FORECLOSURE PREDICTION IN NBFCs

FINAL REPORT

Vikram Radhakrishnan

24th January 2021

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The 2008 financial meltdown was a grim reminder for the need for financial institutions such as banks and Non-Banking Financial institutions to predict the possibility of loan defaults. This would help them improve their risk management capability by having an early warning system for high-risk loans. These systems would provide sufficient time and intelligence to work towards pre-emptive action and allocate resources to preserve their financial health.

FINANCIAL RISK MANAGEMENT AND MOTIVATION

Risk Management has always been a part of the process for managing loans. This was however, largely done using manual methods earlier and decisions were taken based on conservative empirical calculations or experiential human learning. Overall the last few decades many strides have been made in automating the process as well as handling large and rapidly changing data.

Therefore, it is a great need for financial institutions in today's fast paced world, to not only automate the process of prediction of loan defaults but do so with higher confidence and accuracy based on patterns of transactions that have happened in the past.

Machine learning algorithms can be very useful here by 'learning' from past data and predicting the likelihood of a borrower's chance to default from a large number of parameters. This process is not only very fast but can provide better reliability and accuracy compared to human judgement when implemented correctly.

CURRENT STUDY

In our current study, we are provided with aggregate data consisting of multiple parameters, of loans that have been disbursed an NBFC. Each of these entries also has a parameter describing whether that particular loan was foreclosed or not. Using this data as as a reference or learning data, one has to predict whether a particular loan applied or taken would likely result in a default/foreclosure or remains status quo. It is therefore what we refer to in statistical terms, as a binary classification problem with a supervised learning approach.

A BRIEF INTRODUCTION INTO NFBCs, FORECLOSURES

Before we move on to solving the business problem, lets look at some basic information that will help us understand the terminologies and case better.

What is an NBFC?

A Non-Banking Financial Company (NBFC) is a company registered under the Companies Act, 2013 of India, engaged in the business of loans and advances, acquisition of shares, stock, bonds, hire-purchase insurance business or chit-fund business, but does not include any institution whose principal business is that of agriculture, industrial activity, purchase or sale of any goods (other than securities) or providing any services and sale/purchase/construction of immovable property.

How are NBFCs different from Banks?

While NBFCs may appear to be similar to banks, there are distinct differences that set them apart from regular Banks.

- NBFCs provide banking services to people without holding a Bank licence,
- An NBFC cannot accept Demand Deposits,
- An NBFC is not a part of the payment and settlement system
- An NBFC cannot issue Cheques drawn on itself
- Deposit insurance facility of the Deposit Insurance and Credit Guarantee Corporation is not available for NBFC depositors, unlike banks
- An NBFC is not required to maintain Reserve Ratios (CRR, SLR etc.)
- An NBFC cannot indulge primarily in agricultural or industrial activities or sale-purchase, construction of immovable property
- Foreign Investment allowed up to 100 %
- An NBFC accompanies working in Financial Body and Money handling

When or Why are they preferred over banks?

That leads to one to question – why would one approach an NBFC instead of a bank?

- Quick Disbursal of Funds
- Competitive Interest rates
- Lenient Eligibility Criteria
- Relatively minimal Paperwork and Documentation

What is a foreclosure?

Foreclosure is the legal process by which a lender (NBFC in this case) seizes and sells a home or property after a borrower is unable to fulfill his or her repayment obligation.

2. EXPLORATORY DATA ANALYSIS (EDA) and BUSINESS IMPLICATION

BASIC UNDERSTANDING OF DATA PROVIDED

The data provided lists information related to a series of loans authorized or active from **2010 to 2018**, tracks their status (on a particular date) in terms of interest rates, payments, balances e.t.c. Based on the values of the fields provided, it appears that the loan is tracked on a monthly basis.

The data contains both static and dynamic data. Some fields such as agreement ID, authorization date, customer ID, and product e.t.c will remain fixed or static every month, whereas other fields such as paid principal, paid interest, month opening, due day e.t.c will keep changing with time (dynamic).

From a quick perusal of the data, we see observe that there are 53 fields, our columns in our case. 52 of these can be considered as independent variable which influence a single dependent variable, which in this case is 'foreclosure'. Some of these independent variables will influence 'foreclosure' and some may not – for example, agreement ID or customer ID are only identifiers and may not affect the result itself. However, its possible that if a customer defaults on one loan, he may default on the others as well. Therefore its important to understand which features or columns to keep for the analysis and which can be ignored.

There are three 'date-time' fields, 4 'object' or string fields for city, product and NPA. The rest of the fields are numerical in the form of either float or integers.

There are 20012 records or rows in all. More details will follow in EDA.

Basic statistics of the data is also taken out. Information such as mean, median, standard deviation e.t.c are extracted for a quick understanding.

Duplicate records are checked for. No duplicates are found.

A. UNIVARIATE ANALYSIS

Univariate analysis is done on all columns of the dataset. However, in this report, we provide a small set of insights that may prove interesting to the reader.

AUTHORIZATION DATE

We observe that 2018 has the highest number of loans authorized and the amount of authorizations has increased exponentially or steadily Y-O-Y.

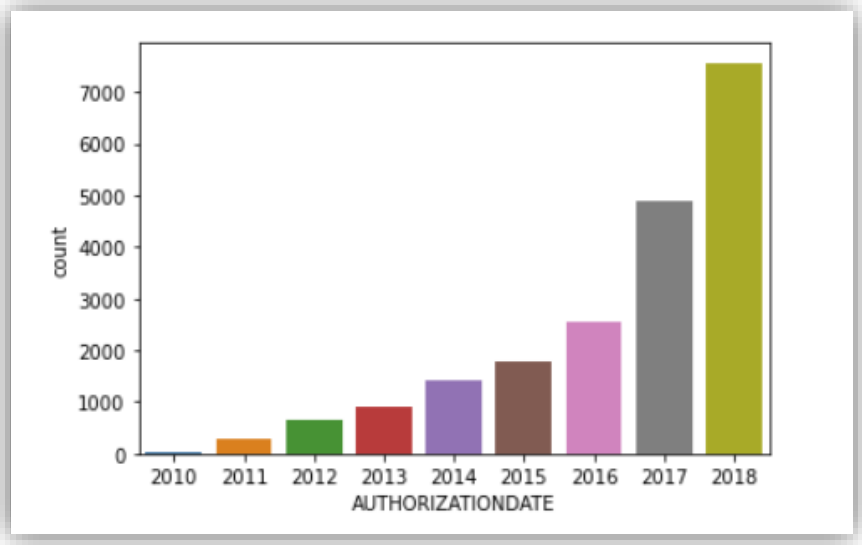

Figure 1: Loans Authorized per year (2010-2018) - Exponential Growth

Highest number of loans appear to be taken in the month of March and the lowest in April.

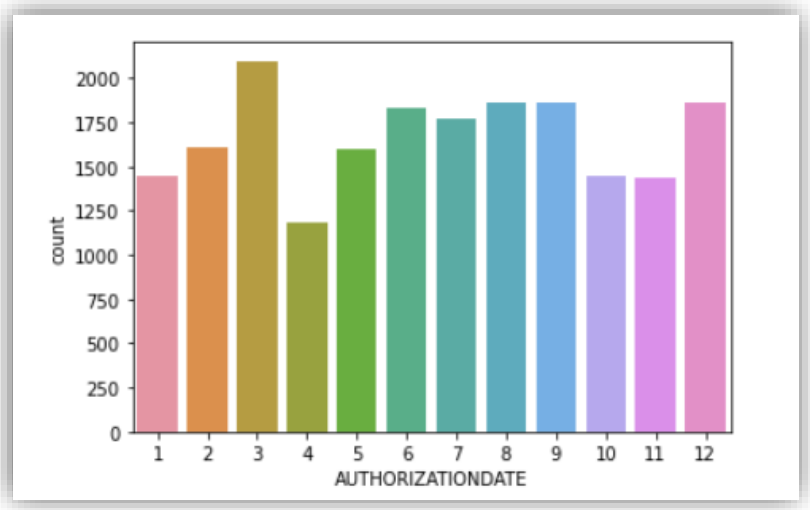

Figure 2 : Loans taken between 2010-2018, Monthwise

To understand the number of foreclosures in the data vs the number of 'status-quo' cases, we can use a pie chart to view the numbers. The number of foreclosures is approximately 10% of the total number of cases.

