Loan Default

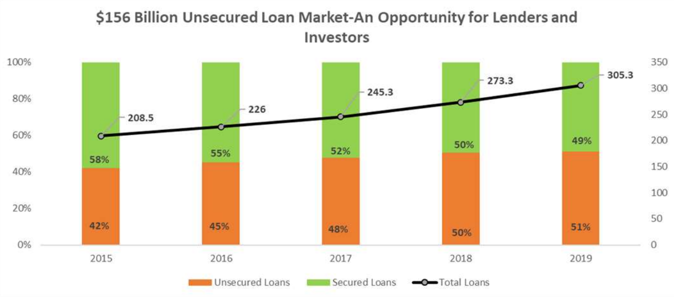
PREDICTION

ABHISHEK GARG | PGP_BABI | 01-March-2020

Introduction

OVERVIEW OF LENDING MARKET

Personal Loan Industry

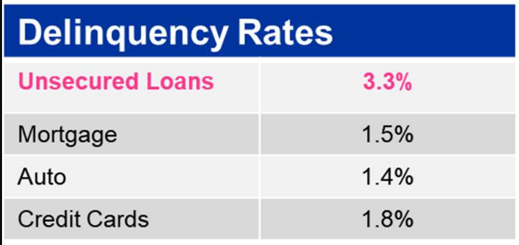


The practice of lending and borrowing is as old as the invention of money. Historically, large-scale lending operations, focused primarily on secured loans. However, with the advent of technology, and changes in risk appetite, unsecured loan has also seen acceleration in growth. **Personal loans (both secured and unsecured) continue to hold their place as the fastest-growing debt category in the U.S., with balances reaching $305 billion in the Q4CY19. That's an increase of 12% year over year—double the growth of credit card debt, the next-highest category, according to Experian data. Specifically, the unsecured personal loans market has been growing to capture a larger share of the pie, rising to a level of 51% of the overall personal loans market in 2019.**

KEY PROBLEM AREA

As its generally said "With great powers comes great responsibilities", similarly in lending marketplace, where a large amount of money is involved, there is a significant risk that lenders take. The most important task for any lender is to predict the probability of default for a borrower. An accurate prediction can help in balancing risk and return for the lender; charging higher rates for higher risks, or even denying the loan when required.

Default Rates in Secured & Unsecured Personal Loans



Delinquency Rates	
Unsecured Loans	3.3%
Mortgage	1.5%
Auto	1.4%
Credit Cards	1.8%

When talking in reference to unsecured personal loans market, the prediction of default becomes quite evident as the default rates in this area is higher than the secured personal loan category.

NEED OF THE STUDY/PROJECT

Loans default often cause huge loss for the banks/lenders. Hence, these lenders pay close attention on this issue and consistently in search of various method to detect and predict default behaviors of their customers. Machine learning algorithms have historically done a pretty good performance on this purpose, which are widely used by the banking. **As a**

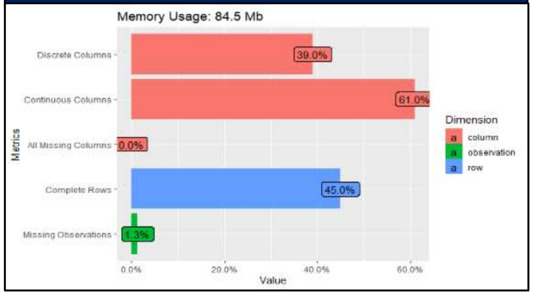
means to further explore this area, the primary purpose of this study is to analyze the loan dataset provided and build a robust machine learning model to effectively predict consumers who are likely to default.

A Overview of the Dataset

DATASET OVERVIEW

The dataset provided contains several variables with plenty of information about the borrowers. Overall, the information includes 226,786 observations across 41 variables (16 discrete such as term of loan, grade of borrower etc.& 25 continuous such as loan amount, funded amount among others). The data seems to be from a peer-to-peer lending provider as one of the variables refers to “Funded Amount by Investors”. Furthermore, information contained includes data from loans provided from 2008 to 2016 (February), as well as current status of the loan (Fully Paid or Default), which is our target variable.

Dataset Summary



Within the dataset, while none of the variables had 100% missing observations, there were nearly 124,802 observations that were missing across different variables (only 45% of the rows have values across all variables). This clearly indicates that the data missing value treatment might be necessary.

IS THE DATA SET GOOD ENOUGH TO ROBUST MODELLING?

Its important to be noted that the dataset could be inherently biased. The primary reason for this inference is that the information contained in the dataset is for the loans that were made (and hence already accepted). Ideally a true unbiased sample of loan applications would include both the applications that have been accepted, as well as rejected.

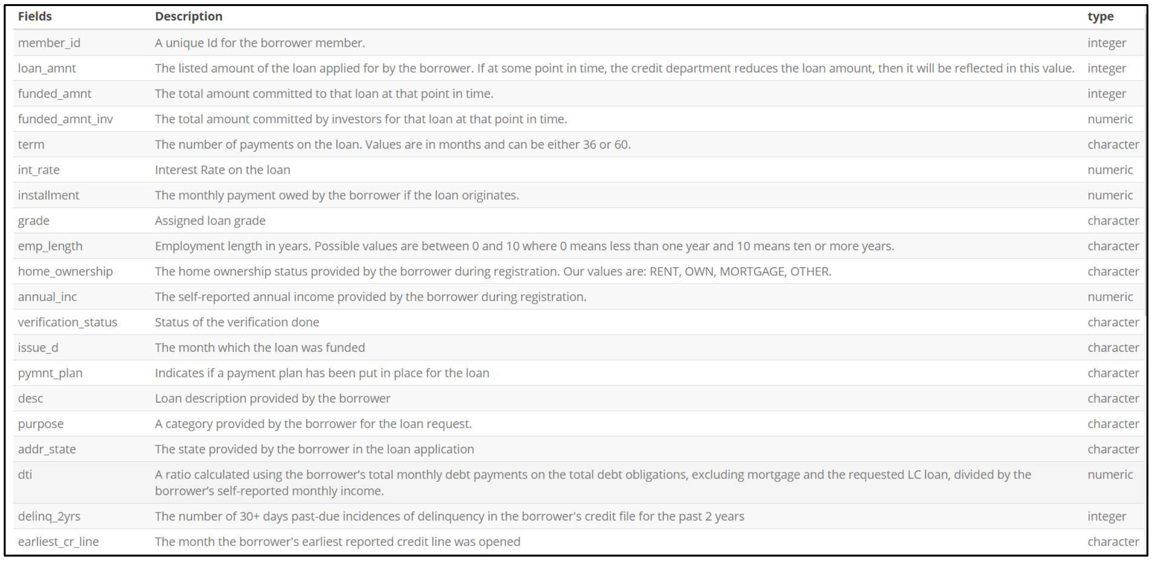
Obviously, since the dataset only include accepted loans, there might be fewer risky looking loan applications than safe looking ones. This will of course be verified as we move forward. However, for now its important to understand that the risky looking loan applications might have been approved after much stricter vetting process, that the safe looking loans might have bypassed. **This indicates that if the model is to be used downstream for approving/rejecting current applications, a variable like grade (which is a category assigned to each borrower after looking at multiple information sources) could actually might not be useful for the final model.**

However, for the purpose of current analysis I would not discard the variable Grade and take this as a note.

