Lending Club: A Data Science Perspective

by Avinash Sooriyarachchi

Executive Summary

Figure 1. *Lending Club is one of the world's first and most established FinTech Companies.*

Lending Club is the world's best known and largest peer-to-peer lending company, and is based in California, USA. Founded in 2016, it is the first peer-to-peer lending company to register its offerings as securities with the Securities Exchange Commission. The fundamental mechanism of operation of the Lending Club is to effectively remove the middleman in the age old lending process. Traditionally, a bank acts as the connecting link between the borrower and the lender, typically reducing the profit earned by the investor in terms of interest collected and increasing the interest charged from the borrower.

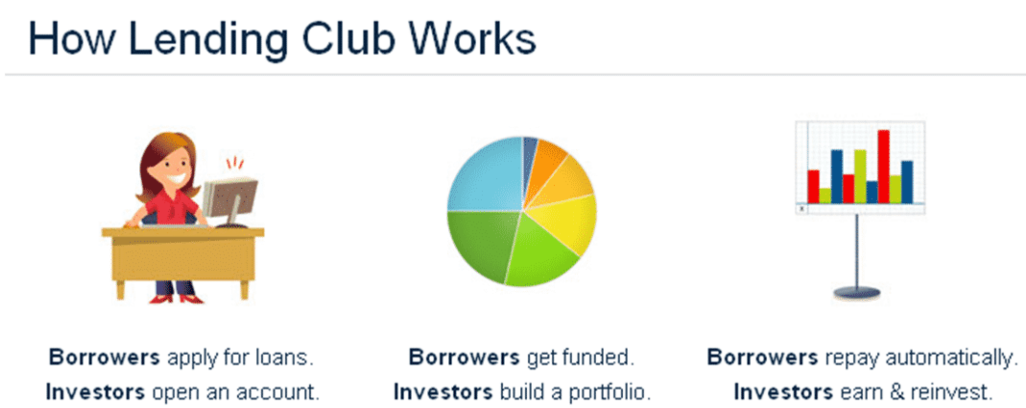


Figure 2*. A simplified graphical representation of the Lending Club's operation[2]*

Given the promising prospect of direct access to potentially millions of borrowers, lending club presents an opportunity for investors, both institutional and non insitutional, to earn profit in the form of interest. US treasury bonds, often regarded as low risk investment opportunities have been observed to offer dwindling interest rates over the past decade or so. For instance, the interest of the 10 year treasury bond has dropped from 4.68 in 2007 to 2.37 in 2017. At the same time, lending club offers interest rates anywhere from 6.03% to 26.06% depending on the credit rate corresponding to the prospective debtor.

In this particular case study, a large set of data pertaining to loans issued during the period 2007-2011 is provided. This data is freely available in the Lending Club website and is straight from their loa database. The data set that has been used for this case study has been 'cleaned', with attributes of loans that feature a large amount of missing values, redundant descriptors and descriptors that increase complexity of data without assisting in modeling or predictive prowess have been removed. The resulting data set following the aforementioned cleaning consisted of data pertaining to 38971 loans 38 attributes pertaining to each.

Here the data thus obtained was further refined through an initial exploratory data analysis process and subjected to a thorough analysis to figure out what variables or in other words, risk factors, signal that a particular loan will be defaulted. The goal is to generate an effective classifying mechanism, which could flag such loans, and the investor could build a portfolio for lending club investments such that loans predicted to have a low risk of defaulting are picked.
This would be a significant opportunity to maximize the profit from investing through lending club while mitigating risk.

For the purpose of 'mining' this data Random Forest technique was primarily resorted to following the aforementioned cleaning. Boosting was also used but the accuracy of the Random Forest technique was at such a high level, such that the former was resorted for both figuring out the main predictors of loan defaulting as well as prediction. The main variables affecting the defaulting of loans were found to be **length of employment, annual income, sub_grade, dti(ratio of the borrower's debt payments and obligations) , revolving credit utilization rate, Total credit revolving balance, installment, interest rate** and **total number of credit lines in the borrower's credit file** and a classifier was proposed using Random Forests.

There were several difficulties in implementing the data mining techniques as there was some ambiguity as to which variables should be converted to categorical variables and which should be left unchanged. Furthermore, in cleaning the data set to build the random forest, the total number of variables was reduced from 38 to a mere 30, thus the probability of losing vital information that could play a key role in modeling the the likelihood of a loan to default could have been lost.

Furthermore, there could be certain time dependant trends for loan defaulting that could possibly be studied from this dataset that have not been explored at any level in this case study.

A more robust and lengthy complete study of this data is bound to reveal key predictors of the likelihood of a loan to default, if performed.

