Final Report LClub

Italo Sayan

8/2/2017

Describing and predicting loan default on the lending club dataset

The loans were issued from 2007 to 2015

Loading the data

```
library(readr)
dataLC <- read_csv("~/datalendingclub.csv")
```

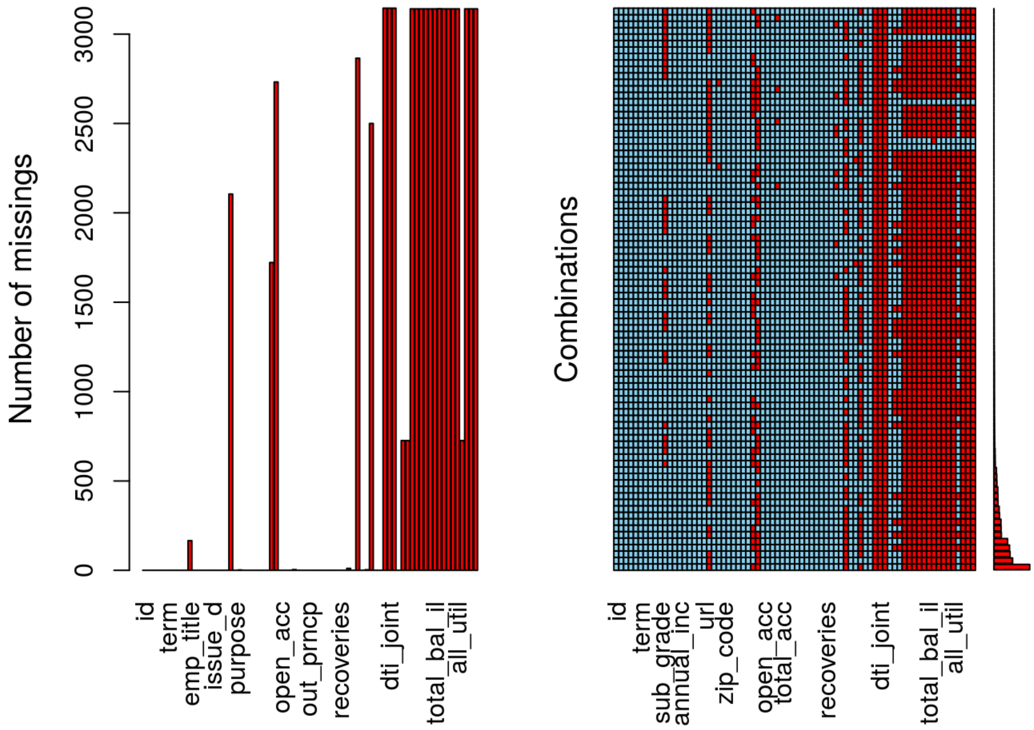
Formatting the loan status variable

```
dataLC <- dataLC[!(dataLC$loan_status == "Current"),]

bad_indicators <- c("Charged Off",
                    "Default",
                    "Does not meet the credit policy. Status:Charged Off",
                    "In Grace Period",
                    "Default Receiver",
                    "Late (16-30 days)",
                    "Late (31-120 days)")

dataLC$loan_status[dataLC$loan_status %in% bad_indicators] <- 'Default'
dataLC$loan_status[dataLC$loan_status == 'Does not meet the credit policy. Status:Fully Paid'] <- 'Fully Pai
d'
dataLC <- dataLC[!dataLC$loan_status == 'Issued',]
rm(bad_indicators)
```

Checking for missing values

```
library(VIM)
```

```
aggr(dataLC, prop= FALSE, numbers = TRUE)
```



Cleaning missing values

```
# Remove variables with more than 20% of mising values
dataLC <- dataLC[!(colMeans(is.na(dataLC)) > 0.2)]

# Data eliminated for:
# Being constant for all values.
# Being imposible to know when the loan in issued
# Irrelevant : URL, id , member id

dataLC$id <-NULL
dataLC$url <- NULL
dataLC$desc <- NULL
dataLC$title <- NULL
dataLC$issue_d <- NULL
dataLC$sub_grade <- NULL
dataLC$member_id <- NULL
dataLC$out_prncp <- NULL
dataLC$emp_title <- NULL
dataLC$revol_bal <- NULL
dataLC$recoveries <- NULL
dataLC$addr_state <- NULL
dataLC$pymnt_plan <- NULL
dataLC$policy_code <- NULL
dataLC$total_pymnt <- NULL
dataLC$funded_amnt <- NULL
dataLC$policy_code <- NULL
dataLC$last_pymnt_d <- NULL
dataLC$next_pymnt_d <- NULL
dataLC$out_prncp_inv <- NULL
dataLC$total_rec_int <- NULL
dataLC$last_pymnt_amnt <- NULL
dataLC$total_pymnt_inv <- NULL
dataLC$total_rec_prncp <- NULL
dataLC$funded_amnt_inv <- NULL
dataLC$application_type <- NULL
dataLC$earliest_cr_line <- NULL
dataLC$earliest_cr_line <- NULL
dataLC$total_rec_late_fee <- NULL
dataLC$last_credit_pull_d <- NULL
dataLC$initial_list_status <- NULL
dataLC$collection_recovery_fee <- NULL
dataLC$verification_status_joint <- NULL
dataLC$collections_12_mths_ex_med <- NULL
dataLC$verification_status <- NULL
```