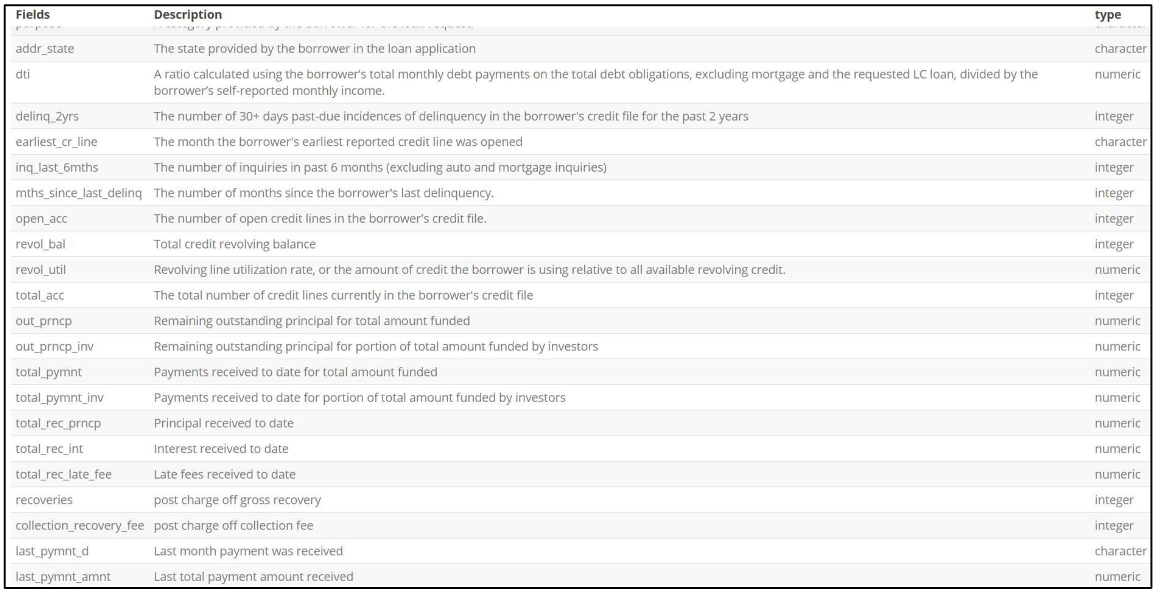
UNDERSTANDING VARIABLES

The 41 variables provided can largely be divided into 3 categories including 1) Loan Characteristics, 2) Consumer Profile, and 3) Payment & Delinquency Information.

Variable Information



Fields	Description	type
member_id	A unique Id for the borrower member.	integer
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.	integer
funded_amnt	The total amount committed to that loan at that point in time.	integer
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.	numeric
term	The number of payments on the loan. Values are in months and can be either 36 or 60.	character
int_rate	Interest Rate on the loan	numeric
installment	The monthly payment owed by the borrower if the loan originates.	numeric
grade	Assigned loan grade	character
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.	character
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.	character
annual_inc	The self-reported annual income provided by the borrower during registration.	numeric
verification_status	Status of the verification done	character
issue_d	The month which the loan was funded	character
pymnt_plan	Indicates if a payment plan has been put in place for the loan	character
desc	Loan description provided by the borrower	character
purpose	A category provided by the borrower for the loan request.	character
addr_state	The state provided by the borrower in the loan application	character
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.	numeric
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years	integer
earliest_cr_line	The month the borrower's earliest reported credit line was opened	character



Fields	Description	type
addr_state	The state provided by the borrower in the loan application	character
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.	numeric
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years	integer
earliest_cr_line	The month the borrower's earliest reported credit line was opened	character
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)	integer
mths_since_last_delinq	The number of months since the borrower's last delinquency.	integer
open_acc	The number of open credit lines in the borrower's credit file.	integer
revol_bal	Total credit revolving balance	integer
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.	numeric
total_acc	The total number of credit lines currently in the borrower's credit file	integer
out_prncp	Remaining outstanding principal for total amount funded	numeric
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors	numeric
total_pymnt	Payments received to date for total amount funded	numeric
total_pymnt_inv	Payments received to date for portion of total amount funded by investors	numeric
total_rec_prncp	Principal received to date	numeric
total_rec_int	Interest received to date	numeric
total_rec_late_fee	Late fees received to date	numeric
recoveries	post charge off gross recovery	integer
collection_recovery_fee	post charge off collection fee	integer
last_pymnt_d	Last month payment was received	character
last_pymnt_amnt	Last total payment amount received	numeric

While most of the variables seem self-explanatory, there are few which require special mention as they might be need some explanation to enhance understanding. These include:

- **Dti:** Dti refers to debt to income ratio, which compares total monthly debt obligations for the borrower to total monthly gross income. This could be an important variable for prediction of defaulting customers, as it shows overall indebtedness.

- **earliest_cr_line:** This describes the date when the first credit line was established. Usually, the longer one has held a credit line, the more desirable as a borrower. This feature might be useful for default prediction, specifically if converted to a measure of how long one has held a credit line.

- **Delinq_2yrs:** Indicates number of times the borrower has been behind on payments in the last 2 years.

- **Inq_last_6mths:** Refers to the number of times a borrower's credit report is accessed by financial institutions, which generally happens when the borrower is seeking a loan or credit line. More inquiries lead to higher rates of nonperformance, perhaps indicating that increased borrower desperation to access credit might highlight poor financial health.

- **revol_util:** Revolving utilization percent is the portion of a borrower's revolving credit limit (i.e. credit card limit) that they actually are using at any given point. For example, if a borrower's total credit limit is $15,000 and their outstanding balance is $1,500 their utilization rate would be 10%. Intuitively, the percentage of non-performing loans steadily increases with utilization rate. Borrowers with high utilization rates are more likely to have high fixed credit card payments which might affect their ability to repay their loans. Also, a high utilization rate often reflects a lack of other financing options, with borrowers turning to peer-to-peer lending as a last resort.

Initial Data Analysis & Treatment

CHECK DATA TYPES & CONVERSION, DATA TRASFORMATION, VARIBALE CREATION

The dataset was imported using data.table::fread(), which uses heuristics to guess the data types, which might not always be correct. This seem to be correct in our case as we can see that variables such as term, grade etc., have been imported as character variables. Furthermore, the attributes that should have been in date format were also imported as characters. To make the attributes useful for analysis, each of them were converted to correct format. Details of variables whose data types were transformed include:

- **Character to Date Format:** issue_d, earliest_cr_line, last_pymnt_d, next_pymnt_d & last_credit_pull_d.

- **Character to Date Format:** term, grade,emp_length,home_ownership,verification_status,pymnt_plan, purpose,addr_state,application_type & loan_status.

Factor Levels: The homeownership variable contained had levels including mortgage, rent and owned, which are self-explanatory. However, levels titled any, other and none, with 1, 36 & 114 observations respectively did not seem clear. They could indicate anything such as the borrower might be living at friend or relative place etc. seemed. To make the levels sensible, all the observation under the 3 levels would be combined into "other". Furthermore, the variable "emp_length" contained a level as "n/a". This was transformed to a level "Unknown".

Additionally, the variable purpose, emp_length, and addr-state have many levels, which might impact model performance because of high cardinality. To make the levels more sensible grouping on levels was undertaken after identification of some logic.

- **Purpose variable**: Upon looking at the data is was note that debt consolidation and credit card represent nearly 82% of the defaulters, and 78% of the total data. Hence, its good to have these categories for sure. Since all other categories represent a very small category of defaulters (individually), as well as total data, these were clubbed in other category, except for home related categories, which were clubbed as home.

- **Emp-length variable**: Upon looking at the data it was noted that 10+ years alone represented ~30% of the cases, and had a likelihood of default at 8.4%. Hence, this seemed to be an important level and was kept as is. On the other hand, the level unknown, although it showed the highest liklihood of default (12.2%), it was just present in 3.7% of the cases (overall) and represent ~5% of the default cases. But since no reasonable logic could be seen to club it with other level, this was also retained. Remining employment years showed a likelihood of default , as well as representation in single digits. Hence, they were clubbed into 0-4 Yrs and 5-9 yrs.

- **add_state variable:** For address state we used the k-means clustering technique to divide the states into 5 categories.

