# GBM

```
## Loading required package: lattice
## Loading required package: ggplot2
```

## Stratified sampling

```
set.seed(123456)
inTrain = createDataPartition(y=data$target, p=0.75, list=FALSE)
training = data[inTrain,]
testing = data[-inTrain,]
dim(training);dim(testing)

## [1] 358  19

## [1] 119  19
```

## GBM-Generalized Boosted Models

```
formular1 = as.factor(target)~new_outstanding_principal_balance+initial_loan_
amount+fico+sales_channel__c+type+current_collection_method+term+average_bank
_balance__c+lender1

my_gbm<-train(formular1,method="gbm",data=training,verbose=FALSE)

## Loading required package: gbm
## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: splines
## Loading required package: parallel
## Loaded gbm 2.1.1
## Loading required package: plyr

print(my_gbm)

## Stochastic Gradient Boosting
##
## 358 samples
##  18 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
```

```
##
## Summary of sample sizes: 358, 358, 358, 358, 358, 358, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.6821306  0.2167149  0.03165156
##   1                  100      0.6736017  0.2245771  0.03208255
##   1                  150      0.6668548  0.2223738  0.03229595
##   2                   50      0.6691064  0.2220034  0.03805920
##   2                  100      0.6509272  0.1992480  0.04052494
##   2                  150      0.6386641  0.1793080  0.04599246
##   3                   50      0.6562275  0.2021453  0.03787260
##   3                  100      0.6384527  0.1840980  0.04567254
##   3                  150      0.6273741  0.1635808  0.04643658
##   Kappa SD
##   0.05781436
##   0.06091473
##   0.06576576
##   0.08485054
##   0.08887856
##   0.10508528
##   0.07018339
##   0.09662363
##   0.10012778
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 50, interaction.depth
##  = 1, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
my_gbm$finalModel
```

```
## A gradient boosted model with bernoulli loss function.
## 50 iterations were performed.
## There were 12 predictors of which 8 had non-zero influence.
```

# Apply GBM to the testing data

```
gbm_pred <- predict(my_gbm, testing)
gbm_pred_df <- data.frame(gbm_pred, target=testing$target)

confusiongbm=table(gbm_pred_df$gbm_pred, gbm_pred_df$target)
print(confusiongbm)
```

```
##
##      0  1
```

```
##   0  6  5
##   1 28 80
```

```
Observed_Accuracy = (confusiongbm[1,1]+confusiongbm[2,2])/sum(confusiongbm)
print(Observed_Accuracy)
```

```
## [1] 0.7226891
```

```
Expected_Accuracy = (sum(confusiongbm[,1])*sum(confusiongbm[1,])/sum(confusio
ngbm)+sum(confusiongbm[,2])*sum(confusiongbm[2,])/sum(confusiongbm))/sum(conf
usiongbm)

kappa = (Observed_Accuracy-Expected_Accuracy)/(1-Expected_Accuracy)
print(kappa)
```

```
## [1] 0.1476015
```

```
#prop.table(table(pred2DF$pred2, pred2DF$target),1)
```

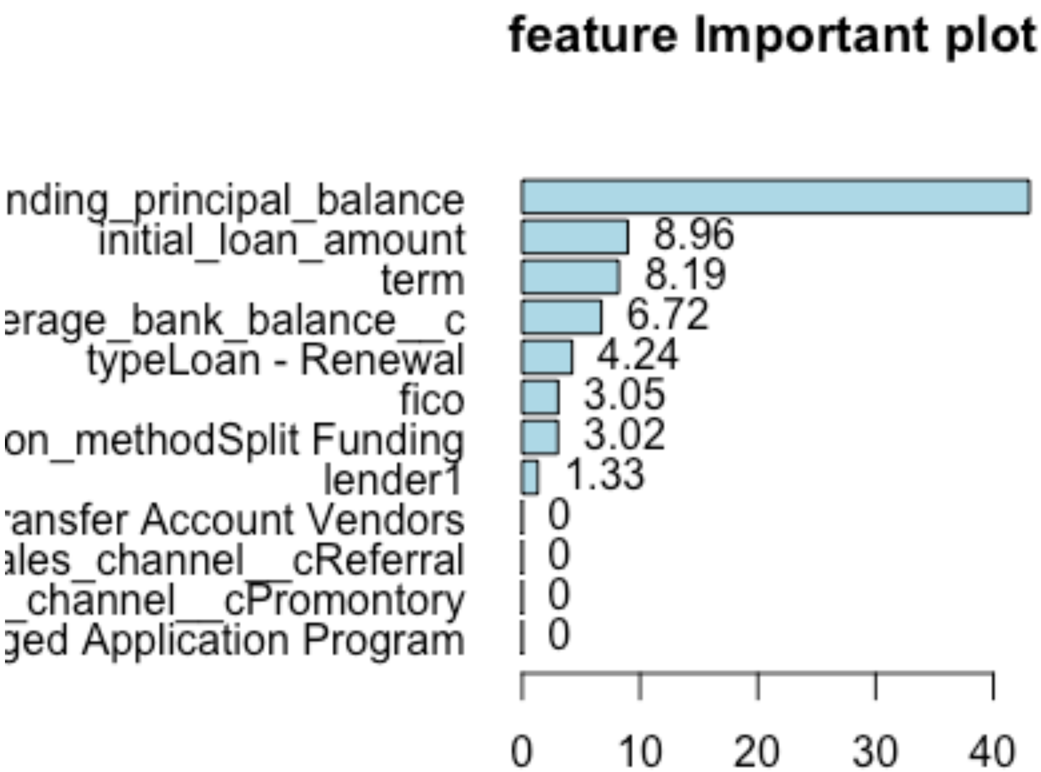
GBM variable importance measures and plots

```
gbmImp <- round(varImp(my_gbm$finalModel, scale = FALSE),2)

gbmImp1=gbmImp[order(gbmImp$Overall),,drop=FALSE]
gbmImp[order(-gbmImp$Overall),,drop=FALSE]
```

```
##                                                    Overall
## new_outstanding_principal_balance                    43.05
## initial_loan_amount                                   8.96
## term                                                  8.19
## average_bank_balance__c                               6.72
## typeLoan - Renewal                                    4.24
## fico                                                  3.05
## current_collection_methodSplit Funding                3.02
## lender1                                               1.33
## sales_channel__cFAP: Managed Application Program      0.00
## sales_channel__cPromontory                            0.00
## sales_channel__cReferral                              0.00
## current_collection_methodTransfer Account Vendors     0.00
```

```
counts <- gbmImp1[,1]
par(mai=c(1,2,1,1))
bp=barplot(counts, main="feature Important plot", horiz=TRUE, names.arg=rowna
mes(gbmImp1),col="light blue",las=1)
text(x= counts, bp, labels=as.character(counts), pos=4)
```



feature Important plot
nding_principal_balance
initial_loan_amount
term
erage_bank_balance__c
typeLoan - Renewal
fico
on_methodSplit Funding
lender1
ransfer Account Vendors
ales_channel__cReferral
_channel__cPromontory
ged Application Program
8.96
8.19
6.72
4.24
3.05
3.02
1.33
0
0
0
0
0
10
20
30
40