Formatting the variables

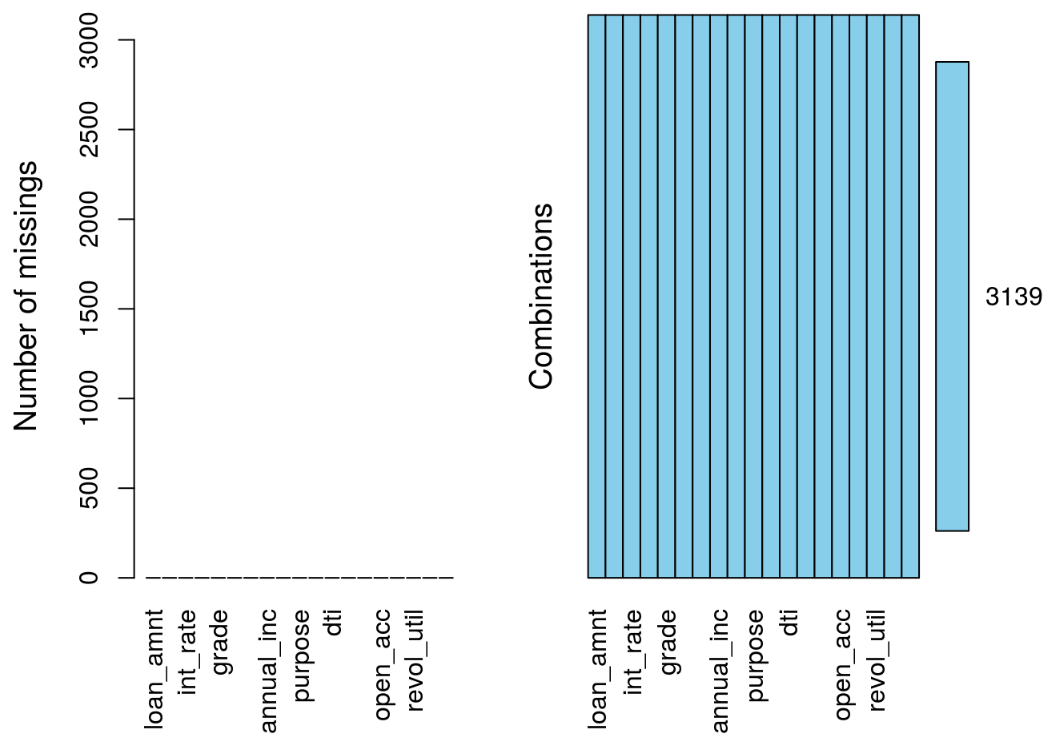
```
# Formatting each of the variables that will be used in the model

# Term
# Replace months word and making the variable numeric
dataLC$term <- as.numeric(gsub(" months","", dataLC$term))

# Grade
# Turning it to a factor variable
dataLC$grade <- factor(dataLC$grade)

# Employment length
# It has multiple issues
dataLC$emp_length[1:100]
```

```
##   [1] "5 years"   "3 years"   "1 year"    "5 years"   "10+ years"
##   [6] "10+ years" "4 years"   "10+ years" "5 years"   "10+ years"
##  [11] "1 year"    "2 years"   "10+ years" "10+ years" "10+ years"
##  [16] "10+ years" "10+ years" "1 year"    "1 year"    "10+ years"
##  [21] "< 1 year"  "2 years"   "10+ years" "< 1 year"  "8 years"
##  [26] "10+ years" "8 years"   "3 years"   "6 years"   "4 years"
##  [31] "3 years"   "< 1 year"  "< 1 year"  "1 year"    "1 year"
##  [36] "2 years"   "1 year"    "10+ years" "2 years"   "4 years"
##  [41] "8 years"   "6 years"   "n/a"       "10+ years" "2 years"
##  [46] "1 year"    "5 years"   "< 1 year"  "7 years"   "7 years"
##  [51] "10+ years" "< 1 year"  "10+ years" "4 years"   "4 years"
##  [56] "10+ years" "5 years"   "10+ years" "10+ years" "6 years"
##  [61] "6 years"   "3 years"   "5 years"   "2 years"   "10+ years"
##  [66] "3 years"   "3 years"   "2 years"   "8 years"   "10+ years"
##  [71] "1 year"    "3 years"   "3 years"   "4 years"   "1 year"
##  [76] "10+ years" "10+ years" "< 1 year"  "3 years"   "8 years"
##  [81] "4 years"   "10+ years" "7 years"   "2 years"   "6 years"
##  [86] "7 years"   "n/a"       "10+ years" "2 years"   "1 year"
##  [91] "1 year"    "1 year"    "8 years"   "10+ years" "5 years"
##  [96] "10+ years" "9 years"   "5 years"   "< 1 year"  "5 years"
```

```
# First if employment length is less than 1 year replace it with 0.5
dataLC$emp_length <- ifelse(dataLC$emp_length == '< 1 year', 0.5 ,dataLC$emp_length)
# Then if employment length is more than 10 years use a random number between 10:19
# The way of handling emp_length can vary
dataLC$emp_length <- ifelse(dataLC$emp_length == "10+ years",sample(10:19, nrow(dataLC[dataLC$emp_length ==
"10+ years",]),replace = TRUE ),dataLC$emp_length)
# Remove the n/a with 0 employment length
dataLC$emp_length <-gsub("n/a",0,dataLC$emp_length)
# Eliminate any left words using regex
dataLC$emp_length <- gsub('[ a-z]','',dataLC$emp_length)
# Making employment length a numeric variable
dataLC$emp_length <- as.numeric(dataLC$emp_length)

# Remove the "OTHER" category from home ownership
table(dataLC$home_ownership)
```

```
##
## MORTGAGE    OTHER      OWN     RENT
##     1504        2      287     1352
```

```r
dataLC <- dataLC[!(dataLC$home_ownership == "OTHER"),]
dataLC$home_ownership <- factor(dataLC$home_ownership)

#Purpose
dataLC$purpose <- factor(dataLC$purpose)

#Loan Status
dataLC$loan_status <- factor(dataLC$loan_status)

#Zip Code
dataLC$zip_code <- gsub('xx', '', dataLC$zip_code)
dataLC$zip_code <- as.integer(dataLC$zip_code)

#Dti: monthly payments divided by monthly income
dataLC$dti <- as.numeric(dataLC$dti)

#Delinquencies in 2yrs
dataLC$delinq_2yrs <- as.integer(dataLC$delinq_2yrs)

#Inq_last_6mths
dataLC$inq_last_6mths <- as.integer(dataLC$inq_last_6mths)

#open_acc
dataLC$open_acc <- as.integer(dataLC$open_acc)

#pub_rec
dataLC$pub_rec <- as.integer(dataLC$pub_rec)

#total_acc
dataLC$total_acc <- as.integer(dataLC$total_acc)

#Annual Income and final cleaning
dataLC <- dataLC[complete.cases(dataLC$annual_inc),]
dataLC <- dataLC[complete.cases(dataLC),]
dataLC$annual_inc <- as.numeric(dataLC$annual_inc)

aggr(dataLC, prop= FALSE, numbers = TRUE)
```




```r
str(dataLC)
```

