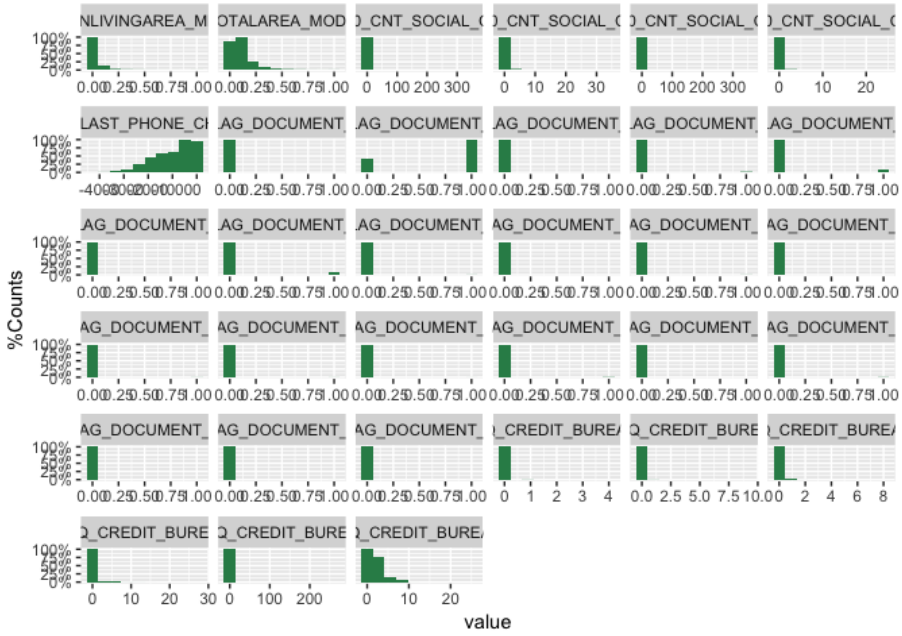
```
  geom_histogram(fill='seagreen4',bins=10)    +
  scale_y_continuous(labels = percent_format(),name='%Counts')
```




```
ggplot(loans_train.melt,aes(x=value,..ncount..),na.rm=TRUE)+
#facet_wrap(~variable,scales='free_x') +
  facet_wrap_paginate(~variable,scales='free_x',ncol = 6, nrow = 6, page = 2) +
  geom_histogram(fill='seagreen4',bins=10)    +
  scale_y_continuous(labels = percent_format(),name='%Counts')
```




```
ggplot(loans_train.melt,aes(x=value,..ncount..),na.rm=TRUE)+
#facet_wrap(~variable,scales='free_x') +
  facet_wrap_paginate(~variable,scales='free_x',ncol = 6, nrow = 6, page = 3) +
  geom_histogram(fill='seagreen4',bins=10)    +
  scale_y_continuous(labels = percent_format(),name='%Counts')
```



2.1 Create a Training and Test Samples

```
set.seed(1)
train <- sample(1:nrow(loans_tr), nrow(loans_tr)*0.80)
test <- sample(1:nrow(loans_tr), nrow(loans_tr)*0.20)
```

2.2 Baselines Incidence Rate(BIR)

```
bir.test.count <- data.frame(addmargins(xtabs(~ loans_tr[test,]$TARGET, data=loans_tr[test,]
# add row/col summary (default is sum)
kable(bir.test.count, "html",caption ="Baseline Incidence Rate - Test Sample -Counts", bookt
      col.names = c("TARGET","Count"),
      format.args = list(big.mark=",")
      ) %>%
kable_styling(bootstrap_options = "striped","hover" ,full_width = F, position = "left")
```

Baseline Incidence Rate - Test Sample - Counts

TARGET	Count
0	56,587
1	4,915
Sum	61,502

```
bir.test.prop <- data.frame(prop.table(xtabs(~ loans_tr[test,]$TARGET, data=loans_tr[test,])
# show counts as proportions of total

kable(bir.test.prop, "html",caption ="Baseline Incidence Rate - Test Sample -Prop",booktabs
      col.names = c("TARGET","Proportion")
      ) %>%
kable_styling(bootstrap_options = "striped","hover" ,full_width = F, position = "left")
```