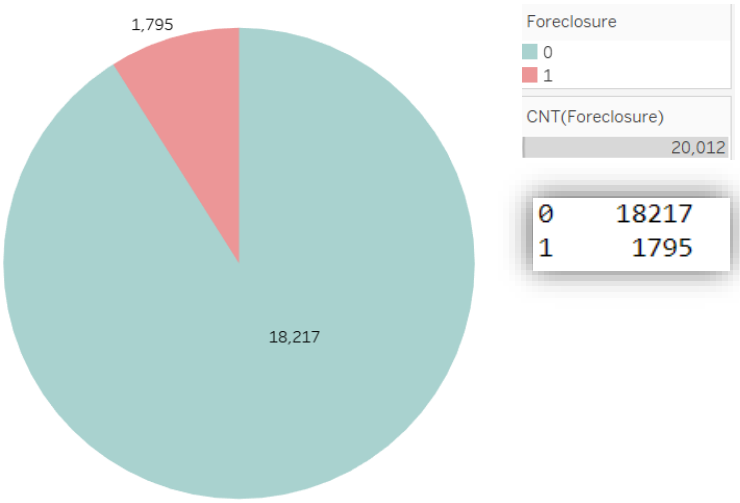

Figure 3 : Number of foreclosures vs. Open Loans in Dataset

TENURE

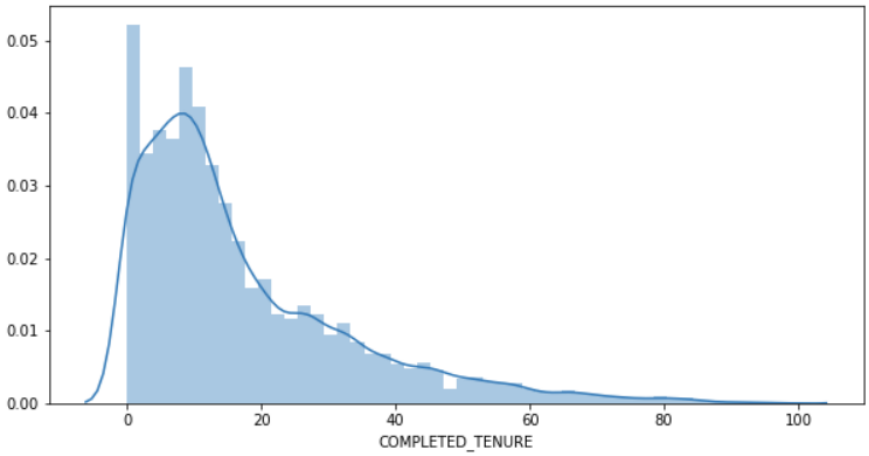

Figure 4: Distribution plot of loans, by completed tenure

The *completed tenure* histogram is left skewed which shows that a large number of active loans have been taken in the last few years.

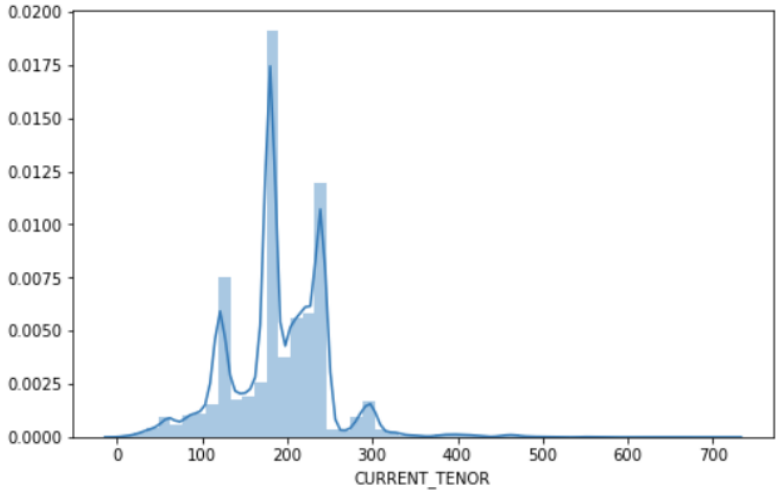

Figure 5: Distribution of loans as per current tenure (in months)

If we observe the *current tenure*, it is seen that the largest density is found in loans that are around 10,15 and 20 years.

B. BIVARIATE ANALYSIS

To start with, a heat map is drawn to find out the correlation between the variables. However, it is difficult to view the heatmap in the report in a single page.

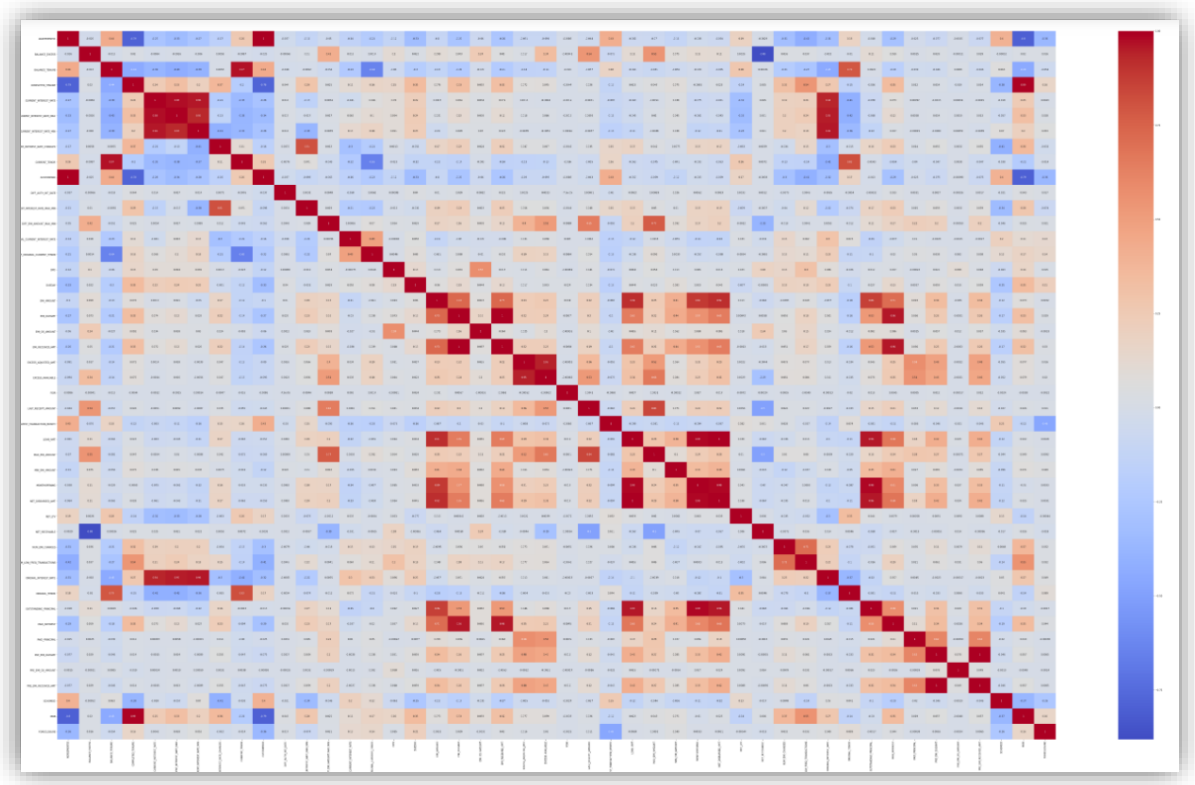

Figure 6: Correlation heatmap of all features in dataset - red hues indicate high correlation

