Application Scorecard Development for Mortgage Products Using Different Statistical Modelling Techniques

Dissertation

Submitted as the requirements for the degree of

Master of Technology

By

Sushant Kulkarni

Abstract

With an unprecedented increase in competition in the credit lending space it has become very important to select the most creditworthy applicants in a short time frame. Hence, it is no longer possible to adjudicate the application by scanning the demographic data, liability data and the bureau report of the applicant manually. Analytics is playing a very crucial role in automating the entire process of adjudication and in meeting the turnaround time for the customers. On the marketing side, it has also become very important to retain existing customers as well as target new customers to increase the market share of the institution by offering new, exciting and attractive offers, where again analytics is playing a crucial role.

This particular article deals with the entire process of development of application scorecard for the mortgage products of the bank. It explains in detail different stages of scorecard development in general, from understanding the product from business point of view to data collection, cleaning, authenticating to fitting the model using analytical tools and to finally implementing the scorecard. It also talks about the basic theory of scorecard development supported by the practical interpretations using different statistical measures and intuitive experience of the scorecard developer.

Acknowledgement

I take this opportunity to express a deep sense of gratitude towards my industry guides **Ms. Samanwita Maity** and **Ms. Shreyashi Ganguly,** and academic guide **Prof. Sarada Samantaray**, for providing excellent guidance, encouragement and inspiration throughout the project work. Without their invaluable guidance, this work would never have been a successful one. I would also like to thank all my colleagues and professors for their valuable suggestions and helpful discussions.

Sushant Kulkarni

M.Tech Data Science

NMIMS

Contents

Introduction:

Application Scorecard Development

With increase in competition and growing pressures for revenue generation, it has led to increase in credit granting and search for some effective ways to attract new creditworthy customers and at same time control losses. The aggressive marketing strategy have resulted deeper penetration of the risk pool of potential customers and need to process them as fast as possible. This has led to rapid and effective credit application and adjudication which in turn has cause automation of the credit application process. The challenge for Risk Manager to produce risk adjudication solutions that can assess the creditworthiness of the applicant satisfactorily, keeps the per-unit processing cost low and reduces the turnaround time for the customers, has increased a lot. Additionally, the automation model or process should be such that it should minimize the denial of credit to the creditworthy customers, while keeping away the delinquent customers as possible.

With respect to the customer management level the companies are striving very hard to maintain their relationship with existing customers by providing attractive additional offers and enhance services. Risk Managers are called in for help to select the "right" (i.e. low risk) customers and reject customers who show negative behaviour (i.e. fraud or non-payment) from various characteristics analysed. Risk Managers are expected to devise strategies to minimize risk and to maximize the profit.

The risk scorecards offer a very powerful and empirically derived solution to business needs. The risk scorecards are widely used in industry for predicting delinquency (i.e. non-payment), bankruptcy i.e. fraud and collections i.e. recovery of NPA amount owed to bank.

Earlier the credit risk scorecards were prepared obtained made outside bank from certain vendors. This had a very high chance of data breach, since data was provided to vendors for model development. Nowadays many big financial institutions develop the risk scorecards internally. Also, due to availability of different analytical and machine learning tools scorecard development has become very easy and do not require core programming expertise. Different analytical coding tools such as SAS, R and Python can be used by model developer. Different machine learning tools like SAS-Enterprise Miner, Microsoft Azure, Certain libraries in python and R can be used by model developer.

This article majorly concentrates on model development procedure done sing SAS Enterprise Guide and SAS Enterprise Miner. This project is to develop Application Scorecard for the mortgage products of the bank. The scorecard is a grouped one, i.e. the raw variables are grouped and then provided as input to the scorecard. The application scorecard is used to score every application based applicant's personal data, bureau credit information report, transaction level data (in case of existing current or savings account customers) and demographic data. After scoring every application is either accepted or rejected based on the score. The application scorecard acts as an adjudicator or underwriter who assesses the creditworthiness of the applicant and provides the final decision whether to accept the applicant or reject it. The model development process starts with preparing certain questionnaire to ask the risk team and business team on the basis of initial exploratory analysis to know about the mortgage product.

Once the developer is well equipped with all the necessary information regarding the business and has thorough knowledge about the product, the developer can start with model development steps. Initially basic analysis such as roll rate and vintage analysis is performed and then the bad definition, performance window and sample window is decided. The variable level univariate analysis and the distribution on year on year basis are performed to

check on the quality of data and understand the variables. The next most important task is variable reduction and selection of variables which includes shortlisting all the strong variables which will finally be entered into the model. Once the final set of strong variables are finalised, the modeller can further procced to fit any of the classification or machine learning techniques like logistic regression, boosting algorithms, neural networks, etc. This again is a very critical stage which requires a lot of human expertise in deciding the best scorecard to implement based on the statistical metrics such as Gini, KS, etc. and the variables in the model making business sense.

As we have briefly understood the process of creating a model in earlier section. The model is actually prepared in different stages; each of these stages acts as the basis to the next stage. This process of model development is quite iterative and model developer needs to go back and forth asper the recommendations of the stake holders. All the above stages are used to develop a model on the accept base which is called as "Known Good/ Bad" model. This model becomes a benchmark model for the next stage of the "All Good/ Bad" model which is developed on the through the door population. The performance of the AGB model should be always better than the KGB model as it consist of through the door population with inferred bad either from bureau delinquency information or from other reject inference techniques used. AGB model is developed over the accepts as well as rejects, hence it will have a very good predictive and classification power as compared to KGB model. After finalising the final AGB model the risk rankings and Gini lift charts are prepared and analysed to validate the performance of the model and finally the AGB model is implemented as scorecard. Also the exercise of retro scoring, swap-set analysis and year on year analysis of GINI is done to double check on the performance and confirm regarding the robustness of the model.

The scorecard is implemented on different platforms in different organisations. It all depends on the in house infrastructure available with the bank. After the implementation of the model

the risk and the business teams perform retro scoring and cut-off analysis to decide the cut-off based on the risk rankings. Once the cut-off is set the adjudicators will accept the application above the cut-off value and reject below the cut-off value. The final note is then prepared and is send across to the stake holders for the sign off.

Chapter 2

Data Review and Project Parameters

2.1 DATA AVAILIBILITY AND QUALITY

The most common problem faced by the financial organizations is regarding the data availability. This is in the context of the quality and quantity of the data. For the scorecard development, reliable and clean data is needed with the minimum number of "goods" and "bads". This problem can be tackled by development of data marts or data warehouses.

The quantity of data needed for the scorecard development varies, but in general the numbers should be sufficient enough to fulfill statistical significance and randomness. However, for the development of scorecard, as a rule of thumb approximately 2000 "bad" accounts and 2000 "good" accounts can randomly be selected from the base of approved accounts within a defined time frame. Whereas for the application scorecard, from the base of rejected applications, more 2000 applications are added to the base which would be further required for the reject inference.

There are different sources of data which are at the disposal for the development of the model. The data obtained from credit bureau (for example CIBIL or Experian), the liability data i.e. transaction level data, balance related data of the existing customers of the organization and the data of the individual having different crossholdings with bank (such as insurance, mutual funds, etc.) and the demographic data (i.e. application level data). Usually, the data obtained from credit bureau is of good quality and reliable. Also, the transaction level and crossholding data obtained in-house is of good quality due to proper warehousing. The application level data

obtained from the application form is much needed to be verified and its authenticity is needed to be confirmed from the policy and sales team. The income related information and the product specific information in secured products (e.g. Auto loan or home loan) such as area size or the mortgage price is usually verified and then digitized. Demographic data in the organizations upto some extent is digitized which can be used; some more demographics such as real estate, geographical information, individual's organizational information can be used.

At the end of this phase, when it is determined that both internal and external data sources are evaluated, quantified and defined, we can begin with gathering initial data for defining project parameters.

2.2 DATA GATHERING FOR DEFINITION OF PROJECT PARAMETERS

In order to develop an application scorecard, the data must be gathered in form of a database which can be further used in any software development environment. Primarily, project parameters include decision regarding "good" and "bad" definition, the performance and sample windows and exclusions in the development sample.

The following data for development is usually collected for last two to five years from the large database:

- Unique identification number (Account number or Application id)
- Date of account opening and Application date
- Underwriting status (whether accept or reject)
- Product and segment identifiers

The secondary objective is to understand the business and the product for which the scorecard is to be developed from different teams. This step will help in collection of new demographic variables and create a good understanding for creation of new derived variables.

2.3 DEFINITION OF PROJECT PARAMETERS

The following analyses are done to define the project parameters and to understand business through data.

2.3.1 Exclusions

In general, the development sample should include only those accounts which would be scored in normal day-to-day credit granting and would consist of the intended customers. Accounts which have abnormal performance or fraud accounts or those which are approved using some non-score-dependent criteria should not be part of development base. Depending on the product for which the application scorecard is to be developed, certain accounts such as staff, preapproved, NRI accounts, VIPs, lost or stolen cards an voluntary cancellations should be removed.

Another way to understand exclusions is to overcome the sample bias issue. For example, if the scorecard is to be developed for certain product or group of individuals, we should only include those kinds of individuals. This process will also help to get rid of all the outliers up to certain extent, and would help to have an unbiased and normal data. This would in turn help to get proper predictions and robust model.

In this Home Loan model we have excluded all the non-individuals. The non-individuals like public firms, partnership firms, private firms, etc. are to be excluded and only individuals such as

Indian residents, NRIs, sole proprietorship firms. The decision is taken after constant discussions with the product and policy team. As when the application scorecard would be implemented it would only be used to score the individuals.