Data Summary/ EDA

The original data set collected from 2007 to 2011 is made available through https://www.lendingclub.com/ and has been the subject of numerous data mining exercises by professional and amateur data miners alike. The particular refined and cleaned dataset was presented in its current form by a group of students for their final project for the course 'Modern Data Mining' taught by Professor Linda Zhao at the Wharton School, University of Pennsylvania.

The cleaned data set subjected to exploratory data analysis in this case study consisted of 38971 separate observations and 38 attributes. The goal of the exercise was to figure out which of these attributes are the risk factors that signal that a loan will be defaulted.

All variables pertaining to post-loan data, except for **loan_status**, were removed in order to perform the analysis, as it is nonsensical and erroneous to attempt to predict the default-proneness of a loan for investment purposes, with post-loan data.
This effectively gets rid of 12 variables.

In addition to this, several other variables that from a heuristic standpoint, could be deemed as merely contributing noise to the data were removed. Zip code of the requester for the loan is one such, as variability at the level of a single such zipcode would be non existent. Purpose was another as it seemed to be rather arbitrary and have questionable reliability. The month which the loan was funded, given by issue_d was also ignored. So was the variable pertaining to earlierst_creditline.

This effectively leaves 20 possible predictors for predicting the loan status.
These variables are as indicated in the following figure.

```
'data.frame':   38971 obs. of  21 variables:
 $ loan_amnt          : int  5000 2500 2400 10000 3000 5000 7000 3000 5600 5375 ...
 $ term               : chr  "36_months" "60_months" "36_months" "36_months" ...
 $ int_rate           : num  0.106 0.153 0.16 0.135 0.127 ...
 $ installment        : num  162.9 59.8 84.3 339.3 67.8 ...
 $ grade              : chr  "B" "C" "C" "C" ...
 $ sub_grade          : chr  "B2" "C4" "C5" "C1" ...
 $ emp_length         : chr  "10+ years" "< 1 year" "10+ years" "10+ years" ...
 $ home_ownership     : chr  "RENT" "RENT" "RENT" "RENT" ...
 $ annual_inc         : num  24000 30000 12252 49200 80000 ...
 $ verification_status : chr  "Verified" "Source Verified" "Not Verified" "Source Verified" ...
 $ loan_status        : chr  "Fully Paid" "Charged Off" "Fully Paid" "Fully Paid" ...
 $ addr_state         : chr  "AZ" "GA" "IL" "CA" ...
 $ dti                : num  27.65 1 8.72 20 17.94 ...
 $ delinq_2yrs        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ inq_last_6mths     : int  1 5 2 1 0 3 1 2 2 0 ...
 $ open_acc           : int  3 3 2 10 15 9 7 4 11 2 ...
 $ pub_rec            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ revol_bal          : int  13648 1687 2956 5598 27783 7963 17726 8221 5210 9279 ...
 $ revol_util         : num  0.837 0.094 0.985 0.21 0.539 0.283 0.856 0.875 0.326 0.365 ...
 $ total_acc          : int  9 4 10 37 38 12 11 4 13 3 ...
 $ pub_rec_bankruptcies: int  0 0 0 0 0 0 0 0 0 0 ...
```

Figure 3. *The final list of potential predictors and the response variable* ***loan_status***

A very superficial perusal of this dataset reveals that, defaulting too common in the world of Lending Club loans, a positive sign for the cautious investor. This can be seen in the following histogram. However, the ability to predict will give the investor an edge because the default rate is in the 15% to 20% range.

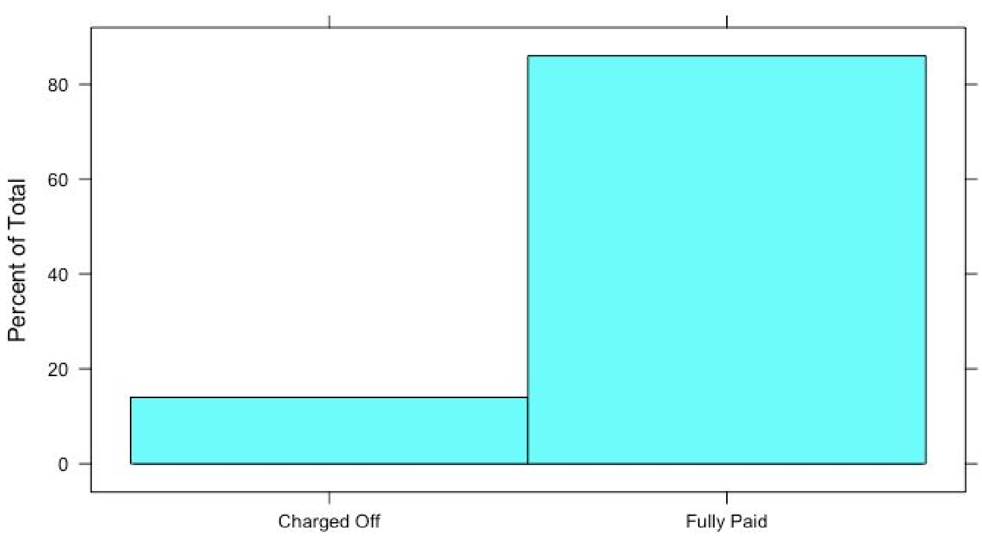


Figure 5. *Relative portions of defaulted and paid off loans in the dataset from the lending club*