Baseline Incidence Rate - Test Sample -Prop

TARGET	Proportion
0	0.9200839
1	0.0799161

```
bir.train.count <- data.frame(addmargins(xtabs(~ loans_tr[train,]$TARGET, data=loans_tr[trai
# add row/col summary (default is sum)
kable(bir.train.count, "html",caption ="Baseline Incidence Rate - Train Sample -Counts", boo
      col.names = c("TARGET","Count"),
       format.args = list(big.mark=",")) %>%
kable_styling(bootstrap_options = "striped","hover",full_width = F, position = "left")
```

Baseline Incidence Rate - Train Sample -Counts

TARGET	Count
0	226,156
1	19,852
Sum	246,008

```
bir.train.prop <- data.frame(prop.table(xtabs(~ loans_tr[train,]$TARGET, data=loans_tr[train
# show counts as proportions of total

kable(bir.train.prop, "html",caption ="Baseline Incidence Rate - Training Sample -Prop",book
      col.names = c("TARGET","Proportion")) %>%
kable_styling(bootstrap_options = "striped","hover",full_width = F, position = "left")
```

Baseline Incidence Rate - Training Sample -Prop

TARGET	Proportion
0	0.9193034
1	0.0806966

3.1 Train a logistic regression model:

```
logit_model <- glm( TARGET~NAME_CONTRACT_TYPE+FLAG_OWN_CAR+FLAG_OWN_REALTY+AMT_CREDIT+
                    AMT_ANNUITY+AMT_GOODS_PRICE,
                    data = loans_tr[train,],
                    family = "binomial")

summary(logit_model)
```

```
##
## Call:
## glm(formula = TARGET ~ NAME_CONTRACT_TYPE + FLAG_OWN_CAR + FLAG_OWN_REALTY +
##     AMT_CREDIT + AMT_ANNUITY + AMT_GOODS_PRICE, family = "binomial",
##     data = loans_tr[train, ])
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8968  -0.4447  -0.4103  -0.3479   3.0178
##
## Coefficients:
##                                    Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                       -2.232e+00  2.029e-02 -110.012  < 2e-16
## NAME_CONTRACT_TYPERevolving loans -3.216e-01  3.126e-02  -10.288  < 2e-16
## FLAG_OWN_CARY                     -1.608e-01  1.628e-02   -9.881  < 2e-16
## FLAG_OWN_REALTYY                  -7.444e-02  1.606e-02   -4.634 3.58e-06
## AMT_CREDIT                         3.016e-06  1.203e-07   25.071  < 2e-16
## AMT_ANNUITY                        1.093e-05  8.331e-07   13.115  < 2e-16
## AMT_GOODS_PRICE                   -4.106e-06  1.388e-07  -29.584  < 2e-16
##
## (Intercept)                       ***
## NAME_CONTRACT_TYPERevolving loans ***
## FLAG_OWN_CARY                     ***
## FLAG_OWN_REALTYY                  ***
## AMT_CREDIT                        ***
## AMT_ANNUITY                       ***
## AMT_GOODS_PRICE                   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137864  on 245779  degrees of freedom
## Residual deviance: 136221  on 245773  degrees of freedom
##   (228 observations deleted due to missingness)
## AIC: 136235
##
## Number of Fisher Scoring iterations: 5
```

3.2 Score the logistic regression model on a Test dataset

```
y_hat <- predict(logit_model, newdata = loans_tr[test,])
p_hat <- exp(y_hat)/(1 + exp(y_hat))  # convert the logodd output into probabilities
glm.pred <- rep(0, length(test))
glm.pred[p_hat > 0.0806] <- 1
```