Missing Observations & Treatement: The variables mths_since_last_delinq (124638 NA's), revol_util (164 NA's), next_pymnt_d (207,723 NA's) and last_credit_pull_d (16 NA's) had missing observations. Each variable with missing values were treated based on the reasoning below:

- **mths_since_last_delinq:** If we think about this in more detail, it may be reasonable to assume that NA values for the variable mths_since_last_delinq actually indicates that there was no event/record of any missed payment so there cannot be any time value. However, if we impute zeros in place of NA, it might indicate that an event (delinquncy just happened last month) has just happened.

Hence, we imputed NA's in this variable by a very an unusual number say "-1" to indicate no missed payments. Furthermore, it would be important to document these assumptions carefully.

- **Revol_util & last_credit_pull_d:** Na's in revol_util are present in just 0.07% of the rows and in last_credit_pull_d are present in just 0.01% of observations. Hence removing these rows might not cause a serious issue.

- **next_pmnt_date:** next_pmnt_date.variable contained a large percentage of NA's and this might have been because the borrower has fully paid the loan. This was confirmed by looking at the distribution of defaulters vs non defaulters per this variable. Hence, logically they would not have a due payment date. So the best treatment was to remove the variable.

- **Last_pmnt_date:** Interestingly all the observations, where this variable was blank belonged to default category. So, if the rows with missing observations were removed, we might end up losing valuable information (given we are trying to predict default). After looking at the the summary of last_pymnt_d of defaulters, a large range was seen, hence no reasonable imputation seems possible. Hence this variable was also excluded.

Variable Creation: After removal of the variables "next_pymnt_d", as well as "last_pymnt_d", there were 3 variablels within our dataset which belong to the "Date" category viz: 1) issue-d (date on which loan was issued to the customer), 2) earliest_cr_line (The month the borrower's earliest reported credit line was opened) 3 last_cr_pull_d (The most recent month credit was pulled from the account). Based on domain understanding, it can be infered that people who have longer credit history are less likely to default. So given the issue_d and earliest_cr_line variables "Age of Credit Line" was derived, that is the number of days since the borrower has credit history. This was a difference of issue date and earliest credit line.

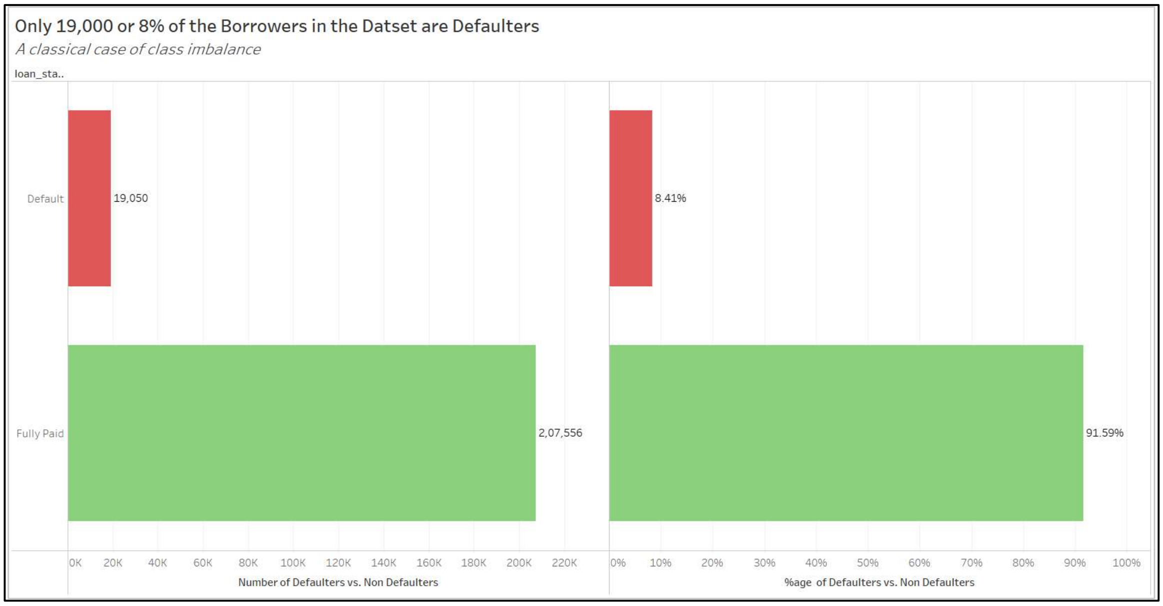
Number of Zeros:

The quantity of zeros, NA, Inf, unique values as well as the data type may lead to a good or bad model. The NA values for the dataset have aready been worked on, however there are quite a few variables which have high number of zero values **such as total_rec_late_fee(97.93%), out_prncp (91.65%), out_prncp_inv (91.65%), delinq_2yrs (83.48%) and inq_last_6mths (50.14%)**.Variables with lots of zeros may not be useful for modeling and, in some cases, they may dramatically bias the model. To determine whether these variables might be good predictors of default, their distribution as well as significance testing using wilcox test was implemented. Finally it was determined that the variable inq_last_6mths might not be a significant predictor, hence was removed. Additionally, recoveries & collection_recovery_fee, with 100% zero observations were removed.

Default Variable: The variable loan_status contained the status of the loan as "Fully Paid" and "Default" and is our target variable. The distribution shows that only 8% of total observations are defaulters. The variable has been converted to dummy variable.

This seems to be a good case of class imbalance, thus indicating that balancing the dataset would be required to make a good model.

Loan Status Variable-Target Variable



Only 19,000 or 8% of the Borrowers in the Datset are Defaulters

A classical case of class imbalance

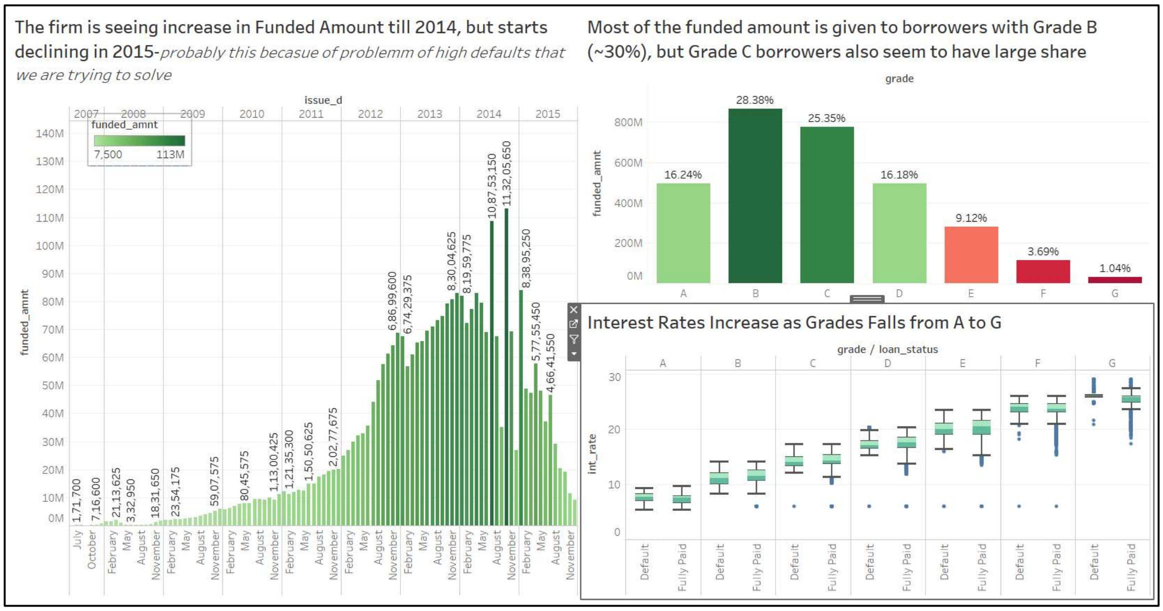
Variables Removed: There were certain variables that were removed as well as the reasoning mentioned: 1) id: Represent unique value to borrower. Might not be useful until we are trying to see top/bottom borrowers, 2) desc: textual information, 3) loan_status: converted to dummy variable, 4) last_pymnt_d: Large number of NA's, 5) pymnt_plan: Low variation (only 6 rows vs others) and 6) application type: Low variation (only 6 rows vs others). In addition to the above, other variables were also removed based on checking their significance in predicting default variable, and other inferences.