Instead of viewing the heatmap graphically with multiple variables, the highly correlated feature pairs are printed instead. In this case, **we print values that have a correlation value of more than 0.95.**

```
('AGREEMENTID', 'CUSTOMERID')
('BALANCE_TENURE', 'CURRENT_TENOR')
('COMPLETED_TENURE', 'MOB')
('CURRENT_INTEREST_RATE', 'CURRENT_INTEREST_RATE_MAX')
('CURRENT_INTEREST_RATE', 'CURRENT_INTEREST_RATE_MIN')
('CURRENT_INTEREST_RATE_MAX', 'CURRENT_INTEREST_RATE')
('CURRENT_INTEREST_RATE_MAX', 'ORIGNAL_INTEREST_RATE')
('CURRENT_INTEREST_RATE_MIN', 'CURRENT_INTEREST_RATE')
('CURRENT_INTEREST_RATE_MIN', 'ORIGNAL_INTEREST_RATE')
('CURRENT_TENOR', 'BALANCE_TENURE')
('CUSTOMERID', 'AGREEMENTID')
('EMI_DUEAMT', 'EMI_RECEIVED_AMT')
('EMI_DUEAMT', 'PAID_INTEREST')
('EMI_RECEIVED_AMT', 'EMI_DUEAMT')
('EMI_RECEIVED_AMT', 'PAID_INTEREST')
('LOAN_AMT', 'MONTHOPENING')
('LOAN_AMT', 'NET_DISBURSED_AMT')
('LOAN_AMT', 'OUTSTANDING_PRINCIPAL')
('MONTHOPENING', 'LOAN_AMT')
('MONTHOPENING', 'NET_DISBURSED_AMT')
('MONTHOPENING', 'OUTSTANDING_PRINCIPAL')
('NET_DISBURSED_AMT', 'LOAN_AMT')
('NET_DISBURSED_AMT', 'MONTHOPENING')
('NET_DISBURSED_AMT', 'OUTSTANDING_PRINCIPAL')
('ORIGNAL_INTEREST_RATE', 'CURRENT_INTEREST_RATE_MAX')
('ORIGNAL_INTEREST_RATE', 'CURRENT_INTEREST_RATE_MIN')
('OUTSTANDING_PRINCIPAL', 'LOAN_AMT')
('OUTSTANDING_PRINCIPAL', 'MONTHOPENING')
('OUTSTANDING_PRINCIPAL', 'NET_DISBURSED_AMT')
('PAID_INTEREST', 'EMI_DUEAMT')
('PAID_INTEREST', 'EMI_RECEIVED_AMT')
('PRE_EMI_DUEAMT', 'PRE_EMI_RECEIVED_AMT')
('PRE_EMI_RECEIVED_AMT', 'PRE_EMI_DUEAMT')
('MOB', 'COMPLETED_TENURE')
```

Figure 7 : Sets/Pairs of Highly correlated features (>0.95)

If we adjust the value of correlation threshold to 0.98, the following values are observed to be well correlated. Some of the highly correlated features in pairs can be removed.

```
('AGREEMENTID', 'CUSTOMERID')
('COMPLETED_TENURE', 'MOB')
('CUSTOMERID', 'AGREEMENTID')
('EMI_DUEAMT', 'EMI_RECEIVED_AMT')
('EMI_RECEIVED_AMT', 'EMI_DUEAMT')
('LOAN_AMT', 'NET_DISBURSED_AMT')
('MONTHOPENING', 'NET_DISBURSED_AMT')
('MONTHOPENING', 'OUTSTANDING_PRINCIPAL')
('NET_DISBURSED_AMT', 'LOAN_AMT')
('NET_DISBURSED_AMT', 'MONTHOPENING')
('OUTSTANDING_PRINCIPAL', 'MONTHOPENING')
('PRE_EMI_DUEAMT', 'PRE_EMI_RECEIVED_AMT')
('PRE_EMI_RECEIVED_AMT', 'PRE_EMI_DUEAMT')
('MOB', 'COMPLETED_TENURE')
```

Figure 8: Sets/Pairs of Highly correlated features (>0.98)

CITIES AND FORECLOSURE

The top 5 cities with the highest number of loans taken are the following:

CITIES - Highest Number of Loans	
MUMBAI	2028
HYDERABAD	1567
AHMEDABAD	1396
SURAT	1391
PUNE	1202

It is observed that the highest number of loans are seen in cities which are business hubs.

Mumbai leads the list with the highest number of loans, followed by Hyderabad.

In terms of foreclosures, Mumbai and Hyderabad again lead the pack, with a few cities which figure out in the highest number of loans list, making an entry into this list as well.