3.3 Confusion Matrix - Logistic Regression

```
glm.logit.cm <- caret::confusionMatrix(factor(glm.pred), factor(loans_tr[test,]$TARGET))
#names(glm.logit.cm) # [1] "positive" "table"    "overall"  "byClass"  "mode"     "dots"
glm.logit.cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 25615  1734
##          1 30972  3181
##
##                Accuracy : 0.4682
##                  95% CI : (0.4643, 0.4722)
##     No Information Rate : 0.9201
##     P-Value [Acc > NIR] : 1
```

```
##
##                  Kappa : 0.0269
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.45267
##             Specificity : 0.64720
##          Pos Pred Value : 0.93660
##          Neg Pred Value : 0.09314
##              Prevalence : 0.92008
##          Detection Rate : 0.41649
##    Detection Prevalence : 0.44468
##       Balanced Accuracy : 0.54993
##
##        'Positive' Class : 0
##
```

```
data.frame(glm.logit.cm$table) %>%
kable("html",caption ="Confusion Matrix - Logit",align='c',booktabs =T) %>%
kable_styling(bootstrap_options = "striped","hover",full_width = F, position = "left")
```

Confusion Matrix - Logit

Prediction	Reference	Freq
0	0	25615
1	0	30972
0	1	1734
1	1	3181

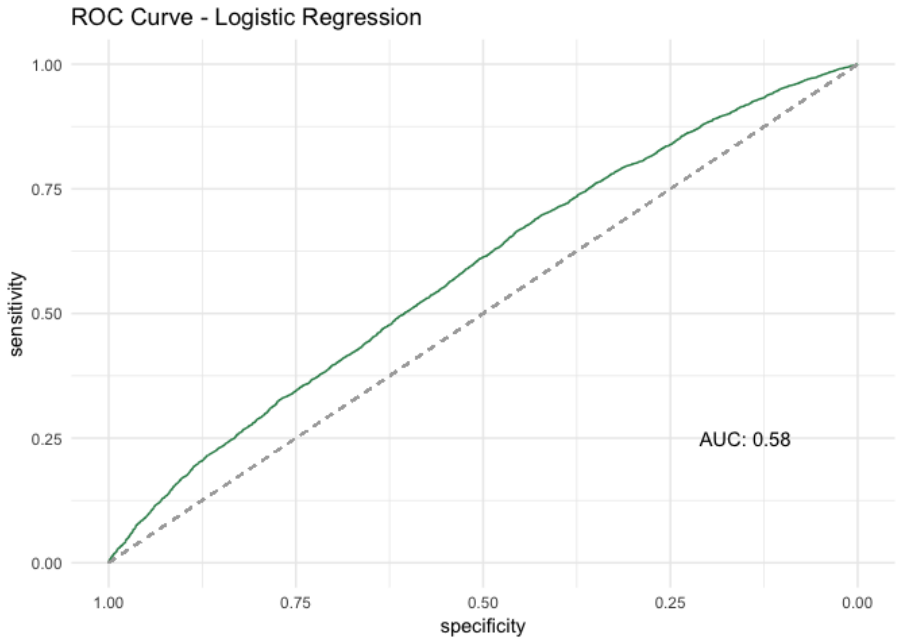
```
data.frame(glm.logit.cm$byClass) %>%
kable("html",caption ="Confusion Matrix - Sensitivity/Specificity",align='c',booktabs =T) %>
kable_styling(bootstrap_options = "striped","hover",full_width = F, position = "left")
```

Confusion Matrix - Sensitivity/Specificity

	glm.logit.cm.byClass
Sensitivity	0.4526658
Specificity	0.6472024
Pos Pred Value	0.9365973
Neg Pred Value	0.0931397
Precision	0.9365973
Recall	0.4526658
F1	0.6103460
Prevalence	0.9200839
Detection Rate	0.4164905
Detection Prevalence	0.4446847
Balanced Accuracy	0.5499341