Initial Analysis of Loans and Other Variables

It can be seen clearly the firm has seen an increase in total amount funded till 2014, however it declines in 2015. Probably this could be because of higher defaults being faced by the firm, which results in lost investor confidence (hence lees number of investors willing to put their money through the firm).

Most of the amount funded is given out to Grade B borrowers , but Grade C also show a big share. Given that interest rates rise from Grade A to Grade G, it can be inferred that Grade A is the best grade and Grade G is the worst grade.

Basic Analysis of Loan and Other Variables



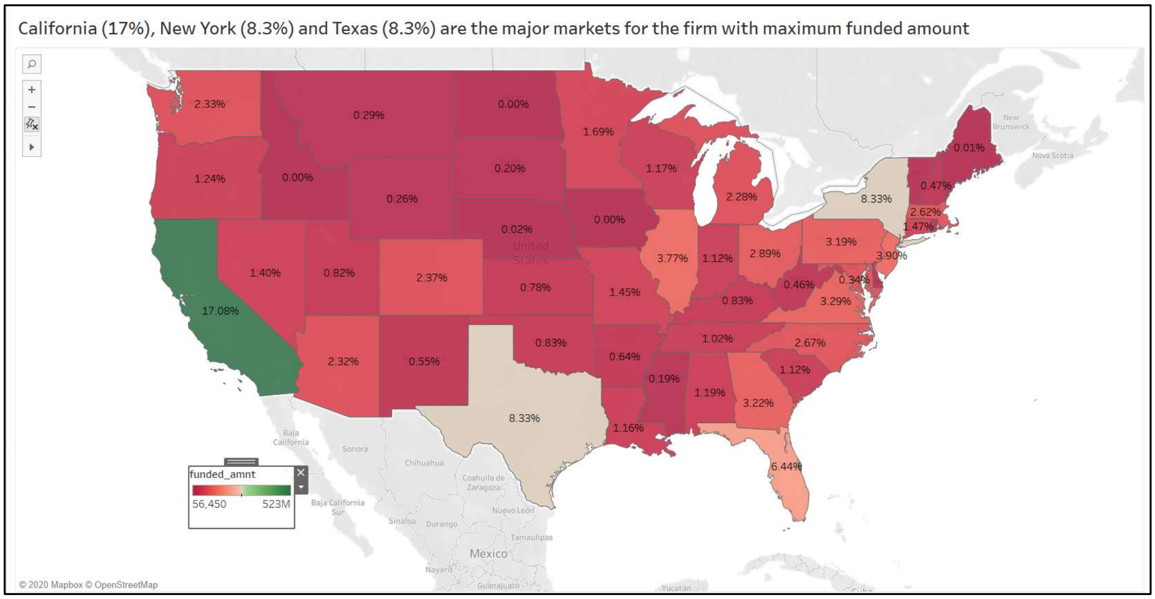
The firm is seeing increase in Funded Amount till 2014, but starts declining in 2015-*probably this becasue of problemm of high defaults that we are trying to solve*

Most of the funded amount is given to borrowers with Grade B (~30%), but Grade C borrowers also seem to have large share

Interest Rates Increase as Grades Falls from A to G

Additionally, it can also be seen that California (with ~17% of total amount funded), new York (with 8.3%) and Texas (with 8.3%) are the biggest markets for the firm.

Basic Analysis of Loan and Other Variables



California (17%), New York (8.3%) and Texas (8.3%) are the major markets for the firm with maximum funded amount

Data Split to Train & Test

Ideally, while performing modelling, each observation should either be used for exploration or confirmation, not both. One can use an observation as many times for exploration, but should only use it once for confirmation. As soon as an observation twice, the task has switched from confirmation to exploration. This is necessary because to confirm a hypothesis it is essential that data must be used independent of the data that has been used to generate the hypothesis. Otherwise this would lead to over optimistic results. In a strict sense, this requires us to split the data into different sets:

This means that even for exploratory data analysis (EDA), we would only look at parts of the data. All exploratory analysis will be performed on the training data only.

One note of caution is necessary here. Since not all data is used for model fitting, the test data may have labels that do not occur in the training set and with same rationale features may have unseen values. So before using the test data for prediction, it would be necessary to prepare the same to be in line with training data.

I have used caret::createDataPartition() function to split my data to training and test sets. One of the primary reasons for choosing the same is that the data is imbalanced, i.e. only a few lenders default while many more do not. The last fact requires a non-random split considering the class label (default / non-default). The caret::createDataPartition() uses stratified sampling by default , which ensured that the proportion of defaulters to non-defaulters in the training and testing data remain same.

Stratified Sampling (proportion of defaulters and non-defaulters in training and testing data		
	Non-Defaulters	**Defaulters**
Original Data	0.91593338	0.08406662
Training Data	0.91593381	0.08406619
Testing Data	0.91593239	0.08406761

Exploratory Data Analysis-Univariate, Bivariate, Multivariate

Exploratory Data Analysis is an important part of any model building exercise as it ensures that the dataset has been understood in term of distributions, frequency etc. A visual look at the data should is an excellent starting point and usually always precede any model considerations. Looking at variables individually would lead to greater insights from the available dataset. Doing so, would clearly enable us to look at distribution patterns and, whether or not they have presence of significant outliers Also consider group sizes and differences between median and mean driven by outliers. Especially when drawing conclusions from summarized / aggregated information, we should be aware of group size.

UNIVARIATE & BIVARIATE ANALYSIS

Numerical Attributes-Profiling

Profiling of numerical variables using funModeling::profiling_num () function clearly indicated significant skew amongst many variables. Specifically, variables like annual income, total_rec_late_fee, revol_bal & delinq_2yrs have a significant positive skew and more likely the distribution is to have outliers. On the other hand, variables like revol_util, dti and interet rates seem to have a relatively normal distribution.

Numerical variables ans skewness			
Variable	Skewness	Variable	Skewness
annual_inc	41.2605192	total_pymnt_inv	1.0009095
total_rec_late_fee	13.9510138	total_rec_prncp	0.8712099
revol_bal	12.1203514	funded_amnt	0.8523403
delinq_2yrs	5.7860568	funded_amnt_inv	0.8507359
out_prncp	4.8297786	loan_amnt	0.8457541
out_prncp_inv	4.8296449	total_acc	0.8177186
total_rec_int	2.6582897	mths_since_last_delinq	0.4082122
inq_last_6mths	1.4607315	int_rate	0.3886775
last_pymnt_amnt	1.3192483	dti	0.2486475
open_acc	1.1615537	revol_util	-0.0187064
installment	1.0156758	ecoveries	NaN
total_pymnt	1.0011758	collection_recovery_fee	NaN

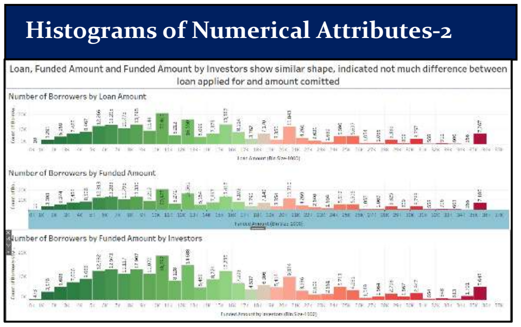
Numerical Attributes-Univariate & Bivariate Plots and Observations

The plots of key numerical variables and insights are included below:

Loan Amount, Funded Amount & Funded Amount by Investors

The distributions of all three variables seem similar in shape suggesting not too much divergence between the loan amount applied for, the amount committed, and the amount committed by investors.