Several variables, that have character and integer data types , have been converted to factors, such that it’s conducive to treat them as categorical variables for the purpose of prediction. These variables are loan_status (which is the response variable) and **term**, **grade**, **sub_grade**, **emp_length**, **home_ownership**, **verification_status**, **loan_status** and **addr_state**, which are among independent variables.

The main problem that could be observed in this data set for predicting the default-proneness of a loan is that, we have effectively almost halved the attributes collected for each loan from 37 to 20. This is a big loss of data and perhaps there are better ways to rearrange the data and maximize the input ariables.

To reemphasize, As far as input variables are concerned, **loan_status** is the dependant variable whereas **term**, **int_rate**, **installment**, **grade**, **sub_grade**, **emp_length**, **home_ownership**, **annual_inc**, **verification_status**, **loan_status**, **addr_state**, **dti**, **delinq_2yrs**, **inq_last_6mths**, **open_acc**, **pub_rec**, **revol_bal**, **revol_util**, **total_acc** and **pub_rec_bankruptcies**.

Analyses

The main machine learning tool used for analysis and classification was that of Random Forests. This choice was made because this data set possesses a large number of predictors for the independent variable, loan_status, and random forests are adept at dealing with such data types with such a large number of features and feature selection. Furthermore, overfitting is a non issue with random forests.

First the data was split to training and testing data, 75% and 25% respectively. For the first run of random forests, the default number of trees was kept i.e. 500 and mtry was 4, which is approximately equal to the square root of the number of predictor variables. For these parameters, the out of bag estimate of error was rather high, at 42.21%. The confusion matrix was as given below. As can be seen, the classification errors were high as well.

Confusion matrix:

	Charged Off	Fully Paid	class.error
Charged Off	2730	1402	0.3393030
Fully Paid	10936	14160	0.4357667

When predicting using the above generated random forest, an accuracy of 65% was obtained, which is rather low. The confusion matrix for predicting using the training data was as follows.

reference

Prediction	Charged Off	Fully Paid
Charged Off	3703	9664
Fully Paid	429	15432

As can be seen above, the lack of accuracy was quite concerning. The number of misclassifications, i.e. actual charged off loans predicted to be fully paid and vice versa are too high.

E.g. in the above confusion matrix, the number of Fully Paid loans correctly predicted as fully paid was 15,432 whereas 9664 loans that were actually fully paid were predicted to default, which certainly raises a keen investors brow, if this indeed is meant to be used as a means to mitigate risk.

When the prediction task was performed with the testing data the accuracy observed was even lowered, to a meagre 59.06% and an even higher misclassification error than before.