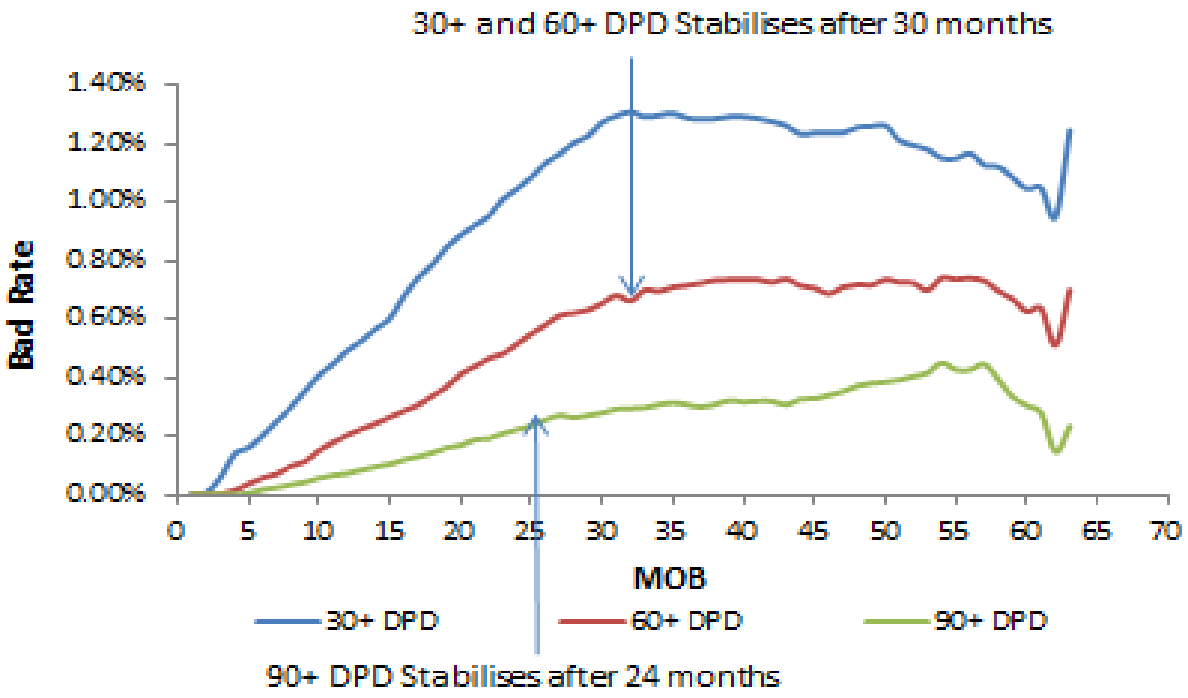
2.3.2 Performance and Sample Windows and "Bad" Definition

The scorecards are developed with a basic assumption that "Future performance will reflect past performance". Hence, in order to predict the performance of the future accounts, the performance of the previous accounts is analyzed. For this purpose, the accounts opened in a specific time frame are being analyzed and their performance is monitored for another specific time period to determine whether they can be defined as good or bad.

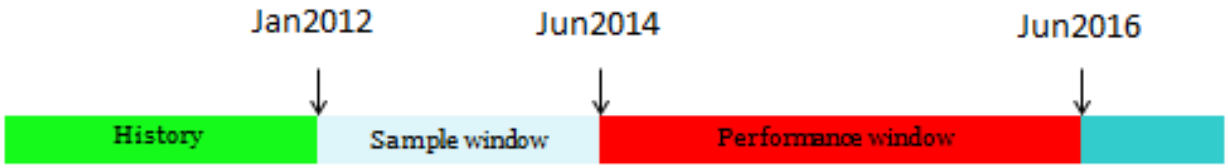
The time window where the performance of accounts opened during a specific time frame (i.e. the sample window) is being monitored to define the class (target i.e. good or bad) is called the "Performance Window".

Whereas, the time window from which the known good and bad cases are selected for development is called as "Sample Window".

The best way to define the performance and sample window is to analyze the payment or delinquency of the portfolio with a certain "bad" definition (e.g. 90 days delinquency). The method which is used to define the performance window is called as "Vintage Analysis". In vintage analysis, the bad rate with certain bad definition (e.g. 90 days delinquency) is plotted against the month-on-book.



This chart shows the stability of the portfolio. The chart shows that the bad rate with respect to month-on-book. From the chart we can see that the bad rates for ever 90 plus days past due gets stable after 24 months and the bad rate starts to level off. The bad rates for 30 plus days past due and 60 plus days past due gets stable at 30 months. Since mostly for all models the bad definition used is ever 90 in 24 months for this project also we consider the same for operational and business ease.



The above image shows the timeframe selected for the sample window and performance window. The green block and white block shows the total applications of the products till June

2014.The white block specifically shows the sample window i.e. the applications to be considered in the development base which is from January 2012 to June 2014. The next block i.e. red block shows the performance of the applications seen in the sample window; the performance is seen on rolling window basis for 2 years for all applications as per the bad definition discussed in next section. The performance is being observed for all applications till June2016.

2.3.3 Definition of "Bad"

This phase categorizes performance of an account into three groups: "Bad", "Good" and "Indeterminate". The definition of bad is straightforward for charge-off, bankruptcy or fraud. For different definitions of "bad", we can get different count of "bad" accounts.

Using some factors below, we can define some bad definitions:

- The definition is defined base on the organizational objectives. If the organization wants to increase the profitability at the cost of risk then the definition of risk can be set to point of delinquency. If delinquency detection is the aim, then there would be much simpler definition (e.g., ever 60 or ever 90).
- The definition depends on the product and objective of the scorecard.
- A "looser" definition which can be 30 days delinquent, where in we can get more number of bad account count, but this definition may not be good differentiator between good and bad accounts.
- A "tighter" definition which can be write-off or 120 days delinquency, where in we may get comparatively less number of bad account count, but can provide more extreme and precise definition.

- In some cases the organization may select definition based on accounting policies on write-offs.

  Choosing a simple bad definition makes it easier for the management and decision making. The definitions like ever90 or ever60 or ever90 in 24months or ever90 in 12months can be considered as the simple bad definition for the scorecard development.

- It is always beneficial with perspective of the organization to maintain same bad definition for the scorecards across the products and segments, for easier management, development and programming of the scorecards.

In this project, the definition is decided to be

- Ever 90 in 24 months to be bad, i.e. if an account goes 90 days delinquent in 24 months of the performance window then it is termed to be bad account.
- 30 to 89 days past due in first 24 months or 90 days past due post 24 months of the performance window are termed to be indeterminate.
- Those accounts who have been less than 30 days past due and less than 89 days past due post 24 months are termed to be good.

2.3.4 Confirming the "Bad" Definition

Once the initial "bad" definition is decided using the analysis done in the last section, this section emphasizes upon confirming the same definition. Different types of consensus and analytical method are used to check whether the accounts which are classified as bad are truly bad.

- **Consensus Method**

  This method involves various stakeholders from Risk, Marketing, Policy and other Operational teams, who based on the past experience and operational conditions. As well

as considering the past analysis, come to the judgmental or consensus decision whether to continue with the same definition or to change it.

- **Analytical Method**

  There are two types of analytical method to confirm the "bad" definition:

  1. Roll rate analysis
  2. Current versus worst delinquency comparison

Roll Rate analysis

Roll rate analysis involves comparing the worst delinquency amongst the performance window.

Comparing the "pervious x" months of delinquency with the "next x" months of delinquency, and calculating the percentage of accounts that maintain their worst delinquency, get better i.e. "roll backward" into previous delinquency bucket, or "roll forward" into next delinquency bucket.

Primarily, roll rate analysis is done to find out the "point of no return" i.e. the point at which the accounts roll forward to next delinquency bucket and never comeback. Typically, accounts that get into 90 days delinquency bucket never roll back and, they might roll forward and get even worse.

According to Basel II Accord, "default" is defined as any point at which bank feels that the obligor is unlikely to repay the loan in full, and specified 90bdays past as a definition (which can be changed by regulator upto 180 for certain products).

The roll rate analysis can be interpreted either from table or from stack plot chart. Both of these approaches involve comparing the maximum delinquencies between two fixed timeframes. Given below are the examples of the two approaches:

Table based approach

Max Delq.in 1st 18m	Max Delq.in next 12m					
	current	X	30	60	90	120+
current	96.20%	3.23%	0.37%	0.14%	0.03%	0.04%
X	69.03%	22.71%	5.22%	2.07%	0.38%	0.59%
30	27.21%	17.47%	28.59%	19.09%	3.17%	4.47%
60	14.42%	7.21%	10.93%	36.74%	13.26%	17.44%
90	6.25%	7.50%	5.00%	20.00%	26.25%	35.00%
120+	7.91%	3.39%	2.82%	1.69%	1.13%	83.05%

The roll rate analysis can be interpreted from the table above in which we are comparing the maximum delinquency of first 18months with next 12 months (i.e. performance window is of 30 months):