3.4 Receiver Operating Characteristic Curve / Area under the Curve - Logistic Regression

```
logit.ROC <- roc(response=loans_tr[test,]$TARGET,
                 predictor=p_hat,
                 levels=rev(levels(factor(loans_tr[test,]$TARGET))))
# names(logit.ROC)
# [1] "percent"           "sensitivities"     "specificities"     "thresholds"
#  [5] "direction"         "cases"             "controls"          "fun.sesp"
#  [9] "auc"               "call"              "original.predictor" "original.response"
# [13] "predictor"         "response"          "levels"
auc.logit <- format(logit.ROC$auc,digits=2)


g <- ggroc(logit.ROC,colour="seagreen4",linetype=1)
g + theme_minimal()+ annotate("text", x=0.15, y=0.25, label=paste('AUC:',auc.logit)) +
  ggtitle("ROC Curve - Logistic Regression")+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1),
                 color="darkgrey", linetype="dashed")
```



Gradient Boosting Model: Introduction

The family of boosting methods is based on an additive strategy of ensemble formation. The main idea of boosting is to add new models to the ensemble sequentially. At each particular iteration, a decision tree is trained with respect to the error of the whole ensemble learnt so far.

Steps:

1. Fit a decision tree to the data: F1(x) = y,
2. We then fit the next decision tree to the residuals of the previous: h1(x) = y − F1(x),
3. Add this new tree to our algorithm: F2(x) = F1(x) + h1(x),
4. Continue this process until some mechanism (i.e. cross validation) tells us to stop.

Advantages:

1. Often provides predictive accuracy that cannot be beat, Win's most kaagle.com competitions with respect to predictive accuracy.
2. Lots of flexibility - can optimize on different loss functions and provides several hyper-parameter tuning options that make the function fit very flexible.
3. No data pre-processing required - often works great with categorical and numerical values as is.
4. Handles missing data - imputation not required

Disadvantages:

1. GBMs will continue improving to minimize all errors. This can overemphasize outliers and cause over fitting. Must use cross-validation to neutralize.
2. Computationally expensive - GBMs often require many trees (>1000) which can be time and memory exhaustive.
3. The high flexibility results in many parameters that interact and influence heavily the behavior of the approach (number of iterations, tree depth, regularization parameters, etc.). This requires a large grid search during tuning.
4. Less interpretable although this is easily addressed with various tools (variable importance, partial dependence plots, LIME, etc.).

Source: http://uc-r.github.io/gbm_regression

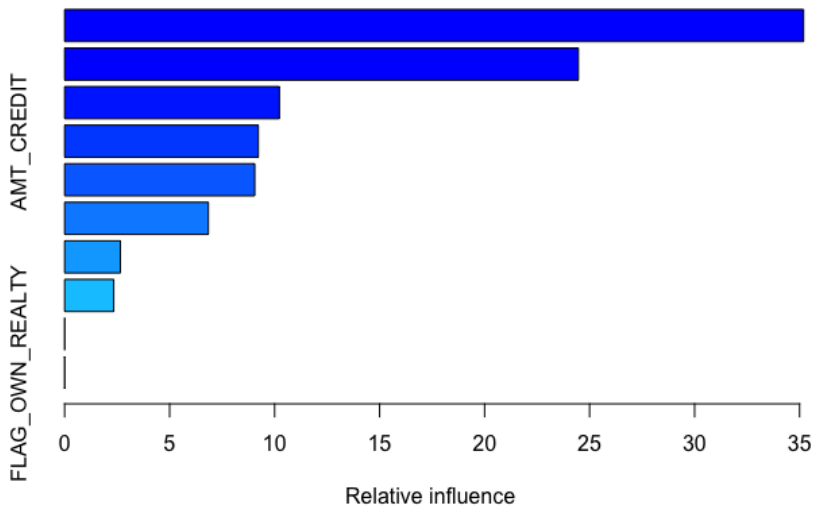
4.1 Simulation for Finding Optimal shrinkage factor/learning Rate

```
# set.seed(103)
# pows = seq(-10, -0.2, by = 0.1)
# lambdas = 10^pows
# length.lambdas = length(lambdas)
# train.errors = rep(NA, length.lambdas)
# test.errors = rep(NA, length.lambdas)
# for (i in 1:length.lambdas) {
#     boost.prelim = gbm(TARGET ~ ., data = loans_tr[train,], n.trees = 1000,  shrinkage = l
#     train.pred = predict(boost.prelim, loans_tr[train,], n.trees = 1000)
#     test.pred = predict(boost.prelim, loans_tr[test,], n.trees = 1000)
#     train.errors[i] = mean((loans_tr[train,]$TARGET - train.pred)^2)
#     test.errors[i] = mean((loans_tr[test,]$TARGET - test.pred)^2)
# }
#

# print("Shrinkage by Train MSE")
# train.errors
# plot(lambdas, train.errors, type = "b", xlab = "Shrinkage", ylab = "Train MSE",
#     col = "blue", pch = 20, xlim=c(10^-7,0.004), ylim=c(6,6.8))
# axis(side=1,at=c(seq(10^-6, 0.004, by = 10^-0.001)),
#      labels=round(c(seq(10^-6, 0.004, by = 10^-0.001)),digits=6),
#      las=2)
# title(cex.lab=0.6)
# min(test.errors)
#
#
# lambdas
# test.errors
# min(test.errors)
# plot(lambdas, test.errors, type = "b", xlab = "Shrinkage", ylab = "Test MSE",
#     col = "red", pch = 20, xlim=c(10^-7,0.004), ylim=c(6.2,6.5))
# axis(side=1,at=c(seq(10^-6, 0.004, by = 10^-0.001))
```

```
#      #labels=round(c(seq(10^-6, 0.004, by = 10^-0.001)),digits=5),
#      )
# lambdas[which.min(test.errors)]
#
```