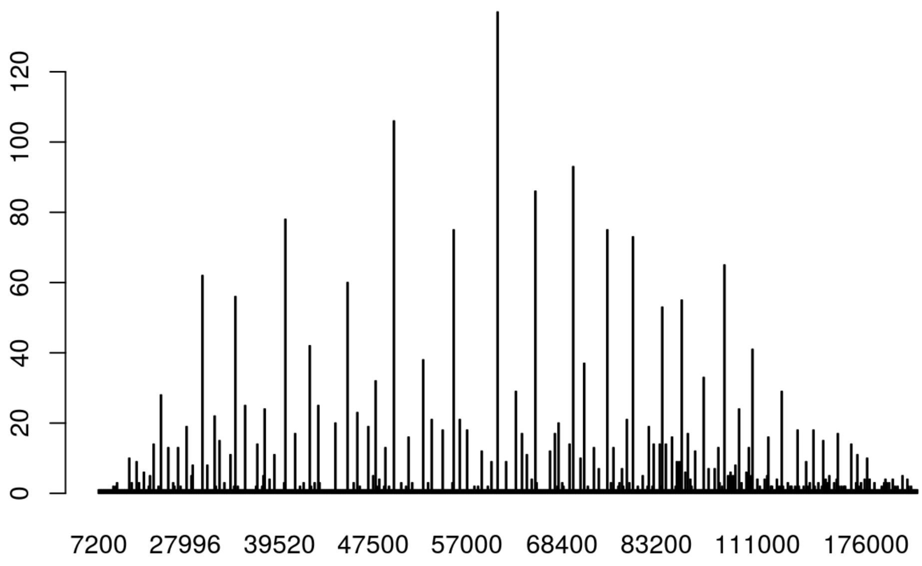
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    3139 obs. of  19 variables:
##  $ loan_amnt     : int  32000 8000 17000 32500 30000 20000 20000 14000 14700 6000 ...
##  $ term          : num  60 36 60 36 60 36 36 36 36 36 ...
##  $ int_rate      : num  17.9 15 16.3 13 12.4 ...
##  $ installment   : num  810 277 416 1095 673 ...
##  $ grade         : Factor w/ 7 levels "A","B","C","D",..: 4 3 4 3 3 3 1 2 4 1 ...
##  $ emp_length    : num  5 3 1 5 14 13 4 19 5 11 ...
##  $ home_ownership: Factor w/ 3 levels "MORTGAGE","OWN",..: 1 3 3 2 1 1 1 3 1 2 ...
##  $ annual_inc    : num  80000 55000 40000 295000 110000 75000 100000 80000 66000 47000 ...
##  $ loan_status   : Factor w/ 2 levels "Default","Fully Paid": 1 1 1 2 2 1 2 2 2 2 ...
##  $ purpose       : Factor w/ 14 levels "car","credit_card",..: 3 2 3 3 3 2 3 3 3 2 ...
##  $ zip_code      : int  448 917 940 112 761 488 226 467 298 275 ...
##  $ dti           : num  39.75 6.15 10.38 10.05 10.54 ...
##  $ delinq_2yrs   : int  5 0 0 1 1 0 0 1 0 0 ...
##  $ inq_last_6mths: int  0 2 0 1 0 1 0 0 1 0 ...
##  $ open_acc      : int  22 7 5 16 18 13 18 11 10 5 ...
##  $ pub_rec       : int  1 0 0 0 0 0 0 2 1 0 ...
##  $ revol_util    : num  9.2 74.5 40.1 84.2 36.6 79.6 54.3 56.1 65 36.8 ...
##  $ total_acc     : int  41 17 8 28 31 42 30 22 24 14 ...
##  $ acc_now_delinq: int  0 0 0 0 0 0 0 0 0 0 ...
```

Exploration

```
library(DescTools)
#Analizing annual income
Desc(dataLC$annual_inc, main = "Annual income distribution",plotit = FALSE)
```

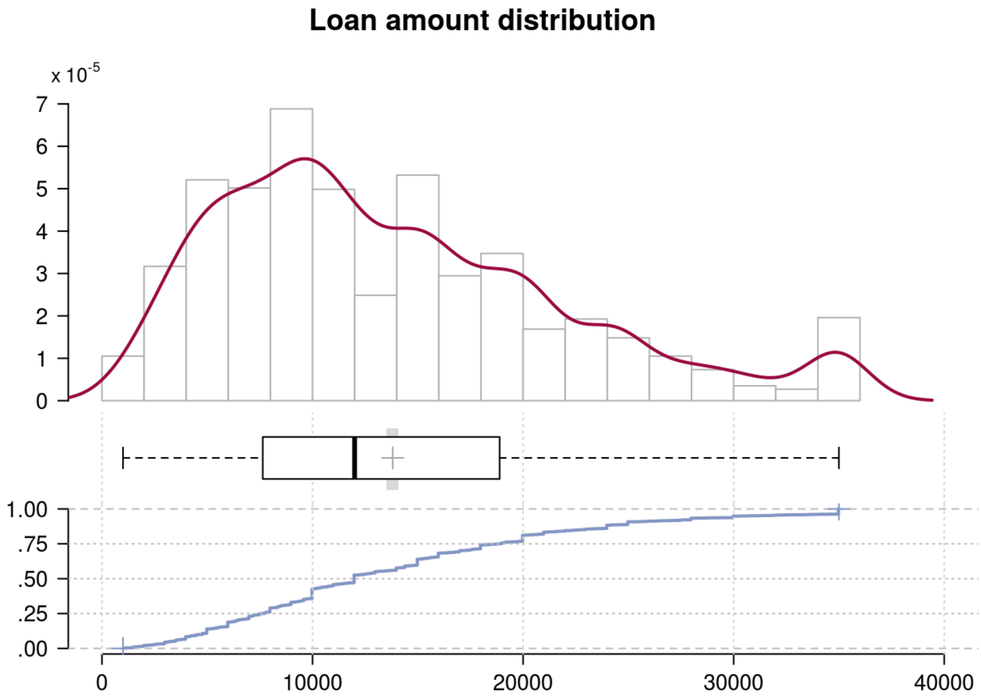
```
## ------------------------------------------------------------------------- 
## Annual income distribution
## 
##         length          n        NAs     unique         0s         mean
##          3'139      3'139          0        672          0    74'055.29
##                    100.0%       0.0%                   0.0%             
## 
##            .05        .10        .25     median        .75          .90
##      27'000.00  32'000.00  45'000.00  62'000.00  87'000.00   120'000.00
## 
##          range         sd      vcoef        mad        IQR         skew
##   4'992'800.00 100'109.26       1.35  29'652.00  42'000.00        38.42
## 
##       meanCI
##    70'551.85
##    77'558.73
## 
##          .95
##   148'000.00
## 
##         kurt
##     1'866.76
## 
## lowest : 7'200.0, 9'240.0, 10'043.0, 10'200.0, 10'560.0
## highest: 500'000.0, 650'000.0, 735'000.0, 750'000.0, 5'000'000.0
```