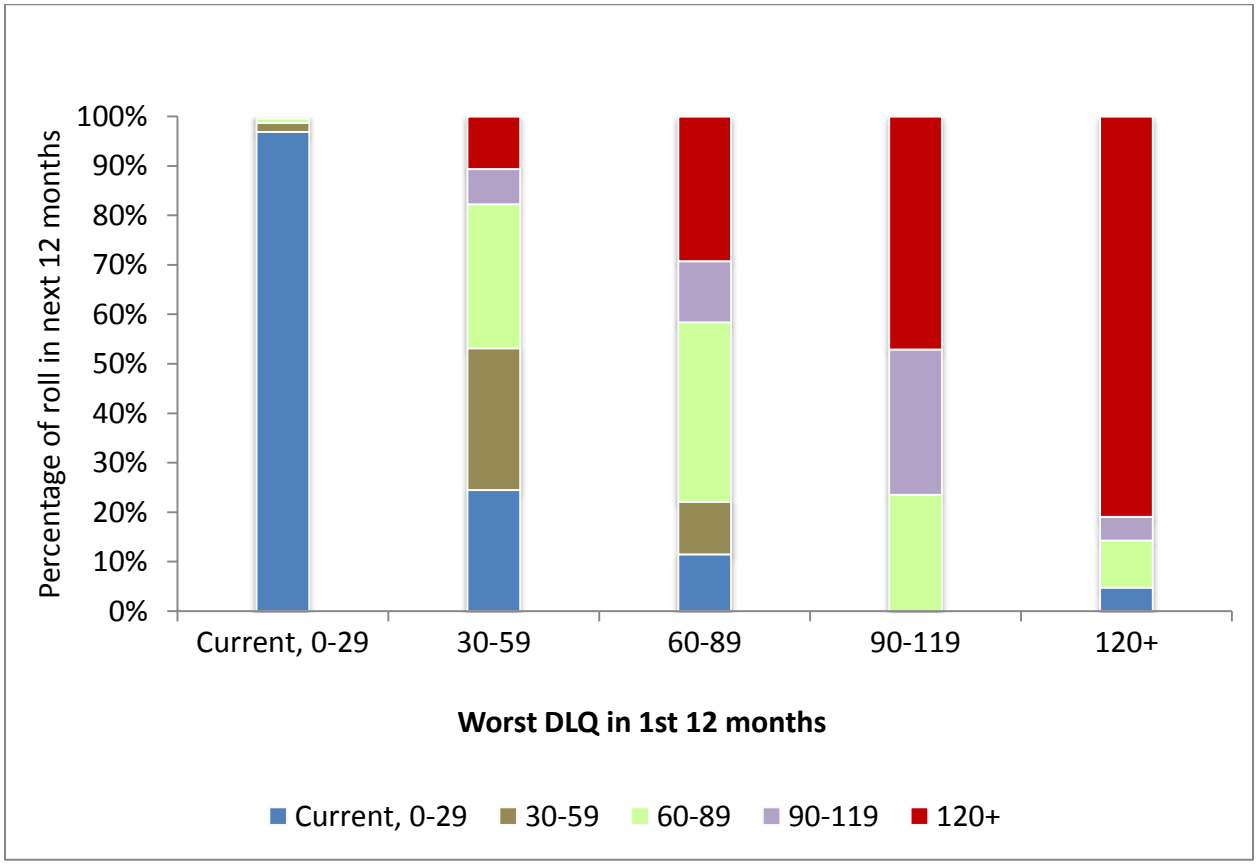
1. Out of total accounts which were current in first 18 months 96.2% of the accounts still remain in current in next 12 months, but 3.23% roll forward in X, 0.37% roll forward in 30 days past due bucket, 0.14% roll forward in 60 days past due bucket, 0.03% roll forward in 90 days past due bucket, 0.04% roll forward in 120 plus days past due bucket in next 12 months.

2. Out of total accounts which were X in first 18 months 22.71% of the accounts still remain in X in next 12 months, but 5.22% roll forward in 30 days past due bucket, 2.07% roll forward in 60 days past due bucket, 0.38% roll forward in 90 days past due bucket, 0.59% roll forward in 120 plus days past due bucket in next 12 months. The 69.03% of total accounts in first 18 months roll back in next 12 months.
3. Out of total accounts which were 30 days past due bucket in first 18 months 28.59% of the accounts still remain in 30 days past due bucket in next 12 months, but 19.09% roll forward in 60 days past due bucket, 3.17% roll forward in 90 days past due bucket, 4.47% roll forward in 120 plus days past due bucket in next 12 months. The 27.21% and 17.47% roll back to current and X bucket respectively in next 12 months.
4. Out of total accounts which were 60 days past due bucket in first 18 months 36.74% of the accounts still remain in 60 days past due bucket in next 12 months, but 13.26% roll forward in 90 days past due bucket, 17.44% roll forward in 120 plus days past due bucket in next 12 months. The 14.42%, 7.21% and 10.93% roll back to current, X and in 30 days past due bucket respectively in next 12 months.
5. Out of total accounts which were 90 days past due bucket in first 18 months 26.25% of the accounts still remain in 90 days past due bucket in next 12 months, but 35% roll forward in 120 plus days past due bucket in next 12 months. The 6.25%, 7.50%, 5 and 20% roll back to current, X , 30 days past due bucket and 60 days past due respectively in next 12 months.

6. Out of total accounts which were 120 plus days past due in first 18 months 83.05% of the accounts still remain in 120 plus days past due bucket in next 12 months. The 7.91%, 3.39%, 2.82%, 1.69% and 1.13% roll back to current, X , 30 days past due bucket, 60 days past due and 90 days past due respectively in next 12 months.

In this approach 90 days past due is the point of no return, since 35% of the applicants move forward in 120 plus days past due and very less applicants come back to the lower buckets.

Chart based approach



This approach is also similar to the earlier approach and can be interpreted by comparing the maximum delinquency of last 12months with next 12 months (i.e. the performance window is of 24 months) as:

1. Out of the total applicants approximately 96% of the applicants stay in the same bucket of current and 0-29 days past due in next 12 months as compared to last 12 months, as seen from chart and rest forward can be analyzed from the chart.
2. Out of the total applicants approximately 23% of the applicants roll backward to current and 0-29 days past due, 27% stay in the same bucket of 30-59 days past due, 26% roll forward in the next bucket of 60-89 days past due, 9%roll forward in next bucket of 90-119 days past due bucket and 15% in the next bucket of 120 plus days past due bucket.

This can be continued further for all the buckets and we can get the point of no return. In the above chart, we can see that 90-119 days past due bucket seems to be point of no return since large proportion of applicants have moved forward to the 120 plus bucket and there are very few applicants moving back to earlier buckets.

The initial approach of table analysis is much better to analyze since we can compare the actual numbers and is easier to interpret and understand. But the second approach is more over a visual approach but sometimes it becomes difficult to interpret and understand.

2.3.5 "Good" and "Indeterminate"

Once the "bad" accounts are defined same kind of analysis can be done to define "good" accounts. Again the good definition should be as per the organizational decisions as discussed earlier. Defining the good definition has involved less analytical exercise as the definition is

usually but obvious. Usually 0 to 29 days past due is considered to be good definition. Some characteristics of a good account:

- Never delinquent or delinquent to a point where forward roll rate is less
- Positive NPV or profitable
- Never bankrupt
- No fraud

While the good accounts need to retain their status till throughout the performance window, the bad accounts can be tagged so whenever they reach the point of specified delinquency stage. This definition is as per the "ever" definition. The indeterminate accounts are the ones which are neither good nor bad i.e. they are the grey area, neither white nor black. Consideration of these accounts in the development base will lead to the confusion in classification of the target by the model, as there would be some accounts with mild delinquency which are neither fully good but are slightly bad. Indeterminate accounts can be:

- Accounts which are neither in 0 to 29 days past due bucket nor in 90 plus days past due bucket i.e. the accounts which lie in 30 to 89 days past due bucket are considered to be indeterminate.
- The accounts which are inactive or which are pre-closed or taken for short period can also be considered as indeterminate.

Some Scorecard Developers usually do not consider the loan not taken up cases and reject cases in the development base. But these cases should be considered since excluding then may lead to huge loss of data. Hence these cases of loan not taken up are considered as indeterminate by some developers.

Adding of the indeterminate to the model development base may lead to confusion to the model for classification and result in huge misclassification. The indeterminate are considered in the analysis only when we are analyzing through the door population.

2.4 SEGMENTATION

In some cases, using different scorecards for scoring the applications provides better risk differentiation than using a single scorecard on all applications. This usually happens when the base consists of applications with different level of risks and different characteristics which will create different sub-population. The process of identification of this sub-population is called as segmentation. The segmentation can broadly be classified into two ways:

1. Generating segmentation ideas based on experience and industry knowledge, and then validating these ideas using analytics.
2. Generating unique segments using statistical techniques such as clustering or decision trees.

In both the cases the population should be divided into sub-population such that each section should consist of some finite number of applications and considerate bad counts to develop a mode, so that the model can efficiently predict the target.

Segmentation, whether using experience or statistical methods, should also be done with future plans in mind. Most analysis and experience is based on the past, but scorecards need to be implemented in the future, on future applicant segments. One way to achieve this is by adjusting segmentation based on, for example, the organization's intended target market. Traditionally, segmentation has been done to identify an optimum set of segments that will maximize performance—the approach suggested here is to find a set of segments for which the

organization requires optimal performance, such as target markets. This approach underscores the importance of trying to maximize performance where it is most needed from a business perspective and ensures that the scorecard development process maximizes business value. This is an area where marketing staff can add value and relevance to scorecard development projects. The Basel II Capital Accord defines segments as "homogeneous risk pools". This provides authority to banks to define the segments asper the perception of the banks.

2.4.1 Experience-Based (Heuristic) Segmentation

Heuristic segmentation is being done on the basis the business knowledge, experience and operational ideas. This can include the insights from the Risk and Marketing team who usually keep on detecting risk on profiles in a specific segment. It may also depend on the future marketing strategies and new product development. We also need to consider some predefined groups differently. Typical segmentation areas used in industry can include:

- **Demographics**

  Regional (i.e. state, pin code, etc.), age, vintage on bureau, vintage with bank

- **Product Type**

  Different types of cards, products and sub-products (e.g. in mortgage we can have Home Loan and Loan Against Property, both come under the umbrella of mortgage products but two different scorecards can be prepared base on the volumes and the bad counts for these sub-products)

- **Sources of Business**

  Loan taken from branch, agents, dealers, etc.

- **Applicant type**

  Existing or New customer to bank, profession, self-employed or salaried, Vintage on Bureau (i.e. hit with bureau or not)

- **Product Owned**

  One with mortgage loan applying for credit cards at same bank

2.4.2 Statistically-Based Segmentation

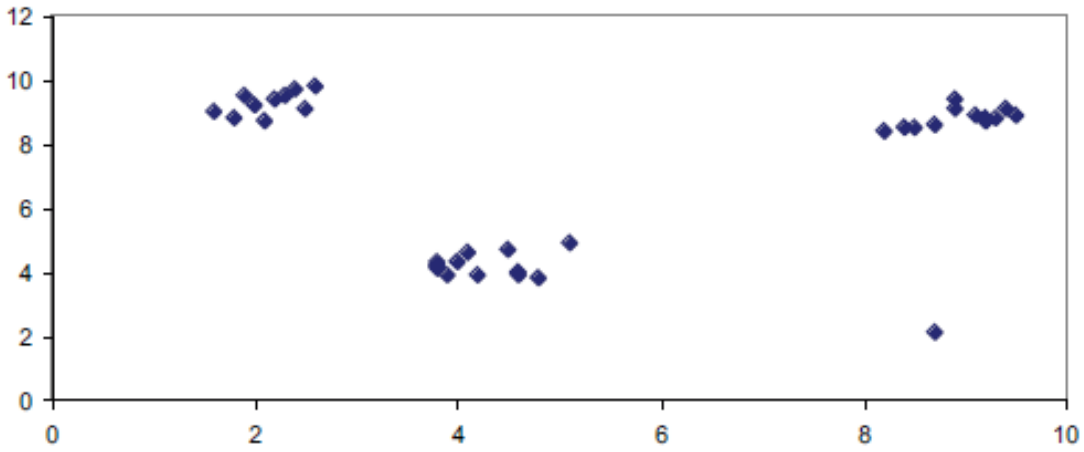
There are mainly two techniques used for statistical-based segmentation:

- **Clustering**

  This technique is widely used in the industry to find out the segments in absence of the heuristic approach. Various clustering techniques are used for segmenting the different observations into clusters depending on the characteristics of each applicant. Thus all the applicants belonging to one cluster will certainly have same characteristics, but the applicants belonging to clusters will have different characteristics. Hence the observations within a cluster are homogeneous while observations across different cluster are heterogeneous. The methods of clustering used are K-means clustering and Self-organizing maps (SOMs).

