Mid Project Report

Renuka Ramachandran

13 November 2017

Loading required libraries

```
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.3
```

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

```
library(readr)
library(DT)
```

```
## Warning: package 'DT' was built under R version 3.3.3
```

```
library(rgdal)
```

```
## Warning: package 'rgdal' was built under R version 3.3.3
```

```
## Loading required package: sp
```

```
## Warning: package 'sp' was built under R version 3.3.3
```

```
## rgdal: version: 1.2-15, (SVN revision 691)
```

```
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.0, released 2017/04/28
##  Path to GDAL shared files: C:/Users/Renuka/Documents/R/win-library/3.3/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Users/Renuka/Documents/R/win-library/3.3/rgdal/proj
##  Linking to sp version: 1.2-5
```

```
library(choroplethrMaps)
```

```
## Warning: package 'choroplethrMaps' was built under R version 3.3.3
```

```
library(choroplethr)
```

```
## Warning: package 'choroplethr' was built under R version 3.3.3
```

```
## Loading required package: acs
```

```
## Warning: package 'acs' was built under R version 3.3.3
```

```
## Loading required package: XML
```

```
## Warning: package 'XML' was built under R version 3.3.3
```

```
##
## Attaching package: 'acs'
```

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

```
## The following object is masked from 'package:base':
##
##     apply
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.3.3
```

```
library(rpart)
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
```

```
##     alpha
```

Data Preparation

Load the data

```
loan_data <- read_csv("C:/Users/Renuka/Desktop/coursework/AdvStats/Project/LoanStats3a.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   loan_amnt = col_integer(),
##   funded_amnt = col_integer(),
##   funded_amnt_inv = col_double(),
##   installment = col_double(),
##   annual_inc = col_double(),
##   dti = col_double(),
##   delinq_2yrs = col_integer(),
##   inq_last_6mths = col_integer(),
##   mths_since_last_delinq = col_integer(),
##   mths_since_last_record = col_integer(),
##   open_acc = col_integer(),
##   pub_rec = col_integer(),
##   revol_bal = col_integer(),
##   total_acc = col_integer(),
##   out_prncp = col_integer(),
##   out_prncp_inv = col_integer(),
##   total_pymnt = col_double(),
##   total_pymnt_inv = col_double(),
##   total_rec_prncp = col_double(),
##   total_rec_int = col_double()
##   # ... with 14 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
dim(loan_data)
```

```
## [1] 42535   145
```

Removing columns with no data

```
## [1] 42535    63
```

Removing columns which consist of only one value and other columns with repetitive data

```
loan_data <- loan_data %>%
             select(-c(funded_amnt, funded_amnt_inv, grade, sub_grade, emp_title,
                       verification_status, pymnt_plan,title, zip_code, earliest_cr_line, revo
l_bal,
                       initial_list_status, out_prncp,out_prncp_inv, recoveries,
                       collection_recovery_fee, last_pymnt_amnt, total_pymnt_inv, application_
type,
                       collections_12_mths_ex_med, policy_code, delinq_amnt, hardship_flag,
```

```
                        collections_12_mths_ex_med, policy_code, delinq_amnt, hardship_flag,
                        acc_now_delinq, chargeoff_within_12_mths, debt_settlement_flag, disburs
ement_method, tax_liens))

dim(loan_data)
```

```
## [1] 42535    35
```

Percentage of Missing values in each column

```
percent_missing <- round(colMeans(is.na(loan_data))*100,2)
percent_missing
```

```
##                 loan_amnt                      term
##                      0.00                      0.00
##                  int_rate                installment
##                      0.00                      0.00
##                emp_length            home_ownership
##                      0.00                      0.00
##                annual_inc                    issue_d
##                      0.01                      0.00
##               loan_status                      desc
##                      0.00                     31.78
##                   purpose                 addr_state
##                      0.00                      0.00
##                       dti                delinq_2yrs
##                      0.00                      0.07
##            inq_last_6mths    mths_since_last_delinq
##                      0.07                     63.30
##    mths_since_last_record                  open_acc
##                     91.42                      0.07
##                   pub_rec                 revol_util
##                      0.07                      0.21
##                 total_acc                total_pymnt
##                      0.07                      0.00
##           total_rec_prncp             total_rec_int
##                      0.00                      0.00
##        total_rec_late_fee              last_pymnt_d
##                      0.00                      0.20
##              next_pymnt_d        last_credit_pull_d
##                     93.54                      0.01
##      pub_rec_bankruptcies debt_settlement_flag_date
##                      3.21                     99.64
##         settlement_status           settlement_date
##                     99.64                     99.64
##         settlement_amount     settlement_percentage

##                     99.64                     99.64
##           settlement_term
##                     99.64
```

Remove columns with missing values greater than 50%

```
loan_data <- loan_data %>%
             select(-c(debt_settlement_flag_date, settlement_status,settlement_date,settlement_
amount,
                       settlement_percentage, settlement_term, next_pymnt_d, mths_since_last_delin
```

```
q,desc,
                    mths_since_last_record))
dim(loan_data)
```

```
## [1] 42535    25
```

Number of rows with missing values

```
nrows <- sum(!complete.cases(loan_data))
nrows
```

```
## [1] 1501
```

Removing rows with missing data ~3% of the data

```
loan_data <- loan_data[complete.cases(loan_data), ]
dim(loan_data)
```

```
## [1] 41034    25
```

Structure of the data

```
str(loan_data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    41034 obs. of  25 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            : chr  "10.65%" "15.27%" "15.96%" "13.49%" ...
##  $ installment         : num  162.9 59.8 84.3 339.3 67.8 ...
##  $ 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 ...
##  $ issue_d             : chr  "Dec-11" "Dec-11" "Dec-11" "Dec-11" ...
##  $ loan_status         : chr  "Fully Paid" "Charged Off" "Fully Paid" "Fully Paid" ...
##  $ purpose             : chr  "credit_card" "car" "small_business" "other" ...
##  $ 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_util          : chr  "83.70%" "9.40%" "98.50%" "21%" ...
##  $ total_acc           : int  9 4 10 37 38 12 11 4 13 3 ...
##  $ total_pymnt         : num  5863 1015 3006 12232 4067 ...
##  $ total_rec_prncp     : num  5000 456 2400 10000 3000 ...
##  $ total_rec_int       : num  863 435 606 2215 1067 ...
##  $ total_rec_late_fee  : num  0 0 0 17 0 ...
##  $ last_pymnt_d        : chr  "Jan-15" "Apr-13" "Jun-14" "Jan-15" ...
##  $ last_credit_pull_d  : chr  "Oct-17" "Oct-16" "Jun-17" "Apr-16" ...
##  $ pub_rec_bankruptcies: int  0 0 0 0 0 0 0 0 0 0 ...
```

Changing data type of columns

```
loan_data$int_rate <- as.numeric(gsub("%","",loan_data$int_rate))
loan_data$revol_util <- as.numeric(gsub("%","",loan_data$revol_util))
```

Exploring Loan Status

```
table(loan_data$loan_status)
```

```
##
##                                         Charged Off
##                                                5468
## Does not meet the credit policy. Status:Charged Off
##                                                 534
##  Does not meet the credit policy. Status:Fully Paid
##                                                1529
##                                          Fully Paid
##                                               33503
```

Removing loans which do not meet credit policy to avoid ambiguity - removed all the records that did not meet credit thresholds since these loans were not endorsed by Lending Club, and so are less important.

```
loan_data <- filter(loan_data, !grepl('Does not meet the credit policy.',loan_status))
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
table(loan_data$loan_status)
```

```
##
## Charged Off  Fully Paid
##        5468       33503
```

Exploratory Data Analysis

Exploring categorical variables

Frequency statistics of the different categorical variables.

```
# Checking column types to see if they are categorical or contiunous
all_vars <- unlist(lapply(loan_data,class))

# Extracting categorical variables
remove <- c("issue_d","addr_state","last_pymnt_d","last_credit_pull_d")
loan_data_ref <- loan_data[,!(names(loan_data) %in% remove)]
cat_vars <- unlist(lapply(loan_data_ref,class))
cat_info <- lapply(1:sum(cat_vars == "character"), function(inx) {
  Category <- loan_data[,names(cat_vars[cat_vars == "character"])[inx]]

  # Getting frequency counts and sorting in decreasing order
  counts_df <- data.frame(table(Category)) %>% arrange(desc(Freq))
  counts_df$Category <- as.character(counts_df$Category)
  dfs <- data.frame(Name = names(cat_vars[cat_vars == "character"])[inx],
```

```
                    counts_df, stringsAsFactors = F)
  dfs$`Freq %` <- round(100*dfs$Freq/sum(dfs$Freq))
  dfs
}) %>% bind_rows()

# Formatting into interactive HTML table
datatable(cat_info)
```

Show 10 entries

Search:

	Name	Category	Freq	Freq %
1	term	36 months	28301	73
2	term	60 months	10670	27
3	emp_length	10+ years	8770	23
4	emp_length	< 1 year	4404	11
5	emp_length	2 years	4299	11
6	emp_length	3 years	4026	10
7	emp_length	4 years	3385	9
8	emp_length	5 years	3239	8
9	emp_length	1 year	3140	8
10	emp_length	6 years	2195	6

Showing 1 to 10 of 34 entries

Previous 1 2 3 4 Next

Inferences: 1. Most of the borrowers almost 92% are do not own homes, they are under rent or mortgage. 2. Higher number of loans are issued for a 30 month term than a 60 month term. 3. The number of 'Charged Off' loans are way lower when compared to 'Fully Paid' loans, thus we will need to oversample the data in order to balance the dataset.