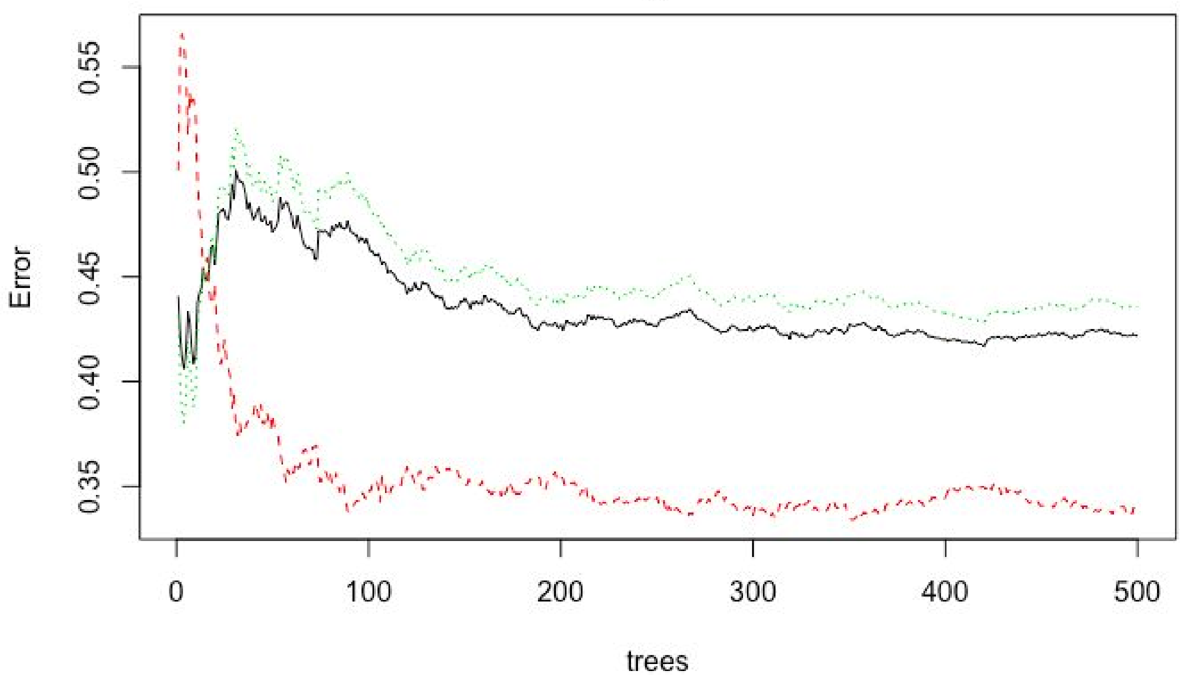


Figure 6: *The propagation of Error vs, the number of trees*

The above graph implies that increasing the number of trees would not necessarily significantly improve the accuracy of the model for prediction purposes.

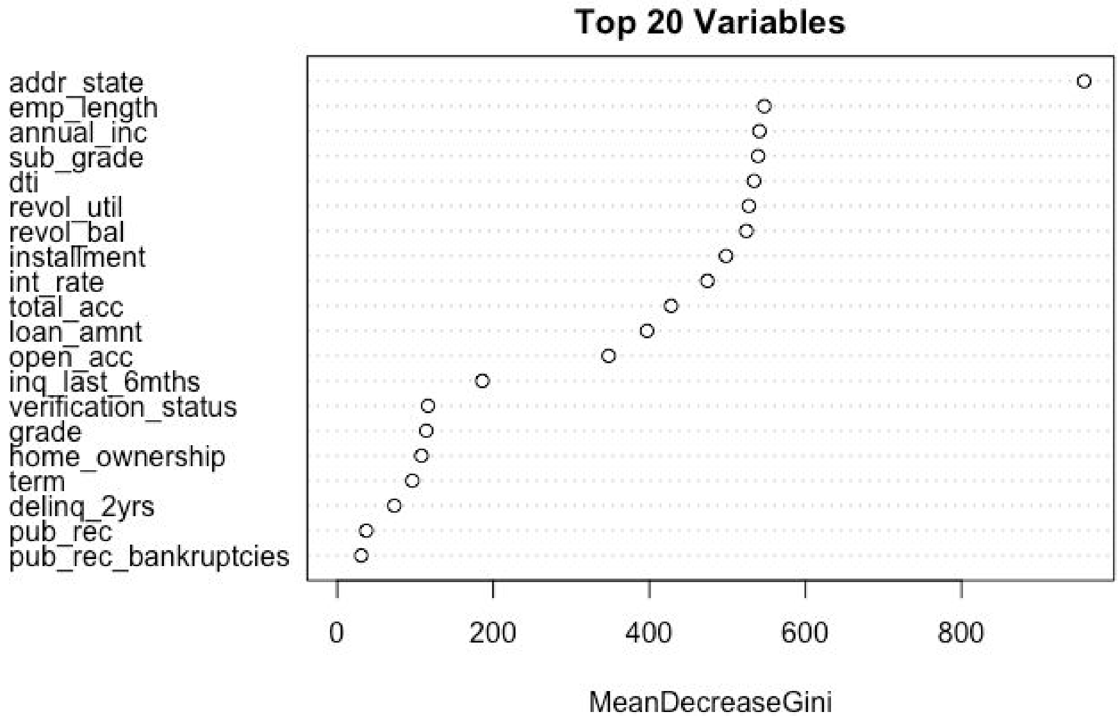


Figure 7. Mean Decrease Gini for the random forest model

Based on the above plot, the variables **addr_state, emp_length, annual_inc, sub_grade, dti , revol_util, revol_bal, installment, int_rate total_amnt** and **total_acc** variables seem to have the biggest impact on the purity of the nodes at the end of the tree without each variable, which is how Mean Decrease in Gini can be interpreted.

addr_state, which is the state in which the address of the debtor belongs to sees to have an anamoously high gini value. Thus a random forest with the above variables except for addr_state was generated.

For the purpose of a classifier, the following selection of variables is to be used.

length of employment, annual income, sub_grade, dti(ratio of the borrower's debt payments and obligations) , revolving credit utilization rate, Total credit revolving balance, installment, interest rate and **total number of credit lines in the borrower's credit file.**

To reduce the probability of investing in loan that may default loans requested by longer employed borrowers, with low dti figures, small installments, small interest rates, high annual income, high number of credit lines, high total credit revolving balance and revolving credit utilization rate.