  In the K-means method the clustering is performed based on the Euclidean distances computed based on the quantitative variables and the observations are divided such that each observation belong only to one cluster.

  In the SOM the observations are mapped based on the spatial relations between the characteristics of observations. The different clusters formed on the spatial region resemble similar characteristics. This method also helps us to know about any outliers.



The above image shows the different clusters formed on the chart based on the similar characteristics, whereas an outlier can be seen on bottom right hand side of the image.

- **Decision Trees**

  This technique is used to define different segments based on the performance criteria (i.e. differentiate between "good" and "bad"). The decision tree technique is used to isolate the segments based on the performance. This is as simple to understand and interpret technique. It also helps to find out optimum breakpoints for each characteristics, hence is as very powerful and convenient method for segmentation. The segmentation based on bad rate for existing/new customer, vintage on bureau, etc. can be done efficiently.

2.4.3 Choosing Segments

After developing the segmented scorecards it is very important and critical decision whether to implement the scorecard or not. The factors involved to consider in implementing scorecards are:

1. Cost of development
2. Cost of implementation
3. Processing
4. Strategy development and monitoring

Pertaining to these problems it is only profitable to implement the scorecard if the performance of the model is significant. Hence the scorecard with a good improvement in the performance and predictive power is only implemented.

Chapter 3

Development Database Creation

3.1 DEVELOPMENT SAMPLE SPECIFICATION

Once the parameters, segmentation and methodology for the scorecard development is finalized in the preceding stage, following are specified and documented:

- Number of scorecards required and specification of each segment, and how to identify various segments
- Definitions of “good”, “bad” and “indeterminate”
- Portfolio bad rate and approval rates for each segment
- Performance and Sample windows
- Definitions of exclusions

3.1.1 Selection of characteristics

The selection of characteristic in the development base is the most important and critical task of the model development process. The characteristics should be selected various factors which are:

- **Expected Predictive Power**

  This information of predictive power comes from alot of experience and collection of information from the various stake holders from business, risk teams and monitoring.

- **Reliability and Robustness**

  Some variables are obtained from the demographic information collected from the applicants and there exists doubt about the authentication of the data since it is not verified by the authentic documents, hence these variables cannot be relied upon for consideration in the model. Whereas some variables are too data specific and may not be constant over time. For example, considering pin code or region in the model may affect the model as the risk related to the region may be data and time specific and may not be constant, thus the variable will not be robust and may flip over time after the implementation. The robustness and reliability can be confirmed from the year on year charts and by confirming with the operations and risk team.

- **Interpretability**

  The demographic variables always have the problem of authenticity and robustness. The nominal or the categorical variables are always the one's which face these problems. These demographic variables are binned in groups based on the risk. The trend of risk over these bins is considered by intuitive trends. But in some cases the intuitive trend decided by the model developer would be different from that expected by stake holders and may lead to conflict over the variable. For example, the variable like age, the intuitive trend expected is that the low age group (i.e. less than 20) and elder age group (i.e. greater than 60) should have higher bad rate. Since low age group must not have stable source of income and the elder age group must have either retired from job or would have very less time left on job and are nearing retirement. But the business team may say that from their experience these guys may have another source of income and have low bad rate on other products so they can be given loan. Hence, the variables to be considered while modelling should be such that

they should be interpretable and robust; they should not flip after the implementation of the model.

- **Creation of Ratios Based on Business Reasoning**

  The raw variables can be used to create some ratio variables which can capture information of more than one variable in one single variable. These liability and bureau variables can be used to create ratio variables. For example, Proportion of credit amount in last 6 months to the average balance in last 6 months. This variable captures the information regarding credit amount and the balance in the account in one single variable. Many times there are very few variables coming up in model, hence in above example may be only average balance or credit amount would be there in the scorecard. But if we create a proportion variable we can get the information of both variables by only one variable in the scorecard.

- **Future Availability**

  Before using any characteristic in the model it should be ensured that the same variables will be captured and maintained in the database in future.

- **Changes in Competitive Environment**

  There are some characteristics which are not strong at time but may turn to be good predictors in coming future. Such kind of variables should be considered in the model. For these variables the modeler should consult the operations team and carry out some year on year analysis to ensure its power of prediction in future.

Now, we know that we need to use the past three to four years of data for the model development and we expect the model to be working efficiently for next two to three years. Hence, we need to analyze the trends of the variables over the years to ensure its stability and robustness of the variables. The liability and bureau variables are expected to have a monotonic trend, whereas the nominal and categorical variables should follow the intuitive trend as expected by the modeler.

3.2 SAMPLING

This step includes decision for two tasks, which splitting the sample base into development and validation, and deciding the proportion of goods, bads and rejects to be included in the total samples.

3.2.1 Development/Validation

The total sample is firstly split into development base and out of time base. The out of time base should be of 3months minimum, to test the performance of the model. Since the whole process of model development takes a quite long time we can extend our out of time base, constraint being we have the sufficient performance window of those applications. The development base is further split randomly into 70% and 30% or 80% and 20% of training and validation sample respectively, with similar proportion of bad counts in training as well as in validation base. The scorecard is developed on the training base and then it is tested for its performance on the validation and out of time base. The model goes through its first test of performance when it is scored on the validation base and later it is scored on the out of time base. The out of time base is said to have applications which are not seen by the development

base and are considered to have recent applications. Hence, the performance on the out of time base is of utmost importance and they testify the performance of the model.

3.2.2 GOOD/BAD/REJECT

As described in the earlier phase, typically to develop a scorecard there should be minimum of 2000 goods, bads and rejects respectively. This method is called as oversampling. This method reduces the multicollinearity impact and makes logistic regression result more significant. But in reality in the industry for the secured products, the bad counts is not as high as 2000. With some significant count which would have proportion of bad as low as 0.5% of 1% can be used to develop a model. This method is called proportional sampling. In this case the care should be taken while splitting the development base to keep the bad proportion similar in the training and development sample. As this project intends to develop a grouped characteristic scorecard, the end objective of this sampling method is to ensure that there are sufficient number of goods and bads in each group for the intuitive analysis. This includes the "weight of evidence" calculation for individual groups and the intuitive trend across the attributes.

3.3 DEVELOPMENT DATA COLLECTION AND CONSTRUCTION

As per the decision from the earlier section the development base is selected and the data related to the applications is selected. The liability variables from the liability base are mapped to the base. The bureau variables obtained from CIBIL credit information report (CIR) and are mapped to the base. The end result of this base will consist of all the variables imputed and cleaned. It should contain all the characteristics to be used in the model. It

should also contain target (i.e. good or bad indicator). It should also contain other application level variables, ratio and other derived variables.

3.3.1 Random and Representative

The selection of applications and accounts must be random, representative of the segment for which the scorecard is to be developed (i.e., representative of the population that will be scored in the future), and must exclude the specified account types. A dataset skewed to a certain region or age group may not be effective for the overall applicant population, as it may penalize others or make them seem a lesser relative risk.

3.3.2 Non-segmented Dataset

For projects where more than one scorecard is being developed, a separate dataset needs to be constructed for every segment, as well as one that is non-segmented. The non-segmented dataset is used for analysis to measure any additional "lift," or advantage, provided by segmented scorecards.

3.3.3 Data Quirks

While gathering and collating the data, the modeler should be aware of the historical changes in the database. Since the data is historical and is of past three to four years, the modeler should be especially aware of the codes, formats and formulas which are changed over the year for the specific variables. For examples, if the product code used earlier for home loan was "HL" but in recent times it has changed to "HomeLoan", then due to such kind of discrepancies there would be loss of data or in some cases where emi is calculated certain

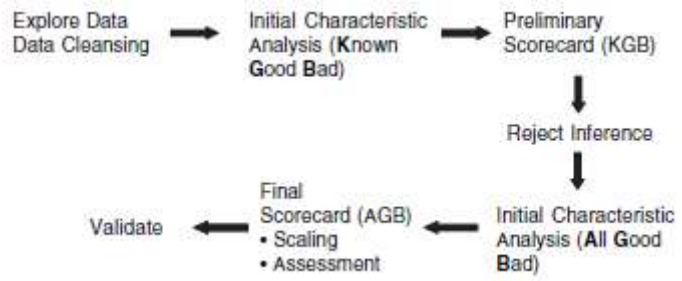
field which are used are populated wrongly or there is any change in the formula lead to wrong data appearing in the specified calculated field. Some variables which are being collected since last two years and before that these variables were not even digitized, this case will lead to lot missing fields. Such kinds of data quirks or format changes are made many times and are not documented. Hence, a modeler should always check for the distribution of the variable at every stage and ask across to concerned teams regarding the observed quirk, so that there are no surprises after preparing whole base.

Chapter 4

Scorecard Development

The scorecard development database created in the previous phase includes all the characteristics and the target variable. There are numerous methods which can be used to develop the scorecard, but all of them involve establishing and quantifying relationship between the characteristics and the target variable.

The following chapter will mainly focus on model development methodology using grouped characteristics. It also emphasises on the algorithm used for development of scorecards, such as logistic regression and additive logistic regression, and the model comparisons.