```
barplot(table(dataLC$annual_inc))
```



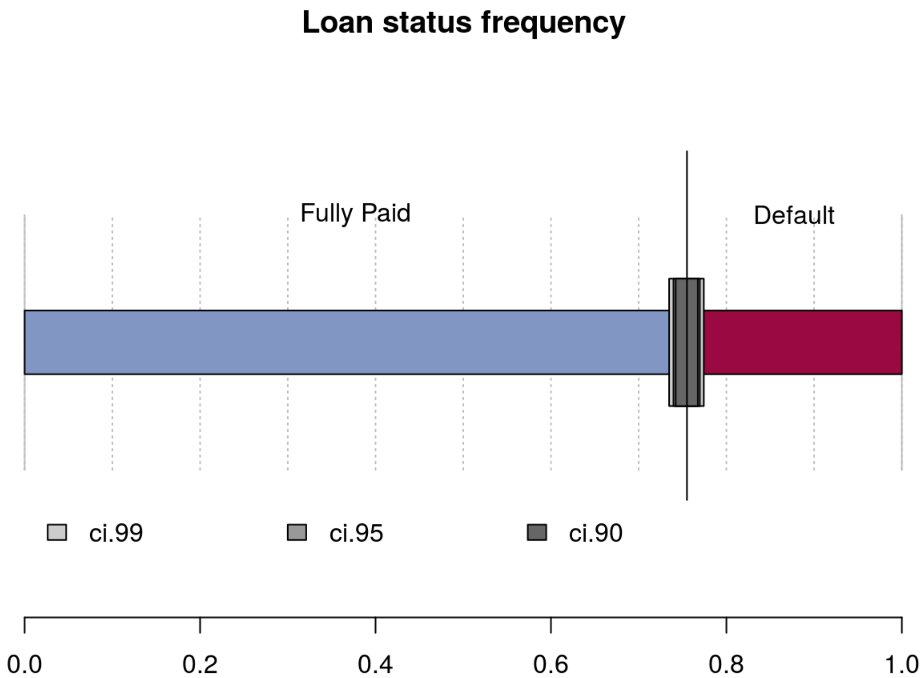

```
#Analizing loan amounts
Desc(dataLC$loan_amnt, main = "Loan amount distribution", plotit = TRUE)
```

```
## ----------------------------------------------------------------------------
## Loan amount distribution
##
##       length         n        NAs     unique         0s        mean
##        3'139     3'139          0        531          0   13'807.73
##                 100.0%       0.0%                  0.0%
##
##          .05       .10        .25     median        .75         .90
##     3'300.00  4'745.00   7'637.50  12'000.00  18'887.50   25'000.00
##
##        range        sd      vcoef        mad        IQR        skew
##    34'000.00  8'201.17       0.59   8'154.30  11'250.00        0.82
##
##       meanCI
##    13'520.72
##    14'094.74
##
##          .95
##    30'755.00
##
##         kurt
##         0.11
##
## lowest : 1'000 (9), 1'200 (3), 1'400 (4), 1'450 (3), 1'500 (11)
## highest: 33'950 (2), 34'100 (2), 34'475 (2), 34'500, 35'000 (118)
```



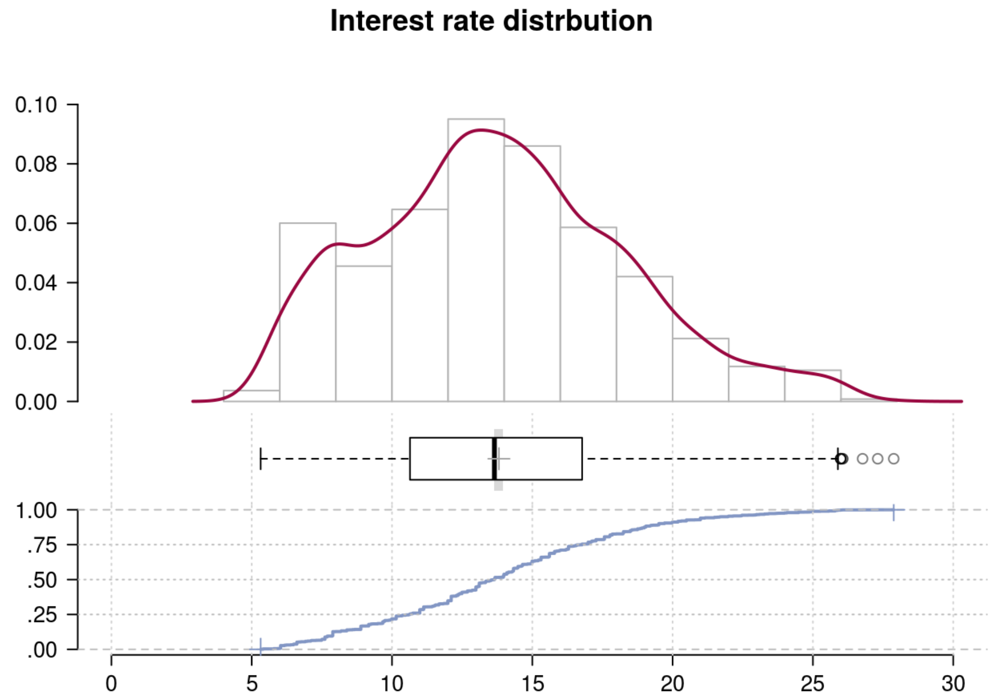

