Kiva Final

R Gogliotti

December 5, 2018

The following libraries were used in this analysis

```
library(data.table)
library(corrplot)
library(tidyverse)
library(reshape2)
library(treemap)
library(stringr)
library(ggthemes)
library(countrycode)
library(gdata)
library(ggrepel)
library(ggwordcloud)
library(randomForest)
library(caret)
library(DMwR) #SMOTE Hybrid Model
```

Read in data

```
kiva <- fread("BDS_WK11_kiva_loans.csv", header=T)
```

General Exploratory Analysis

```
head(kiva)

##          id funded_amount loan_amount            activity         sector
## 1:  653051           300         300 Fruits & Vegetables           Food
## 2:  653053           575         575            Rickshaw Transportation
## 3:  653068           150         150      Transportation Transportation
## 4:  653063           200         200          Embroidery           Arts
## 5:  653084           400         400          Milk Sales           Food
## 6: 1080148           250         250            Services       Services
##
use
## 1:                                           To buy seasonal, fresh
fruits to sell.
## 2:                     to repair and maintain the auto rickshaw used in
their business.
## 3: To repair their old cycle-van and buy another one to rent out as a
source of income
## 4:         to purchase an embroidery machine and a variety of new
embroidery materials.
## 5:                                                          to purchase
```

```
one buffalo.
## 6:                                purchase leather for my business
using ksh 20000.
##    country_code  country       region currency partner_id
## 1:           PK Pakistan       Lahore      PKR        247
## 2:           PK Pakistan       Lahore      PKR        247
## 3:           IN    India    Maynaguri      INR        334
## 4:           PK Pakistan       Lahore      PKR        247
## 5:           PK Pakistan Abdul Hakeem      PKR        245
## 6:           KE    Kenya                   KES         NA
##                  posted_time            disbursed_time
## 1: 2014-01-01 06:12:39+00:00 2013-12-17 08:00:00+00:00
## 2: 2014-01-01 06:51:08+00:00 2013-12-17 08:00:00+00:00
## 3: 2014-01-01 09:58:07+00:00 2013-12-17 08:00:00+00:00
## 4: 2014-01-01 08:03:11+00:00 2013-12-24 08:00:00+00:00
## 5: 2014-01-01 11:53:19+00:00 2013-12-17 08:00:00+00:00
## 6: 2014-01-01 10:06:19+00:00 2014-01-30 01:42:48+00:00
##                  funded_time term_in_months lender_count
## 1: 2014-01-02 10:06:32+00:00             12           12
## 2: 2014-01-02 09:17:23+00:00             11           14
## 3: 2014-01-01 16:01:36+00:00             43            6
## 4: 2014-01-01 13:00:00+00:00             11            8
## 5: 2014-01-01 19:18:51+00:00             14           16
## 6: 2014-01-29 14:14:57+00:00              4            6
##                            tags borrower_genders repayment_interval
## 1:                                        female          irregular
## 2:                                female, female          irregular
## 3: user_favorite, user_favorite           female             bullet
## 4:                                        female          irregular
## 5:                                        female            monthly
## 6:                                        female          irregular
##          date
## 1: 2014-01-01
## 2: 2014-01-01
## 3: 2014-01-01
## 4: 2014-01-01
## 5: 2014-01-01
## 6: 2014-01-01
```

```
str(kiva)
```

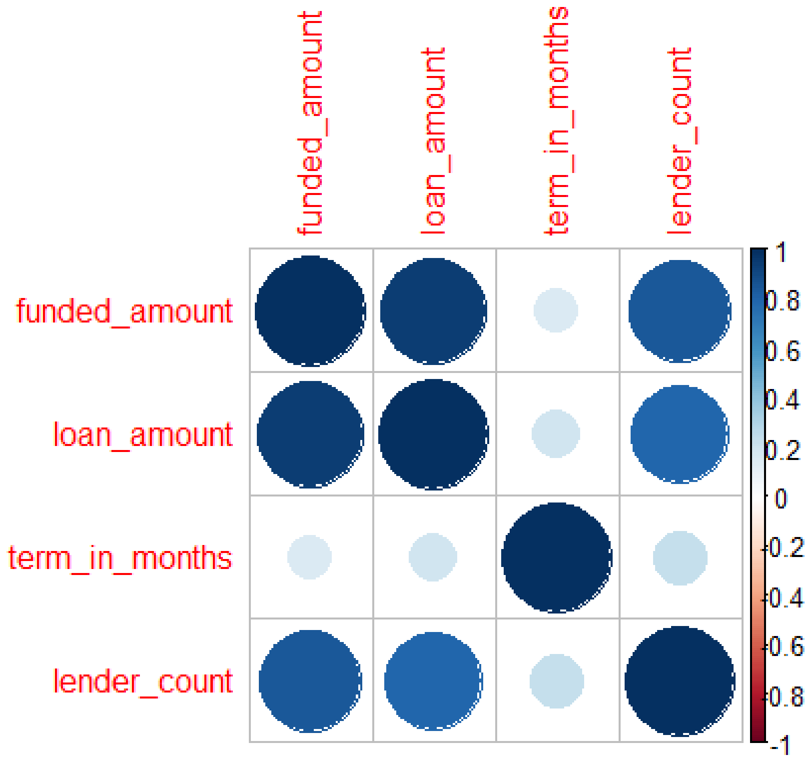
```
## Classes 'data.table' and 'data.frame':   671205 obs. of  20 variables:
##  $ id                 : int  653051 653053 653068 653063 653084 1080148
653067 653078 653082 653048 ...
##  $ funded_amount     : num  300 575 150 200 400 250 200 400 475 625 ...
##  $ loan_amount       : num  300 575 150 200 400 250 200 400 475 625 ...
##  $ activity          : chr  "Fruits & Vegetables" "Rickshaw"
"Transportation" "Embroidery" ...
##  $ sector            : chr  "Food" "Transportation" "Transportation"
"Arts" ...
```

```
##  $ use               : chr  "To buy seasonal, fresh fruits to sell. " "to repair and maintain the auto rickshaw used in their business." "To repair their old cycle-van and buy another one to rent out as a source of income" "to purchase an embroidery machine and a variety of new embroidery materials." ...
##  $ country_code      : chr  "PK" "PK" "IN" "PK" ...
##  $ country           : chr  "Pakistan" "Pakistan" "India" "Pakistan" ...
##  $ region            : chr  "Lahore" "Lahore" "Maynaguri" "Lahore" ...
##  $ currency          : chr  "PKR" "PKR" "INR" "PKR" ...
##  $ partner_id        : num  247 247 334 247 245 NA 334 245 245 247 ...
##  $ posted_time       : chr  "2014-01-01 06:12:39+00:00" "2014-01-01 06:51:08+00:00" "2014-01-01 09:58:07+00:00" "2014-01-01 08:03:11+00:00" ...
##  $ disbursed_time    : chr  "2013-12-17 08:00:00+00:00" "2013-12-17 08:00:00+00:00" "2013-12-17 08:00:00+00:00" "2013-12-24 08:00:00+00:00" ...
##  $ funded_time       : chr  "2014-01-02 10:06:32+00:00" "2014-01-02 09:17:23+00:00" "2014-01-01 16:01:36+00:00" "2014-01-01 13:00:00+00:00" ...
##  $ term_in_months    : num  12 11 43 11 14 4 43 14 14 11 ...
##  $ lender_count      : int  12 14 6 8 16 6 8 8 19 24 ...
##  $ tags              : chr  "" "" "user_favorite, user_favorite" "" ...
##  $ borrower_genders  : chr  "female" "female, female" "female" "female" ...
##  $ repayment_interval: chr  "irregular" "irregular" "bullet" "irregular" ...
##  $ date              : chr  "2014-01-01" "2014-01-01" "2014-01-01" "2014-01-01" ...
##  - attr(*, ".internal.selfref")=<externalptr>

dim(kiva)

## [1] 671205     20
```