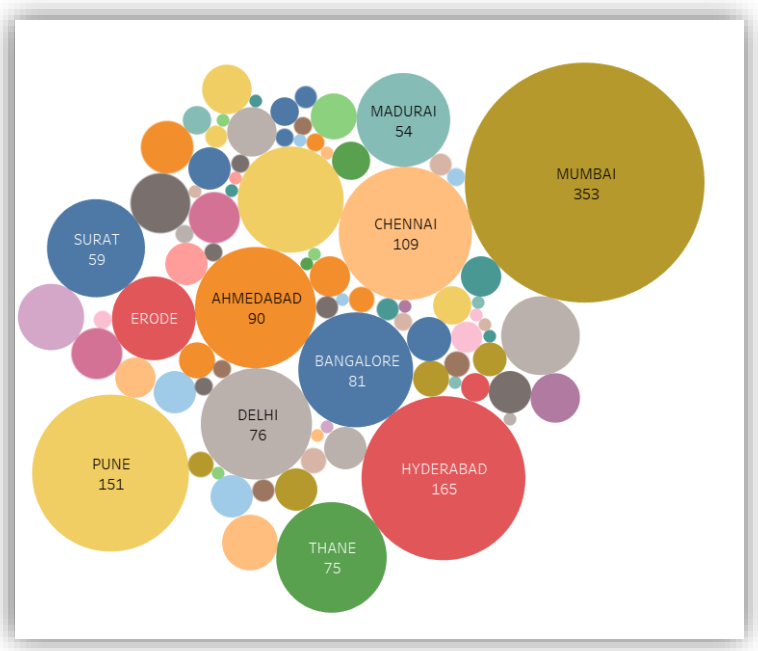

Figure 9: Bubble chart - Cities with highest number of Loans

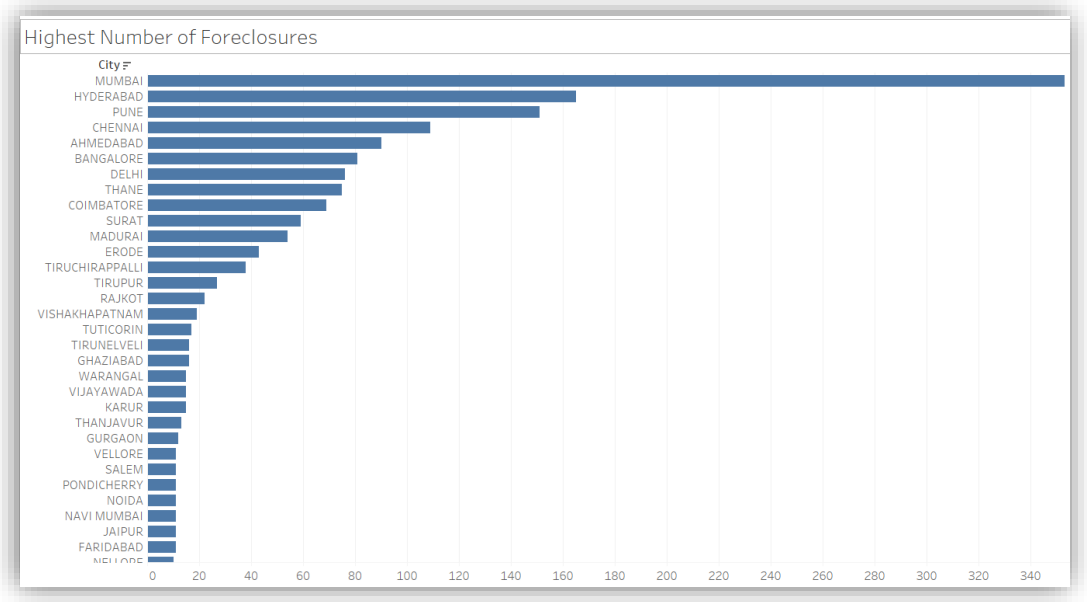

Figure 10: Bar plot View - Count of loans per city in descending order

When seen as a **ratio of defaults to loans applied per city,** smaller cities are likely to default more. In terms of big cities (metros), Mumbai seems to be high on the list of likely default.

RATIO OF DEFAULTS TO LOANS APPLIED	
NATHAM	1.000000
VEPPAMPATTU	1.000000
CHIDAMBARAM	1.000000
SABARKANTHA	0.250000
DHARMAPURI	0.250000
SRIPERAMBUDUR	0.250000
TIRUCHARAPALLI	0.236000
MEDAK	0.200000
ERODE	0.191111
MUMBAI	0.174063
GHAZIABAD	0.173913

PRODUCT

If we look at the numbers product-wise, we observe that LAP and STHL forms roughly a third each all loan disbursed. HL and STLAP appear to be less popular forming roughly a sixth each, of the total number of loans disbursed throughout.

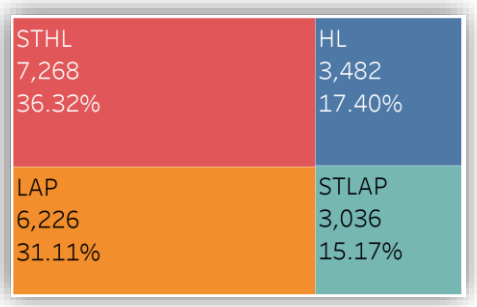

Figure 11: Loans disbursed during the entire period of study, individual counts of each product and their percentage as part of total

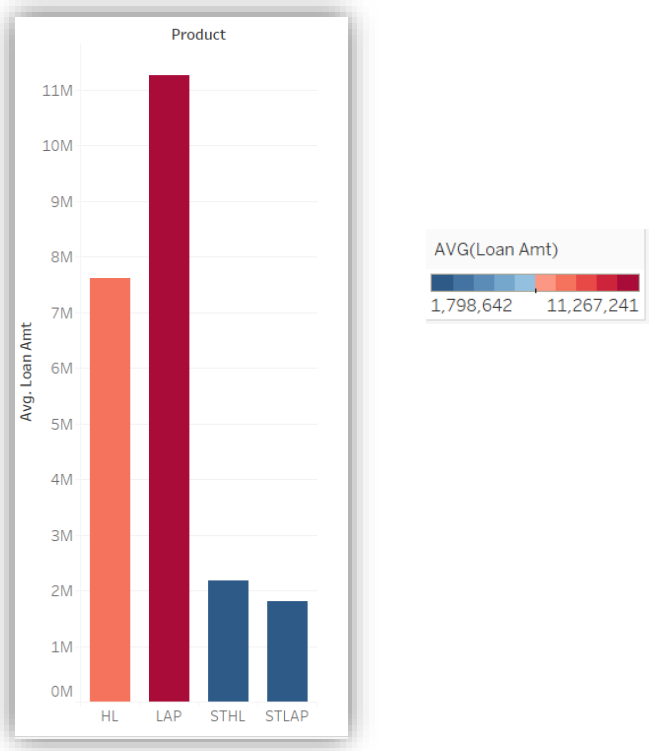

Figure 12 : Average amount of loan amount disbursed through each product

Its observed that STLAP has the highest interest rate amongst all products with a whopping 18%. LAP and STHL both have interest rates of around 14% each. The HL product is offered at the least rate of interest at a rate of 12%.

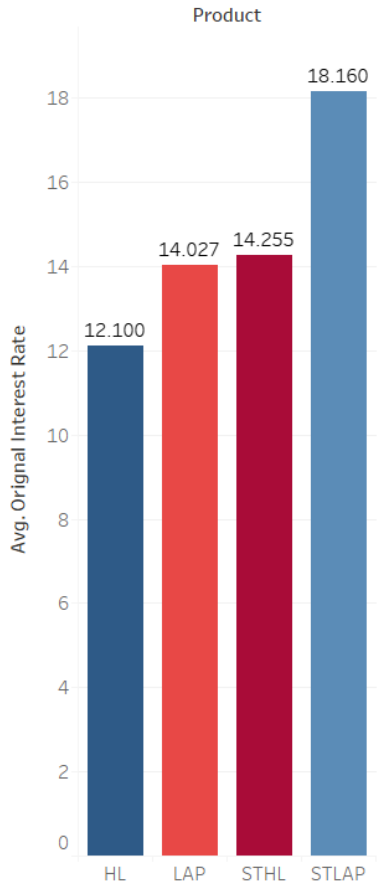

Figure 13: Original home loan rates per product

In terms of loan amount disbursed per loan product, the LAP product forms the largest share with almost half the of the total loan amount disbursed during the total period of consideration. HL comes in second with about 32 % of total loans disbursed. STHL and STLAP together form around 18% of the total amount of loans amount disbursed.

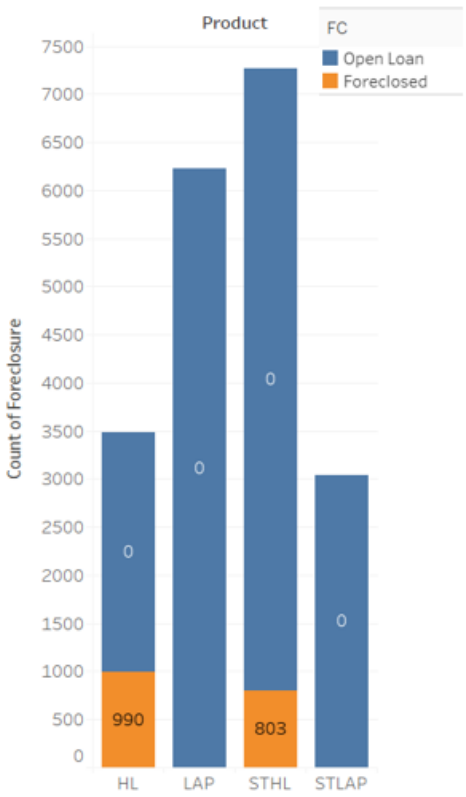

Figure 14: Proportion of Foreclosures to Open loans in each product

The plot of foreclosures reveals a very surprising trend. Only two products show significant amount of foreclosures. LAP has seen a negligible amount of foreclosures in throughout the period and STLAP has seen none. This information can be very useful to understand which products to work on to prevent foreclosures and we will be discussing more on this in the recommendations section.

3. DATA CLEANING AND PRE-PROCESSING

Once we have an general understanding of the data, we can proceed with the process of 'cleaning' the data in order to make it more suitable for analysis and 'fix' the issues or errors in the data. This will make the modeling process more accurate and faster.

The following are a few steps that are done in this effort:

a) **Removal of unwanted variables:**

Certain columns which do not have sufficient data such as NPA_IN_LAST_MONTH & NPA_IN_CURRENT_MONTH are removed. These will not play a big role in prediction.