4.2 Training a Gradient Boosting Model

```
boost.loans <-gbm(TARGET ~ . -SK_ID_CURR,
                  data = loans_tr[train,],
                  n.trees = 1000,
                  shrinkage = 0.01,
                  distribution = "bernoulli",
                  verbose=FALSE)

# names(boost.loans)
# [1] "initF"             "fit"               "train.error"       "valid.error"
#  [5] "oobag.improve"     "trees"             "c.splits"          "bag.fraction"
#  [9] "distribution"      "interaction.depth" "n.minobsinnode"    "num.classes"
# [13] "n.trees"           "nTrain"            "train.fraction"    "response.name"
# [17] "shrinkage"         "var.levels"        "var.monotone"      "var.names"
# [21] "var.type"          "verbose"           "data"              "Terms"
# [25] "cv.folds"          "call"              "m"

varimp <- data.frame(summary(boost.loans ) )
```




```
str(varimp)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ var    : Factor w/ 10 levels "AMT_ANNUITY",..: 3 6 1 2 7 9 4 5 10 8
##  $ rel.inf: num  35.19 24.47 10.23 9.22 9.05 ...
```

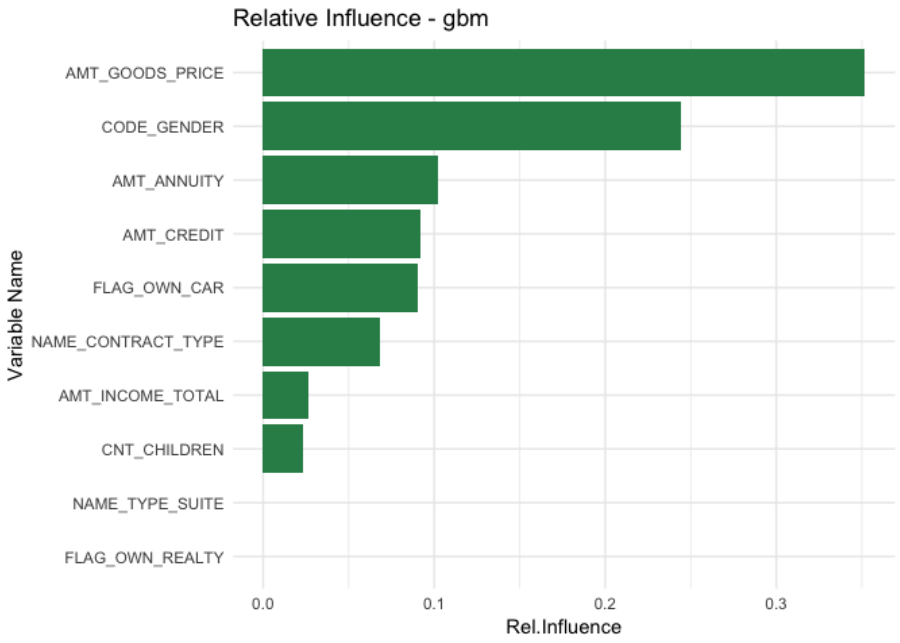
```
printvi <- varimp %>%
  arrange(desc(rel.inf)) %>%
kable("html",caption ="Variable - Relative Influence",align='c',booktabs =T) %>%
kable_styling(bootstrap_options = "striped","hover",full_width = F, position = "left")
```

```
printvi
```