4.1 EXPLORE DATA

Before actual model development, it is always better to start with some exploratory analysis of the sample data. For this exploratory analysis purpose, some basic statistics are to be calculated such as mean, median, range of values of the characteristics and proportion of missing. This analysis is the better ways to start, as it helps us understand the business attributes in the development base and to reassure the integrity of the data. This type of analysis can be done on the year on year basis to compare the year on year distribution with

the whole portfolio distribution. The missing values in the data should be handled by either replacing them with mean, median or mode or by replacing them with some special values. In case of missing values imputation by special values (e.g. 99 or 999), proper documentation should be prepared for the imputed values for each attributes for further use. Hence, this exercise again confirms the understanding of all the aspects of the data, including the data quirks.

4.2 MISSING VALUE AND OUTLIERS

In the real life scenario in the financial industry, the data contains a lot of missing values or some other data quirks (such as wrong entries). In the financial industry mostly the data is being collected from the applicants through the forms which are further digitized. Hence in many cases the data is missing due to few attributes which are not filled up or which are not applicable to the applicant or miss-keyed during the digitization of form or some cases of simple outliers.

While in some statistical techniques (e.g. decision trees) the missing values are taking care of while developing a model. But since this model building process concentrates mostly on logistic regression we need a complete dataset with no missing values to suffice the requirement of complete case analysis. Mainly, there are four ways to deal with missing values:

1. Exclude all observations from data with missing values. Since in industry, it is more likely to have a lot of missing values. Hence, if we exclude the data with missing values we would be left with very less data for model development.
2. Exclude the attributes or characteristics which have significant amount of missing values (e.g. more than 70% of missing data).

3. Include attributes or characteristics with missing values. The missing values can be treated as by creating a separate dummy variable, which would be then use in the regression as an input. The scorecard will be then allowed to assign the weights to the variable, which be neutral in some cases or mean value, but might be closer to another variable.
4. Imputing missing values using some statistical techniques.

Usage of option 1 and 2 may lead to huge loss of data. But option 2 can be adopted if we have many attribute, and we can afford the loss of the attributes by this condition. Option 3 offers many benefits since it creates dummy variables which would help in interpretations, but this can only be adopted when we have few characteristics or else it may lead to confusion. Option 4 is seems to be the most practical option where in the missing values are imputed using statistical techniques. This can be done by either imputing missing values by mean, median or mode depending on the requirement or by assigning some predefined special values for different categories of missing values. The data mining software used for the model development has some algorithms to impute missing values. It has an imputation node which provides the option of different algorithms such as tree-based imputation and replacement of missing values by mean or median values of the respective attributes. Assigning the mean or most frequent values to the missing may lead to spike in the data; hence it would lead to difficulty in differentiating between data with actual values and mean values imputed. The better way to impute is to first find out the reason for the missing values and then assign the respective missing values to these categories. For example, if the liability variable is missing it may be because the applicant is not an existing customer for current or saving account or the applicant may not have done transaction through his account. In this case some predefined value can be assigned to missing value, if he is not an existing

customer then we can assign -11111 or if he has not made any transactions then we can assign -999. The rule of thumb for deciding the predefined values is that they should be out of the normal range, which can be some very low negative values like -999 or -11111. Since we intend to develop a grouped characteristics scorecard it would be beneficial to use this method of imputation. While in the process of selection of the variable, these special values will help to understand the trend across the variable and find out whether it makes any business sense to consider the particular variable in model development.

The outlier values which fall outside the normal range should be imputed because it may cause some negative effect on the regression results. They can be imputed by mean or medium, but since mean may lead to a spike if the outliers are considered in its calculation. Hence it is good to use median which will smoothen the effect of the outliers. Before any imputation it is necessary to understand the outlier whether it is a genuine data quirk or any fraud. After confirming the reason of the outliers, in case of any data quirks this value can be either imputed by median or can be considered as missing and assigned some special values.

Finally, after the scorecard development the attributes having the special values will be treated differently in the regression and the special value will be assigned different score accordingly.

4.3 ASSUMPTIONS AND PROBLEM OF CORRELATION

The scorecard development is predominantly done using logistic regression; hence we need to satisfy the basic assumptions of the algorithm. The following are the assumptions which need to be satisfied before fitting the model:

1. In binary logistic regression the dependant variable should be necessarily binary and in ordinal logistic regression the dependant variable should be ordinal.

2. The assumption of logistic regression is that, P(Y=1) is the probability of event occurring. Hence, it is necessary to code the dependant variable accordingly representing factor 1 to be the desired outcome.
3. The model should be correctly fitted to get the best prediction outcome. The data should neither be over fitting nor should be under fitting. The independent variable should consist only of those meaningful variables which make business sense and can easily be interpreted. The best way to approach this problem is by using stepwise method to estimate logistic regression.
4. The error term should be independent and model should have no or little multicollinearity. This explains that each observation should be independent and should not be derived from past or future observation. Also the variables should be independent of each other. However, some variables can have interacted effect of two or more variables. This problem could be overcome by deducting the mean of each variable, using principal component analysis (variable clustering) or using factor analysis with orthogonal rotation.
5. The logistic regression assumes linearity of dependent variables and log odds. This means dependent and independent variables need not be linearly related, however independent variable should be linearly related to the log odds.
6. Logistic regression requires a large sample size because maximum likelihood estimates are less powerful than ordinary least squares.

In the banking industry due to regulatory norms and to perform the due diligence a huge amount of data is being collected and documented. Hence there is immense amount of data which can be used for modelling, which results in large number of independent variables. Due to large number of independent variables there always arises problem of

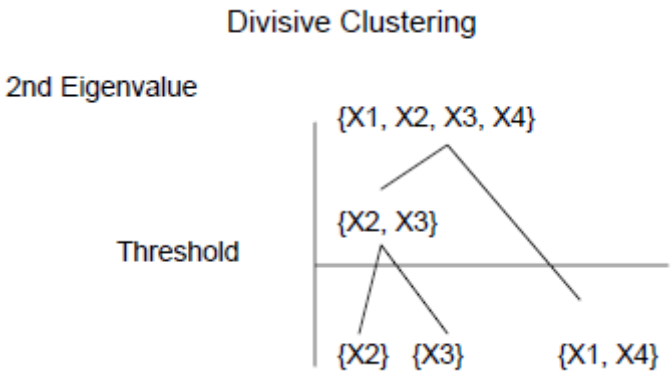
multicollinearity. To overcome this problem there are many techniques, but the method we use is called variable clustering which is a type of principal component analysis. This method can be implemented using PROC VARCLUS in sas environment.

4.4 DIMENSION REDUCTION

With high dimensional datasets it is more difficult to identify the relevant inputs then identifying redundant inputs. Since there are hundreds and thousands of variables available at our disposal to use for model development, it becomes difficult find out the correct relationship between the models. It is often seen that some variables as highly correlated to another. This leads to wastage of time in finalizing the variables which have no or very less multicollinearity problem, represent all pool of information in the base and having business sense. Hence, to speed up this process PROC VARCLUS is used to get the clusters of correlated variables and then select few variables from the clusters according to the business requirements.

PROC VARCLUS is strongly related to principal component analysis and hence is a method for eliminating redundant dimensions. This is a type of variable clustering in which we find group of variables called cluster, which are highly correlated amongst themselves, while the variables in the adjacent cluster are not correlated. If the eigenvalue for the cluster is greater than specified threshold then the cluster is split into two different dimensions. The reassignment of the cluster is done in two phases. In the first phase the nearest component sorting is done, similar to the principle of nearest centroid sorting algorithm. This is an iterative process in which the cluster components are computed and each variable is assigned to component with highest squared correlation. In the second phase search algorithm in executed, wherein each variable is tried and tested by assigning it to different clusters and see amount of variance explained. If the variance explained is increased then the variable is

added to the respective cluster and other aspects are recalculated before testing of next variable.



Note: If the second eigenvalue is larger than the given threshold, more than one dimension exists in the cluster. In the above example initial cluster is broken up into three different clusters.

Rule of thumb for setting up threshold:

1. Larger eigenvalue results in fewer clusters
2. Smaller eigenvalue results in more clusters

Solution to the problem of correlation is explained with a live industry example using PROC VARCLUS in SAS. Also, the theory behind its computation and interpretation of the results with the rule of thumb to be considered are explained below.

5 Clusters		R-squared with		1-R**2
Cluster	Variable	Own Cluster	Next Closest	Ratio
Cluster 1	**Variable1**	0.9991	0.9011	0.0095
1	**Variable2**	0.999	0.9011	0.0101
1	**Variable3**	0.9952	0.8984	0.0477
1	**Variable4**	0.995	0.8982	0.0494
1	**Variable5**	0.839	0.7718	0.7058
Cluster 2	**Variable6**	0.9965	0.942	0.0598
2	**Variable7**	0.9965	0.9421	0.0605
2	**Variable8**	0.9945	0.9389	0.0904
2	**Variable9**	0.861	0.8011	0.6989
Cluster 3	**Variable10**	0.9995	0.6583	0.0015
3	**Variable11**	0.9994	0.658	0.0017
3	**Variable12**	0.9991	0.6577	0.0027
3	**Variable13**	0.9984	0.6494	0.0044
3	**Variable14**	0.9984	0.6504	0.0046
Cluster 4	**Variable15**	0.9996	0.7581	0.0017
4	**Variable16**	0.9995	0.7586	0.002
4	**Variable17**	0.9994	0.7578	0.0024
4	**Variable18**	0.9981	0.7361	0.0072
4	**Variable19**	0.9982	0.7492	0.0072
4	**Variable20**	0.9977	0.7439	0.009
4	**Variable21**	0.9966	0.7985	0.0167
Cluster 5	**Variable22**	0.9993	0.71	0.0023
5	**Variable23**	0.9991	0.7127	0.0032
5	**Variable24**	0.9973	0.7312	0.0099
5	**Variable25**	0.9973	0.7439	0.0105
5	**Variable26**	0.539	0.3223	0.6802

The above shown is an example of variable reduction using PROC VARCLUS. There are about 26 variables in the base which after applying PROC VARCLUS get segmented into five clusters each consisting of highly correlated variables. This proc statement performs an iterative process as explained above and the final optimized output which we desire is shown above.