```
#Analizing loan status
Desc(dataLC$loan_status,main = "Loan status frequency" ,plotit = T)
```

```
## ------------------------------------------------------------------------
## Loan status frequency
##
##    length       n     NAs unique
##     3'139   3'139       0      2
##           100.0%    0.0%
##
##                 freq   perc  lci.95  uci.95'
## Fully Paid  2'370  75.5%   74.0%   77.0%
## Default       769  24.5%   23.0%   26.0%
##
## ' 95%-CI Wilson
```



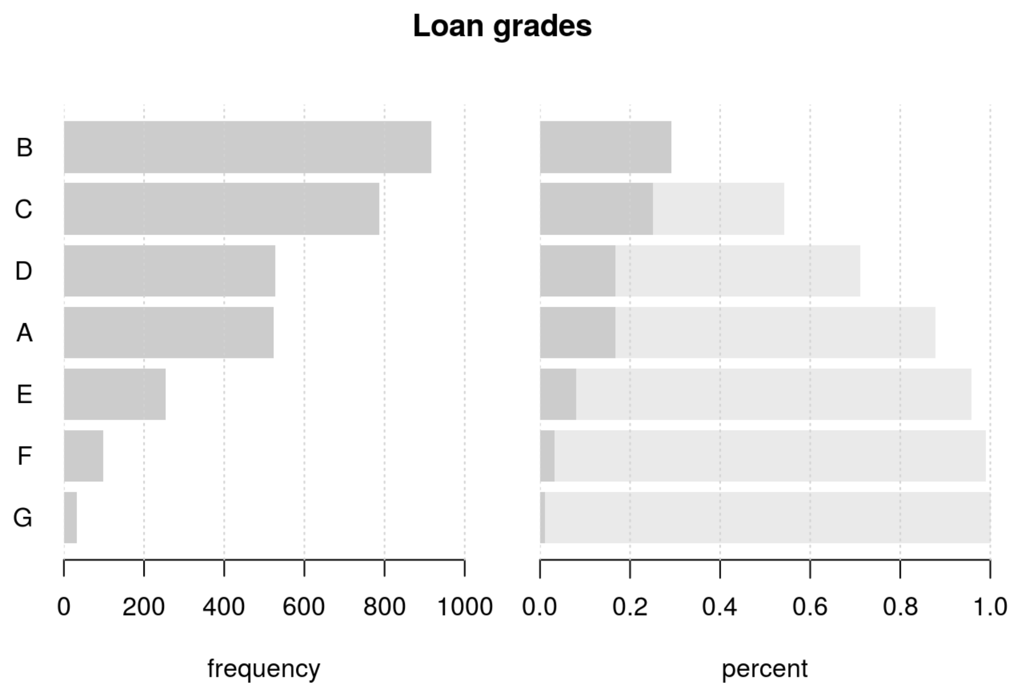
```r
#Interest rate distribution
Desc(dataLC$int_rate ,main = "Interest rate distrbution" ,plotit = T)
```

```
## ------------------------------------------------------------------------- 
## Interest rate distrbution
##
##    length        n      NAs   unique       0s     mean   meanCI
##     3'139    3'139        0      316        0   13.811   13.654
##             100.0%     0.0%              0.0%            13.968
##
##       .05      .10      .25   median      .75      .90      .95
##     6.620    7.890   10.640   13.650   16.775   19.520   21.764
##
##     range       sd    vcoef      mad      IQR     skew     kurt
##    22.560    4.480    0.324    4.507    6.135    0.357   -0.242
##
## lowest : 5.32 (3), 5.42 (8), 5.79 (6), 5.93 (2), 5.99 (4)
## highest: 25.99 (4), 26.06 (2), 26.77, 27.31, 27.88
```




```r
#Loan Grade graph
Desc(dataLC$grade, main = "Loan grades", plotit = TRUE)
```

```
## ------------------------------------------------------------------------- 
## Loan grades
##
##    length       n     NAs unique levels  dupes
##     3'139   3'139       0      7      7      y
##            100.0%    0.0%
##
##     level  freq   perc  cumfreq  cumperc
## 1       B   917  29.2%      917    29.2%
## 2       C   786  25.0%    1'703    54.3%
## 3       D   528  16.8%    2'231    71.1%
## 4       A   524  16.7%    2'755    87.8%
## 5       E   253   8.1%    3'008    95.8%
## 6       F    99   3.2%    3'107    99.0%
## 7       G    32   1.0%    3'139   100.0%
```



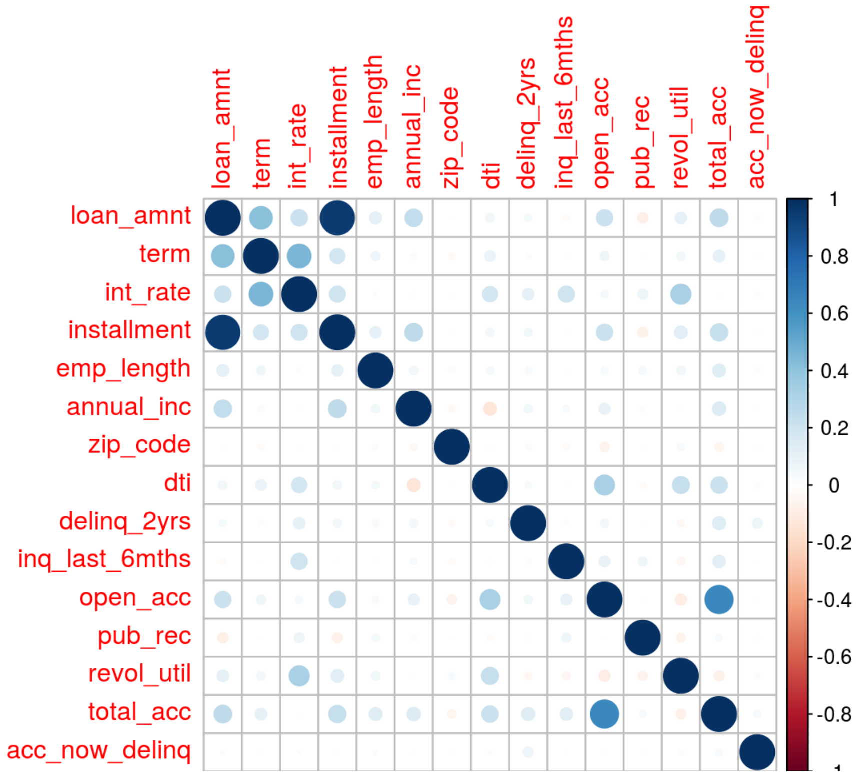
Comparative statistics between defaulted and paid loans