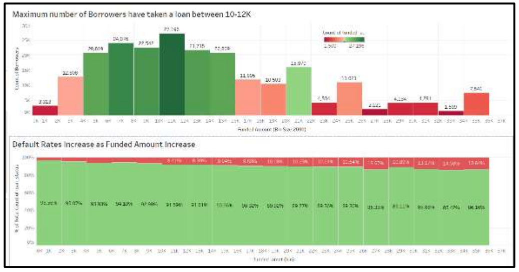
Histograms of Numerical Attributes-2



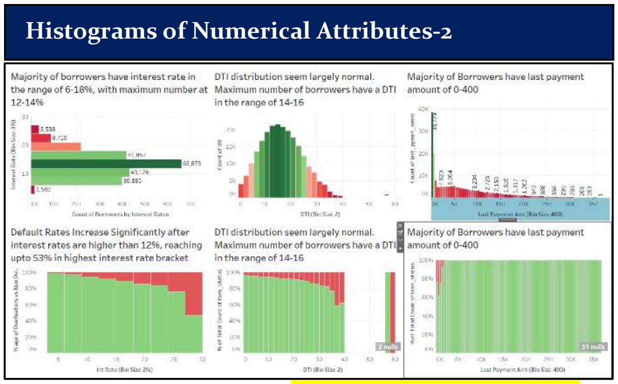
Funded Amount

Maximum number of borrowers have taken a loan between 10-12K. Other funded amount ranges which have large number of borrowers include 6-8K, 8-10K, 12-16K etc.. **It can be clearly seen that default rates increase as funded amount increase.**

Histograms of Numerical Attributes-2



Interest Rates, DTI, & Last Payment Amount



Histograms of Numerical Attributes-2

Majority of borrowers have interest rate in the range of 6-18%, with maximum number at 12-14%

DTI distribution seem largely normal. Maximum number of borrowers have a DTI in the range of 14-16

Majority of Borrowers have last payment amount of 0-400

Default Rates Increase Significantly after interest rates are higher than 12%, reaching upto 53% in highest interest rate bracket

DTI distribution seem largely normal. Maximum number of borrowers have a DTI in the range of 14-16

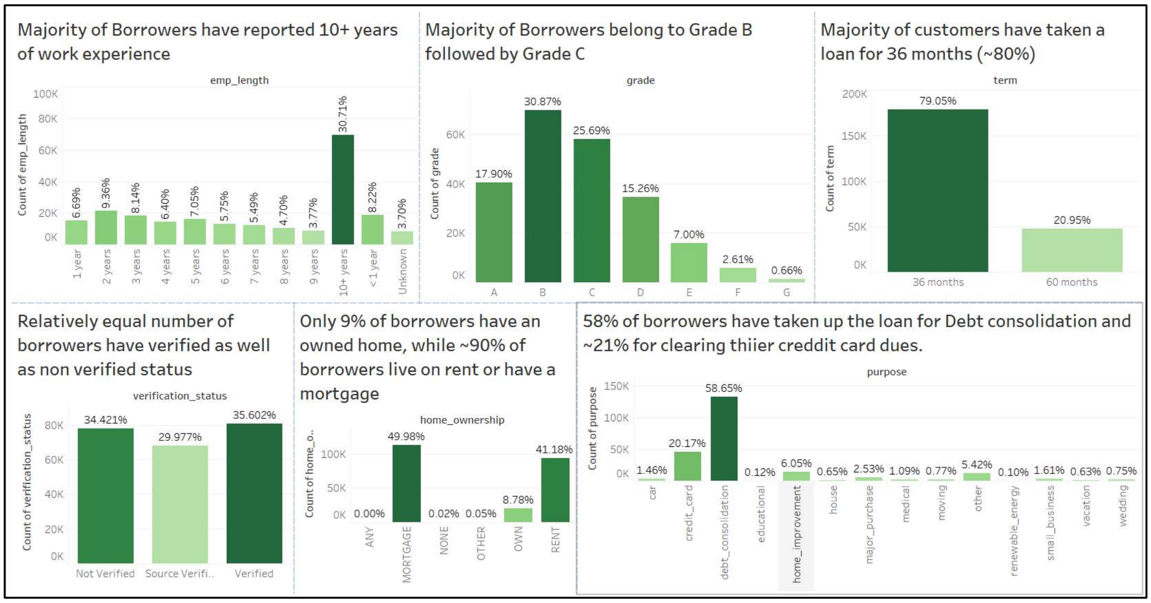
Majority of Borrowers have last payment amount of 0-400

The distribution of interest rates and DTi seem largely normal. With respect to interest rates maximum number of borrowers have an interest rate of 12-14%. **The default rates increase as interest rates increase.** Similarly, for DTI, it can be seen that the distribution is largely normal, with majority of borrowers with a DTI of 14-16. **Default rates for DTI also increase as the DTI increases.** For last payment amount however it can be seen that the distribution is significantly positively skewed. **Default rates are much higher for people who have a last payment amount of 0-400.**

Similar observations can be made for other variables as well.

Categorical Attributes-Univariate & Bivariate Plots and Observations

Profiling as well as plots of categorical data shows throw some useful insights. Majority of the borrowers have a term of 36 months, belong to grade B, have 10+ years' experience, live in mortgage property, use the loans for debt consolidation, and stay in California. Looking at the purpose and add-state variables showcases that have large number of fields/categories, hence it might be useful to group these categories before modelling. These steps would be included in next stage.



Majority of Borrowers have reported 10+ years of work experience

Majority of Borrowers belong to Grade B followed by Grade C

Majority of customers have taken a loan for 36 months (~80%)

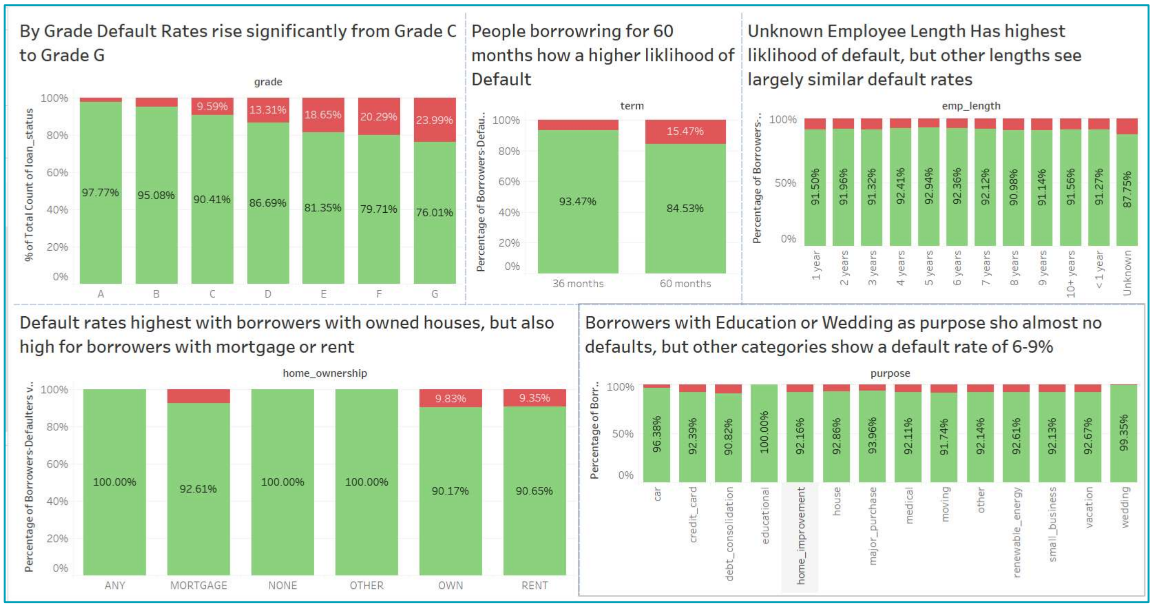
Relatively equal number of borrowers have verified as well as non verified status

Only 9% of borrowers have an owned home, while ~90% of borrowers live on rent or have a mortgage

58% of borrowers have taken up the loan for Debt consolidation and ~21% for clearing thiier creddit card dues.