The datset is large. To more easily explore it, we subset the data

Subset data for easier computational analysis in EDA

```
#set seed so reproducible
set.seed(1876)
# subsetting data
p <- .1 # proportion of data for EDA
w <- sample(1:nrow(kiva), nrow(kiva)*p, replace=F)
kiva_exp_subset <-kiva[w,]

rm(kiva) #remove the larger dataset Full dataset will be brought back in later

#numeric subset
kiva_num <- kiva_exp_subset[, c("funded_amount", "loan_amount", "term_in_months", "lender_count")]
kiva_num %>% cor() %>% corrplot(.)
```

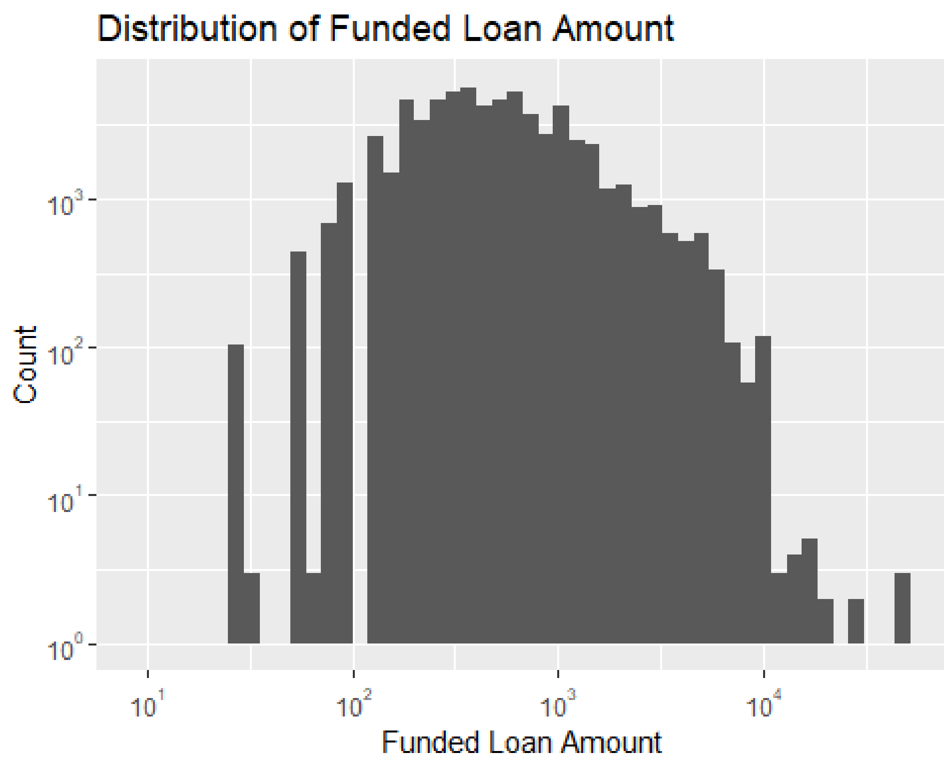


From the corrplot we see a correlation in the variables you might expect, such as loan amount and funded amount. Term in months is uncorrelated with the other variables. This confirms further assumptions

Next we explore the normality of the funded amount

```
fundedLoanAmountDistribution <- function(kiva_exp_subset)
{
 kiva_exp_subset %>%
    ggplot(aes(x = funded_amount) )+
    scale_x_log10(
      breaks = scales::trans_breaks("log10", function(x) 10^x),
      labels = scales::trans_format("log10", scales::math_format(10^.x))
    ) +
    scale_y_log10(
      breaks = scales::trans_breaks("log10", function(x) 10^x),
      labels = scales::trans_format("log10", scales::math_format(10^.x))
    ) +
    geom_histogram(bins=50) +
    labs(x = 'Funded Loan Amount' ,y = 'Count', title = paste("Distribution
of", "Funded Loan Amount"))
}

fundedLoanAmountDistribution(kiva_exp_subset)

## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Removed 334 rows containing non-finite values (stat_bin).

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 10 rows containing missing values (geom_bar).
```




```
kiva <- fread("BDS_WK11_kiva_loans.csv", header=T)
kiva2 <- kiva
```

Preprocessing

We begin by creatng new variables from the existting variables to express thier characterisaics for easier processing.We also remove all variables that have to do with the loan being funded such as the number of borrowers and dat it was funded. These are not helpful inpredicting if the loan will be funded because they would not be availible when predicting if a new loan application will be funded

```
kiva <- kiva2 %>%
  mutate( Funded= ifelse(funded_amount/loan_amount==1, "Yes", "No")) %>%
  mutate(number_borrower= str_count(borrower_genders, ',')+1 ) %>%
  mutate(number_tags= ifelse(tags=="",0, str_count(tags, ',')+1 ))%>%
  mutate(month= month(posted_time))%>%
  mutate(weekday=wday(posted_time, label = TRUE))

kiva$female_borrower <-  ifelse(grepl("female",kiva$borrower_genders), 1, 0)
kiva$male_borrower <-  ifelse(grepl("male",kiva$borrower_genders), 1, 0)
```

```r
kiva$continent <- factor(countrycode(sourcevar = kiva[, "country"],
                                     origin = "country.name",
                                     destination = "continent"))

#correct NA's indused by Kosovo not being recognized
kiva$continent[kiva$country=="Kosovo"]  <- "Europe"
kiva$continent[kiva$country=="Virgin Islands"]  <- "Americas"

kiva <- kiva%>%
  select(Funded, loan_amount, sector, continent,
repayment_interval,term_in_months,number_tags,tags,
number_borrower,female_borrower,male_borrower, month, weekday)

kiva_exp_subsetH2 <- kiva2 %>%
  mutate( Funded= ifelse(funded_amount/loan_amount>=1, "Yes", "No")) %>%
  dplyr::select(id, tags, Funded) %>%
  unnest(y = strsplit(tags, ","))

kiva_exp_subsetH2 $y <- gsub("\\#", "", kiva_exp_subsetH2 $y)
#kiva_exp_subsetH2 $y <- gsub("\\_", " ", kiva_exp_subsetH2 $y) # remove
hashtags replace with space to join like for like

kiva_exp_subsetH2 $y <- trim(kiva_exp_subsetH2 $y)

kiva_exp_subsetH2 <- kiva_exp_subsetH2 %>%
  group_by(y,Funded) %>%
  summarise(count=n()) %>%
  cast(y~Funded) %>%
  mutate(TotalTag= Yes+No) %>%
  arrange(desc(TotalTag))  %>%
  filter(TotalTag>=50) %>% # Only keep tags used at least 50 times
  mutate(PercentFunded=Yes /TotalTag) %>%
  arrange(desc(PercentFunded))
#make names easier to work with by removing spaces
names <- kiva_exp_subsetH2$y
names <- gsub("\\-", "_", names)
names <- gsub("\\ ", "_", names)
rm(kiva2)

R <- nrow(kiva_exp_subsetH2)
N <- nrow(kiva)
C <- ncol(kiva)

for (i in 1:R){
 kiva[,(C+i)] <-
ifelse(grepl(as.character(kiva_exp_subsetH2[i,1]),kiva$tags), 1, 0)
 kiva[,(C+i)] <- as.factor(kiva[,(C+i)])
 colnames(kiva)[(C+i)] <- names[i]
}
kiva <- kiva %>% select(-tags) #remove tags
```