```
library(psych)
describeBy(dataLC[,c('loan_amnt','int_rate','annual_inc','emp_length')], group=dataLC$loan_status)
```

```
##
##  Descriptive statistics by group
## group: Default
##            vars   n     mean       sd   median  trimmed      mad     min
## loan_amnt     1 769 14738.98  8351.45 12875.00 14020.71  7894.84 1200.00
## int_rate      2 769    15.85     4.32    15.61    15.72     4.27    5.42
## annual_inc    3 769 65572.97 41594.87 60000.00 60677.89 29652.00 7200.00
## emp_length    4 769     6.53     5.59     5.00     5.95     5.93    0.00
##                  max     range skew kurtosis      se
## loan_amnt   35000.00  33800.00 0.70    -0.14  301.16
## int_rate       27.31     21.89 0.23    -0.26    0.16
## annual_inc 735000.00 727800.00 6.29    87.78 1499.95
## emp_length     19.00     19.00 0.71    -0.67    0.20
## --------------------------------------------------------
## group: Fully Paid
##            vars    n     mean        sd   median  trimmed      mad     min
## loan_amnt     1 2370 13505.57   8130.70 12000.00 12658.04  8154.30 1000.00
## int_rate      2 2370    13.15      4.33    12.99    12.94     4.40    5.32
## annual_inc    3 2370 76807.57 112619.87 65000.00 67574.30 31134.60 9240.00
## emp_length    4 2370     7.04      5.59     6.00     6.54     5.93    0.00
##                  max      range  skew kurtosis      se
## loan_amnt  3.500e+04   34000.00  0.87     0.21  167.01
## int_rate   2.788e+01      22.56  0.42    -0.17    0.09
## annual_inc 5.000e+06 4990760.00 35.58  1539.20 2313.35
## emp_length 1.900e+01      19.00  0.62    -0.79    0.11
```

Correlation between varibles

```
#function that filters numberic variables
getNumericColumns<-function(t){
    tn = sapply(t,function(x){is.numeric(x)})
    return(names(tn)[which(tn)])
}
library(corrplot)
#correlation of numeric variables
corrplot(cor(dataLC[getNumericColumns(dataLC)],use="na.or.complete"))
```



Modelling how grade is determined

```
#logistic regression
set.seed(2)

train <- as.vector(sample(1:nrow(dataLC), nrow(dataLC)/3))

glm.grade <- lm(as.numeric(dataLC$grade) ~  term + emp_length + home_ownership + annual_inc + purpose + del
inq_2yrs + revol_util , data=dataLC ,subset=train)
summary(glm.grade)
```

```
##
## Call:
## lm(formula = as.numeric(dataLC$grade) ~ term + emp_length + home_ownership +
##     annual_inc + purpose + delinq_2yrs + revol_util, data = dataLC,
##     subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7540 -0.7935 -0.1574  0.6895  3.7303
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.481e+00  3.751e-01  -3.948 8.43e-05 ***
## term                       6.364e-02  3.474e-03  18.317  < 2e-16 ***
## emp_length                -1.148e-02  6.181e-03  -1.858 0.063514 .
## home_ownershipOWN          3.255e-01  1.284e-01   2.534 0.011415 *
## home_ownershipRENT         2.281e-01  7.511e-02   3.037 0.002451 **
## annual_inc                -1.283e-06  7.269e-07  -1.764 0.077947 .
## purposecredit_card         4.480e-01  3.445e-01   1.300 0.193752
## purposedebt_consolidation  7.807e-01  3.375e-01   2.313 0.020919 *
## purposeeducational         3.875e+00  1.153e+00   3.362 0.000803 ***
## purposehome_improvement    8.230e-01  3.736e-01   2.203 0.027848 *
## purposehouse               6.317e-01  5.948e-01   1.062 0.288454
## purposemajor_purchase      9.575e-01  4.119e-01   2.325 0.020283 *
## purposemedical             9.558e-01  5.132e-01   1.862 0.062832 .
## purposemoving              1.015e+00  5.203e-01   1.950 0.051429 .
## purposeother               1.218e+00  3.685e-01   3.304 0.000985 ***
## purposerenewable_energy    9.341e-01  8.476e-01   1.102 0.270662
## purposesmall_business      1.619e+00  3.934e-01   4.114 4.20e-05 ***
## purposevacation            2.485e+00  6.474e-01   3.839 0.000131 ***
## purposewedding             8.285e-01  5.361e-01   1.545 0.122587
## delinq_2yrs                1.997e-01  3.891e-02   5.133 3.41e-07 ***
## revol_util                 1.678e-02  1.409e-03  11.906  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.101 on 1025 degrees of freedom
## Multiple R-squared:  0.3638, Adjusted R-squared:  0.3514
## F-statistic: 29.31 on 20 and 1025 DF,  p-value: < 2.2e-16
```

```
#Adjusted R-squared:  0.3462
glm.int <- lm(dataLC$int_rate ~  grade , data=dataLC ,subset=train)
summary(glm.int)
```

```
##
## Call:
## lm(formula = dataLC$int_rate ~ grade, data = dataLC, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5411 -0.8628  0.0702  0.9489  2.8396
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.4828     0.1056   70.88   <2e-16 ***
## gradeB        3.9857     0.1307   30.49   <2e-16 ***
## gradeC        7.1084     0.1354   52.50   <2e-16 ***
## gradeD       10.0805     0.1474   68.40   <2e-16 ***
## gradeE       12.7776     0.1818   70.30   <2e-16 ***
## gradeF       16.0683     0.2479   64.82   <2e-16 ***
## gradeG       17.4728     0.4668   37.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.364 on 1039 degrees of freedom
## Multiple R-squared:  0.9085, Adjusted R-squared:  0.908
## F-statistic:  1719 on 6 and 1039 DF,  p-value: < 2.2e-16
```

Grade clearly determines interest rate Adjusted R-squared: 0.904

Logistic Regression