Categorical Attributes with Target: Looking at the categorical variables with respect to target, rates of default steadily increase as the loan grades worsen from A to G, as expected. Propensity to default is higher in people taking loan for 60 months. other variables with default can be seen below.

Bivariate Analysis of Categorical Variables



By Grade Default Rates rise significantly from Grade C to Grade G

People borrowring for 60 months how a higher liklihood of Default

Unknown Employee Length Has highest liklihood of default, but other lengths see largely similar default rates

Default rates highest with borrowers with owned houses, but also high for borrowers with mortgage or rent

Borrowers with Education or Wedding as purpose sho almost no defaults, but other categories show a default rate of 6-9%

Outlier Treatment

Outliers are extreme values that deviate from other observations on data, they may indicate a variability in a measurement, experimental errors or a novelty. In other words, an outlier is an observation that diverges from an overall pattern on a sample.

Many machine learning algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of machine learning algorithms resulting in longer training times, less accurate models and ultimately poorer results.

Many machine learning models, like linear & logistic regression, are easily impacted by the outliers in the training data. However, tree-based algorithms such as decision tree and random forest are generally robust to outliers. But the sake of consistency, I have applied outlier treatment on the datasets used for all algorithms.

While outliers can be treated in a number of ways, I have used winsorizaiton technique to cap the outliers at 1 percentile and 99 percentile respectively.

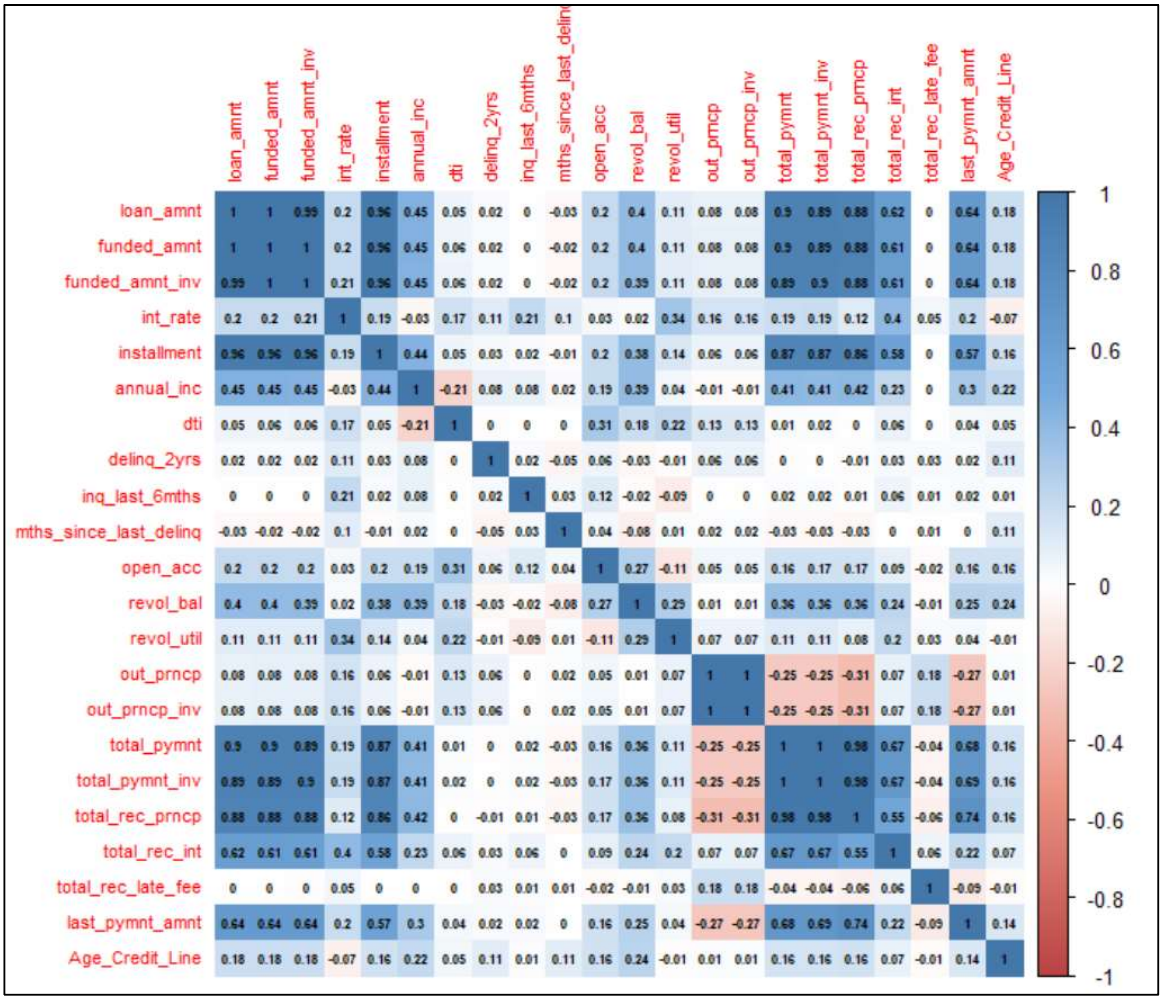
Multivariate Analysis-Correlation

Collinearity is infamously famous for inflating the variance of at least one estimated regression coefficient, which can cause the model to predict erroneously and in a business setup it can have an unrepairable consequence. Multicollinearity can also effect the sign of the relationship (i.e. positive or negative) and the degree of effect on the independent variable.

Many models rely on the notion of correlation between independent and dependent variables so a natural exploratory visualization would be a correlation plot or correlogram. However, its important to note that correlation plot shows the strength of relationship between variables and not causation. But this plot gives a good idea on which of the variables might be highly dependent on each other.

We can produce a correlation plot which looks as below:

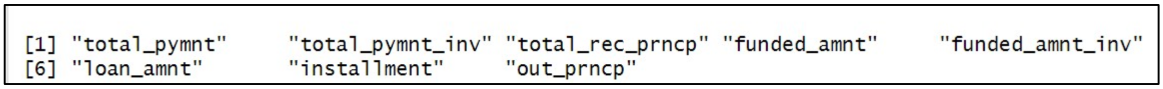
Collinearity Analysis



As it can be seen clearly that many variables such as loan_amnt, funded_amnt, funded_amnt_inv, total_payment, total_pymnt_inv, installment etc. have extremely strong relationship amongst each other (~90%).

Rather than a visual inspection, I have used the technique of an (automatic) inspection of correlations and removal of highly correlated features via function caret::findCorrelation() with a defined cutoff parameter. If two variables have a high correlation, the function looks at the mean absolute correlation of each variable and removes the variable with the largest mean absolute correlation. Using exact = TRUE will cause the function to re-evaluate the average correlations at each step while exact = FALSE uses all the correlations regardless of whether they have been eliminated or not. So, upon using the function the variables below were identified which needed to be removed.

Variable removed after collinearity analysis



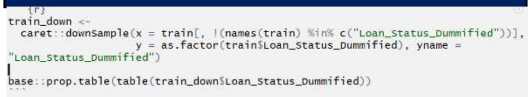
```
[1] "total_pymnt"     "total_pymnt_inv" "total_rec_prncp" "funded_amnt"     "funded_amnt_inv"
[6] "loan_amnt"       "installment"     "out_prncp"
```

Data-Set Imbalance Correction

USING TECHNIQUE TO BALANCE DATA

Before we can think about any modeling, we have to realize that the dataset is highly imbalanced, i.e. the target variable has a very low proportion of defaults (namely 0.08406617). This can lead to several issues in many algorithms, e.g.