Reasons for success and proposals for continued success

Lending Club is successful because the low interest offered to borrowers and the high interest earned to investors compared to alternative investment strategies with comparable risk. Furthermore, it managed to spearhead the use of the world wide web right from it's inception in 2006 when ebanking was just gaining attention. Furthermore, they built investor credibility with the Securities Exchange Commission's Approval. Thus investments from both retail and institutional investors va Lending Club has almost doubled every year, except for a couple of minor hiccups along the way [3].

Figure 8. *Not the only player in the market anymore : Lending Club has competition in both the US and elsewhere*

However, the landscape has changed and competitors have emerged. Marcus by Goldman Sachs, Avant, Prosper and a number of others have taken market share away from the lending club.

Based on what I have seen from the data mining exercise conducted above, one way to lure institutional investors, who aim to invest substantial amounts, in one case $60 million, is to reduce the risk of defaulting. However, in an increasingly competitive landscape, it would be unwise to turn loan requestors away by having too high thresholds for income, employment duration, credit scores and other variables seen in the model above.

What I propose for lending club, supported by data science, slightly increase the required income levels, employment durations and the rest of the above variables without deterring borrowers.

At the same time, they should explore the option of creating a separate lending business targeting upper middle class borrowers who wish to borrow loan amounts exceeding the currently offered values for interest rates higher than usual but less than banks. This service could be promoted among select retail investors and high capacity institutional investors/ high networth individuals.

Conclusions

The data mining exercise carried out with the Lending Club data set from 2007 to 2011, using the random forest technique generated modest predictive ability. Better data and more robust machine learning techniques need to be employed to achieve greater accuracy to help the cautious investor spot the loan most likely to default. However, the variables **length of employment, annual income, sub_grade, dti(ratio of the borrower's debt payments and obligations) , revolving credit utilization**

rate, Total credit revolving balance, installment, interest rate and **total number of credit lines in the borrower's credit file,** seemed to have the biggest effect on the response variable **loan_status.**

Based on the above selection of variables, it is recommended that the lending club impose slightly tighter requirements on each of the above such that loss of market share to competitors is minimal and increases investor confidence. Furthermore, I highly suggest that Lending Club create a separate 'premium experience' targeting upper middle class borrowers with the possibility of higher loan amounts, higher interest rates and more visibility among investors likely to invest large amounts of capital.

References

[1]https://s.thestreet.com/files/tsc/v2008/photos/contrib/uploads/a07fb0f8-f38d-11e6-834b-f9d13741005e_600x400.jpg
[2]http://echeck.org/wp-content/uploads/2016/12/Showing-how-the-lending-club-works-and-makes-money-1.png
[3]https://www.economist.com/blogs/schumpeter/2013/01/lending-club
[4]https://letstalkpayments.com/us-peer-to-peer-p2p-lending-market-a-crisp-report/

Appendix (next page)

Untitled

Avi Sooriyarachchi

11/19/2017

```
rm(list=ls())
library(leaps)  # regsubsets() for model selection
library(car)    # Anova()
library(glmnet) # glmnet() and cv.glmnet()
```

```
## Warning: package 'glmnet' was built under R version 3.4.2
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-13
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:car':
##
##     recode
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(histogram)
```

```
data1 <- read.csv("LoanStats_07_11_Clean.csv", sep=",", header=T, as.is=T)
data<-data1[,-c(2,3,28,29,30,31,32,33,34,35,36,37,9,14,16,17,21)]
```

Convert to factors

```
data$loan_status <- as.factor(data$loan_status)
data$term <- as.factor(data$term)
data$grade <- as.factor(data$grade)
data$sub_grade <- as.factor(data$sub_grade)
data$emp_length <- as.factor(data$emp_length)
data$home_ownership <- as.factor(data$home_ownership)
data$verification_status <- as.factor(data$verification_status)
data$loan_status <- as.factor(data$loan_status)
data$addr_state <- as.factor(data$addr_state)
```

```
str(data)
```