Next, we can look at variables that are highly correlated and then proceed to remove some features that may not be necessary. When there are pairs of data that are correlated, one of the two members can be retained and the other can be discarded after careful inspection as required.

There are some columns such as CUSTOMER_ID or AGREEMENT_ID that may not be necessary and are simply required as primary identifiers. These again can be removed from the features used for modeling .

Datetime values such as 'AUTHORIZATIONDATE', 'INTEREST_START_DATE' can be removed since their difference is already captured under a variable DIFF_AUTH_INT_DATE.

b) **Null/Missing Value treatment :**

A null value check is done and the following features have null values. The number of null values in each features has also been provided in the table.

CUSTOMERID	281
DIFF_EMI_AMOUNT_MAX_MIN	89
LAST_RECEIPT_AMOUNT	247
LAST_RECEIPT_DATE	75
LATEST_TRANSACTION_MONTH	75
MAX_EMI_AMOUNT	89
MIN_EMI_AMOUNT	89
SCHEMEID	281
NPA_IN_LAST_MONTH	19893
NPA_IN_CURRENT_MONTH	19893

Treatment of null values:

Treatment of missing values by imputation or removal is important in performing logistic regression. CART may be able to handle missing data since it has in-built algorithms to impute missing values with surrogate variables. We will we look at treatment further during modeling process. However a few points on a couple of features:

CUSTOMER ID : The null values in this column cannot be treated using the usual ways of imputation since these are not statistical values. There are two options here – Either remove this column or keep the column as it is for later reference. We choose the second option.

DIFF_EMI_AMOUNT_MAX_MIN: This shows the difference between two columns

"MAX_EMI_AMOUNT" and "MIN_EMI_AMOUNT". Therefore the missing values can be derived from these two columns, as long as they have values.

We do not drop any columns or any impute any values at the moment.

c) **Outlier treatment :**

The 1.5 IQR technique can be used to treat outliers, where 1.5xIQR is subtracted from the 1st quartile and 1.5 x IQR is added to the 3rd quartile value. Any number outside these limits is treated as an outlier and moved to the nearest outlier limit.

BEFORE TREATMENT

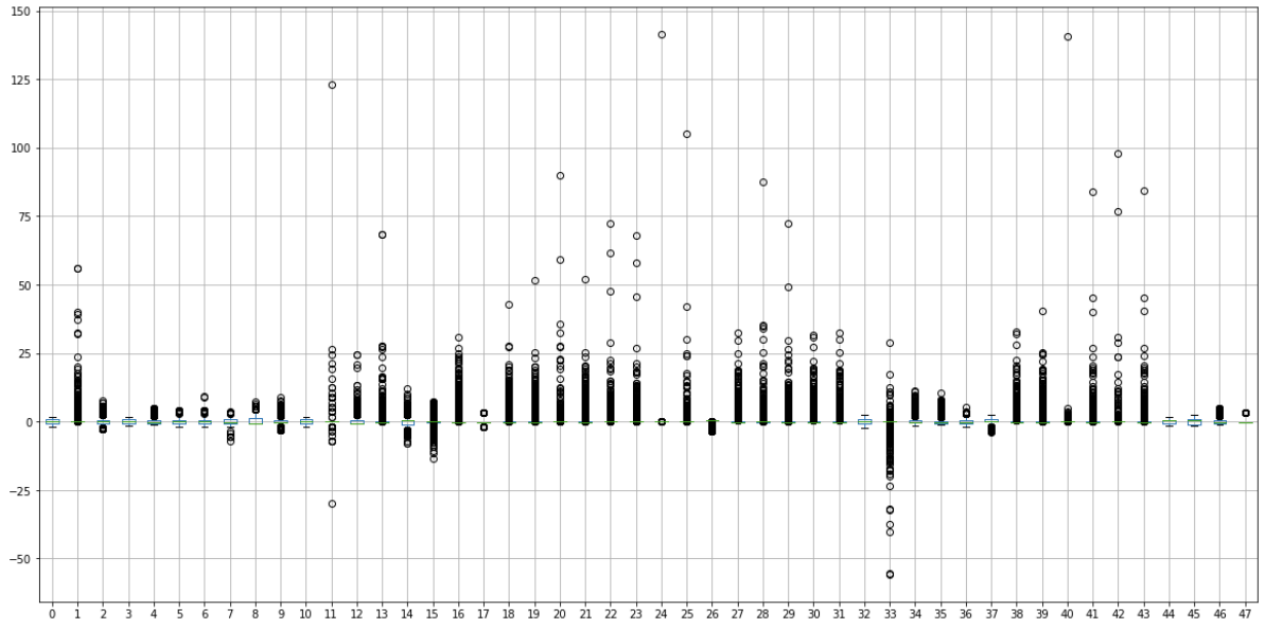

Figure 15: Plot of outliers in features (Before Treatment)

AFTER OUTLIER TREATMENT

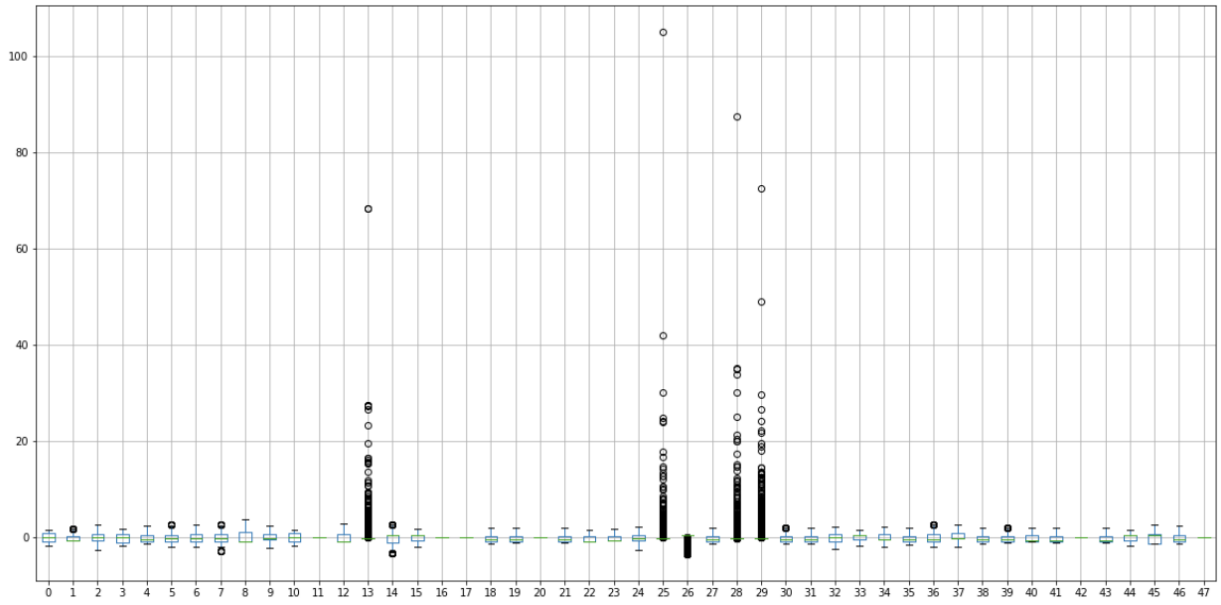

Figure 16 : Data after treating outliers in selected features

d) **Variable transformation**

In our case, we have made only a few transformations to variables such as converting few categorical variables into numerical variables through encoding.