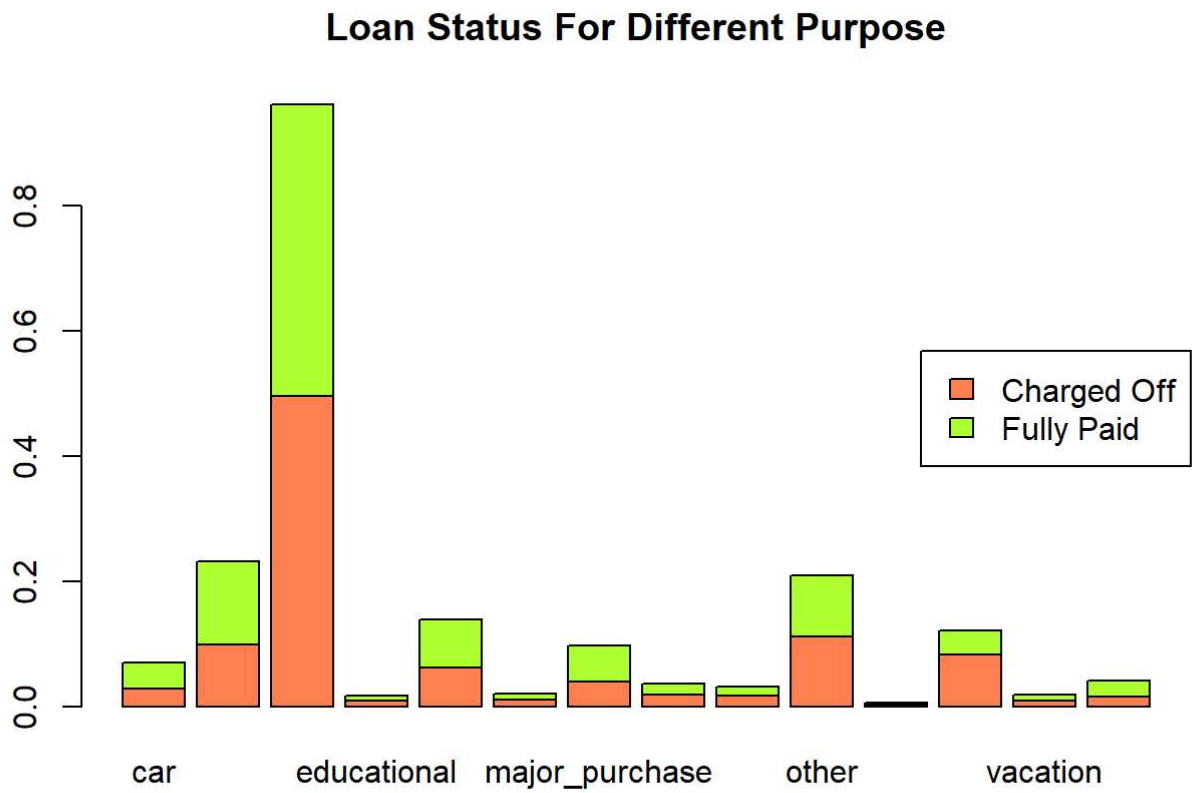
Relationship between Loan Status and Purpose of Loan

```
tab <- table(loan_data$loan_status,loan_data$purpose)
# Uses column margin (proportion); margin=1 uses row margins (proportion).
tab1<-prop.table(tab, margin=1)
tab1
```

```
##
##                      car credit_card debt_consolidation educational
##   Charged Off 0.028529627 0.098024872        0.495245062 0.009144111
##   Fully Paid  0.041070949 0.134256634        0.466734322 0.007491866
##
##               home_improvement       house major_purchase     medical
##   Charged Off      0.061448427 0.010241405    0.038771031 0.019019751
##   Fully Paid       0.077396054 0.009521535    0.058084351 0.017311883
##
```

```
##
##                    moving       other renewable_energy small_business
##   Charged Off 0.016276518 0.111741039      0.003291880    0.082662765
##   Fully Paid  0.014416619 0.097424111      0.002507238    0.039280064
##
##                   vacation     wedding
##   Charged Off 0.009692758 0.015910753
##   Fully Paid  0.009581232 0.024923141
```

```
barplot(tab1, col=c("coral", "greenyellow"), main="Loan Status For Different Purpose")
legend("right",
    legend = c("Charged Off","Fully Paid"),
    fill = c("coral", "greenyellow"))
```



Inference: - From the charts above it can be seen that majority of the loans are taken to tackle 'debt_consolidation' of which almost half of the borrowers have defaulted.

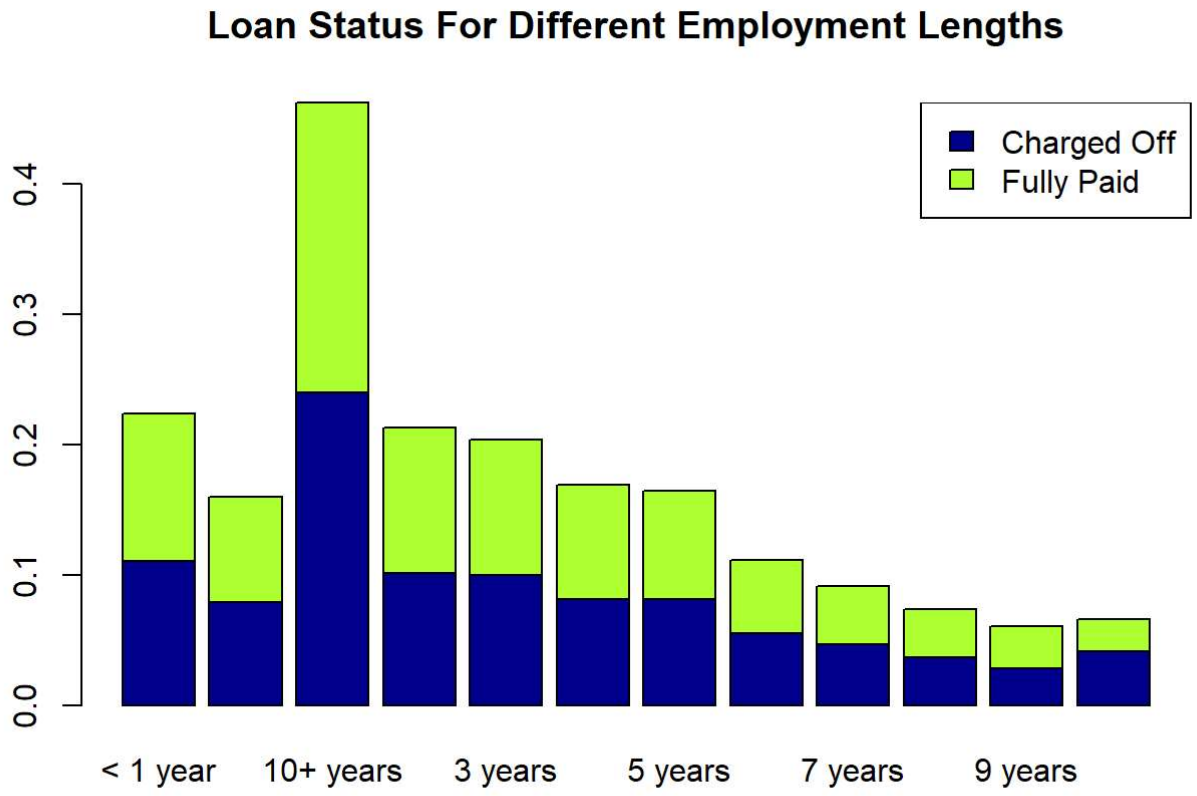
Relationship between Loan Status and Employment Duration of the borrower

```
tab <- table(loan_data$loan_status,loan_data$emp_length)
# Uses column margin (proportion); margin=1 uses row margins (proportion).
tab1<-prop.table(tab, margin=1)
tab1
```

```
##
##                < 1 year     1 year  10+ years    2 years    3 years
##   Charged Off 0.11027798 0.07937089 0.23957571 0.10113387 0.09967081
##   Fully Paid  0.11345253 0.08076889 0.22266663 0.11181088 0.10390114
##
##                 4 years    5 years    6 years    7 years    8 years
##   Charged Off 0.08138250 0.08119071 0.05486467 0.04700972 0.03630256
```

```
##   Charged Off 0.08138259 0.08119971 0.05486467 0.04700073 0.03639356
##   Fully Paid  0.08775334 0.08342536 0.05656210 0.04453333 0.03748918
##
##                  9 years        n/a
##   Charged Off 0.02816386 0.04096562
##   Fully Paid  0.03238516 0.02525147
```

```
barplot(tab1, col=c("darkblue", "greenyellow"), main="Loan Status For Different Employment Lengths")
legend("topright",
    legend = c("Charged Off","Fully Paid"),
    fill = c("darkblue", "greenyellow"))
```