Variable - Relative Influence

var	rel.inf
AMT_GOODS_PRICE	35.189680
CODE_GENDER	24.465440
AMT_ANNUITY	10.228470
AMT_CREDIT	9.223400
FLAG_OWN_CAR	9.053912
NAME_CONTRACT_TYPE	6.845110
AMT_INCOME_TOTAL	2.655213
CNT_CHILDREN	2.338774
NAME_TYPE_SUITE	0.000000
FLAG_OWN_REALTY	0.000000

```
varimp %>%
  arrange(desc(rel.inf)) %>%
 ggplot() +
  geom_bar(mapping = aes(x = reorder(var,rel.inf), y = ..prop.., weight = rel.inf, group=1),
           fill="seagreen4") +
  theme_minimal()+
  coord_flip()+
  labs(y="Rel.Influence",x="Variable Name",title="Relative Influence - gbm")
```



4.3 Score the Gradient Boosting Model on a Test dataset

```
boost.prob = predict(boost.loans, loans_tr[test,], n.trees = 1000, type = "response")
boost.pred <- rep(0, length(test))
```

```
boost.pred[boost.prob > 0.0806] <- 1
```

4.4 Confusion Matrix - Gradient Boosting

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 33100  2009
##          1 23487  2906
##
##                Accuracy : 0.5854
##                  95% CI : (0.5815, 0.5893)
##     No Information Rate : 0.9201
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0588
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5849
##             Specificity : 0.5913
##          Pos Pred Value : 0.9428
##          Neg Pred Value : 0.1101
##              Prevalence : 0.9201
##          Detection Rate : 0.5382
##    Detection Prevalence : 0.5709
##       Balanced Accuracy : 0.5881
##
##        'Positive' Class : 0
##
```