```
set.seed(2)

train <- as.vector(sample(1:nrow(dataLC), nrow(dataLC)/3))

temp <- model.matrix(loan_status~ 0+ ., data=dataLC)

loan_status <- dataLC$loan_status == 'Default'

dataLCLog <- as.data.frame(cbind(loan_status,temp))

glm.fit=glm(loan_status ~.,data=dataLCLog,family=binomial,subset=train)

summary(glm.fit)
```

```
##
## Call:
## glm(formula = loan_status ~ ., family = binomial, data = dataLCLog,
##     subset = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.55234  -0.76147  -0.55474  -0.00025   2.56805
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.569e+01  6.862e+02  -0.023 0.981759
## loan_amnt                 -1.724e-04  7.974e-05  -2.162 0.030636 *
## term                       6.777e-02  1.968e-02   3.444 0.000573 ***
## int_rate                  -1.189e-01  6.068e-02  -1.960 0.049978 *
## installment                5.116e-03  2.447e-03   2.090 0.036582 *
## gradeA                    -3.087e+00  1.284e+00  -2.404 0.016202 *
## gradeB                    -2.261e+00  1.090e+00  -2.074 0.038041 *
## gradeC                    -1.360e+00  9.583e-01  -1.419 0.155852
## gradeD                    -1.206e+00  8.641e-01  -1.396 0.162729
## gradeE                    -6.918e-01  8.061e-01  -0.858 0.390761
## gradeF                     4.901e-02  8.019e-01   0.061 0.951264
## gradeG                            NA         NA      NA       NA
## emp_length                -1.942e-02  1.472e-02  -1.319 0.187282
## home_ownershipOWN          3.559e-01  2.829e-01   1.258 0.208379
## home_ownershipRENT         4.052e-01  1.750e-01   2.316 0.020564 *
## annual_inc                -1.671e-06  2.262e-06  -0.739 0.460169
## purposecredit_card         1.521e+01  6.862e+02   0.022 0.982312
## purposedebt_consolidation  1.513e+01  6.862e+02   0.022 0.982407
## purposeeducational        -2.409e+00  2.496e+03  -0.001 0.999230
## purposehome_improvement    1.503e+01  6.862e+02   0.022 0.982525
## purposehouse               1.620e+01  6.862e+02   0.024 0.981159
## purposemajor_purchase      1.538e+01  6.862e+02   0.022 0.982117
## purposemedical             1.525e+01  6.862e+02   0.022 0.982270
## purposemoving              1.595e+01  6.862e+02   0.023 0.981460
## purposeother               1.500e+01  6.862e+02   0.022 0.982557
## purposerenewable_energy   -3.871e-01  1.784e+03   0.000 0.999827
## purposesmall_business      1.605e+01  6.862e+02   0.023 0.981344
## purposevacation           -6.870e-01  1.333e+03  -0.001 0.999589
## purposewedding             1.467e+01  6.862e+02   0.021 0.982940
## zip_code                  -5.999e-04  2.505e-04  -2.395 0.016623 *
## dti                        2.595e-02  1.146e-02   2.264 0.023578 *
## delinq_2yrs                9.734e-02  8.429e-02   1.155 0.248124
## inq_last_6mths             4.734e-02  7.353e-02   0.644 0.519695
## open_acc                  -1.857e-02  2.270e-02  -0.818 0.413322
## pub_rec                    8.245e-02  1.362e-01   0.605 0.545049
## revol_util                 6.234e-03  3.658e-03   1.704 0.088338 .
## total_acc                 -9.260e-03  9.327e-03  -0.993 0.320834
## acc_now_delinq            -1.647e+01  2.400e+03  -0.007 0.994523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1168.7  on 1045  degrees of freedom
## Residual deviance: 1032.6  on 1009  degrees of freedom
## AIC: 1106.6
##
## Number of Fisher Scoring iterations: 15
```

Low Sensitivity Logistic Regression Model

```
library(caret)
glm.probs=predict(glm.fit,newdata=dataLCLog[-train,],type="response")
glm.pred=ifelse(glm.probs>0.5,"Default","Fully Paid")

groundtrue <- dataLC$loan_status[-train]

confusionMatrix(table(glm.pred,groundtrue))
```

```
## Confusion Matrix and Statistics
##
##               groundtrue
## glm.pred      Default Fully Paid
##   Default          79         81
##   Fully Paid      432       1501
##
##                Accuracy : 0.7549
##                  95% CI : (0.7359, 0.7732)
##     No Information Rate : 0.7559
##     P-Value [Acc > NIR] : 0.5523
##
##                   Kappa : 0.1347
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.15460
##             Specificity : 0.94880
##          Pos Pred Value : 0.49375
##          Neg Pred Value : 0.77651
##              Prevalence : 0.24415
##          Detection Rate : 0.03774
##    Detection Prevalence : 0.07645
##       Balanced Accuracy : 0.55170
##
##        'Positive' Class : Default
##
```

High Sensitivity Logistic Regression Model

```
glm.pred=ifelse(glm.probs>0.2,"Default","Fully Paid")

confusionMatrix(table(glm.pred,groundtrue))
```

```
## Confusion Matrix and Statistics
##
##               groundtrue
## glm.pred      Default Fully Paid
##   Default         374        730
##   Fully Paid      137        852
##
##                Accuracy : 0.5858
##                  95% CI : (0.5643, 0.607)
##     No Information Rate : 0.7559
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1942
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7319
##             Specificity : 0.5386
##          Pos Pred Value : 0.3388
##          Neg Pred Value : 0.8615
##              Prevalence : 0.2441
##          Detection Rate : 0.1787
##    Detection Prevalence : 0.5275
##       Balanced Accuracy : 0.6352
##
##        'Positive' Class : Default
##
```

Random Forest