Inferences 1. Maximum number of borrowers have an employment length of more than 10+ years 2. It can be noticed that there are quite a number of borrowers having < 1 year employment duration

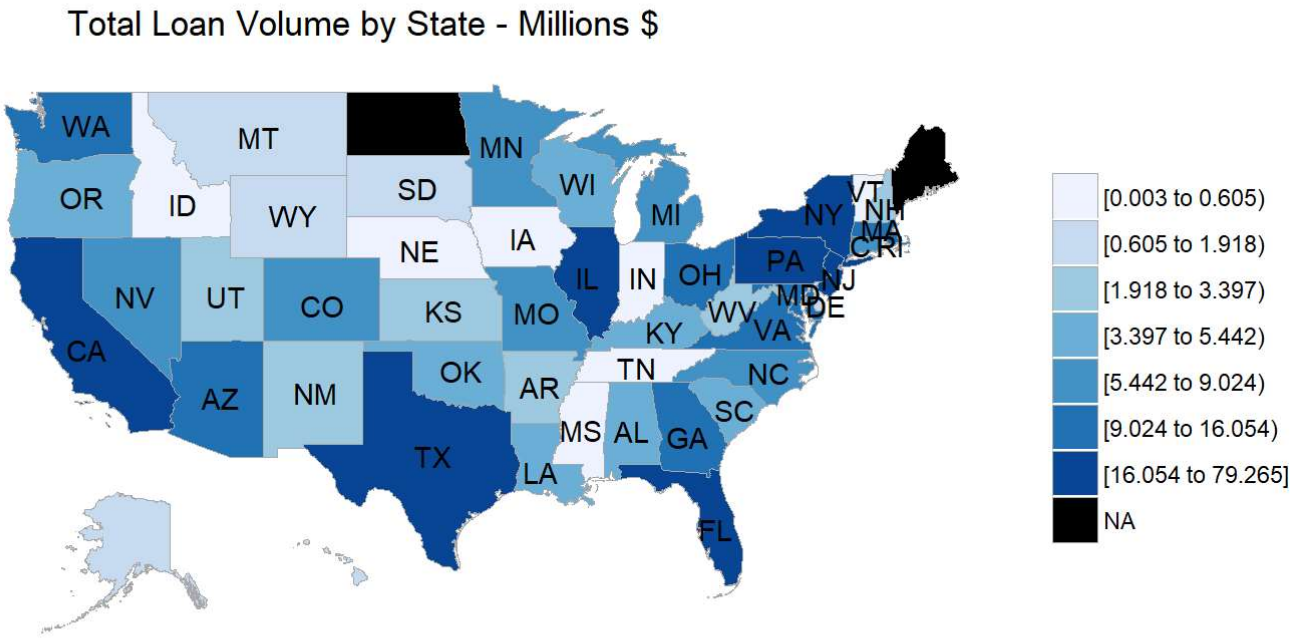
Total Loan Amount Volumes by State - to understand in which states the borrowers live in:

```
# Aggregating up by state
loan_by_state <- loan_data %>%
                 group_by(addr_state) %>%
                 summarize(`Total Loans ($)` = sum(loan_amnt)/1e6) %>%
                 arrange(desc(`Total Loans ($)`))
colnames(loan_by_state) <- c("region","value")
# Getting summary percentage of top 4 regions
top4_states <- round(100*sum(loan_by_state$value[1:4])/sum(loan_by_state$value),1)

# Replacing out the state codes with their full names for plotting
data("state.regions")
loan_by_state$region <- sapply(loan_by_state$region, function(state_code) {
    inx <- grep(pattern = state_code, x = state.regions$abb)
    state.regions$region[inx]
```

```
})
# Plotting US map with values
state_choropleth(loan_by_state, title = "          Total Loan Volume by State - Millions $")
```

```
## Warning in self$bind(): The following regions were missing and are being
## set to NA: north dakota, maine
```



Inference: - More number of borrowers live in the states of California, Texas, Florida, Pennsylvania, New York and Illinois. This can be attributed to the fact that the population in these states are high compared to the rest of the country.

Exploring numerical variables

Summary Statistics of numerical variables

```
# Enters zero NAs for summary when there are none so the summary data structures can be combine
d
custom_summary <- function(var) {
    res <- summary(var)
    return(res)
}
# Extracting continuous variables
cont_info <- lapply(loan_data[,all_vars == "numeric"], custom_summary)

# Formatting summaries into uniform data structure and combining

cont_names <- names(all_vars[all_vars == "numeric"])
cont_info <- lapply(1:length(cont_info), function(inx) {
  new_vect <- c(cont_names[inx],round(cont_info[[inx]],2))
  names(new_vect)[1] <- "Var Name"
```

```
  new_vect
});
cont_info <- do.call(rbind,cont_info)


# Formatting into interactive HTML table
datatable(cont_info)
```

Show 10 entries

Search:

Var Name	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
int_rate	5.42	9.25	11.86	12.04	14.61	24.59
installment	16.08	167.4	280.9	325.1	431.4	1305
annual_inc	4000	40800	59390	69040	82500	6e+06
dti	0	8.25	13.46	13.37	18.63	29.99
revol_util	0	25.6	49.5	48.98	72.5	99.9
total_pymnt	33.97	5605	9992	12300	16690	58890
total_rec_prncp	0	4727	8000	9904	14000	35000
total_rec_int	6.22	668.4	1360	2296	2875	23890
total_rec_late_fee	0	0	0	1.35	0	180.2

Showing 1 to 9 of 9 entries

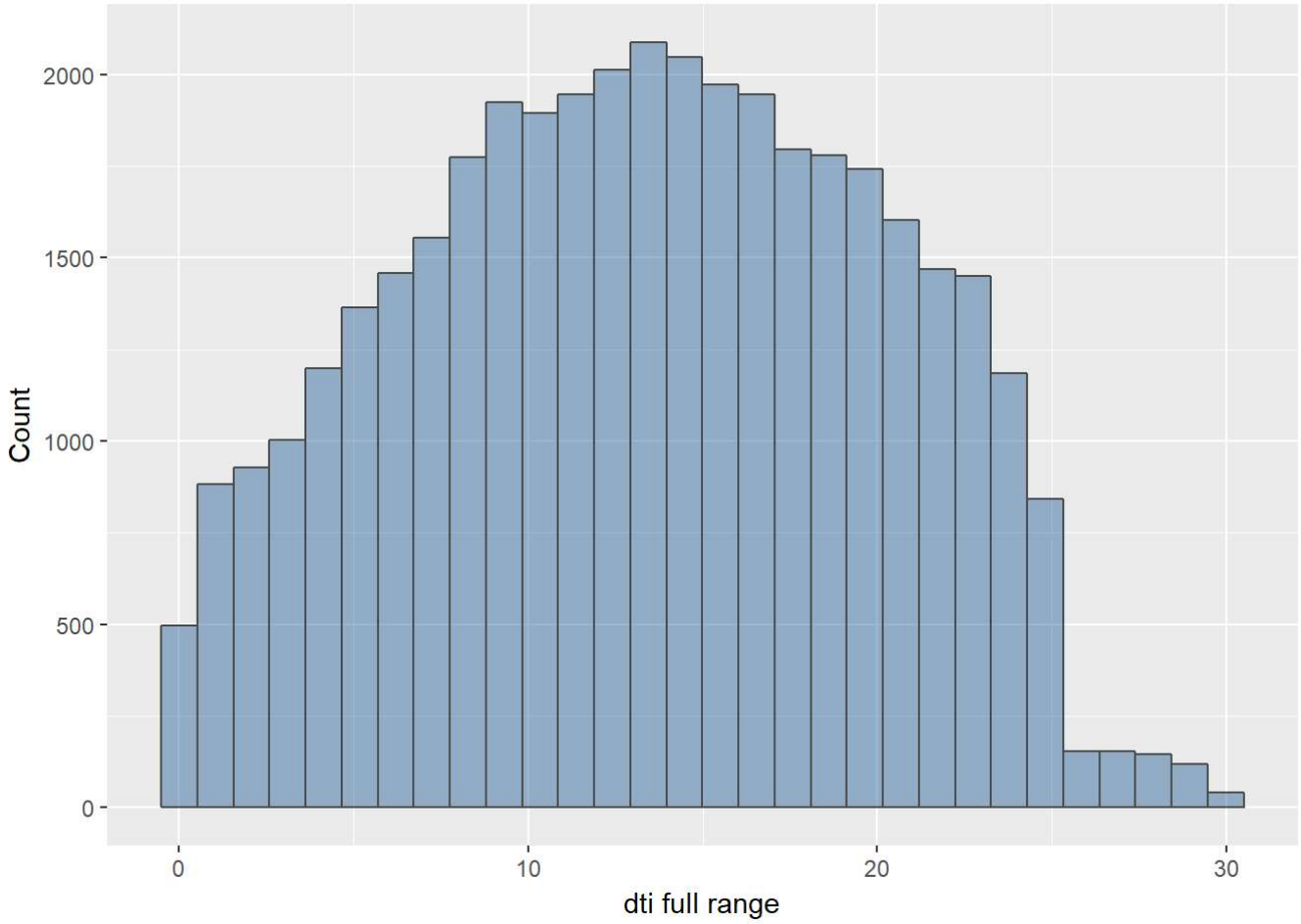
Previous 1 Next

Observation: - The total_rec_late_fee has value 0 most of the cases and will not be useful in the analysis