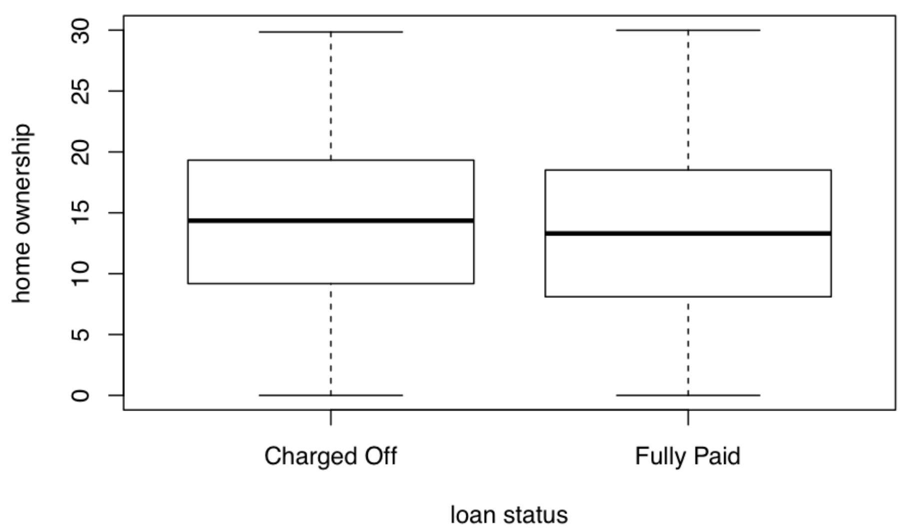
```
## 'data.frame':    38971 obs. of  21 variables:
##  $ loan_amnt          : int  5000 2500 2400 10000 3000 5000 7000 3000 5600 5375 ...
##  $ term               : Factor w/ 2 levels "36_months","60_months": 1 2 1 1 2 1 2 1 2 2 ...
```

```
## $ int_rate           : num  0.106 0.153 0.16 0.135 0.127 ...
## $ installment        : num  162.9 59.8 84.3 339.3 67.8 ...
## $ grade              : Factor w/ 7 levels "A","B","C","D",..: 2 3 3 3 2 1 3 5 6 2 ...
## $ sub_grade          : Factor w/ 35 levels "A1","A2","A3",..: 7 14 15 11 10 4 15 21 27 10 ...
## $ emp_length         : Factor w/ 12 levels "< 1 year","1 year",..: 3 1 3 3 2 5 10 11 6 1 ...
## $ home_ownership     : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 4 4 4 4 4 4 4 3 4 ...
## $ annual_inc         : num  24000 30000 12252 49200 80000 ...
## $ verification_status : Factor w/ 3 levels "Not Verified",..: 3 2 1 2 2 2 1 2 2 3 ...
## $ loan_status        : Factor w/ 2 levels "Charged Off",..: 2 1 2 2 2 2 2 2 1 1 ...
## $ addr_state         : Factor w/ 49 levels "AK","AL","AR",..: 4 11 15 5 36 4 27 5 5 42 ...
## $ dti                : num  27.65 1 8.72 20 17.94 ...
## $ delinq_2yrs        : int  0 0 0 0 0 0 0 0 0 0 ...
## $ inq_last_6mths     : int  1 5 2 1 0 3 1 2 2 0 ...
## $ open_acc           : int  3 3 2 10 15 9 7 4 11 2 ...
## $ pub_rec            : int  0 0 0 0 0 0 0 0 0 0 ...
## $ revol_bal          : int  13648 1687 2956 5598 27783 7963 17726 8221 5210 9279 ...
## $ revol_util         : num  0.837 0.094 0.985 0.21 0.539 0.283 0.856 0.875 0.326 0.365 ...
## $ total_acc          : int  9 4 10 37 38 12 11 4 13 3 ...
## $ pub_rec_bankruptcies: int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
set.seed(150) # Set Seed so that same sample can be reproduced in future also
# Now Selecting 75% of data as sample from total 'n' rows of the data
sample <- sample.int(n = nrow(data), size = floor(.75*nrow(data)), replace = F)
train <- data[sample, ]
test  <- data[-sample, ]
```

Parsimonious model

```r
boxplot(data$dti~data$loan_status, ylab ="home ownership", xlab = "loan status")
```




```r
#random forest
library(randomForest)
```

```
## randomForest 4.6-12
```

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

##
## Attaching package: 'randomForest'

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

```
set.seed(222)
rf<-randomForest(loan_status~.,data=train)
print(rf)
```

```
##
## Call:
##  randomForest(formula = loan_status ~ ., data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 42.21%
## Confusion matrix:
##             Charged Off Fully Paid class.error
## Charged Off        2730       1402   0.3393030
## Fully Paid        10936      14160   0.4357667
```

```
attributes(rf)
```

```
## $names
##  [1] "call"            "type"            "predicted"
##  [4] "err.rate"        "confusion"       "votes"
##  [7] "oob.times"       "classes"         "importance"
## [10] "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"
## [16] "y"               "test"            "inbag"
## [19] "terms"
##
## $class
## [1] "randomForest.formula" "randomForest"
```

```
#prediction and confusion matrix-train data
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:histogram':
##
##     histogram

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin
```

```r
p1<-predict(rf,train)
```

```r
confusionMatrix(p1,train$loan_status)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Charged Off Fully Paid
##   Charged Off        3703       9664
##   Fully Paid          429      15432
##
##                Accuracy : 0.6547
##                  95% CI : (0.6492, 0.6601)
##     No Information Rate : 0.8586
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2643
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8962
##             Specificity : 0.6149
##          Pos Pred Value : 0.2770
##          Neg Pred Value : 0.9730
##              Prevalence : 0.1414
##          Detection Rate : 0.1267
##    Detection Prevalence : 0.4573
##       Balanced Accuracy : 0.7555
##
##        'Positive' Class : Charged Off
##
```