The analyst can then start selection of variables from the clusters, which can be done by looking at

1-R**2 ratio. The formula for this ratio is:

$$\text{1-R**2 ratio} = \frac{1-R^2 own\ cluster}{1-R^2 next\ cluster} = \frac{1-(High\ Value)}{1-(Low\ Value)} = \frac{Low\ Value}{High\ Value} = Low\ Value$$

Hence as per the above formula, variable selected from each cluster should have high correlation within own cluster and low correlation with other clusters. Hence the variable with lowest 1-R**2 ratio is selected from the cluster. If the variables in the cluster do not make any business sense then no variable is selected from that cluster. Sometimes there are too many variables in the cluster which are to be selected depending on the business requirements and 1-R**2 ratio. Hence variable selection is a very tricky stage where in large amount of business interaction supported with some statistical measures is required. This can be done in two stages wherein, in first stage the variables within the cluster are sorted in ascending order of 1-R**2 ratio but in all the cases the ratio is too close and it becomes very difficult for an analyst to decide the selection of variable. Hence, in the second stage the IV i.e. information value of the variables is calculated and the variables are selected based on the lowest 1-R**2 ratio and highest IV.

5 Clusters		R-squared with		1-R**2	
Cluster	Variable	Own Cluster	Next Closest	Ratio	IV
Cluster 1	**Variable1**	0.9991	0.9011	0.0095	0.186
1	**Variable2**	0.999	0.9011	0.0101	0.194
1	**Variable3**	0.9952	0.8984	0.0477	0.05
1	**Variable4**	0.995	0.8982	0.0494	0.101
1	**Variable5**	0.839	0.7718	0.7058	0.28
Cluster 2	**Variable6**	0.9965	0.942	0.0598	0.067
2	**Variable7**	0.9965	0.9421	0.0605	0.044
2	**Variable8**	0.9945	0.9389	0.0904	0.035
2	**Variable9**	0.861	0.8011	0.6989	0.037
Cluster 3	**Variable10**	0.9995	0.6583	0.0015	0.038
3	**Variable11**	0.9994	0.658	0.0017	0.051
3	**Variable12**	0.9991	0.6577	0.0027	0.065
3	**Variable13**	0.9984	0.6494	0.0044	0.047
3	**Variable14**	0.9984	0.6504	0.0046	0.099
Cluster 4	**Variable15**	0.9996	0.7581	0.0017	0.144
4	**Variable16**	0.9995	0.7586	0.002	0.159
4	**Variable17**	0.9994	0.7578	0.0024	0.052
4	**Variable18**	0.9981	0.7361	0.0072	0.104
4	**Variable19**	0.9982	0.7492	0.0072	0.076
4	**Variable20**	0.9977	0.7439	0.009	0.09
4	**Variable21**	0.9966	0.7985	0.0167	0.062
Cluster 5	**Variable22**	0.9993	0.71	0.0023	0.044
5	**Variable23**	0.9991	0.7127	0.0032	0.035
5	**Variable24**	0.9973	0.7312	0.0099	0.04
5	**Variable25**	0.9973	0.7439	0.0105	0.082
5	**Variable26**	0.539	0.3223	0.6802	0.046

In the above table, the variables marked with yellow are selected. For example, in cluster1 Variable2 is selected on basis of low 1-R**2 ratio and high IV, whereas Variable5 is selected on basis of high IV. Sometimes though the 1-R**2 ratio is high and IV is low the variable is selected just because it is to be included in the model due to business requirement (e.g. Vintage variable). Whereas in some cases the 1-R**2 ratio is low but the IV is high hence the variable is selected. Hence it is up to the analyst to set the rules for selection of the variables. Practically when there is large number of variables it takes a very long time for the execution of the variable clustering and hence to get output. The solution to this problem is to execute PROC VARCLUS on pool of variables separately and then after selection of variables from each pool run a final PROC VARCLUS on those selected variables. For example, PROC VARCLUS statement is run on the bureau variables and liability variables separately and then again in step 2 PROC VARCLUS is executed on the selected variables from the step1. Hence the time execution is saved and the output is obtained faster with accurate results.

4.5 INITIAL CHARACTERISTIC ANALYSIS

This kind of analysis involves mainly two tasks:

1. To check on the strength of the variables individually depending on the performance of the predictor. This also known as univariate screening, usually done to remove the weak and variables not related to the business.
2. To group all the strong variables depending on the predictor performance. This is done for both continuous and discrete variables.

The grouping of the variables is done for the ease to relate it to the scorecard format. The scorecard can be created using the continuous variables as well, but grouping of the variables provides following advantages:

- Offers an easier way to tackle with outliers in the interval variables, missing values and other rare classes of data quirks.
- Grouping of variables helps in understanding the relationship of variable with the predictor. A univariate chart between the attribute and the predictor performance results into a powerful tool than simple attribute statistics to understand the strength of variable. It also helps to state and prove the hypothesis of the trend of the respective attribute with respect to the event rate in each group of the variable, which in turn helps us understand the business importance of the variable to be selected in the model.
- It allows unprecedented control over the development process, by shaping the groups based on the business requirements and the event rate in each group which in turn lead to the inputs to the final model.
- The grouping of variables helps the modeller understand the behaviour of risk predictor and increases the knowledge of portfolio which certainly helps in developing a better model.

At the end of this stage the scorecard developer will be done with the selection of all strong variables which are grouped and ranked, and further becomes the base for the independent variable input to the regression model.

The strength of the attributes is gauged with following four parameters:

1. Predictive power of each attribute. Weight of evidence (WOE) measure is used for this purpose.
2. Range and trend of weight of evidence across groups of the attributes within a characteristic.

3. Predictive power of the characteristic. Information Value (IV) measure is used for this purpose.
4. Operational and business considerations (e.g. grouping states based on the risk over the portfolio or grouping the self-employed and salaried differently based on their income levels.

In some cases over and above these techniques other methods are followed, such as rank ordering according to the predictive power using Chi Square or R-Square prior to grouping characteristics. This method gives an indication of strength of variables and alerts in cases where IV is too high or too low compared to other statistics.

The initial characteristic analysis involves a lot of interactions with the business and operations team, to get insights about the unexpected behaviour pattern and better ways to enhance the groupings.

As the variables are selected from the earlier PROC VARCLUS output, they are further screened in the current stage by binning each variable into groups based on the event count in respective bins. Then the variables are rank ordered based on the IV. The binning can be done in SAS E-Miner using Interactive Grouping Node. The other method which can be used to create bins is decision tree, which is mainly used for discrete variables.

4.5.1 Statistical Measures

The purpose of calculating the WOE and IV is to find out the strength of variable to separate between the good and bad accounts. It is a measure of difference between proportion of goods and bads in each attribute (i.e. odds of an applicant with attribute being good or bad). The WOE is based on log odds calculation:

(Distribution of Good / Distribution of Bad)

The WOE can hence be calculated as follows:

$$[\ ln\left(\frac{Distribution\ of\ Good}{Distribution\ of\ Bad}\right)]\ \times 100$$

Multiplication by 100 is done to make numbers easier to work with. If the WOE is negative then it indicates that a particular attribute has higher proportion of bads than goods.

Information Value (IV) comes from information theory which helps to know the strength of the characteristics and can be formulated as follows:

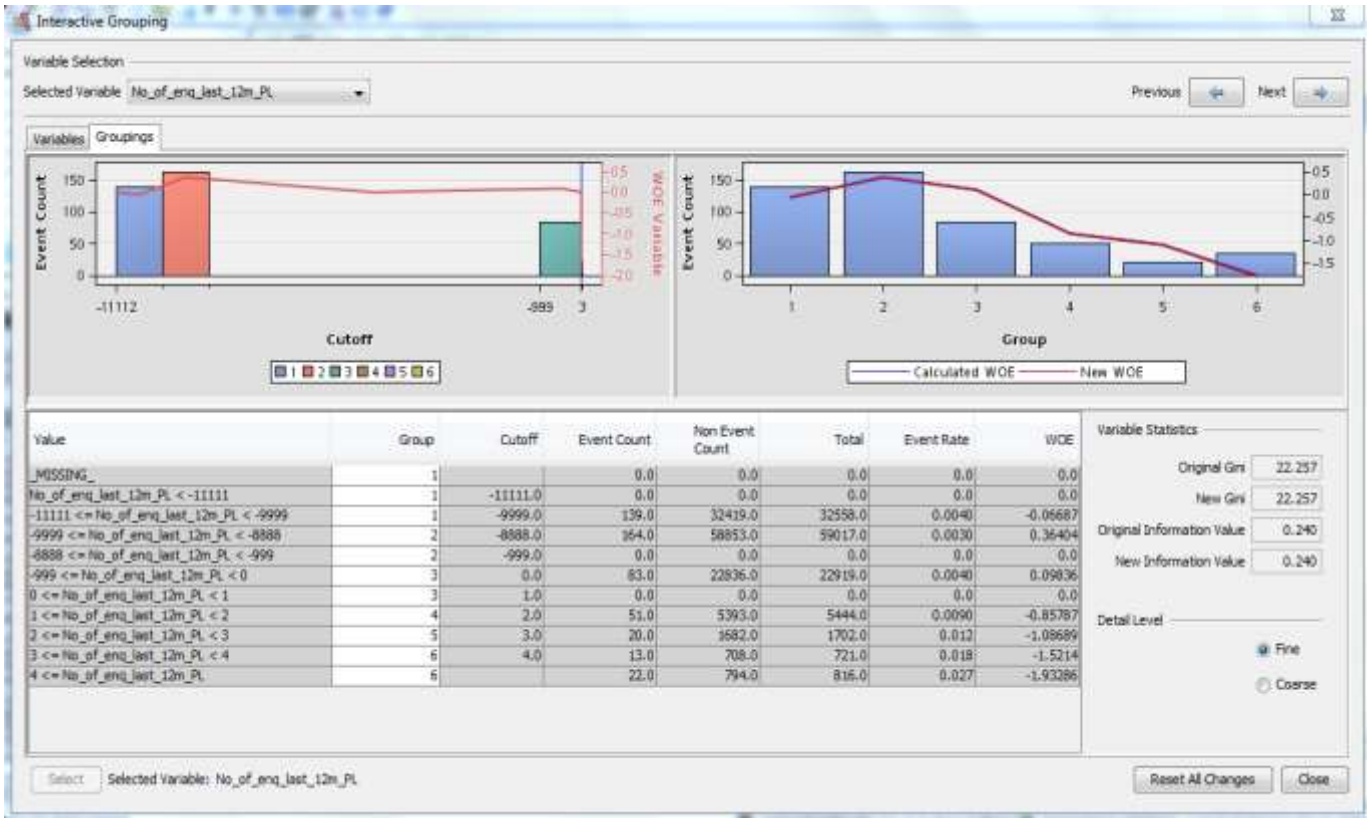
$$\sum_{i=1}^{n}(\ Distribution\ of\ Good_i - Distribution\ of\ Bad_i\ )\times\ ln\left(\frac{Distribution\ of\ Good_i}{Distribution\ of\ Bad_i}\right)$$