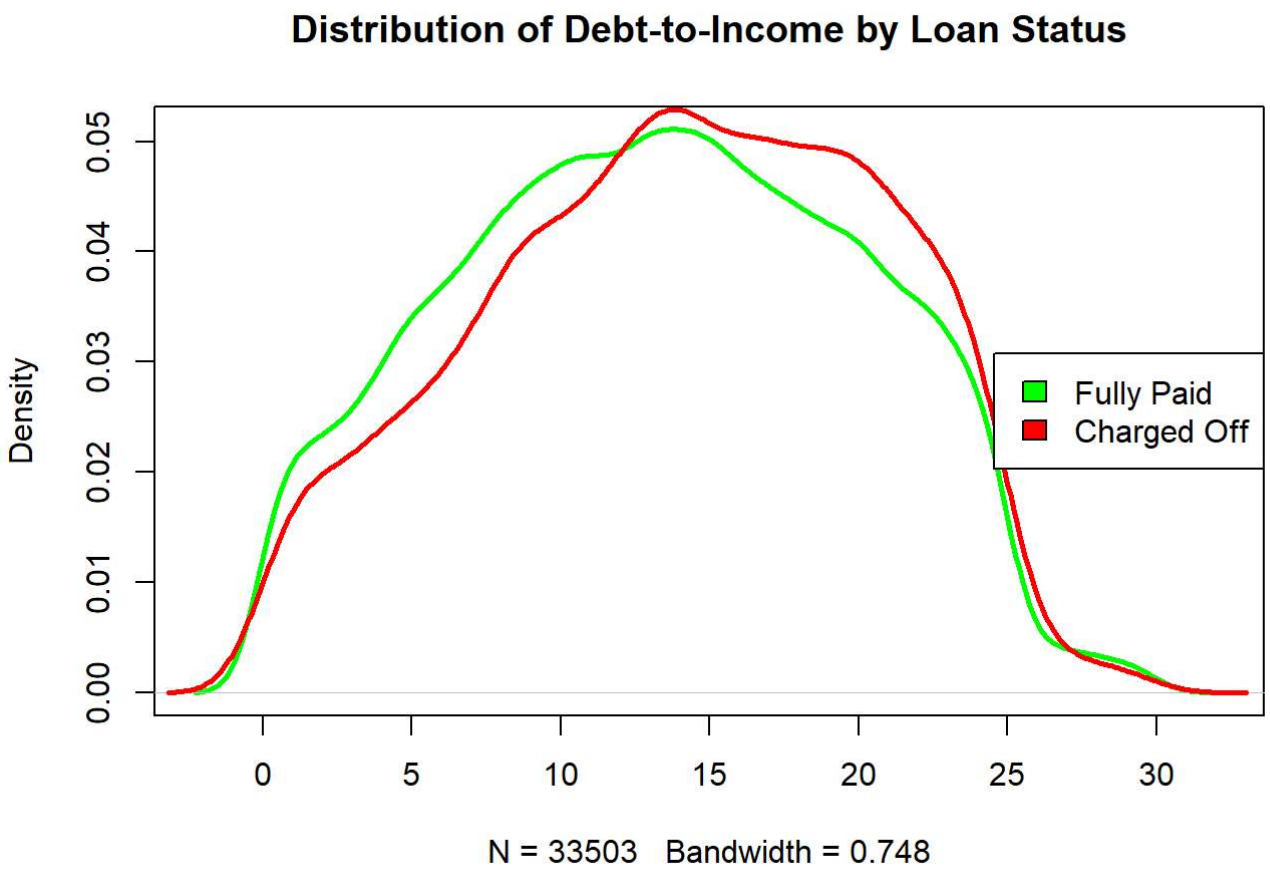
Debt to Income Ratio Distribution

```
dti_raw <- loan_data$dti
dens1 <- qplot(dti_raw, fill = I("dodgerblue4"),
               alpha = I(0.4), col = I("grey29")) + xlab("dti full range") + ylab("Count")
dens1
```

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



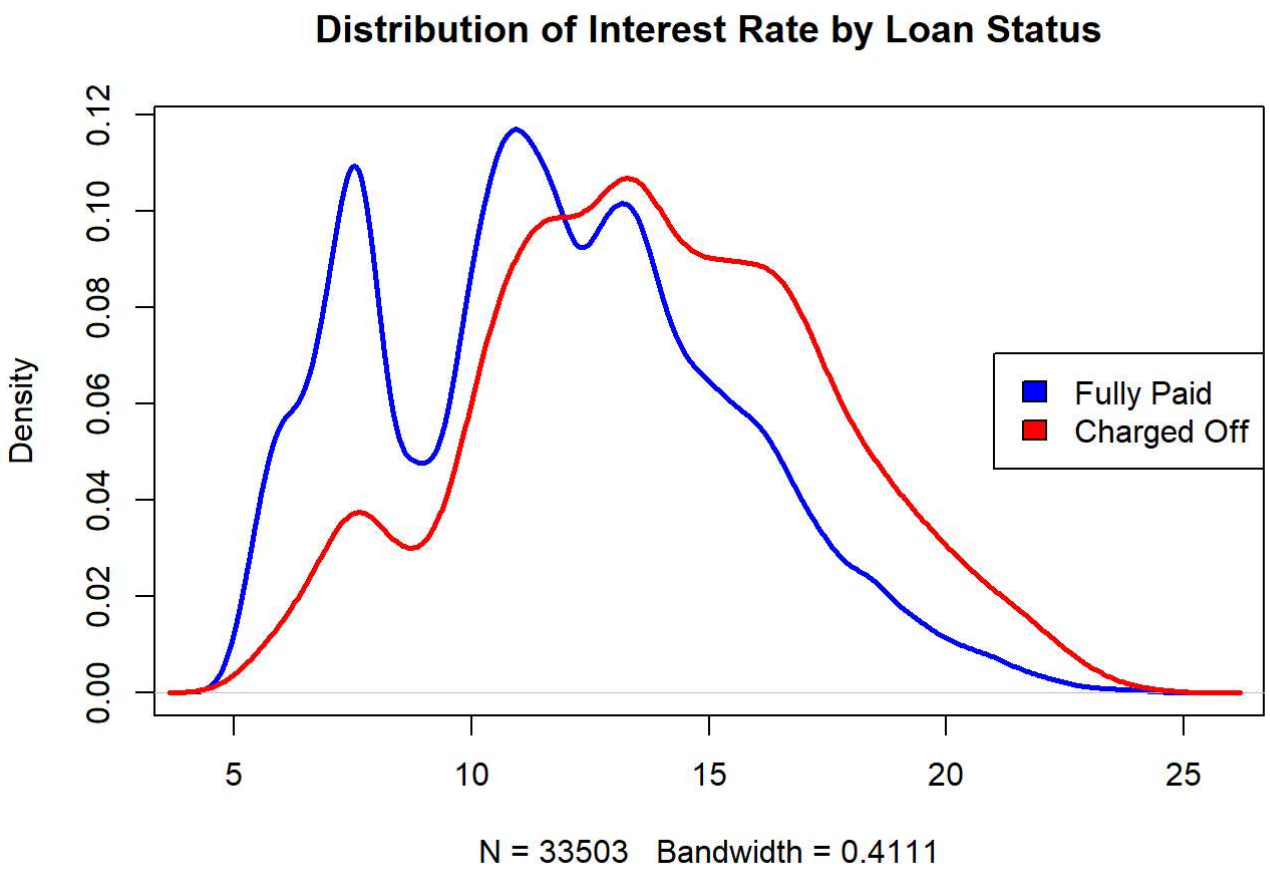

```
p <-loan_data[loan_data$loan_status=="Fully Paid",]
d <-loan_data[loan_data$loan_status=="Charged Off",]
plot(density(p$dti), col="green", lwd=2.5, main="Distribution of Debt-to-Income by Loan Status"
)
lines(density(d$dti), col="red", lwd=2.5)
legend("right", legend = c("Fully Paid", "Charged Off"), fill = c("green","red"))
```



Inference - Debt to Income ratio is almost normally distributed - The distribution of debt_to_income ratio is very similar for 'Fully Paid' and 'Charged Off' loans