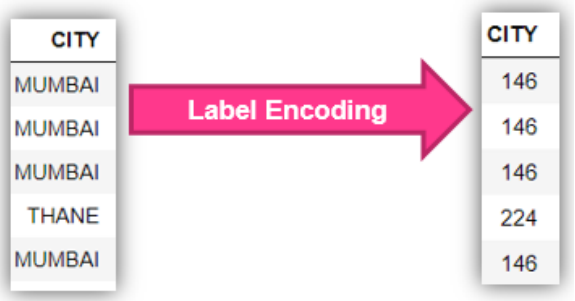

Figure 17: Example of Label Encoding of Cities

e) **Addition of new variables (if required)**

Addition of new variables has not been done during the pre-processing stage and can be carried out in the modeling stage to reduce the number of features and extract more information. One may compute new variables or create new features based on interaction of the other features.

4. MODEL BUILDING

For the current data-set we are dealing with, **Supervised Learning** algorithms would be appropriate. Here we model dependencies and relationships between a target prediction output and input features.

This problem also requires a **Classification** approach to solving. In classification, we create predictive models from training data which have features and class labels. These predictive models in-turn use the features learnt from training data on new, previously unseen data to predict their class labels. The output classes are discrete (not continuous).

It is not required to use all the 53 features for model building. As mentioned earlier in the document, we can remove redundant features first by using a correlation study. We can then rank the important features through a feature selection criteria. We had used **SCIKIT LEARN's Feature Selection** to come up with the 15 most important features required to model this problem and use it for model building. Some of these features are Balance Tenure, Current Interest Rate, Completed Tenure, Original Interest Rate, Product e.t.c. This would reduce the number of features initially before ranking them through feature selection.

Alternatively, other methods such as PCA or Chi-square tests can be used to do feature selection and removal. Since any method can be used for this exercise, we do not discuss PCA or chi-square methods in this document further.

We further classified the features into **dependent** and **independent** variables ('X' and 'y' respectively).

We then perform a '**TEST-TRAIN-SPLIT'** dividing the rows or data into two segments on whether they will be used for training the model or for testing the 'trained' model.

For this problem, we use **80%** of the data for training and **20%** for testing. Given that the number of rows in our data is approximately 20,000, this would mean that 16,000 rows are used for training and 4,000 records are used for testing.

For many of these models **Cross Validations** are performed. This means that an iterative process is performed to vary the test and train data within the dataset to get a more accurate model.

Other methods such as Tuning of parameters (e.g. through Grid Search), Boosting and Bagging techniques are also employed to improve model performance.

The following basic classification models are employed in this study:

1. **Decision Trees (CART)**
2. **Logistic Regression (Logit)**
3. **Linear Discriminant Analysis (LDA)**
4. **Support Vector Machine (SVM)**
5. **Artificial Neural Networks (ANN)**
6. **Gauss Naïve Bayes (NB)**
7. **K Nearest Neigbour (KNN)**

Other advanced models are used to further improve the performance and accuracy – these are:

8. **Random Forest with tuning (RF)**
9. **Bagging (with RF)**
10. **Boosting (e.g. XGBoost)**

SMOTE has also been tried on the models, but have reported or used this only for SVM since this method is particularly sensitive to data imbalances.

These models are used along with this particular data to determine which of them is most appropriate to determine default using performance metrics such as **Accuracy, Support, F1 Score, Recall and Precision**.

Graphical means of comparing the test to train performances and understanding the applicability of the models can be achieved through tools such as the **ROC curve**.

One of the first metrics we look at to understand how well our model behaves is the ROC curve. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

An ROC curve plots TPR vs. FPR at different classification thresholds.

Another metric associated with the ROC curve is the AUC. **AUC** stands for "Area under the ROC Curve." AUC measures the entire two-dimensional area underneath the entire ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

Let us take the example of the CART model – here we plot the TRAIN ROC vs TEST ROC.

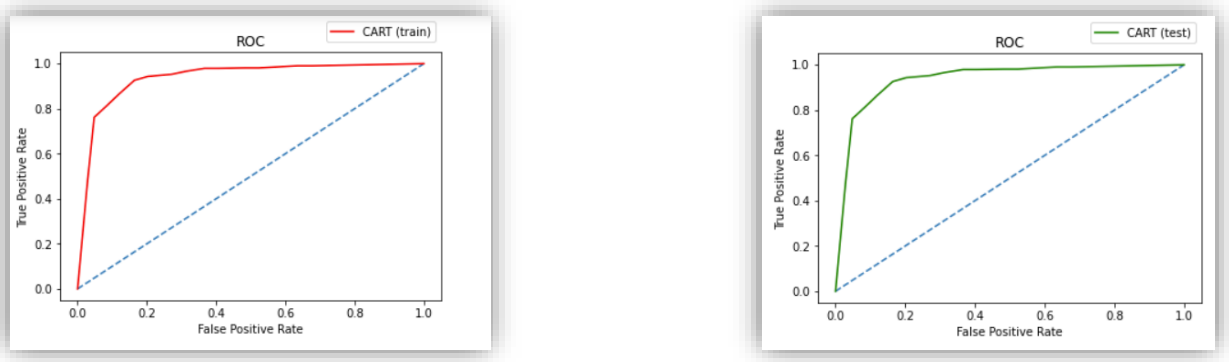

Figure 18: ROC Curves plotted for a CART (Decision Tree) Model

For the two curves plotted, the AUC scores are more or less the same, as they have approximately the same area under the curve.

	CART Train	CART Test
AUC	0.94	0.94

Figure 19: Area Under Curve (AUC) scores for Decision Trees (CART)

Similarly, we can plot ROC curves and get AUC values for other models individually as well.

For the purpose of this report, we do not plot these individually, but represent some of the relatively classic/simpler models in a single graph for Train and Test data.

In the following figure, we have plotted ROC curves in a single graph (each) to compare the model performances in either train or test data.

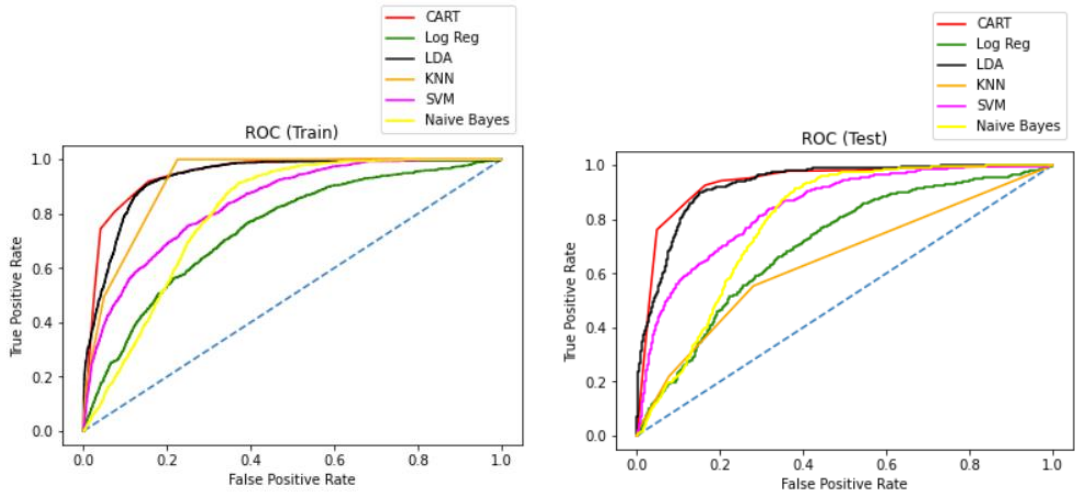

Figure 20: ROC Curves for different ML models with minimal tuning

In the above example, we see that though some curves look similar in terms of shape (such as CART), others have quite different shapes between the Train and Test values such as K Nearest Neighbours (KNN).

	CART Train	CART Test	Logit Train	Logit Reg Test	LDA Train	LDA Test	KNN Train	KNN Test	SVM Train	SVM Test	NB Train	NB Test
AUC	0.94	0.94	0.91	0.91	0.92	0.92	0.92	0.90	0.91	0.91	0.80	0.79

Figure 21: AUC scores of various models

For the models shown above, the AUC scores for test and train appear to be more or less very close to each other.