Convert variables to factors

```
kiva$Funded <- as.factor(kiva$Funded)
kiva$sector <- as.factor(kiva$sector)
kiva$continent <- as.factor(kiva$continent)
kiva$repayment_interval <- as.factor(kiva$repayment_interval)
kiva$female_borrower <-as.factor(kiva$female_borrower)
kiva$male_borrower <-as.factor(kiva$male_borrower)
kiva$month <- as.factor(kiva$month)
save(kiva , file='cleankiva.RData')
```

```
load(file='cleankiva.RData')
str(kiva)
```

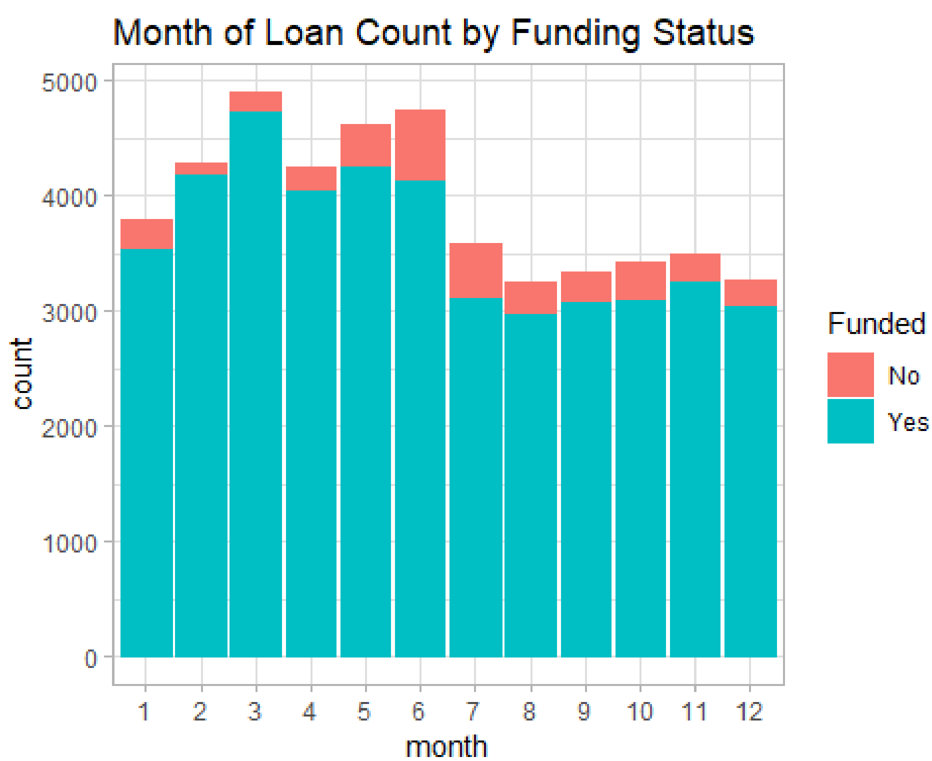
```
## 'data.frame':    671205 obs. of  46 variables:
##  $ Funded               : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2
2 ...
##  $ loan_amount          : num  300 575 150 200 400 250 200 400 475 625 ...
##  $ sector               : Factor w/ 15 levels "Agriculture",..: 7 14 14 2
7 13 1 13 10 7 ...
##  $ continent            : Factor w/ 5 levels "Africa","Americas",..: 3 3 3
3 3 1 3 3 3 3 ...
##  $ repayment_interval   : Factor w/ 4 levels "bullet","irregular",..: 2 2
1 2 3 2 1 3 3 2 ...
##  $ term_in_months       : num  12 11 43 11 14 4 43 14 14 11 ...
##  $ number_tags          : num  0 0 2 0 0 0 2 2 1 0 ...
##  $ number_borrower      : num  1 2 1 1 1 1 1 1 1 1 ...
##  $ female_borrower      : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2
...
##  $ male_borrower        : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2
...
##  $ month                : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1
1 1 1 1 1 1 ...
##  $ weekday              : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 4 4
4 4 4 4 4 4 4 4 ...
##  $ Female_Education     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Widowed              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Health_and_Sanitation: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Orphan               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Eco_friendly         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Post_disbursed       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Fabrics              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ Technology           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
```

```
...
## $ volunteer_like      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ volunteer_pick      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Single_Parent       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Interesting_Photo   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ user_favorite       : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 2 1 2 1
...
## $ Woman_Owned_Biz     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1
...
## $ Inspiring_Story     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Schooling           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Animals             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Repeat_Borrower     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Unique              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Elderly             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1
...
## $ Low_profit_FP       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ First_Loan          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Sustainable_Ag      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Parent              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Biz_Durable_Asset   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Trees               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Vegan               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Single              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Hidden_Gem          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Supporting_Family   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Job_Creator         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Tourism             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
## $ Refugee             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
```

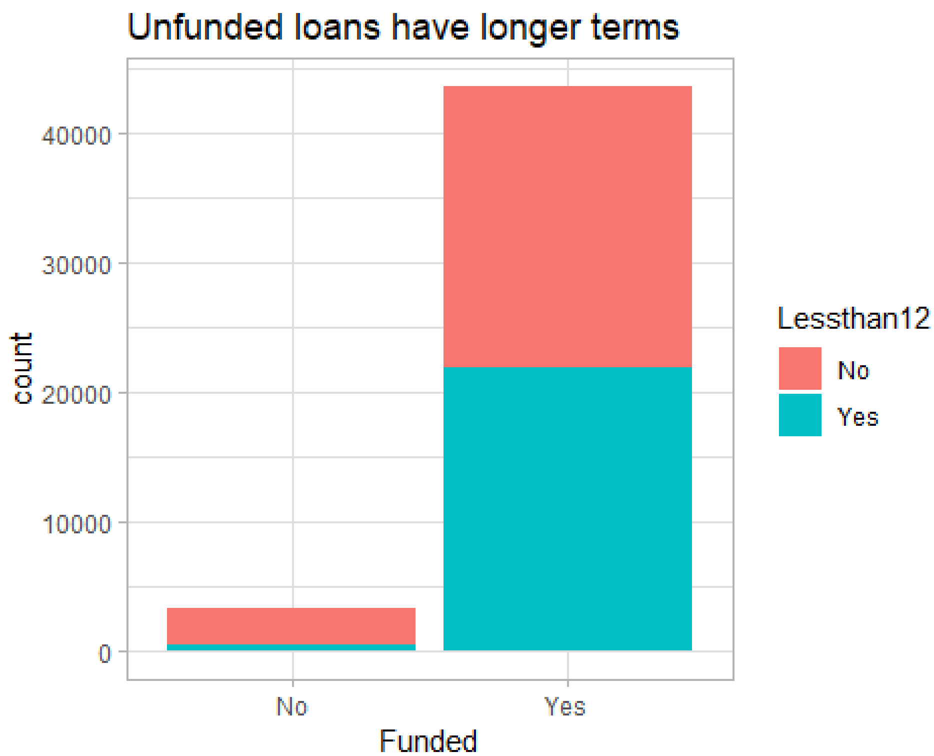
```
...
##  $ Repair_Renew_Replace : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
```