Interest Rate Distribution by Loan Status

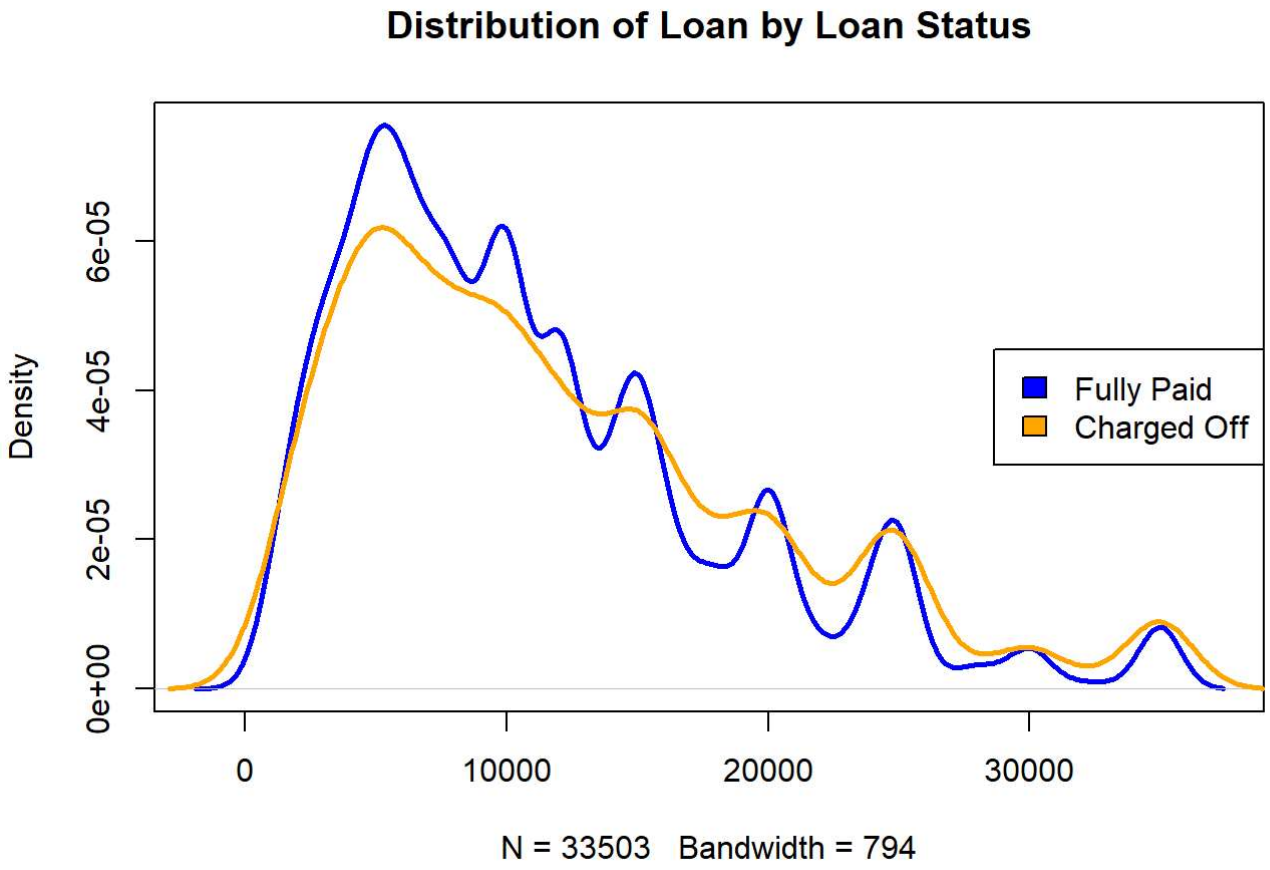
```
p <-loan_data[loan_data$loan_status=="Fully Paid",]
d <-loan_data[loan_data$loan_status=="Charged Off",]
plot(density(p$int_rate), col="blue", lwd=2.5, main="Distribution of Interest Rate by Loan Stat
us")
lines(density(d$int_rate), col="red", lwd=2.5)
legend("right", legend = c("Fully Paid", "Charged Off"), fill = c("blue","red"))
```



Inference: Interest Rate is right skewed in the case of 'Fully Paid' loans whereas it is more spread for the Charged Off loans. This means that more number of 'Fully Paid' have lower interest rate compared to 'Charged Off' loans.

Loan Amount Distribution by Loan Status

```
p <-loan_data[loan_data$loan_status=="Fully Paid",]
d <-loan_data[loan_data$loan_status=="Charged Off",]
plot(density(p$loan_amnt), col="blue", lwd=2.5, main="Distribution of Loan by Loan Status")
lines(density(d$loan_amnt), col="orange", lwd=2.5)
legend("right", legend = c("Fully Paid", "Charged Off"), fill = c("blue","orange"))
```



Inference: - It can be observed that higher number of loans have been issued for smaller loan amounts, thus the distribution is skewed. Also, the distribution dont vary a lot between 'Fully Paid' and 'Charged Off' loans

Additional Metrics

```
#Creating additional metric from the data that can affect the loan status
#openacc_ratio = total number of open credit lines/total number of credit lines
#This can assess the financial strength of the borrower, as in how much debt borrower has.
loan_data$openacc_ratio <- (loan_data$open_acc/loan_data$total_acc)*100

loan_data <- loan_data %>%
             select(loan_status,loan_amnt,int_rate,installment,emp_length,home_ownership,
                    annual_inc,purpose,dti,openacc_ratio)
```

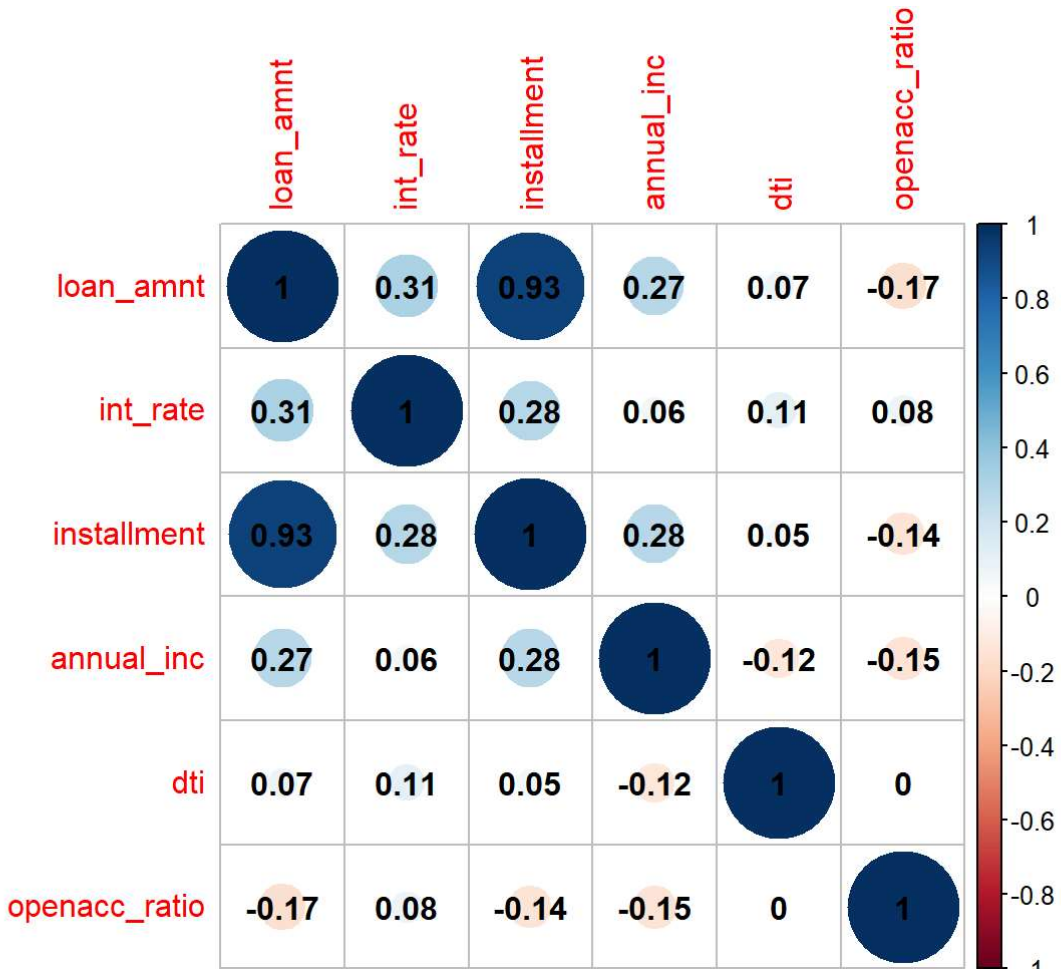
Overall Correlation between Numeric Variables

```
loan_data_num <- loan_data %>%
                  select(loan_amnt,int_rate,installment,annual_inc,dti,openacc_ratio)
cormat <- cor(loan_data_num)
round(cormat, 2) # Rounded to 2 decimals
```

```
##              loan_amnt int_rate installment annual_inc   dti
## loan_amnt         1.00     0.31        0.93       0.27  0.07
## int_rate          0.31     1.00        0.28       0.06  0.11
## installment       0.93     0.28        1.00       0.28  0.05
## annual_inc        0.27     0.06        0.28       1.00 -0.12
## dti               0.07     0.11        0.05      -0.12  1.00
## openacc_ratio    -0.17     0.08       -0.14      -0.15  0.00
```

```
##              openacc_ratio
## loan_amnt            -0.17
## int_rate              0.08
## installment          -0.14
## annual_inc           -0.15
## dti                   0.00
## openacc_ratio         1.00
```

```
corrplot(cormat, method="circle", addCoef.col="black")
```



Observation: - From the correlation matrix we can identify that loan_amnt and installment has very high correlation, this is natural. Hence we need not consider installment separately as a predictor when we build the model since its effect is already captured by 'loan_amnt'.