Based on this methodology, the rule of thumb for IV is:

- If IV is less than 0.02: unpredictive
- 0.02 to 0.1: weak
- 0.1 to 0.3: medium
- 0.3+:strong

Variable with an IV of greater than 0.5 is susceptible, and needs to be checked for over prediction. They should be either kept out from modelling base or should be used in a controlled manner. These rules can be set by the analyst depending on the variables. Generally the demographic variables have a good IV since it has less groups and the variable is good distinguisher of goods and bads.

For example



The above image shows the basic interactive grouping node output for a selected variable. The following rules are to be noted before grouping the raw variables:

- Missing observations should be binned separately.
- The special values such as -11111,-9999,-8888 and -999 should be binned separately. The special values like -11111 represents no hit on bureau i.e. bureau does not have the data for that particular applicant, -9999 represents that the particular applicant have not made any enquiry for loan in last 12 months, -8888 represents that the particular applicant has not made any enquiry for Personal Loan in last 12 months and -999 is a special case where in the bureau is supposed to have data but due to some irregularities the data is missing with the bureau, hence it indicates that the applicant is hit on bureau.
- Further the actual raw data are grouped into bins depending on the event counts.

- In general every bin should consist of atleast 5% of the total population with no group having zero counts of goods and bads. But practically it is not possible to bin the data keeping minimum of 5% population, so it depends on the analyst whether to split the bin further or not. But some constraint of minimum bad and good counts should be kept before splitting the bin.

In the above example, missing and special bins of -11111 kept in group1 and special bin of -8888 and -9999 are kept in group 2, this categorization is done on characteristics of the special values and they resemble the same effect hence are placed in these groups. The -999 resembles data missing with bureau and hence can be clubbed with 0 bin, so group 3 consists of values -999 and 0. As there are no observations with 0, hence there is no point in creating a new bin. The next bin is with observations having value 1 and value 2 which has sufficient amount of bad and good hence separate groups 4 and 5 are created separately. Further split is done with value 3 and 4 in group 6 which has less data but will provide good split of event rate.

We can further understand the splitting of the trend and event rate logic which drives the splitting of the raw variable

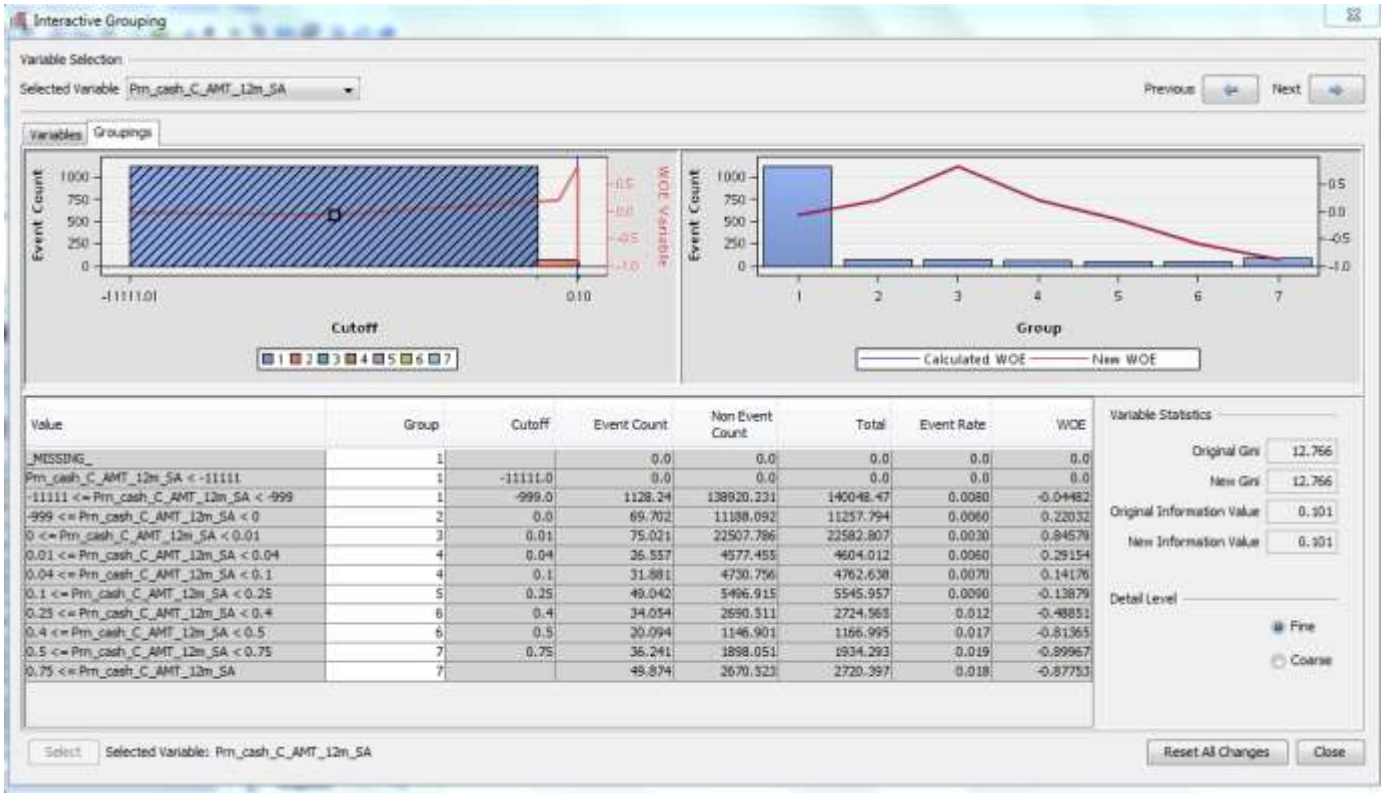
4.5.2 Logical Trend

The WOE and IV is not only a factor of choosing the variables for the further analysis or rank ordering the variable as strong predictor but also is a statistical strength which ensures the logical order of the attribute and helps to know whether it makes operational sense.

For this kind of analysis we need to set some hypothesis about the trend of the variable and then see whether it is true i.e. it is proved or false. In general, for all continuous variable we expect the trend of the variable after grouping to be monotonic i.e. the trend should be either

monotonically increasing or decreasing. But in case of the categorical variable the trend might be monotonic or the variable can be grouped in such a way that the trend becomes monotonic, but in most of the cases the trend is data driven and is to be checked with hypothesis whether the hypothesis is proved or not. For example, variable such as the employment type which consists of whether the applicant is self-employed or salaried, it is being seen across the banking and insurance industry that the self-employed applicants seem to have more chance to default than the salaries guys because the salaried guys have fixed income and can afford to pay the instalments but in case of self-employed due to some income crunch might miss one or two payments. Hence, the hypothesis set in this case would be that the self-employed applicants will be worse than salaried and would have high bad rate than then other. If the hypothesis is proved we can use this variable in the model or else we will have to discard this variable though having the high IV because it will not make any business and operational sense.

We can understand this further with an example



This variable is proportion of cash credit amount to total credit in last 12 months, it is a liability variable. We can understand this better in following manner.

- **Special bins**

  The variable consists of -11111 and -999 as special bins. -11111 represents that the applicant is anew to bank and is not a current account and savings account customer.

  -999 represents that the customer is a current account or savings account customer but has not made any transactions in last 12m.

- **Hypothesis**

  The hypothesis is set that the applicant should have very less cash transactions as compared to the other transactions. Hence, the proportion of cash credits to total credit should be as low as possible. This states that the customers with higher proportion will have higher bad rate than customers with lower proportion.