EDA

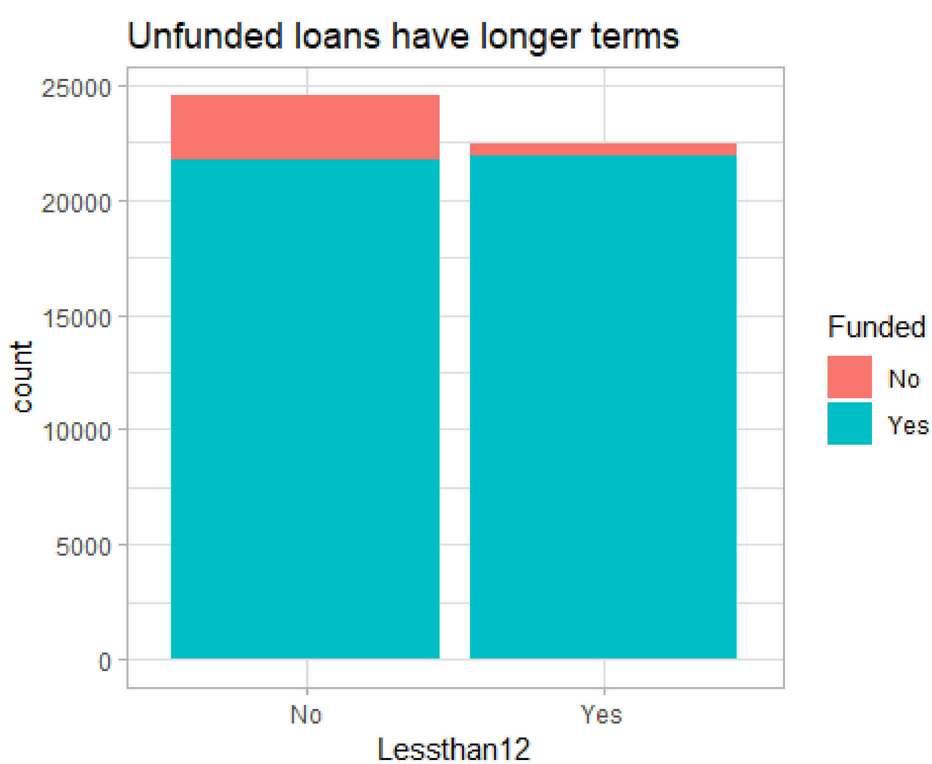
```r
#subset for EDA
set.seed(1900)

p <- .07 # proportion of data for training
w <- sample(1:nrow(kiva), nrow(kiva)*p, replace=F)
eda <-kiva[w,]

ggplot(eda, aes(x=month, color=Funded, fill=Funded))+ geom_bar()+
theme_light()+ labs(title="Month of Loan Count by Funding Status")
```




```r
kiva_exp_subsetH1 <- eda %>%
  mutate(Lessthan12=ifelse(eda$term_in_months<=12,"Yes", "No"))

ggplot(kiva_exp_subsetH1, aes(Funded, fill=Lessthan12))+ geom_bar() +
theme_light()+ labs(title="Unfunded loans have longer terms")
```

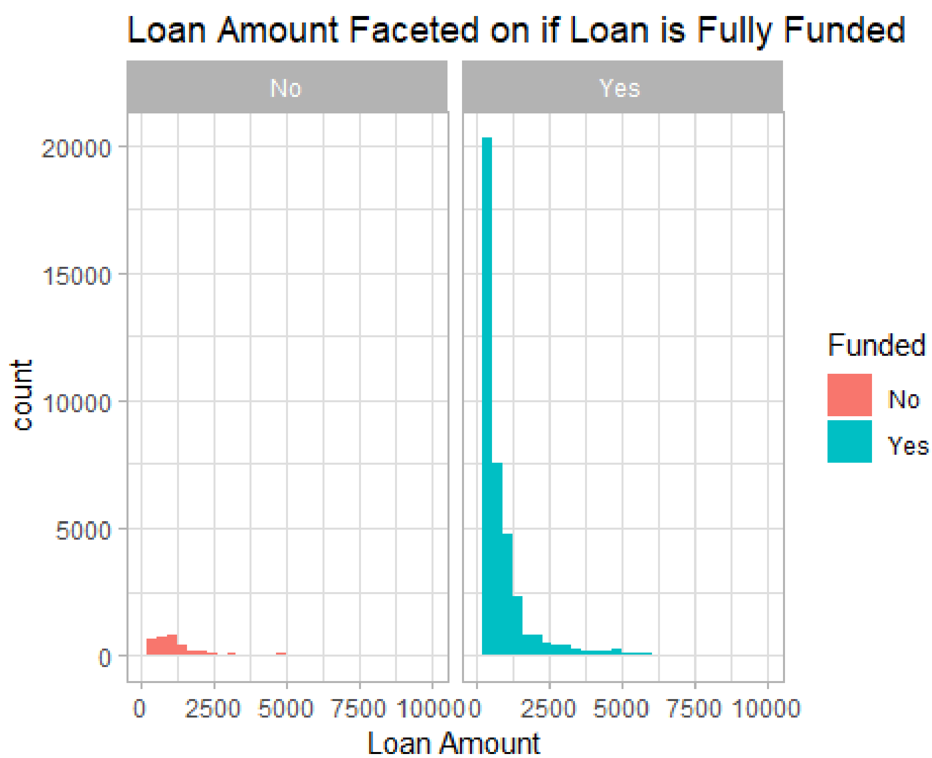



```
ggplot(kiva_exp_subsetH1, aes(Lessthan12,fill=Funded))+ geom_bar()+
theme_light()+ labs(title="Unfunded loans have longer terms")
```