Balancing Data



```
   {r}
train_down <-
  caret::downSample(x = train[, !(names(train) %in% c("Loan_Status_Dummified"))],
                    y = as.factor(train$Loan_Status_Dummified), yname =
"Loan_Status_Dummified")

base::prop.table(table(train_down$Loan_Status_Dummified))
```

- biased accuracy
- loss functions attempt to optimize quantities such as error rate, not taking the data distribution into consideration
- errors have the same cost (not pertaining to imbalanced data only).

Under sampling the majority class may lose information but via decreasing dataset also lead to more computational efficiency. We also tried SMOTE and ROSE but functions DMwR::SMOTE() and ROSE::ROSE() seem to be very picky about their input and so far we failed to run them. As pointed out here a random sampling for either under- or oversampling is not the best idea. It is recommended to use cross-validation and perform over- or under-sampling on each fold independently to get an honest estimate of model performance. For now, we go with under sampling which still leaves a fair amount of training observations (at least for non-deep learning approaches).

Model Building

For building all the models, I have used caret::train() function. The train() function is essentially a wrapper around whatever method we chose. For consistency, I have used 10 fold-cross validation repeated 5 times, and used ROC as metrics for final model selection.

LOGISTIC REGRESSION

Logistic regression is one of the most used algorithms in classification problems. Here, we see our problem is the one belonging to binary classification class. Hence the first algorithm we use to predict default is "Logistic Regression".

Logistic regression predicts the probability of the outcome being true. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. t predicts the probability of occurrence of an event by fitting data to a logit function. Few important points to be noted about logistic regression:

- GLM does not assume a linear relationship between dependent and independent variables. However, it assumes a linear relationship between link function and independent variables in logit model.
- The dependent variable need not to be normally distributed.
- It does not uses OLS (Ordinary Least Square) for parameter estimation. Instead, it uses maximum likelihood estimation (MLE).
- Errors need to be independent but not normally distributed.

PERFORMANCE USING LOGISTIC REGRESSION

In our case when using the logistic regression model, on the training dataset overall seemed to perform well, with an overall accuracy of 89.24%. However, for problems with imbalanced class distribution as ours (only 8% of the instances were defaulters in our case), we better look at other performance metrics.

Since our primary assumption was that the firm wanted to identify as many defaulters correctly as possible, **I would probably be focusing on recall or sensitivity, which is the proportion of actually defaulting customers that the model identifies or predicts correctly**. As it can be seen, the logistic model (on the training data) was able to predict 93% o the customers correctly. Furthermore, the specificity (that is the number of non defaulters our model was able to identify correctly) was high at 84%%, which seems good.

However, we cannot comment on model performance till the time its tested on new data (in our case testing data). On the testing data we see that the overall accuracy was 85.63%, while sensitivity was at 93.12% and specificity was 84.93%.

Since the numbers do not show a significant drop, we can say that the logistic model performed well on the data at hand.

	Logistic Regression-Training	Logistic Regression Testing	Observations
Accuracy	0.8924	0.8563	Overall accuracy do no decline much when model is used on training set, indicating robustness
Sensitivity	0.9387	0.93123	Both on training and testing the model is able to identify ~93% of defaulters correctly, which is good
Specificity	0.8460	0.84937	Also, the model identifies ~84% of non-defaulters both on training and testing set, which seems good
F1 Score	0.8971	0.52136	Logistic model had an overall F1 score o 52.13% on the testing set. This would probably be helpful, in addition to other metrics to identify the best model

CLASSIFICATION TREES

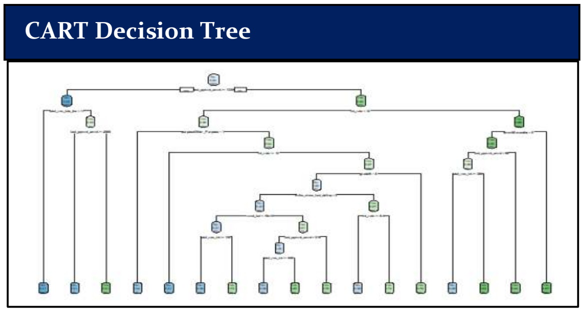
A Decision Tree or specifically the CART model is a Supervised Machine Learning where the data is continuously split according to a certain parameter. Decision trees are built using a heuristic called recursive partitioning, because it splits the data into subsets, which are then split repeatedly into even smaller subsets, and so on and so forth until the process stops when the algorithm determines the data within the subsets are sufficiently homogenous, or another stopping criterion has been met.

For our problem of classifying defaulters, I have adapted this algorithm because it's a popular classification mechanism, its inexpensive to construct, simple to explain, robust to outliers, missing data or data scales. For our problem, the final model would probably be used by the loan officer approving/or disapproving the loan. So, he/she must clearly understand the reasons for taking the decision. However, CART, so of the issues that this model suffers from include propensity to overfitting, instability, as well as probability of being biased toward splits on features having many levels.

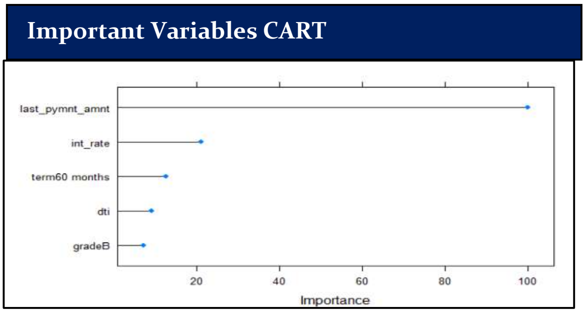
PERFORMANCE OF THE CART MODEL

The CART model in case was able to breakdown the process of classification of a borrower to defaulter in simple steps. The final tree provided clearly gave an indication such as if the last payment amount is below/above a certain amount then a borrower could be out in a bucket with a second set of rules applied and so on. The CART tree and important variables that could be used to identify a defaulter are given below:

CART Decision Tree



Important Variables CART



In our case when using the logistic regression model, on the training dataset overall seemed to perform well, with an overall accuracy of 90.48%. The CART model (on the training data) was able to predict 95.9% of the defaulting customers correctly, which was better then the logistic model. Furthermore, the specificity (that is the number of non defaulters our model was able to identify correctly) was high at 85%%, which seems good and was marginally higher than the logistic model tried above.

On the testing data we see that the overall accuracy was 85.77%, while sensitivity was at 95.29% and specificity was 84.90%.

Since the numbers do not show a significant drop, we can say that the CART model also performed well on the data at hand.

	CART-Training	CART Testing	Observations
Accuracy	0.9048	0.8577	Overall accuracy do no decline much when model is used on training set, indicating robustness
Sensitivity	0.9594	0.95293	Both on training and testing the model is able to identify ~95% of defaulters correctly, which is good
Specificity	0.8502	0.84900	Also, the model identifies ~85% of non-defaulters both on training and testing set, which seems good
F1 Score	0.9098	0.52969	Logistic model had an overall F1 score o 52.96% on the testing set. This would probably be helpful, in addition to other metrics to identify the best model