prediction & confusion matrix- test data

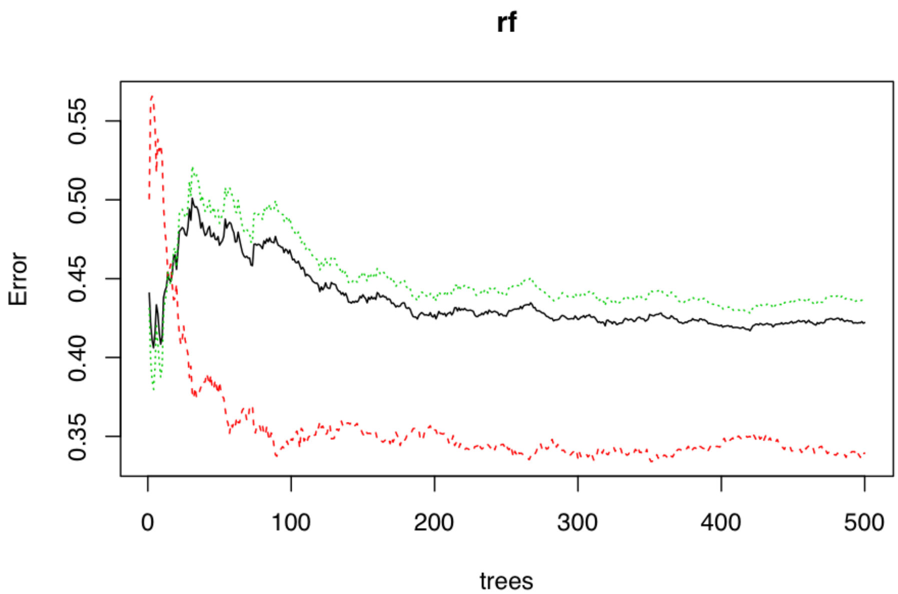
```r
p2<-predict(rf,test)
confusionMatrix(p2,test$loan_status)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Charged Off Fully Paid
##   Charged Off         923       3576
##   Fully Paid          413       4831
##
##                Accuracy : 0.5906
##                  95% CI : (0.5807, 0.6004)
##     No Information Rate : 0.8629
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.133
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.69087
##             Specificity : 0.57464
##          Pos Pred Value : 0.20516
##          Neg Pred Value : 0.92124
```

```
##               Prevalence : 0.13712
##           Detection Rate : 0.09473
##     Detection Prevalence : 0.46177
##        Balanced Accuracy : 0.63275
##
##         'Positive' Class : Charged Off
##
```

Error rate of Random Forest

```
plot(rf)
```



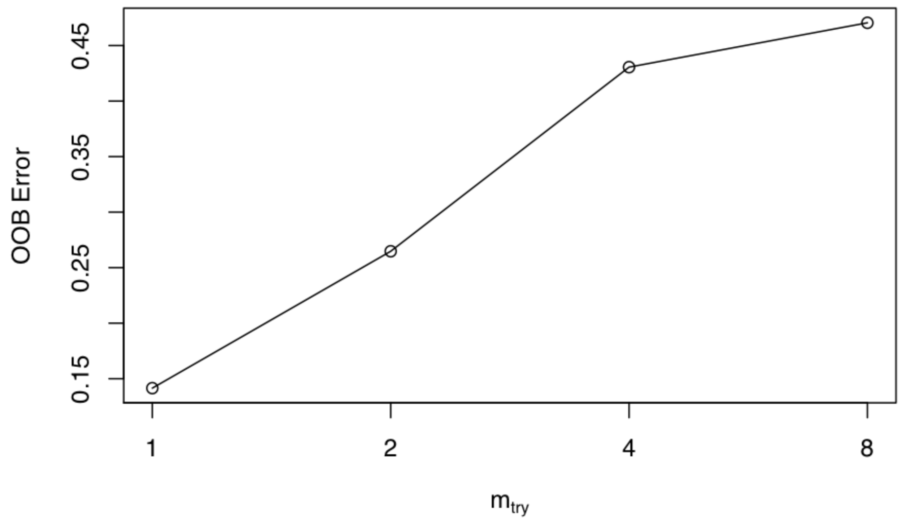

```
t<-tuneRF(train[,-11], train[,11],
          stepFactor=0.5,
          plot=TRUE,
          ntreeTry = 300,
          trace=TRUE,
          improve = 0.05)
```

```
## mtry = 4  OOB error = 43.06%
## Searching left ...
## mtry = 8     OOB error = 47.05%
## -0.09272944 0.05
## Searching right ...
## mtry = 2     OOB error = 26.47%
## 0.385141 0.05
## mtry = 1     OOB error = 14.15%
## 0.4656242 0.05
```

```
## Warning in randomForest.default(x, y, mtry = mtryCur, ntree = ntreeTry, :
## invalid mtry: reset to within valid range
```