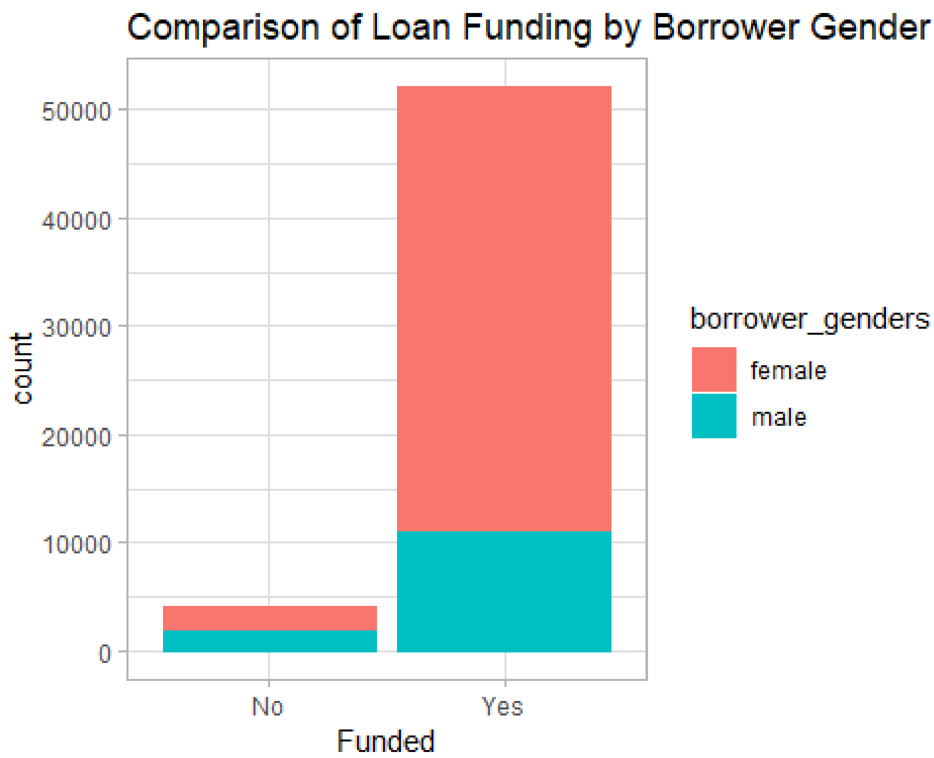


```r
ggplot(eda, aes(x=loan_amount, fill=Funded))+ geom_histogram()+
facet_wrap(~Funded)+ xlim(c(0,10000))+ theme_light()+ labs(x="Loan Amount",
title= "Loan Amount Faceted on if Loan is Fully Funded")
```

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

## Warning: Removed 15 rows containing non-finite values (stat_bin).
```




```r
kiva_exp_subsetH3 <- kiva_exp_subset %>%
  mutate( Funded= ifelse(funded_amount/loan_amount==1, "Yes", "No")) %>%
  dplyr::select(id, borrower_genders, Funded) %>%
  mutate(Single_bor= ifelse(borrower_genders=="female", "Yes",
ifelse(borrower_genders=="male", "Yes", "No"))) %>%
  filter(Single_bor=="Yes")

ggplot(kiva_exp_subsetH3, aes(x=Funded, color=borrower_genders,
fill=borrower_genders))+ geom_bar()+ labs(title= "Comparison of Loan Funding
by Borrower Gender")+ theme_light()
```



Initial Model Testing

```r
set.seed(1876)
#subset in training and testing dataset
p <- .07 # proportion of data for training
w <- sample(1:nrow(kiva), nrow(kiva)*p, replace=F)
train <-kiva[w,]
test <- kiva[-w,]
rm(kiva)

res.rf <- randomForest(Funded~.,data= train, mtry=20, ntree=500,
importance=T, na.action= na.omit)
res.rf

save(res.rf , file='randfor.RData')

load(file='randfor.RData')

importance(res.rf)

##                             No        Yes MeanDecreaseAccuracy
## loan_amount        159.3137021 30.3667533           91.2300661
## sector              59.6060927 50.7562192           72.0587363
## continent           54.1155591  2.2768598           29.6879112
## repayment_interval  51.7091785 43.2011174           66.2276673
## term_in_months     158.1440288 16.0955891           76.7352289
## number_tags         14.1324098 37.9975563           51.5933458
```

```
## number_borrower        17.7227160  15.5334722             22.0626756
## female_borrower       102.5439631 -11.5209976             32.6499086
## male_borrower          23.7671047   1.7449201             12.0790024
## month                  92.7048608  22.5750531             62.5068036
## weekday                14.3154541   3.7158628             10.4016777
## Female_Education        3.0921400  -1.4823576             -0.5310527
## Widowed                -2.4013744  -1.8294056             -2.5968134
## Health_and_Sanitation   4.6821736   3.1843362              4.9860400
## Orphan                  0.0000000  -2.0088163             -2.0097806
## Eco_friendly           11.6825547   4.2081992              9.3626891
## Post_disbursed          2.0489717  -5.3035190             -4.5217510
## Fabrics                 3.9131245  -7.6892915             -6.0776910
## Technology              0.8528648  11.1041294             11.9523264
## volunteer_like          0.8557324  -2.0787635             -1.4667117
## volunteer_pick         -0.1434360  -1.6319615             -1.5181082
## Single_Parent          -1.0800973   4.9848822              4.5093550
## Interesting_Photo      -0.6436776 -11.7321889            -11.6118092
## user_favorite           3.3672671  41.0277233             41.0372096
## Woman_Owned_Biz        25.0938726  17.4614781             25.1271334
## Inspiring_Story         1.4393598  -5.2558133             -4.6524158
## Schooling               6.4885740   7.5363908              9.6654443
## Animals                11.4388625   0.9830051              5.7149211
## Repeat_Borrower        15.4488591   3.7315876             10.1264639
## Unique                  0.4179853  -4.9237601             -4.3887068
## Elderly                15.3655413  -2.3054688              4.5479565
## Low_profit_FP          17.1218174  -7.5342837              2.7827579
## First_Loan              6.1417335  -8.6550760             -5.6328370
## Sustainable_Ag          8.1618941  -3.1237515             -0.6007846
## Parent                 22.3306797  16.0322404             23.2933901
## Biz_Durable_Asset       4.3128187   2.1936210              3.9548051
## Trees                   3.3541731  -3.9924102             -2.6559292
## Vegan                  18.9676636   4.4073326             11.7767247
## Single                 -0.8292126   5.3341977              5.1703212
## Hidden_Gem              2.1757490  -7.3790909             -6.3478581
## Supporting_Family      10.5609622  -5.7150152             -1.4669896
## Job_Creator             8.4015188  -2.0866145              1.7837731
## Tourism                 0.0000000   0.0000000              0.0000000
## Refugee                 4.3770849  -4.5379616             -2.5399975
## Repair_Renew_Replace   33.9016163 -10.0122834             10.8030725
##                       MeanDecreaseGini
## loan_amount               1141.9337107
## sector                     558.8610887
## continent                  240.7265284
## repayment_interval         203.3055813
## term_in_months             721.4456827
## number_tags                492.0618358
## number_borrower            139.2232220
## female_borrower            171.8293121
## male_borrower               27.0769567
## month                      750.9957442
```