Similar for other advanced models such as Neural networks, Ensemble methods (RF) and models with tuning, boosting or bagging, we can plot the AUC scores. It is observed that some models 'OVERFIT' but they can be processed to prevent this from happening. Some ways to prevent overfitting are to work on feature selection, hyper-paramater tuning, cross-validation and other methods.

	CART Train	CART Test	Logit Train	Logit Reg Test	LDA Train	LDA Test	KNN Train	KNN Test	SVM Train	SVM Test	NB Train	NB Test	Neural Net Train	Neural Net Test
AUC	0.98	0.98	0.91	0.91	0.91	0.92	0.92	0.90	0.91	0.91	0.81	0.81	0.82	0.82
Accuracy	1.00	0.97	0.73	0.70	0.93	0.93	0.92	0.66	0.85	0.84	0.81	0.81	0.65	0.64
Recall	0.89	0.85	0.01	0.00	0.48	0.49	0.01	0.00	0.00	0.00	0.42	0.43	0.42	0.42
Precision	0.93	0.95	0.43	0.25	0.53	0.53	0.43	0.25	0.00	0.00	0.21	0.22	0.24	0.23
F1 Score	0.91	0.89	0.02	0.01	0.50	0.51	0.02	0.01	0.00	0.00	0.28	0.29	0.30	0.29

Figure 22: Metrics of different ML models (traditional)

We observe that other than CART, the rest of the models do not perform too well in terms of other metrics such as recall and precision, though AUC and Accuracy may be somewhat acceptable. These models can be further bettered. However, we do not spend too much time doing this, as we have been able to get good results using other methods such as CART (and futher through RF and XGBoost which are discussed later).

It is also observed that methods like SVM perform poorly with default conditions. This is likely due to a small imbalance in data (Only 10% of the data has 'foreclosure' indicated,

while the rest 90% is in status quo). For data imbalances, we usually apply a technique for oversampling or undersampling (as per case) such as SMOTE.

Therefore, we can **examine the performance of selected models with SMOTE**. It must be noted that SMOTE is usually used in cases where less than 5% of the data is unbalanced, since it generates pseudo-data. Therefore, we can use it for models that perform poorly, such as SVM.

	SVM Train	SVM Test
AUC	0.91	0.91
Accuracy	0.85	0.84
Recall	0.00	0.00
Precision	0.00	0.00
F1 Score	0.00	0.00

SVM Without SMOTE

	SVM Train	SVM Test
AUC	0.66	0.65
Accuracy	0.73	0.71
Recall	0.67	0.64
Precision	0.66	0.15
F1 Score	0.67	0.24

SVM with SMOTE

Figure 23: Application and improvement of SVM through SMOTE on data

We also proceed with Model Tuning and Ensemble methods as a next stage to try to obtain more accurate models.

MODEL TUNING AND ENSEMBLE METHODS

We mentioned earlier in this document that Model Tuning can help make our models better in terms of metrics. One of the methods to do this is to employ Ensemble techniques.

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. Some examples of Ensemble methods are :

a) Random Forest (from extending CART or Decision Trees)
b) Bagging (**B**ootstrap **Agg**regat**ing**)
c) Bagging

We have applied all the above three ensemble methods as individual models and evaluated their performance through metrics.

	Rand For (Tuned) Train	Rand For (Tuned) Test	Bagging(RF) Train	Bagging(RF) Test	XGBoost Train	XGBoost Test
AUC	0.99	0.98	0.94	0.09	0.98	0.98
Accuracy	1.00	0.99	0.99	0.56	0.98	0.98
Recall	0.93	0.84	0.96	1.00	0.84	0.79
Precision	0.99	0.97	0.92	0.09	0.94	0.93
F1 Score	0.96	0.90	0.94	0.16	0.89	0.86

Figure 24: Ensemble Models - Metrics

We observe that through these methods, a large change in metrics is observed when compared to the models applied in Part 1 of this document.

For our problem, Random Forest and Boosting (Extreme Gradient Boost) seem to work out the best.

Bagging seems to have issues in the test data set. This can be fixed by modifying the hyperparamters of the model. However, for this exercise, we do not tune this further in the interest of time.

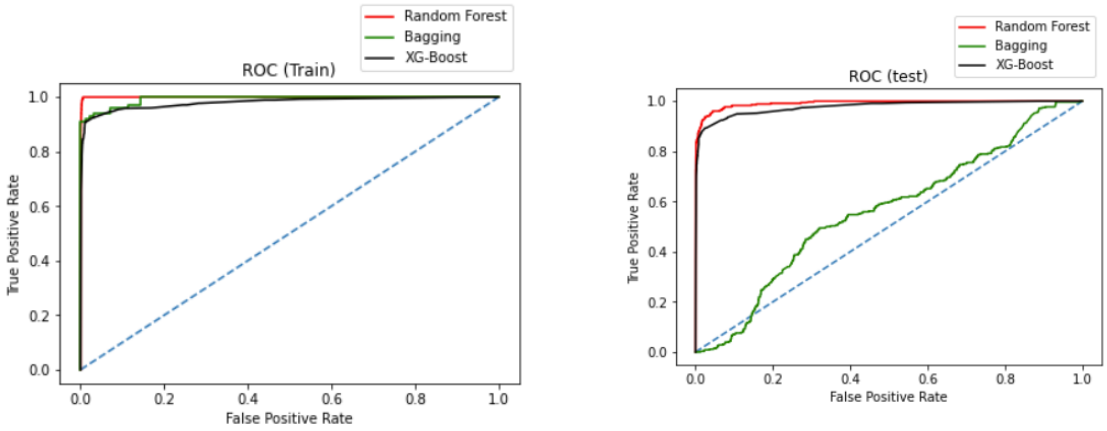

Figure 25: ROC Curves for Ensemble Models (RF, Boosting, Bagging)

From the ROC curves, we observe that we have good performance for RF and XGB, but poor test performance for Bagging. Therefore, we discard bagging for now (it is possible to make this better though, and leave this for a later time).

Below is an explanation for each of the ensemble methods :

Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest outputs a class prediction and the class with the most votes/weightage becomes the model's prediction. A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

It is important that the models have a low correlation between them. Uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. This is because the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction.

Bootstrap Aggregating (Bagging) is also an ensemble method and can be thought of as a precursor to random forest. Random samples of the training data set are created with

replacment (sub sets of training data set). Then, a model is built (classifier or Decision tree) for each sample. Finally, results of these multiple models are combined using average or majority voting. As each model is exposed to a different subset of data and we use their collective output at the end, so we are making sure that problem of overfitting is taken care of by not clinging too closely to our training data set. Thus, Bagging helps us to reduce the variance error. Combinations of multiple models decreases variance, especially in the case of unstable models, and may produce a more reliable prediction than a single model.

Boosting is an iterative technique which adjusts the weight of an observation based on the last classification. If an observation was classified incorrectly, it tries to increase the weight of this observation and vice versa. Boosting in general decreases the bias error and builds strong predictive models. Boosting has shown better predictive accuracy than bagging, but it also tends to over-fit the training data as well. Thus, parameter tuning becomes a crucial part of boosting algorithms to make them avoid overfitting. Boosting is a sequential technique in which, the first algorithm is trained on the entire data set and the subsequent algorithms are built by fitting the residuals of the first algorithm, thus giving higher weight to those observations that were poorly predicted by the previous model.

Other model tuning measures

GRID SEARCH - Grid search is a tuning technique that attempts to compute the optimum values of hyper-parameters. It is an exhaustive search that is performed on specific parameter values of a model. The model is also known as an estimator.Grid search exercise can save us time, effort and resources.