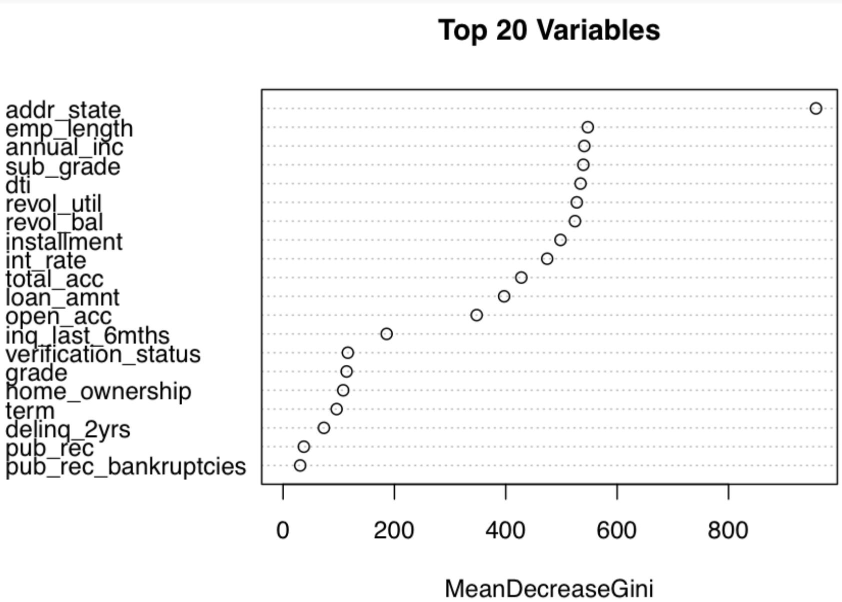
```
## mtry = 0     OOB error = 14.14%
## 0.0004836759 0.05
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```




```
#random fores
```

```
varImpPlot(rf, sort=T,n.var=20,main="Top 20 Variables")
```



```r
importance(rf)
```

```
##                      MeanDecreaseGini
## loan_amnt                   397.08760
## term                         96.20561
## int_rate                    474.40038
## installment                 498.33910
## grade                       114.11283
## sub_grade                   539.29120
## emp_length                  547.35170
## home_ownership              107.99559
## annual_inc                  541.02869
## verification_status         116.31309
## addr_state                  957.21161
## dti                         534.27192
## delinq_2yrs                  73.30211
## inq_last_6mths              186.02776
## open_acc                    347.73390
## pub_rec                      37.31641
## revol_bal                   524.31010
## revol_util                  527.54944
## total_acc                   427.93343
## pub_rec_bankruptcies         30.77227
```

```r
varUsed(rf)
```

```
##  [1] 105524   7503 105881 125808  17527  88209 103859  30621 126661  33994
## [11] 131111 130996  21082  52704  96890   9734 129223 128674 113606   7756
```

```r
library(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 2.1.3
```

Boosting

````
```{r}
#ntree <- 200
#fit.boost <- gbm(loan_status ~., data = data, distribution = "gaussian", n.trees = ntree,
interaction.depth = 2,
#               train.fraction = .7)
```

```{r}
names(fit.boost)
#fit.boost$fit # hat y
#fit.boost$train.error # training errors
#fit.boost$valid.error # testing errors if train.fraction is given
```

```{r}
yhat <- predict(fit.boost, newdata = data, n.trees = ntree)
```

```{r}
ntree <- 2000
#fit.boost <- gbm(loan_status~., data = data, distribution = "gaussian", n.trees = ntree,
interaction.depth = 2,
#               train.fraction = .8)
#gbm.perf(fit.boost, method ="test")
```



```{r}
#n.t <- floor(.8*38971)
#data.train <- data[1:n.t, ]   # n.t <- floor(.8*263)
#data.test <- data[-(1:n.t), ]
```

```{r}
#B <- gbm.perf(fit.boost, method ="test") # optimal number of trees
#yhat <- predict(fit.boost, newdata = data.test, n.trees = gbm.perf(fit.boost, method ="test") )
````
````