```
## weekday                          487.0324810
## Female_Education                   2.9237330
## Widowed                           14.5880002
## Health_and_Sanitation             21.4331906
## Orphan                             1.0297924
## Eco_friendly                      26.2842750
## Post_disbursed                     4.6154028
## Fabrics                           27.9293822
## Technology                        23.2149089
## volunteer_like                    26.4356228
## volunteer_pick                    31.5197921
## Single_Parent                     22.2865459
## Interesting_Photo                 15.4506117
## user_favorite                    125.9066019
## Woman_Owned_Biz                   73.6636798
## Inspiring_Story                    8.4918327
## Schooling                         66.6142631
## Animals                           60.8549253
## Repeat_Borrower                  102.2829057
## Unique                            11.0584507
## Elderly                          100.9785632
## Low_profit_FP                     15.1985066
## First_Loan                        45.7151902
## Sustainable_Ag                    17.1221263
## Parent                           120.9145251
## Biz_Durable_Asset                 62.2524819
## Trees                             19.0984005
## Vegan                             75.7062786
## Single                            52.4601209
## Hidden_Gem                         6.0395391
## Supporting_Family                 54.6882783
## Job_Creator                       35.5240369
## Tourism                            0.0697919
## Refugee                           13.2801821
## Repair_Renew_Replace              54.4583844
```

```
prediction <- predict(res.rf, test, type = "class")
# Checking classification accuracy
table(prediction, test$Funded)
```

```
##
## prediction     No    Yes
##        No    7710   4452
##        Yes  37199 574860
```

```
confusionMatrix(prediction, test$Funded, positive = "No")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     No    Yes
```

```
##        No    7710   4452
##        Yes  37199 574860
##
##                Accuracy : 0.9333
##                  95% CI : (0.9327, 0.9339)
##     No Information Rate : 0.9281
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2471
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.17168
##             Specificity : 0.99232
##          Pos Pred Value : 0.63394
##          Neg Pred Value : 0.93922
##              Prevalence : 0.07194
##          Detection Rate : 0.01235
##    Detection Prevalence : 0.01948
##       Balanced Accuracy : 0.58200
##
##        'Positive' Class : No
##
```

Try to balance

```
table(train$Funded)
```

```
##
##    No   Yes
##  3421 43563
```

```
set.seed(1876)
#Downsampled Model
down_train <- downSample(x = train[,-1],
                         y = train$Funded)
table(down_train$Funded)
```

```
## < table of extent 0 >
```

```
#Upsampled Model
up_train <- upSample(x = train[,-1],
                         y = train$Funded)
table(up_train$Funded)
```

```
## < table of extent 0 >
```

```
set.seed(1876)
#SMOTE
smote_train <- SMOTE(Funded ~ ., data  = train, perc.over = 100,
perc.under=200)
save(smote_train, file = "Smotedat.Rdata")
```

```r
load("Smotedat.Rdata")

set.seed(1876)
downrf <- randomForest(Class~.,data= down_train, mtry=20, ntree=500,
importance=T, na.action= na.omit)
save(downrf, file="down.Rdata")

load(file="down.Rdata")
prediction_down <- predict(downrf, test, type = "class")
confusionMatrix(prediction_down, test$Funded, positive = "No")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     No    Yes
##        No   38977 125429
##        Yes   5932 453883
##
##                Accuracy : 0.7896
##                  95% CI : (0.7885, 0.7906)
##     No Information Rate : 0.9281
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2925
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.86791
##             Specificity : 0.78349
##          Pos Pred Value : 0.23708
##          Neg Pred Value : 0.98710
##              Prevalence : 0.07194
##          Detection Rate : 0.06244
##    Detection Prevalence : 0.26338
##       Balanced Accuracy : 0.82570
##
##        'Positive' Class : No
##

set.seed(1876)

uprf <- randomForest(Class~.,data= up_train, mtry=20, ntree=500,
importance=T, na.action= na.omit)
save(uprf,file= "upf.Rdata")

load(file= "upf.Rdata")
prediction_up <- predict(uprf, test, type = "class")
confusionMatrix(prediction_up, test$Funded, positive = "No")

## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction     No    Yes
##        No   13756  13967
##        Yes  31153 565345
##
##                Accuracy : 0.9277
##                  95% CI : (0.9271, 0.9284)
##     No Information Rate : 0.9281
##     P-Value [Acc > NIR] : 0.8499
##
##                   Kappa : 0.3427
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.30631
##             Specificity : 0.97589
##          Pos Pred Value : 0.49619
##          Neg Pred Value : 0.94777
##              Prevalence : 0.07194
##          Detection Rate : 0.02204
##    Detection Prevalence : 0.04441
##       Balanced Accuracy : 0.64110
##
##        'Positive' Class : No
##
```

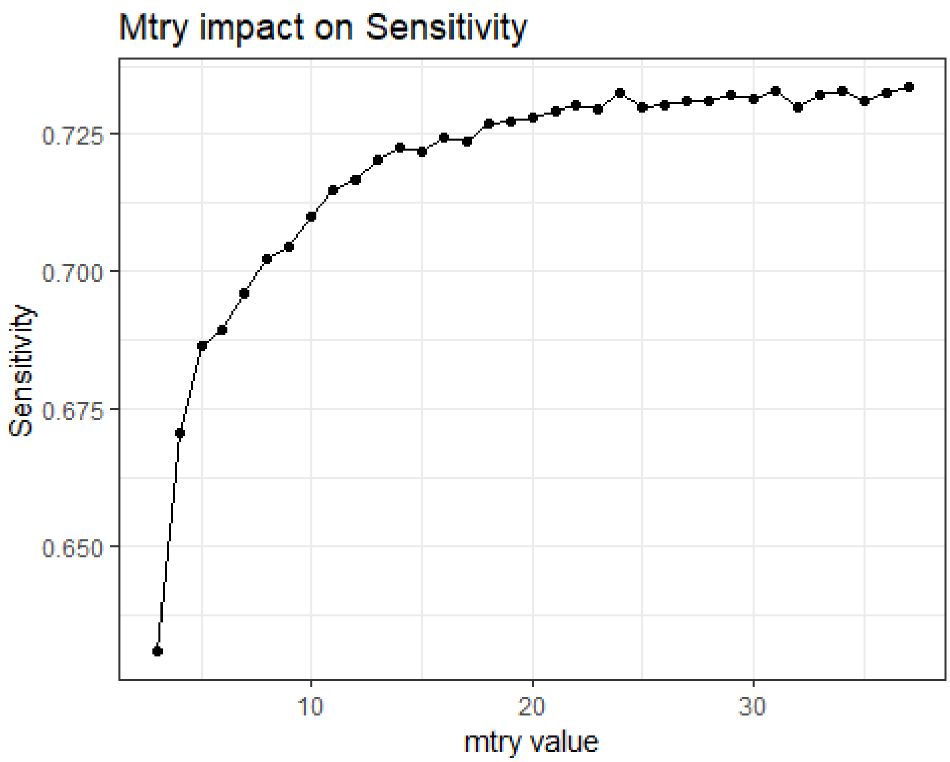
```
set.seed(1876)

smoterf <- randomForest(Funded~.,data= smote_train, mtry=20, ntree=500,
importance=T, na.action= na.omit)
save(smoterf,file= "smote.Rdata")