RANDOM FOREST MODEL

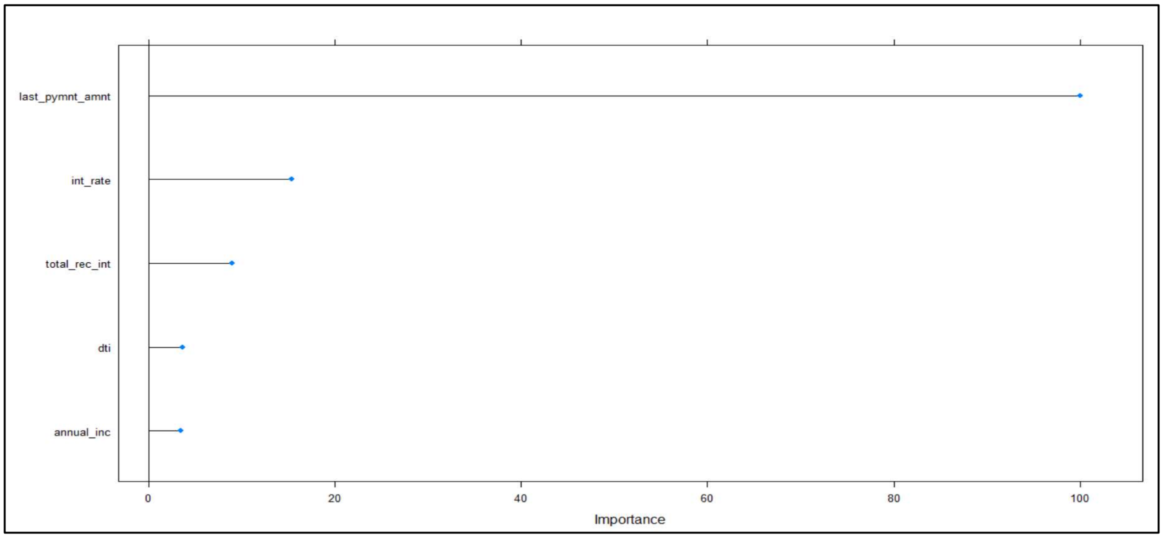
Random forest, like its name implies, consists of many individual decision trees that operate as an ensemble. The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. The main principle behind ensemble methods is that a group of “weak learners” can come together to form a “strong learner”. In fact, random forest algorithm offers predictive performance can compete with the best supervised learning algorithms.

For predicting the loan defaulters, I chose the random forest algorithm as the final algorithm as in addition to being one of the most powerful algorithms, it's easy to learn and use for both professionals and lay people. It provides god accuracy, runs efficiently on relatively large datasets such as ours, robust to outliers, missing data and scales, and is a stable algorithm. However, there are some drawback that we should be aware of that include complexity and longer training period.

PERFORMANCE OF THE RANDOM FOREST MODEL

The random forest model was able to perform well and provided important variables as given below:

Important Variables CART



In our case when using the random forest model, on the training dataset overall seemed to perform extremely well, with an overall accuracy of 99.86%. The meant out of 26670 observation, out model was able to predict 26633 observations correctly. This was good news, however, was a problem as well as it might indicate that the model has overfit, and learn the noise in the data as well. This could only be confirmed once this has been tested on new data and in turn provides much lower accuracy and other metrics. For the time being, the performance on the training dataset on other metrics could be noted.

The random forest model (on the training data) was able to predict 99.9% of the defaulting customers correctly. Furthermore, the specificity (that is the number of non-defaulters our model was able to identify correctly) was high at 99.83%%, which seems good and was marginally higher than the logistic & CART model tried above.

Now comes the important part, if our model has performed well that means it has not overfit the data , the we should not see a significant decline in its performance on the testing dataset. On the testing data we see that the overall accuracy was 92.42%, which is not siginifcatly different from the one seen on training data hence we could conclude that the model seems to be working properly. The sensitivity on the testing data stood at was at 93.96% and specificity was 92.27%.

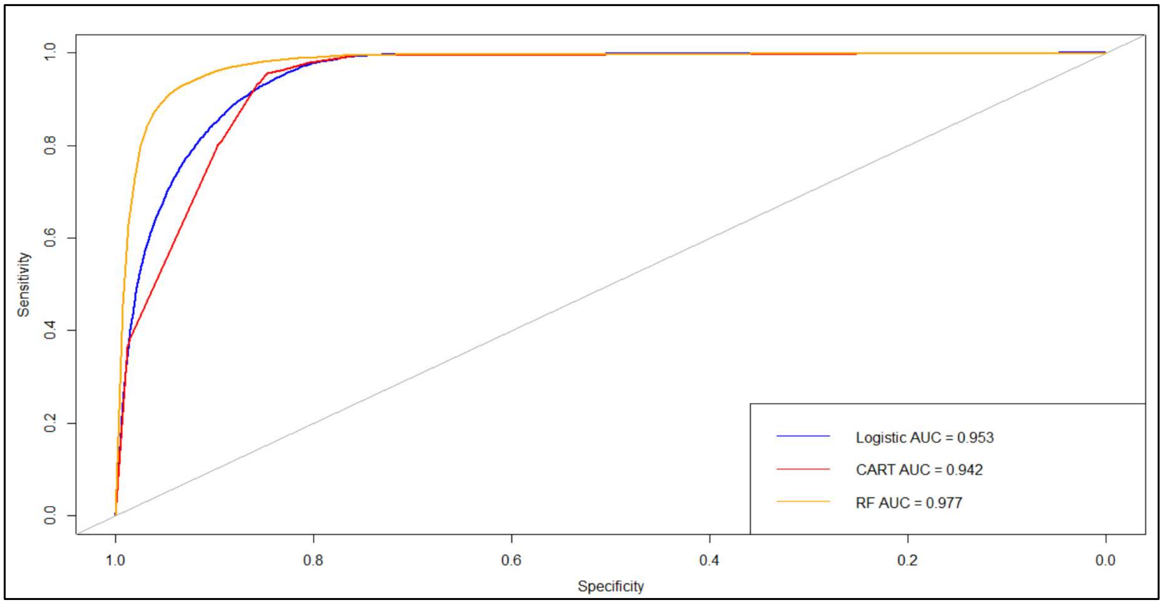
. A comparative performance of the model on training and testing data can be seen below:

	RF-Training	RF-Testing	Observations
Accuracy	0.9986	0.9242	Overall accuracy do no decline much when model is used on training set, indicating robustness
Sensitivity	0.9990	0.93963	On training and testing the model is able to identify ~99% and ~94%of defaulters correctly, which is good
Specificity	0.9983	0.92278	Also, the model identifies ~99% an ~93% of non-defaulters both on training and testing set, which seems good
F1 Score	0.9986	0.67577	RF model had an overall F1 score 67.58% on the testing set. This would probably be helpful, in addition to other metrics to identify the best model

Model Comparison

While all the 3 models performed really well on our data to predict defaulters (all models identified >90% defaulters correctly), given the initial objective that the firm urgently wants to reduce defaulters by identifying then as much as possible (before the loan approval), while not losing out on many good customers (as is the case with almost all business organizations) I would go ahead and suggest the random forest model. The reason being the model was correctly able to identify ~94% of the defaulters correctly (sensitivity). This was marginally higher that sensitivity of logistic model (~93%) although marginally lower than CART model (~95%). On the flip side the Rf model was able to identify ~93% of the non-defaulters as well, which is relatively higher than 85% for both the logistic regression and the CART model. Additionally, based on roc curve as well the Random Forest model performed the best.

ROC Curve- Logistic, CART & Random Forest



However, if one of the additional goals were that the model be also explainable to the loan officers who would be using it in the end, I would prefer logistic model as it would given them a clear idea on changes in probability of a customer truing to defaulter if one of the variables change by some units. Furthermore, this model would also give them the flexibility to change probability thresholds which is an added advantage.

Final Recommendations to the Banks

Given the analysis and exploration of the dataset, following advice to be rendered to the management:

- A relook at the strategies with respect to amount of loan funded, specifically with customers belonging to Grade C to Grade D and probably reduce the amount of loan granted.
- Expanding in states like IA, ID & DC, which have really low default rates.
- Stricter requirements for loans funded recently as default rates seem to be higher with recent customers
- Additional processes to re-evaluate people with last payment amount of $0-500, as this is one of the significant predictors

Additionally, it would also be suggested that the models be tried with datasets containing loan applications that have been approved/rejected as it would provide a better idea on the kind of model that could be deployed. Furthermore, tuning model hyper parameters could also be tried on to achieve better performance.