The RANDOM FOREST model, CART and other models were tuned using **GRID SEARCH** to arrive at the optimum hyperparameters. Please refer to the code to understand how this can be implemented.

CROSS VALIDATION: Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

The three steps involved in cross-validation are as follows :

1. Reserve some portion of sample data-set.
2. Using the rest data-set train the model.
3. Test the model using the reserve portion of the data-set.

To do this numerically, we iteratively can try various CV values and keep increasing the count till it makes no significant difference to the CV score. At this point, we can freeze the CV number and use it for further processing as shown in the following figure. In this example, CV values of 3,5 and 10 are taken and the train and test cross validation scores are observed. We see that by doubling the number from 5 to 10, not much of a difference is seen. So we freeze the CV value at 5.

```
3
Train CV Score : 0.9734524132911998
Test CV Score : 0.9670234171303861
5
Train CV Score : 0.9747641713307502
Test CV Score : 0.968019975031211
10
Train CV Score : 0.975513585259213
Test CV Score : 0.9680224438902743
```

Figure 26: Cross validation Scores for different values of sets/partitions

Amongst all the methods tried out for this dataset, the best methods that we arrive at are **CART, Random Forest and XGBoost**. We can discard other algorthims for this purpose even though the enjoy a a good accuracy score, as their 'other' metrics such as precision & recall are not agreeable.

5. MODEL VALIDATION

In order to choose the right model, we need to check whether :

a) The accuracy score for test and train data are good enough (without overfitting scenario)
b) The other measures such as support, F1 Score, Precision and Recall are reasonable.

Precision talks about how precise/accurate your model is out of those predicted positive, how many of them are actual positive. In our model, its important to have high precision.

Recall actually calculates how many of the Actual Positives our model capture through labeling it as Positive (True Positive). With classification problems such as this, where False Negatives are a lot more expensive than False Positives, **we may want to have a model with a high recall rather than high precision.** In our model, we see poor recall scores for most of the models carried out except for Decision Trees (CART). This seems to be a good model for our problem. We can improve this model further by Tuning.

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives). This seems to be healthy for the CART model.

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. In our traditional models, CART seems to be the best model again here, since it shows a healthy support score.

I**n the case of predicting if a loan would default/foreclose — it would be better to have a high Recall and Accuracy as a priority amongst all metrics** as the banks don't want to lose money and it would be a good idea to for the bank to be conservative and not release the loan if there is a slight doubt about a default.

Low precision, in this case, might be okay.

In advanced or ensemble models, both RANDOM FOREST and XGBOOST models exhibit good RECALL values, as well as healthy in other metrics. Given a choice, one can go for the RANDOM FOREST algorithm as it performs well on all metrics, on both test and train data.

6. RECOMMENDATIONS/ FINAL INTERPRETATION

Through this exercise, we have demonstrated the rapid manner in which financial health of loans can be analyzed and predicted for a potential foreclosure through machine learning and advanced analytics.

When compared to manual methods that are empirical or prone to human error, such automated methods improve confidence in decision making and allow the financial institutions to make quick judgements on how to proceed with a particular loan in a given situation. Such methods when implemented in the risk management process of NBFCs can significantly improve the response to critical situations.

Analytics dashboards based on a single data platform can be set up at various levels – from the executive management level to the customer service executives. At the top level, the dashboard would show macro level metrics

As seen from the Data Analysis in this document, the number of loans and activities of the NBFC are increasing exponentially year-on-year. With increasing number of records in rapid time, financial institutions must be ready to take on analytics on big data scale. These techniques while applicable to loan foreclosure prediction, can be extended to other functions of the financial institution such as assessing new loan applications based on metrics that go beyond the traditional approaches.

Its therefore vital to carry out a digital transformation of the risk management process and incorporate such financial technology to ensure sound financial health with least delays. As a first step, the NBFC could approach consulting companies to study their existing system and further set-up a pilot study to measure the effectiveness of switching to a process with advanced analytics. Once the benefits are established, advanced analytics can be incorporated for as a key step in managing foreclosure risk.

In addition to risk management, the same analysis could be leveraged to improve customer traction by encouraging customers in the low risk categories to opt for or upsell other types of loan products or offerings.

Data Analysis reveals that most number of foreclosures come only from two products – HL And STHL which are likely to be home loans with relaxed collaterals. LAP and STLAP which appear to be loans which are provided against property (as a collateral) appear to have minimal foreclosures. The NBFC therefore has to look into the former two products and peform research on how to reduce the foreclosures in those two products.

	Product			
FC	HL	LAP	STHL	STLAP
Open Loan	2,492	6,224	6,465	3,036
Foreclosed	990	2	803	

APPENDIX

TRADITIONAL MODELS – METRICS WITH AND WITHOUT SMOTE

Without SMOTE

	CART Train	CART Test	Logit Train	Logit Reg Test	LDA Train	LDA Test	KNN Train	KNN Test	SVM Train	SVM Test	NB Train	NB Test	Neural Net Train	Neural Net Test
AUC	0.98	0.98	0.91	0.91	0.91	0.92	0.92	0.90	0.91	0.91	0.81	0.81	0.82	0.82
Accuracy	1.00	0.97	0.73	0.70	0.93	0.93	0.92	0.66	0.85	0.84	0.81	0.81	0.65	0.64
Recall	0.89	0.85	0.01	0.00	0.48	0.49	0.01	0.00	0.00	0.00	0.42	0.43	0.42	0.42
Precision	0.93	0.95	0.43	0.25	0.53	0.53	0.43	0.25	0.00	0.00	0.21	0.22	0.24	0.23
F1 Score	0.91	0.89	0.02	0.01	0.50	0.51	0.02	0.01	0.00	0.00	0.28	0.29	0.30	0.29

With SMOTE

	CART Train	CART Test	Logit Train	Logit Reg Test	LDA Train	LDA Test	KNN Train	KNN Test	SVM Train	SVM Test	NB Train	NB Test	Neural Net Train	Neural Net Test
AUC	0.99	0.97	0.91	0.91	0.90	0.85	0.92	0.90	0.66	0.65	0.76	0.75	0.54	0.90
Accuracy	1.00	0.96	0.73	0.70	0.95	0.94	0.92	0.66	0.73	0.71	0.82	0.81	0.55	0.54
Recall	0.99	0.89	0.01	0.00	0.95	0.91	0.01	0.00	0.67	0.64	0.77	0.74	0.10	0.10
Precision	0.99	0.82	0.43	0.25	0.87	0.36	0.43	0.25	0.66	0.15	0.76	0.22	0.85	0.31
F1 Score	0.99	0.85	0.02	0.01	0.90	0.52	0.02	0.01	0.67	0.24	0.76	0.34	0.18	0.15

ENSEMBLE MODELS – METRICS WITH AND WITHOUT SMOTE

Without SMOTE

	Rand For (Tuned) Train	Rand For (Tuned) Test	Bagging(RF) Train	Bagging(RF) Test	XGBoost Train	XGBoost Test
AUC	0.99	0.98	0.94	0.09	0.98	0.98
Accuracy	1.00	0.99	0.99	0.56	0.98	0.98
Recall	0.93	0.84	0.96	1.00	0.84	0.79
Precision	0.99	0.97	0.92	0.09	0.94	0.93
F1 Score	0.96	0.90	0.94	0.16	0.89	0.86

With SMOTE

	Rand For (Tuned) Train	Rand For (Tuned) Test	Bagging(RF) Train	Bagging(RF) Test	XGBoost Train	XGBoost Test
AUC	0.99	0.98	0.94	0.09	0.98	0.98
Accuracy	1.00	0.99	0.99	0.56	0.98	0.98
Recall	0.93	0.83	0.96	1.00	0.84	0.79
Precision	0.99	0.97	0.92	0.09	0.94	0.93
F1 Score	0.96	0.89	0.94	0.16	0.89	0.86