load(file= "smote.Rdata")
prediction_smote <- predict(smoterf, test, type = "class")
confusionMatrix(prediction_smote, test$Funded, positive = "No")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     No    Yes
##        No   32748  77621
##        Yes  12161 501691
##
##                Accuracy : 0.8562
##                  95% CI : (0.8553, 0.857)
##     No Information Rate : 0.9281
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3559
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.72921
```

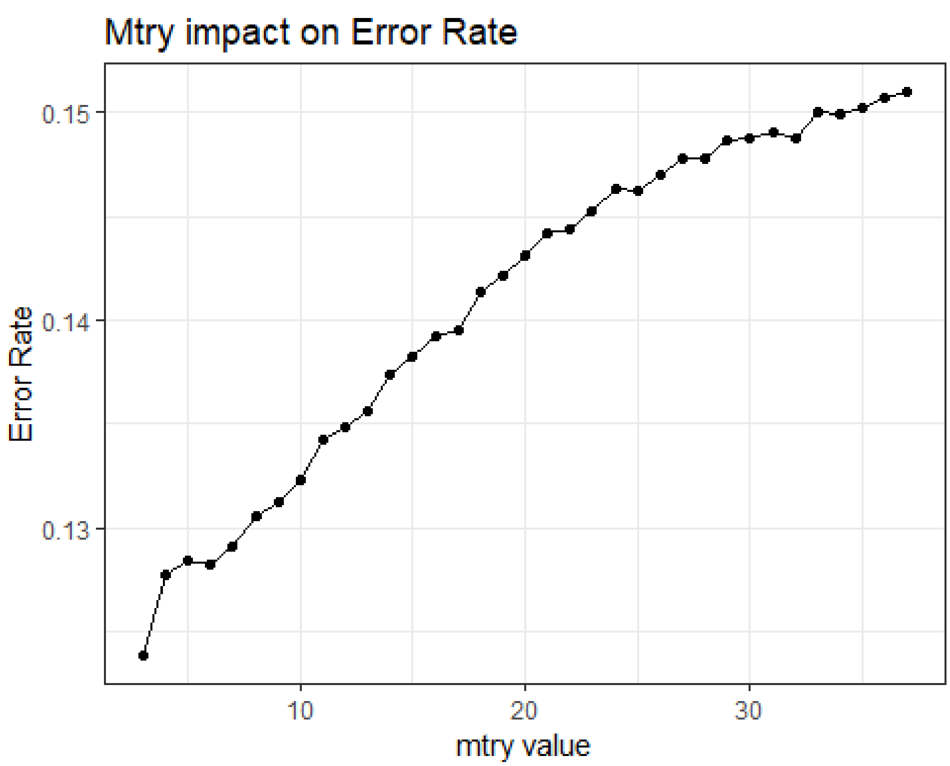
```
##             Specificity : 0.86601
##          Pos Pred Value : 0.29671
##          Neg Pred Value : 0.97633
##              Prevalence : 0.07194
##          Detection Rate : 0.05246
##    Detection Prevalence : 0.17681
##       Balanced Accuracy : 0.79761
##
##        'Positive' Class : No
##
```

Optimize mtry

```
# Using For loop to identify the right mtry for model

i=37
sensitivity_matrix <- matrix(0, ncol=1, nrow=i)
err_matrix <- matrix(0, ncol=1, nrow=i)


for (r in 3:40) {
  model <- randomForest(Funded~.,data= smote_train, mtry=r, ntree=500,
importance=T, na.action= na.omit)
  predValid <- predict(model, test, type = "class")
    cm <- confusionMatrix(predValid, test$Funded, positive = "No")
        #store data

         err_matrix [[r,1]] <-  (cm$table[1,2]+cm$table[2,1])/nrow( test)

         sensitivity_matrix[[r, 1]] <- cm$byClass[1]
}
 save(err_matrix, file="err.Rdata")
 save(sensitivity_matrix, file="ses.Rdata")

load("ses.Rdata")
load("err.Rdata")

graph <- as.data.frame(cbind(sensitivity_matrix, err_matrix,
row.names(err_matrix)))
graph$index <- c(1:37)
ggplot(graph[-c(1:2),], aes(x=index, y=V1))+
geom_line()+geom_point()+labs(x="mtry value", y= "Sensitivity", title= "Mtry
impact on Sensitivity")+ theme_bw()
```

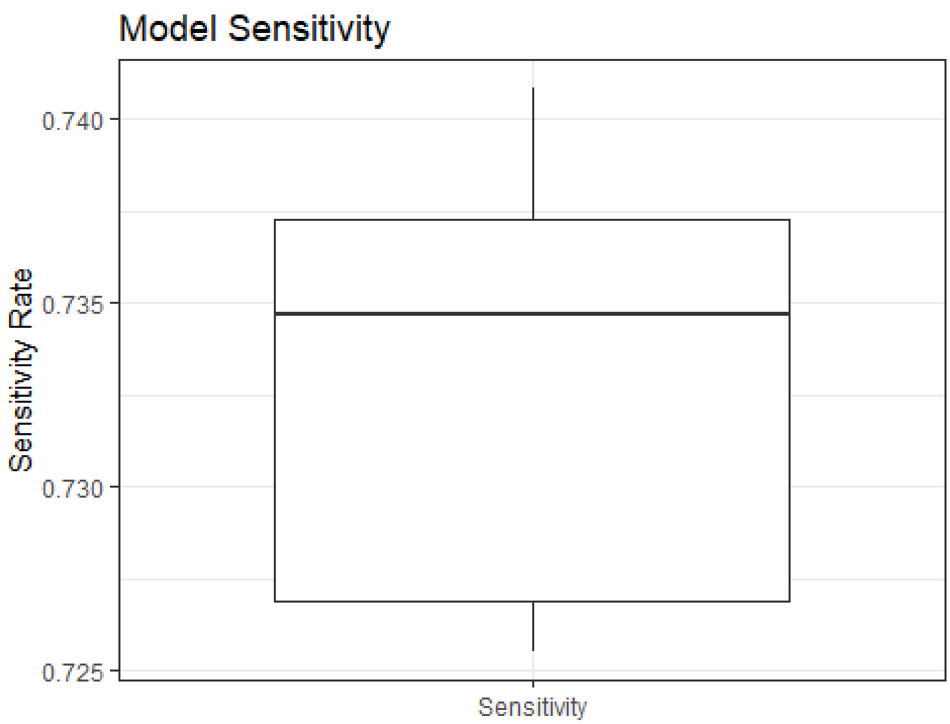