Feature Engineering

Merging categories of the factors - keeping top five buckets as such and combining the rest into one category 'Others'

```
table(loan_data$emp_length)
```

```
##
##  < 1 year    1 year 10+ years   2 years   3 years   4 years   5 years
##      4404      3140      8770      4299      4026      3385      3239
##   6 years   7 years   8 years   9 years       n/a
##      2195      1749      1455      1239      1070
```

```
#Re-bucketing emp_length variable as:
#10+ years, < 1 year , 2-5 years, 6-9 years based on frequency shown below
loan_data_mod <- loan_data
loan_data_mod <- loan_data[,-4]
loan_data_mod$emp_length <- gsub("[2-5] years","2-5 years",loan_data_mod$emp_length)
loan_data_mod$emp_length <- gsub("[6-9] years","6-9 years",loan_data_mod$emp_length)
loan_data_mod <- loan_data_mod%>%
                  filter(emp_length!='n/a')
```

```
#Re-bucketing Purpose variable based on frequency as
#debt_consolidation,credit_card,home_improvement,major_purchase,small_business,other
table(loan_data_mod$purpose)
```

```
##
##                 car        credit_card debt_consolidation
##                1480               4905              17947
##        educational   home_improvement              house
##                 294               2833                362
##     major_purchase            medical             moving
##                2086                659                548
##              other   renewable_energy     small_business
##               3715                 94               1724
##           vacation            wedding
##                345                909
```

```
loan_data_mod$purpose <- gsub("car","other",gsub("educational","other",gsub("house","other",gsu
b("medical","other",gsub("moving","other",gsub("renewable_energy","other",gsub("vacation","othe
r",gsub("wedding","other",loan_data_mod$purpose)))))))
```