```
str(dataLC)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    3139 obs. of  19 variables:
##  $ loan_amnt     : int  32000 8000 17000 32500 30000 20000 20000 14000 14700 6000 ...
##  $ term          : num  60 36 60 36 60 36 36 36 36 36 ...
##  $ int_rate      : num  17.9 15 16.3 13 12.4 ...
##  $ installment   : num  810 277 416 1095 673 ...
##  $ grade         : Factor w/ 7 levels "A","B","C","D",..: 4 3 4 3 3 3 1 2 4 1 ...
##  $ emp_length    : num  5 3 1 5 14 13 4 19 5 11 ...
##  $ home_ownership: Factor w/ 3 levels "MORTGAGE","OWN",..: 1 3 3 2 1 1 1 3 1 2 ...
##  $ annual_inc    : num  80000 55000 40000 295000 110000 75000 100000 80000 66000 47000 ...
##  $ loan_status   : Factor w/ 2 levels "Default","Fully Paid": 1 1 1 2 2 1 2 2 2 2 ...
##  $ purpose       : Factor w/ 14 levels "car","credit_card",..: 3 2 3 3 3 2 3 3 3 2 ...
##  $ zip_code      : int  448 917 940 112 761 488 226 467 298 275 ...
##  $ dti           : num  39.75 6.15 10.38 10.05 10.54 ...
##  $ delinq_2yrs   : int  5 0 0 1 1 0 0 1 0 0 ...
##  $ inq_last_6mths: int  0 2 0 1 0 1 0 0 1 0 ...
##  $ open_acc      : int  22 7 5 16 18 13 18 11 10 5 ...
##  $ pub_rec       : int  1 0 0 0 0 0 0 2 1 0 ...
##  $ revol_util    : num  9.2 74.5 40.1 84.2 36.6 79.6 54.3 56.1 65 36.8 ...
##  $ total_acc     : int  41 17 8 28 31 42 30 22 24 14 ...
##  $ acc_now_delinq: int  0 0 0 0 0 0 0 0 0 0 ...
```

```
library(randomForest)
rf.lendingclub <- randomForest(loan_status~.,data=dataLC , subset=train , mtry=4, importance =TRUE , type =
'classification')

prediction.ontest.rf = predict(rf.lendingclub ,newdata=dataLC[-train ,],type="prob")
glm.rf.pred = ifelse (prediction.ontest.rf[,'Default'] > 0.5,"Default","Fully Paid")
groundtrue <- dataLC$loan_status[-train]
confusionMatrix(table(glm.rf.pred, groundtrue))
```

```
## Confusion Matrix and Statistics
##
##             groundtrue
## glm.rf.pred  Default Fully Paid
##   Default         35         38
##   Fully Paid     476       1544
##
##                Accuracy : 0.7544
##                  95% CI : (0.7354, 0.7727)
##     No Information Rate : 0.7559
##     P-Value [Acc > NIR] : 0.5723
##
##                   Kappa : 0.0627
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.06849
##             Specificity : 0.97598
##          Pos Pred Value : 0.47945
##          Neg Pred Value : 0.76436
##              Prevalence : 0.24415
##          Detection Rate : 0.01672
##    Detection Prevalence : 0.03488
##       Balanced Accuracy : 0.52224
##
##        'Positive' Class : Default
##
```

```
importance (rf.lendingclub)
```

```
##                    Default Fully Paid MeanDecreaseAccuracy MeanDecreaseGini
## loan_amnt      -7.5666449 12.9669829           10.5710806      26.67666553
## term            3.4756293  5.2013452            6.8579335       7.48826762
## int_rate       -0.2672047 12.1449289           12.0985910      37.95662528
## installment    -8.7560284 11.2009549            7.2727817      30.66736030
## grade           1.4404757 14.4748079           15.2003184      20.04006917
## emp_length     -2.3042806  0.4961473           -0.7070731      23.16585575
## home_ownership  0.2549154  0.7866883            0.8773050       7.19368943
## annual_inc     -0.9226895  5.7458487            4.5391167      33.61257675
## purpose         0.2039252 -1.6183931           -1.2292539      17.42790303
## zip_code       -0.1122950 -0.5878187           -0.6061707      33.89521670
## dti             1.7423690  4.0073501            4.3704690      36.67306646
## delinq_2yrs     1.4432859 -0.2280519            0.6747129       6.66617313
## inq_last_6mths  2.9202250  5.7954545            6.7348675      12.20814748
## open_acc       -5.1992043  4.3015443            1.0689381      22.30496376
## pub_rec         1.0953272  0.1666887            0.6835296       5.20496560
## revol_util     -3.8725730  8.7586964            5.7157540      35.83745620
## total_acc      -1.5543929  6.6167348            5.3017297      30.15291866
## acc_now_delinq  0.0000000  0.0000000            0.0000000       0.02019239
```

High sentivity random forest model

```
glm.rf.pred = ifelse (prediction.ontest.rf[,'Default']>0.2,"Default","Fully Paid")

confusionMatrix(table(glm.rf.pred, groundtrue))
```

```
## Confusion Matrix and Statistics
##
##             groundtrue
## glm.rf.pred  Default Fully Paid
##   Default        415        871
##   Fully Paid      96        711
##
##                Accuracy : 0.538
##                  95% CI : (0.5163, 0.5595)
##     No Information Rate : 0.7559
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1728
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8121
##             Specificity : 0.4494
##          Pos Pred Value : 0.3227
##          Neg Pred Value : 0.8810
##              Prevalence : 0.2441
##          Detection Rate : 0.1983
##    Detection Prevalence : 0.6144
##       Balanced Accuracy : 0.6308
##
##        'Positive' Class : Default
##
```

Conclusion:

It is possible to use modern machine learning models to predict loan default. Random forest are slightly more effective than logistic regression . Altering the probability threeshold from 0.5 to 0.2 increased the detection of defaulted loans from 5503 to 35764 in the case of random forest. It is possible to do this because predicting fully paid loans as defaulted is less risky than predicting defaulted loans as paid.

In order to increase the prediction power of the models 2 extension can be made. Joining zip code with census data and conducting TF-IDF on loan description text to identify relevant keywords The work of Shunpo Chang and others should be used as a guide http://cs229.stanford.edu/proj2015/199_report.pdf