Confusion Matrix - GBM

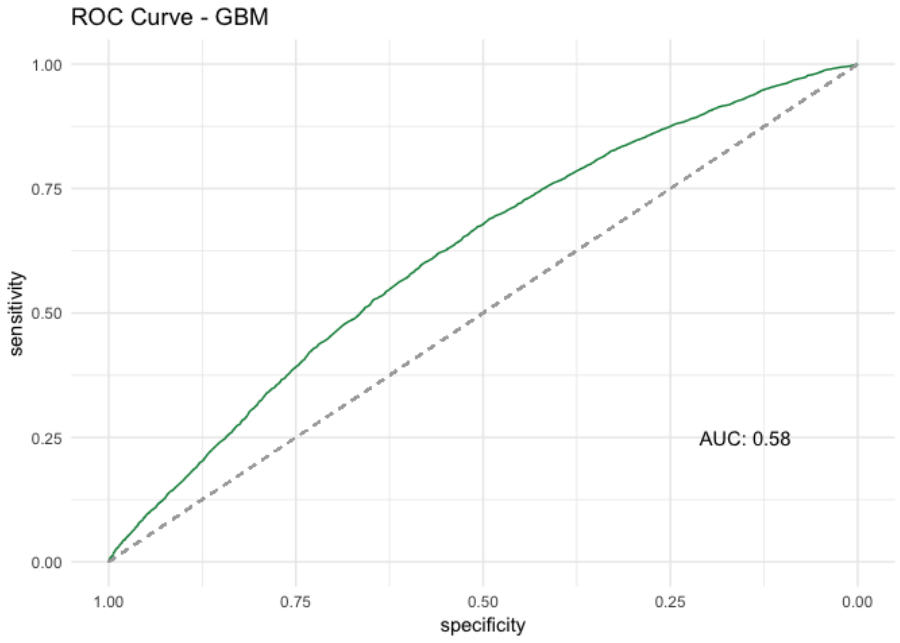
Prediction	Reference	Freq
0	0	33100
1	0	23487
0	1	2009
1	1	2906

Confusion Matrix - Sensitivity/Specificity

	gbm.cm.byClass
Sensitivity	0.5849400
Specificity	0.5912513
Pos Pred Value	0.9427782
Neg Pred Value	0.1101050
Precision	0.9427782
Recall	0.5849400
F1	0.7219508
Prevalence	0.9200839

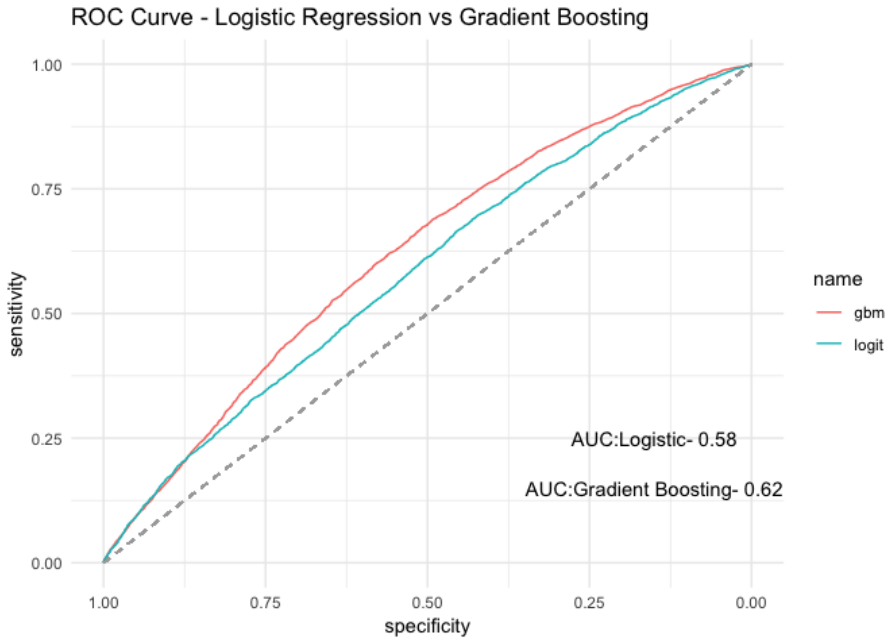
	gbm.cm.byClass
Detection Rate	0.5381939
Detection Prevalence	0.5708595
Balanced Accuracy	0.5880956

4.5 Receiver Operating Characteristic Curve / Area under the Curve - Gradient Boosting



4.6 Compartive Analysis - Logistic Regression vs Gradient Boosting -ROC Curves /AUC:

```
g3 <- ggroc(list(logit=logit.ROC, gbm=gbm.ROC))
g3 + theme_minimal()+ annotate("text", x=0.15, y=0.25, label=paste('AUC:Logistic-',auc.logit
  annotate("text", x=0.15, y=0.15, label=paste('AUC:Gradient Boosting-',auc.gbm))+
  ggtitle("ROC Curve - Logistic Regression vs Gradient Boosting")+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1),
               color="darkgrey", linetype="dashed")
```


ROC Curve - Logistic Regression vs Gradient Boosting
sensitivity
specificity
1.00
0.75
0.50
0.25
0.00
AUC:Logistic- 0.58
AUC:Gradient Boosting- 0.62
name
gbm
logit

Appendix