Model Building

Decision Tree

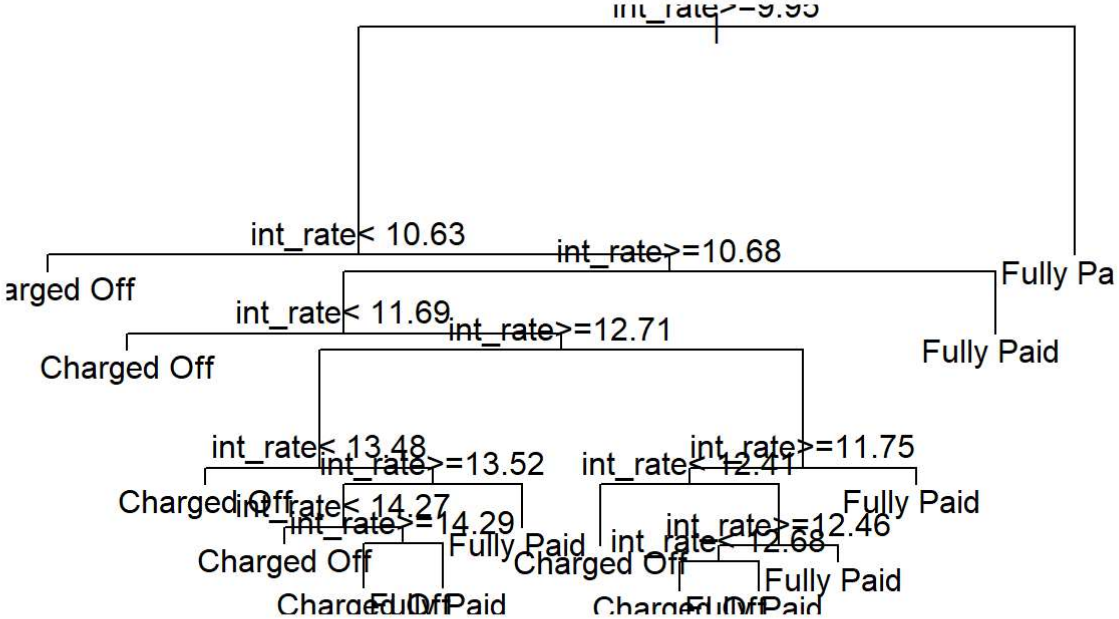
Implemented decision trees first because it implicitly perform variable screening or feature selection for the given data, this wil help me identify important variables in the initial stages itself. Also, it handles categorical variables as such and it is easily interpretable.

```
#Creating train and test dataset
#Balancing dataset to have enough Good and Bad loan observations

loan_data_1 <- loan_data_mod %>%
                  filter(loan_status=="Charged Off")
loan_data_0 <- loan_data_mod %>%
                  filter(loan_status=="Fully Paid")
loan_data_bal <- rbind(loan_data_1,loan_data_0[c(1:5250),])

set.seed(1)
train_rows <- sample(nrow(loan_data_bal), 0.7*nrow(loan_data_bal))
train_dt <- loan_data_bal[train_rows,]
test_dt <- loan_data_bal[-c(train_rows),]
#Creating Decision tree
loan_dtree <- rpart(loan_status ~ ., data = train_dt, control=rpart.control(minsplit=10, minbuc
ket=3))
plot(loan_dtree)
```

```
text(loan_dtree, pretty = 0, cex = 1)
```




```
summary(loan_dtree)
```

```
## Call:
## rpart(formula = loan_status ~ ., data = train_dt, control = rpart.control(minsplit = 10,
##     minbucket = 3))
##   n= 7345
##
##           CP nsplit rel error    xerror       xstd
## 1 0.19762955      0 1.0000000 1.0000000 0.01181046
## 2 0.02687431      1 0.8023705 0.8023705 0.01155462
## 3 0.02646086      5 0.6463616 0.6918412 0.01120397
## 4 0.01846748      7 0.5934399 0.5934399 0.01075291
## 5 0.01254135     11 0.5184675 0.5201213 0.01032142
## 6 0.01000000     13 0.4933848 0.4950386 0.01015307
##
## Variable importance
## int_rate
##       99
##
## Node number 1: 7345 observations,    complexity param=0.1976295
##   predicted class=Fully Paid   expected loss=0.4939415  P(node) =1
##     class counts:  3628  3717
##    probabilities: 0.494 0.506
##   left son=2 (5647 obs) right son=3 (1698 obs)
##   Primary splits:
##       int_rate      < 9.95     to the right, improve=236.27400, (0 missing)
##       purpose       splits as  RLLLLL,       improve= 50.87433, (0 missing)
```

```
##       annual_inc    < 43975    to the left,  improve= 38.76598, (0 missing)
##       openacc_ratio < 73.20513 to the right, improve= 10.95672, (0 missing)

##       loan_amnt     < 5137.5   to the left,  improve= 10.21707, (0 missing)
##   Surrogate splits:
##       dti < 26.65    to the left,  agree=0.771, adj=0.008, (0 split)
##
## Node number 2: 5647 observations,    complexity param=0.02687431
##   predicted class=Charged Off  expected loss=0.436515  P(node) =0.7688223
##     class counts:  3182  2465
##    probabilities: 0.563 0.437
##   left son=4 (204 obs) right son=5 (5443 obs)
##   Primary splits:
##       int_rate   < 10.635   to the left,  improve=80.65622, (0 missing)
##       purpose    splits as  RRLLLL,       improve=50.26144, (0 missing)
##       annual_inc < 74551.5  to the left,  improve=24.90038, (0 missing)
##       loan_amnt  < 9962.5   to the left,  improve=14.31823, (0 missing)
##       dti        < 1.945    to the left,  improve=13.35641, (0 missing)
##
## Node number 3: 1698 observations
##   predicted class=Fully Paid   expected loss=0.262662  P(node) =0.2311777
##     class counts:   446  1252
##    probabilities: 0.263 0.737
##
## Node number 4: 204 observations
##   predicted class=Charged Off  expected loss=0  P(node) =0.027774
##     class counts:   204     0
##    probabilities: 1.000 0.000
##
## Node number 5: 5443 observations,    complexity param=0.02687431
##   predicted class=Charged Off  expected loss=0.4528753  P(node) =0.7410483
##     class counts:  2978  2465
##    probabilities: 0.547 0.453
##   left son=10 (5192 obs) right son=11 (251 obs)
##   Primary splits:
##       int_rate   < 10.68    to the right, improve=99.844740, (0 missing)
##       purpose    splits as  RRRLLL,       improve=46.434070, (0 missing)
##       annual_inc < 74551.5  to the left,  improve=21.661110, (0 missing)
##       dti        < 1.945    to the left,  improve=12.109730, (0 missing)
##       loan_amnt  < 9962.5   to the left,  improve= 9.923462, (0 missing)
##
## Node number 10: 5192 observations,    complexity param=0.02687431
##   predicted class=Charged Off  expected loss=0.4318182  P(node) =0.7068754
##     class counts:  2950  2242
##    probabilities: 0.568 0.432
##   left son=20 (309 obs) right son=21 (4883 obs)
##   Primary splits:
##       int_rate   < 11.685   to the left,  improve=122.52880, (0 missing)
##       purpose    splits as  RRRLLL,       improve= 46.42492, (0 missing)
##       annual_inc < 74551.5  to the left,  improve= 24.01024, (0 missing)
##       loan_amnt  < 9962.5   to the left,  improve= 14.26781, (0 missing)
##       dti        < 1.945    to the left,  improve= 10.35267, (0 missing)
##
## Node number 11: 251 observations
##   predicted class=Fully Paid   expected loss=0.1115538  P(node) =0.03417291
##     class counts:    28   223
##    probabilities: 0.112 0.888
##
## Node number 20: 309 observations
```

```
## Node number 20: 309 observations
##   predicted class=Charged Off  expected loss=0  P(node) =0.04206943
##     class counts:   309     0
##    probabilities: 1.000 0.000
##
## Node number 21: 4883 observations,    complexity param=0.02687431
##   predicted class=Charged Off  expected loss=0.459144  P(node) =0.664806
##     class counts:  2641  2242
##    probabilities: 0.541 0.459
##   left son=42 (3660 obs) right son=43 (1223 obs)
##   Primary splits:
##       int_rate   < 12.71    to the right, improve=120.967600, (0 missing)
##       purpose    splits as  RRRLLL,       improve= 39.754300, (0 missing)
##       annual_inc < 74551.5  to the left,  improve= 22.189660, (0 missing)
##       loan_amnt  < 9962.5   to the left,  improve=  9.160823, (0 missing)
##       dti        < 1.945    to the left,  improve=  9.038330, (0 missing)
##   Surrogate splits:
##       dti < 27.515   to the left,  agree=0.751, adj=0.004, (0 split)
##
## Node number 42: 3660 observations,    complexity param=0.01846748
##   predicted class=Charged Off  expected loss=0.3948087  P(node) =0.4982982
##     class counts:  2215  1445
##    probabilities: 0.605 0.395
##   left son=84 (247 obs) right son=85 (3413 obs)
##   Primary splits:
##       int_rate   < 13.485   to the left,  improve=82.574370, (0 missing)
##       purpose    splits as  RRLLLL,       improve=35.975890, (0 missing)
##       annual_inc < 39360    to the left,  improve=17.490100, (0 missing)
##       loan_amnt  < 8562.5   to the left,  improve=12.009810, (0 missing)
##       dti        < 3.22     to the left,  improve= 6.402455, (0 missing)
##
## Node number 43: 1223 observations,    complexity param=0.02646086
##   predicted class=Fully Paid   expected loss=0.3483238  P(node) =0.1665078
##     class counts:   426   797
##    probabilities: 0.348 0.652
##   left son=86 (861 obs) right son=87 (362 obs)
##   Primary splits:
##       int_rate      < 11.745   to the right, improve=38.556420, (0 missing)
##       annual_inc    < 70549.5  to the left,  improve=10.121280, (0 missing)
##       purpose       splits as  RRRRRL,       improve= 7.862599, (0 missing)
##       dti           < 22.535   to the right, improve= 5.181319, (0 missing)
##       openacc_ratio < 89.44444 to the right, improve= 4.971822, (0 missing)
##
## Node number 84: 247 observations
##   predicted class=Charged Off  expected loss=0  P(node) =0.03362832
##     class counts:   247     0
##    probabilities: 1.000 0.000
##
## Node number 85: 3413 observations,    complexity param=0.01846748
##   predicted class=Charged Off  expected loss=0.4233812  P(node) =0.4646698
##     class counts:  1968  1445
##    probabilities: 0.577 0.423
##   left son=170 (3109 obs) right son=171 (304 obs)
##   Primary splits:
##       int_rate   < 13.52    to the right, improve=58.880520, (0 missing)
##       purpose    splits as  RLLLLL,       improve=33.161390, (0 missing)
##       annual_inc < 79208    to the left,  improve=12.927690, (0 missing)
##       loan_amnt  < 8562.5   to the left,  improve= 6.164905, (0 missing)
##       dti        < 3.22     to the left,  improve= 5.352938, (0 missing)
```

```
##       dti        < 9.22     to the left,  improve=3.992990, (0 missing)
##   Surrogate splits:
##       dti < 27.92    to the left,  agree=0.911, adj=0.003, (0 split)
##
## Node number 86: 861 observations,    complexity param=0.02646086
##   predicted class=Fully Paid   expected loss=0.4297329  P(node) =0.1172226
##     class counts:   370   491
##    probabilities: 0.430 0.570
##   left son=172 (194 obs) right son=173 (667 obs)
##   Primary splits:
##       int_rate   < 12.415   to the left,  improve=159.948000, (0 missing)
##       annual_inc < 70549.5  to the left,  improve=  7.659287, (0 missing)
##       purpose    splits as  RRRRRL,       improve=  6.003782, (0 missing)
##       loan_amnt  < 10037.5  to the left,  improve=  4.702444, (0 missing)
##       dti        < 4.92     to the left,  improve=  4.392018, (0 missing)
##   Surrogate splits:
##       annual_inc    < 230400   to the right, agree=0.777, adj=0.010, (0 split)
##       openacc_ratio < 88.19444 to the right, agree=0.776, adj=0.005, (0 split)
##
## Node number 87: 362 observations
##   predicted class=Fully Paid   expected loss=0.1546961  P(node) =0.04928523
##     class counts:    56   306
##    probabilities: 0.155 0.845
##
## Node number 170: 3109 observations,    complexity param=0.01846748
##   predicted class=Charged Off  expected loss=0.394339  P(node) =0.4232811
##     class counts:  1883  1226
##    probabilities: 0.606 0.394
##   left son=340 (223 obs) right son=341 (2886 obs)
##   Primary splits:
##       int_rate   < 14.265   to the left,  improve=74.713440, (0 missing)
##       purpose    splits as  RRRLLL,       improve=32.616960, (0 missing)
##       annual_inc < 78462.03 to the left,  improve=15.436940, (0 missing)
##       loan_amnt  < 5125     to the left,  improve= 9.171906, (0 missing)
##       dti        < 3.22     to the left,  improve= 5.901157, (0 missing)
##
## Node number 171: 304 observations
##   predicted class=Fully Paid   expected loss=0.2796053  P(node) =0.0413887
##     class counts:    85   219
##    probabilities: 0.280 0.720
##
## Node number 172: 194 observations
##   predicted class=Charged Off  expected loss=0.005154639  P(node) =0.02641253
##     class counts:   193     1
##    probabilities: 0.995 0.005
##
## Node number 173: 667 observations,    complexity param=0.01254135
##   predicted class=Fully Paid   expected loss=0.2653673  P(node) =0.09081007
##     class counts:   177   490
##    probabilities: 0.265 0.735
##   left son=346 (401 obs) right son=347 (266 obs)
##   Primary splits:
##       int_rate      < 12.455   to the right, improve=14.108810, (0 missing)
##       dti           < 22.53    to the right, improve= 4.874771, (0 missing)
##       annual_inc    < 70549.5  to the left,  improve= 3.521784, (0 missing)
##       purpose       splits as  RRRRRL,       improve= 2.393054, (0 missing)
##       openacc_ratio < 52.08696 to the right, improve= 1.516481, (0 missing)
##   Surrogate splits:
##       loan_amnt  < 29850    to the left,  agree=0.613, adj=0.030, (0 split)
```

```
##       loan_amnt   < 15000    to the left,  agree=0.615, adj=0.026, (0 split)
##       annual_inc < 147500   to the left,  agree=0.603, adj=0.004, (0 split)
##
## Node number 340: 223 observations
##   predicted class=Charged Off  expected loss=0  P(node) =0.03036079
##     class counts:   223     0
##    probabilities: 1.000 0.000
##
## Node number 341: 2886 observations,    complexity param=0.01846748
##   predicted class=Charged Off  expected loss=0.4248094  P(node) =0.3929204
##     class counts:  1660  1226
##    probabilities: 0.575 0.425
##   left son=682 (2640 obs) right son=683 (246 obs)
##   Primary splits:
##       int_rate   < 14.285   to the right, improve=68.041270, (0 missing)
##       purpose    splits as  RRRLLL,       improve=30.490650, (0 missing)
##       annual_inc < 79500    to the left,  improve=11.924040, (0 missing)
##       dti        < 3.22     to the left,  improve= 5.520844, (0 missing)
##       loan_amnt  < 5125     to the left,  improve= 5.410629, (0 missing)
##   Surrogate splits:
##       dti < 29.095   to the left,  agree=0.916, adj=0.012, (0 split)
##
## Node number 346: 401 observations,    complexity param=0.01254135
##   predicted class=Fully Paid   expected loss=0.3491272  P(node) =0.05459496
##     class counts:   140   261
##    probabilities: 0.349 0.651
##   left son=692 (91 obs) right son=693 (310 obs)
##   Primary splits:
##       int_rate      < 12.685   to the left,  improve=99.734710, (0 missing)
##       dti           < 18.4     to the right, improve= 3.770883, (0 missing)
##       annual_inc    < 39300    to the left,  improve= 1.971438, (0 missing)
##       openacc_ratio < 47.72257 to the right, improve= 1.929327, (0 missing)
##       loan_amnt     < 2225     to the left,  improve= 1.357071, (0 missing)
##   Surrogate splits:
##       dti            < 1.15     to the left,  agree=0.778, adj=0.022, (0 split)
##       home_ownership splits as  RLRR,         agree=0.776, adj=0.011, (0 split)
##
## Node number 347: 266 observations
##   predicted class=Fully Paid   expected loss=0.1390977  P(node) =0.03621511
##     class counts:    37   229
##    probabilities: 0.139 0.861
##
## Node number 682: 2640 observations
##   predicted class=Charged Off  expected loss=0.3916667  P(node) =0.3594282
##     class counts:  1606  1034
##    probabilities: 0.608 0.392
##
## Node number 683: 246 observations
##   predicted class=Fully Paid   expected loss=0.2195122  P(node) =0.03349217
##     class counts:    54   192
##    probabilities: 0.220 0.780
##
## Node number 692: 91 observations
##   predicted class=Charged Off  expected loss=0  P(node) =0.01238938
##     class counts:    91     0
##    probabilities: 1.000 0.000
##
## Node number 693: 310 observations
##   predicted class=Fully Paid   expected loss=0.1580645  P(node) =0.04220558
```

```
##     class counts:    49   261
##    probabilities: 0.158 0.842
```