```
ggplot(graph[-c(1:2),], aes(x=index, y=V2))+ geom_line()+geom_point()
+labs(x="mtry value", y= "Error Rate", title= "Mtry impact on Error Rate")+
theme_bw()
```

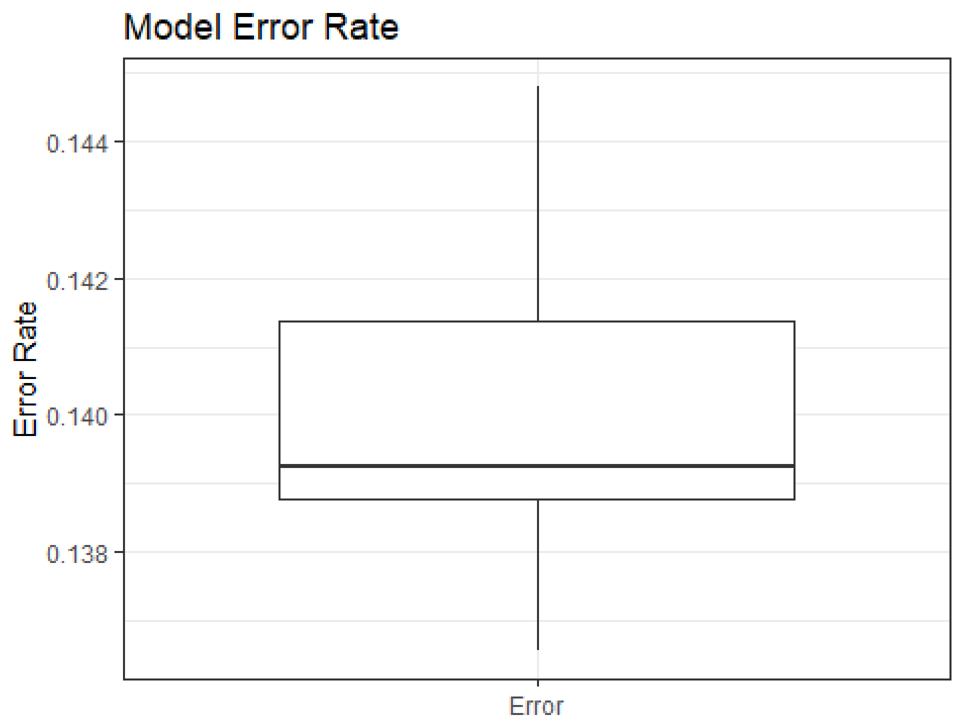


```r
R <- 10 # replications

# create the matrix to store values 1 row per model
err_matrix2 <- matrix(0, ncol=1, nrow=R)

sensitivity_matrix2 <- matrix(0, ncol=1, nrow=R)

fmeasure_matrix2 <- matrix(0, ncol=1, nrow=R)

gmean_matrix2 <- matrix(0, ncol=1, nrow=R)


set.seed(1876)

for (r in 1:R){

# subsetting data to training and testing data
p <- .07 # proportion of data for training
w <- sample(1:nrow(kiva), nrow(kiva)*p, replace=F)
train <-kiva[w,]
test <- kiva[-w,]

smote_train <- SMOTE(Funded ~ ., data  = train, perc.over = 100,
perc.under=200)

#run model
rf_fin <- randomForest(Funded~.,data= smote_train, mtry=15, ntree=500,
importance=T, na.action= na.omit)


#make prediction
prediction_fin <- predict(rf_fin, test, type = "class")
      #create CM
      cm <- confusionMatrix(prediction_fin, test$Funded, positive = "No")
        #store data

         err_matrix2 [[r,1]] <-  (cm$table[1,2]+cm$table[2,1])/nrow( test)

         sensitivity_matrix2[[r, 1]] <- cm$byClass[1]

         fmeasure_matrix2 [[r, 1]] <- cm$byClass[7]

         gmean_matrix2 [[r, 1]] <- sqrt(cm$byClass[1]* cm$byClass[2])

}

save(err_matrix2, file="err2.Rdata")
save(sensitivity_matrix2, file="sens2.Rdata")
```

```r
save(fmeasure_matrix2, file="fmeas2.Rdata")
save(gmean_matrix2, file="gmean2.Rdata")
save(rf_fin, file="Finmod.Rdata")

load(file="sens2.Rdata")
load(file="err2.Rdata")

sens <- as.data.frame(sensitivity_matrix2)
sens$heading <- "Sensitivity"
ggplot(sens, aes(x=heading, y=V1))+ geom_boxplot() +labs(x="", y=
"Sensitivity Rate", title= "Model Sensitivity")+ theme_bw()
```




```r
err <- as.data.frame(err_matrix2)
err$heading <- "Error"
ggplot(err, aes(x=heading, y=V1))+ geom_boxplot() +labs(x="", y= "Error
Rate", title= "Model Error Rate")+ theme_bw()
```



#Final Model

```
set.seed(1988)
p <- .07 # proportion of data for training
w <- sample(1:nrow(kiva), nrow(kiva)*p, replace=F)
train <-kiva[w,]
test <- kiva[-w,]

smote_train <- SMOTE(Funded ~ ., data  = train, perc.over = 100,
perc.under=200)

#run model
rf_fin <- randomForest(Funded~.,data= smote_train, mtry=15, ntree=500,
importance=T, na.action= na.omit)


#make prediction
prediction_fin <- predict(rf_fin, test, type = "class")

confusionMatrix(prediction_fin, test$Funded, positive = "No")

save(rf_fin, file="final.Rdata")
save(prediction_fin, file = "prediction_fin.Rdata")

load( file="final.Rdata")

load( file = "prediction_fin.Rdata")
```

```r
confusionMatrix(prediction_fin, test$Funded, positive = "No")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     No    Yes
##        No   10375  97807
##        Yes  34534 481505
##
##                Accuracy : 0.788
##                  95% CI : (0.787, 0.789)
##     No Information Rate : 0.9281
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0377
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.23102
##             Specificity : 0.83117
##          Pos Pred Value : 0.09590
##          Neg Pred Value : 0.93308
##              Prevalence : 0.07194
##          Detection Rate : 0.01662
##    Detection Prevalence : 0.17331
##       Balanced Accuracy : 0.53109
##
##        'Positive' Class : No
##

load(file='cleankiva.RData')
importance <- (importance(rf_fin)[,4])

#change structure of importance to work with ggplot
importance<- as.list(importance)
names(importance) <- colnames(kiva[,-1])
importance2 <- unlist(importance)
most_sig_stats <- (sort(desc(importance2)))*-1
most_sig_stats<- as.data.frame(most_sig_stats)
row.names(most_sig_stats)

##  [1] "loan_amount"           "term_in_months"
##  [3] "month"                 "sector"
##  [5] "number_tags"           "weekday"
##  [7] "repayment_interval"    "female_borrower"
##  [9] "continent"             "Parent"
## [11] "number_borrower"       "user_favorite"
## [13] "Woman_Owned_Biz"       "Repeat_Borrower"
## [15] "Elderly"               "Vegan"
## [17] "Animals"               "Biz_Durable_Asset"
```

```
## [19] "Schooling"              "Single"
## [21] "Supporting_Family"      "Fabrics"
## [23] "First_Loan"             "Eco_friendly"
## [25] "Repair_Renew_Replace"   "volunteer_pick"
## [27] "Health_and_Sanitation"  "Sustainable_Ag"
## [29] "Single_Parent"          "Technology"
## [31] "volunteer_like"         "male_borrower"
## [33] "Job_Creator"            "Trees"
## [35] "Widowed"                "Interesting_Photo"
## [37] "Refugee"                "Inspiring_Story"
## [39] "Low_profit_FP"          "Unique"
## [41] "Hidden_Gem"             "Female_Education"
## [43] "Post_disbursed"         "Orphan"
## [45] "Tourism"
```

```
most_sig_stats[,2] <- row.names(most_sig_stats)

#Word cloud shows importance of variables
ggplot(most_sig_stats, aes(size= most_sig_stats, label= V2,   color =
factor(sample.int(12, 45, replace = TRUE))))+
  geom_text_wordcloud() +
theme_minimal()
```