Model Evaluation

```
predictions_dt <- (predict(loan_dtree, test_dt, type = "class"))
confusionMatrix(predictions_dt, test_dt$loan_status)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Charged Off Fully Paid
##   Charged Off        1275        426
##   Fully Paid          341       1107
##
##                Accuracy : 0.7564
##                  95% CI : (0.741, 0.7713)
##     No Information Rate : 0.5132
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5118
##  Mcnemar's Test P-Value : 0.002421
##
##             Sensitivity : 0.7890

##             Specificity : 0.7221
##          Pos Pred Value : 0.7496
##          Neg Pred Value : 0.7645
##              Prevalence : 0.5132
##          Detection Rate : 0.4049
##    Detection Prevalence : 0.5402
##       Balanced Accuracy : 0.7555
##
##        'Positive' Class : Charged Off
##
```

Inferences:

1. Interest_rate seems to be the most important variable which is used throught various levels of the tree, this can be seen in the tree diagram.
2. Other important variables as indicated in the summary include purpose, annual_inc, openacc_ratio and loan_amnt.
3. The minsplit=10, minbucket=3 was set because that gave the best accuracy. Tuning of these parameters was done on a trial and error basis.
4. The reusulting accuracy of the model is 75.6%

Logistic Regression

On using logistic regression, I will be further able to understand the impact the various predictors on the dependent variables. Apart from knowing just the important variables, I can now understand whether there is a positive or negative reationship between the predictors and the dependent variable. To build this modeld On-Hot encoding is carried out on the categorical variables.

```
#Categorizing loan_status to be 1 as Charged Off/Default and 0 as Fully Paid/Non-Default
loan_data_mod$loan_status_cat <- ifelse(loan_data_mod$loan_status=='Charged Off',1,0)

#One Hot-encoding
loan_data_final <- cbind(loan_data_mod, model.matrix(~emp_length-1,loan_data_mod), model.matrix
```

```
(~home_ownership-1,loan_data_mod), model.matrix(~purpose-1,loan_data_mod))

remove <- c("loan_status","emp_length","home_ownership","purpose")
loan_data_final <- loan_data_final[,!(names(loan_data_final) %in% remove)]

#Creating train and test dataset
#Balancing dataset to have enough Good and Bad loan observations

loan_data_1 <- loan_data_final %>%
                  filter(loan_status_cat==1)
loan_data_0 <- loan_data_final %>%
                  filter(loan_status_cat==0)
loan_data_bal <- rbind(loan_data_1, loan_data_0[c(1:5250),])
set.seed(1)
train_rows <- sample(nrow(loan_data_bal), 0.7*nrow(loan_data_bal))
train <- loan_data_bal[train_rows, ]
test <- loan_data_bal[-c(train_rows),]
```

Model

```
fit_model1 <- glm(loan_status_cat ~ ., data = train, family = binomial)
summary(fit_model1)
```

```
##
## Call:
## glm(formula = loan_status_cat ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0807  -1.0872  -0.6349   1.1121   3.3279
##
## Coefficients: (3 not defined because of singularities)
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -4.848e-01  1.739e-01  -2.787 0.005316 **
## loan_amnt                 -1.363e-05  3.629e-06  -3.756 0.000173 ***
## int_rate                   1.254e-01  6.834e-03  18.344  < 2e-16 ***
## annual_inc                -4.544e-06  6.940e-07  -6.547 5.88e-11 ***
## dti                       -1.413e-03  3.898e-03  -0.362 0.716979
## openacc_ratio              5.227e-04  1.399e-03   0.374 0.708726
## `emp_length< 1 year`       2.719e-01  9.460e-02   2.874 0.004049 **
## `emp_length1 year`         1.367e-01  1.038e-01   1.317 0.187806
## `emp_length10+ years`      1.923e-01  7.428e-02   2.588 0.009646 **
## `emp_length2-5 years`      1.542e-01  6.888e-02   2.239 0.025184 *
## `emp_length6-9 years`             NA         NA      NA       NA
## home_ownershipMORTGAGE     3.372e-02  5.534e-02   0.609 0.542356
## home_ownershipOTHER        1.345e+01  1.327e+02   0.101 0.919284
## home_ownershipOWN          2.210e-02  9.862e-02   0.224 0.822717
## home_ownershipRENT                NA         NA      NA       NA
## purposecredit_otherd      -1.479e+00  1.288e-01 -11.488  < 2e-16 ***
## purposedebt_consolidation -9.581e-01  1.139e-01  -8.412  < 2e-16 ***
## purposehome_improvement   -7.050e-01  1.469e-01  -4.798 1.60e-06 ***
## purposemajor_purchase     -7.002e-01  1.681e-01  -4.166 3.10e-05 ***
## purposeother              -6.049e-01  1.228e-01  -4.926 8.38e-07 ***
## purposesmall_business             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10181.3  on 7344  degrees of freedom
## Residual deviance:  9536.3  on 7327  degrees of freedom
## AIC: 9572.3
##
## Number of Fisher Scoring iterations: 12
```

Model Evaluation

```
p <- predict(fit_model1, test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
p_model <- ifelse(p > 0.5, "1", "0")
t <- table(p_model, test$loan_status_cat)
t
```

```
##
## p_model   0   1
##       0 980 624
##       1 553 992
```

```
accuracy <- (980+992)/sum(t)
accuracy
```

```
## [1] 0.6262305
```

```
error_rate <- 1-accuracy
```

Inferences: 1. From the summary it can be observed that the most significant variables are loan amount, interest rate, annual_income and purpose of loan. 2.The model gives a slightly lower accuracy of 62.6% when compared to decision tree method.

Support Vector Machines - SVM

Random Forest was also experimented with, but the results were very poor. Random forest is intrinsically suited for multiclass problems, while SVM is intrinsically two-class. SVM is the third model I chose to implement.

```
svm_model <- ksvm(loan_status ~ .,
                  data = train_dt,
                  kernel = "rbfdot",
                  kpar = list(sigma=0.003909534),
                  C = 0.1,
                  prob.model = TRUE,
                  scaled = FALSE)


summary(svm_model)
```

```
## Length  Class   Mode
```

```
##      1   ksvm     S4
```

Model Evaluation

```
predict_loan_status_svm <- predict(svm_model,test_dt,type="probabilities")
predict_loan_status_svm <- as.data.frame(predict_loan_status_svm)$"Fully Paid"
predict_loan_status_label <- ifelse(predict_loan_status_svm < 0.5,"Charged Off","Fully Paid")
c <- confusionMatrix(predict_loan_status_label,test_dt$loan_status,positive = "Fully Paid")
c
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Charged Off Fully Paid
##   Charged Off        1269       1187
##   Fully Paid          347        346
##
##                Accuracy : 0.5129
##                  95% CI : (0.4952, 0.5305)
##     No Information Rate : 0.5132
##     P-Value [Acc > NIR] : 0.5214
##
##                   Kappa : 0.0111
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.2257
##             Specificity : 0.7853
##          Pos Pred Value : 0.4993
##          Neg Pred Value : 0.5167
##              Prevalence : 0.4868
##          Detection Rate : 0.1099
##    Detection Prevalence : 0.2201
##       Balanced Accuracy : 0.5055
##
##        'Positive' Class : Fully Paid
##
```

Inferences: 1. Since the given dataset does not have a lot of variables, SVM is not a very apt method. It is more suited when working with dataset of high demiensionality. 2. SVM resulted in the least accuracy with 51.9%.

Model Comparison and Final Results

1. As per the analysis conducted the factors that we need to look out for to detect default loan cases are interest_rate, loan_amount, purpose and annual income
2. The best accuracy was obtained for decision trees model compared to logistic and SVM with 75% accuracy.