- **Grouping**

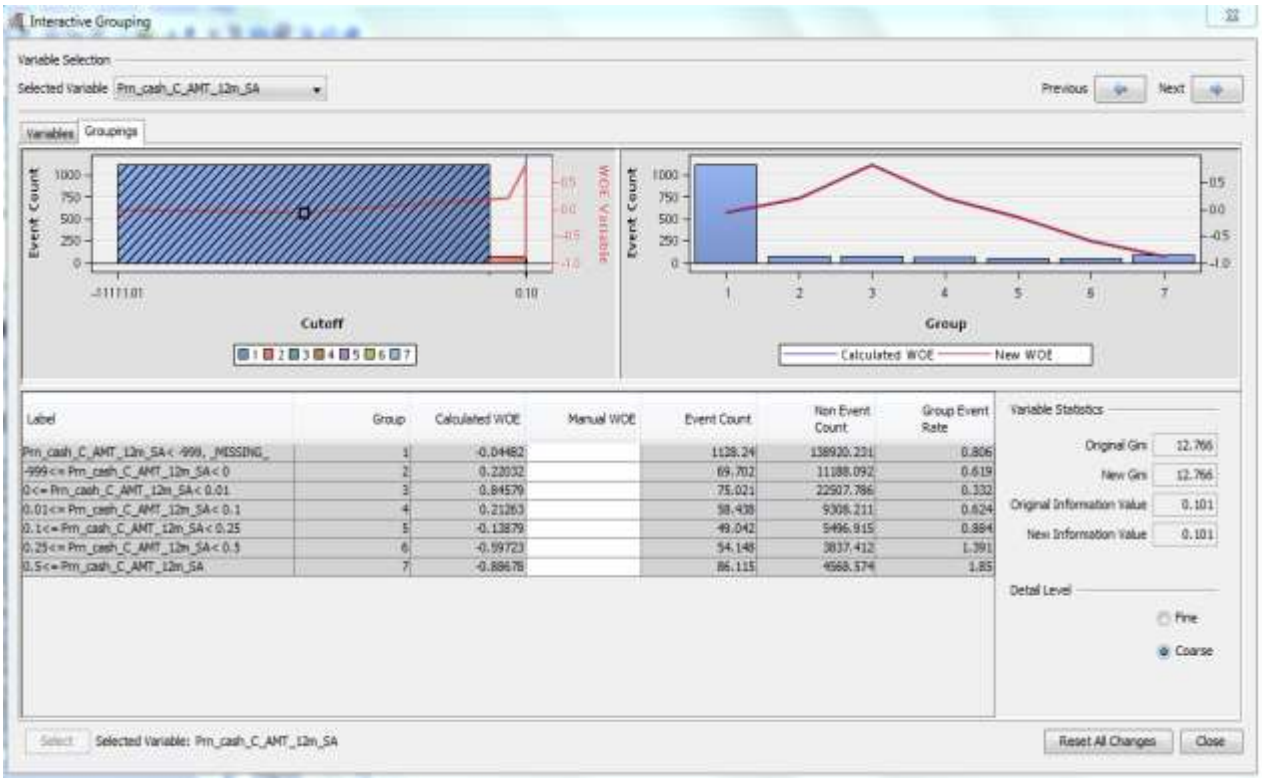
  The Missing and -11111 are grouped to be in same bin as they both represent the effect of missing data and are kept in group 1. -999 is a special bin which represents that there are no transactions made in last 12 months which is placed in group 2. Further the raw proportion values are split into groups 3, 4, 5, 6 and 7 depending on the event rate, such that the event rate has a monotonic increase. This can be seen in the fine groupings.

  Since proportion from 0 to 0.01 has very low bad rate of 0.0030, it is grouped separately as group 3. Proportion 0.01 to 0.04 has bad rate of 0.0060 and proportion 0.04 to 0.1 has bad rate of 0.0070, since these two have very close bad rate we keep them in same group 4. Proportion 0.1 to 0.25 has a marginally higher bad rate hence can be placed in separate group5. While proportion 0.25 to 0.4 has bad rate of 0.012, proportion 0.4 to 0.5 has bad rate of 0.017, proportion 0.5 to 0.75 has bad rate of 0.019 and proportion greater than 0.75 has bad rate of 0.018, all these splits have very close bad rates. The first split of 0.012 bad rate should be placed separately and second can be grouped with the rest of the bins but by doing this the coarse binning will not have a significant difference which can be seen in the next step. Hence split two and three are binned into group6. The forth and the fifth bin has marginal difference hence can be binned together. The rule of thumb is that if the next split has lower bad rate it should be binned with the earlier bin and if it has higher bad rate then it should be binned with the forthcoming group.

- **Trend**

  According to the hypothesis the trend of this variable should be monotonically increasing i.e. lower proportion value should have lower bad rate and higher proportion value should have higher bad rate. The split amongst the values is done in

such a manner that there are considerable numbers of good and bad counts in each bin. This can further be seen by coarse binning. This is shown in the image below:



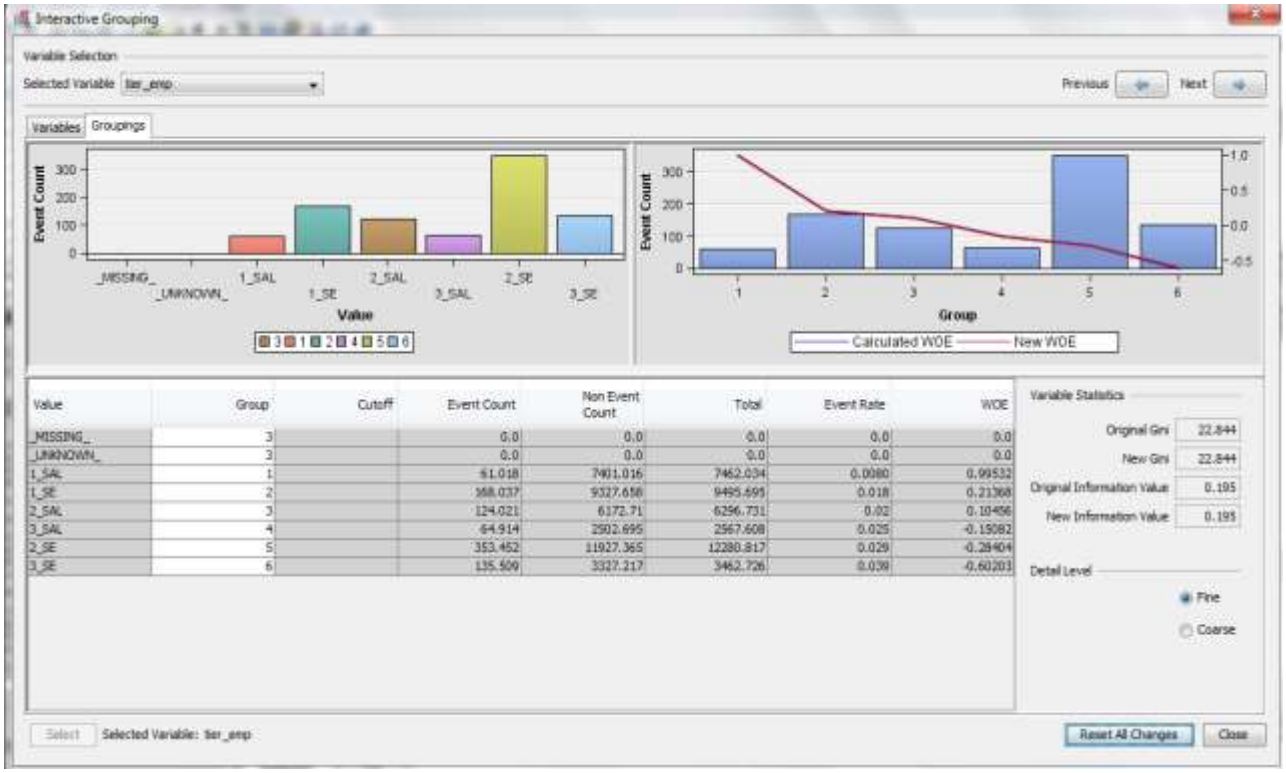
The most important to note is that the values should be grouped in such a manner that there should be significant difference between adjacent group event rate and should be monotonically trending.

In the above diagram, the group 1 has slightly higher event rate than group 2 which is true according to our hypothesis that bad rate of the applicants who are new to bank is more than those who are existing customer to bank but have not made any transactions. While in the bins where there are actual values i.e. from group 3 to group 7 then event rate is monotonically decreasing, hence there is a decreasing trend which is also proved according to our hypothesis.

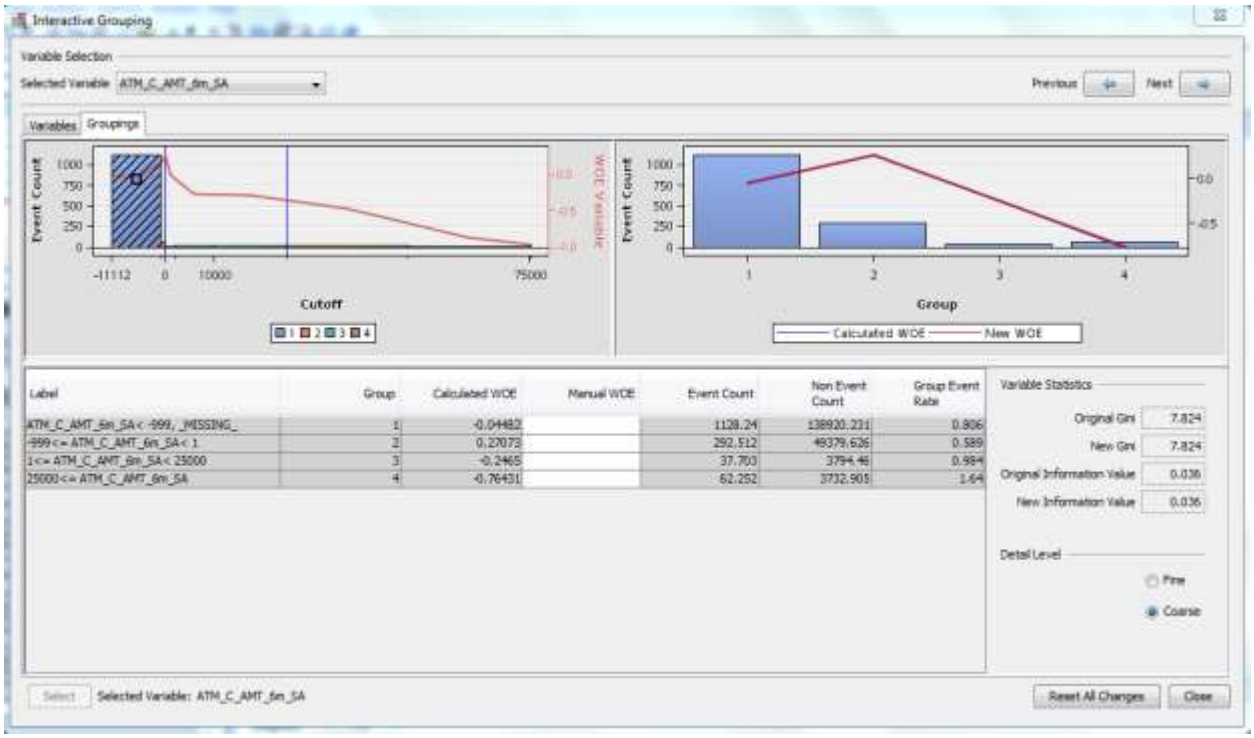
Label	Group	Group Event Rate
Prn_cash_C_AMT_12m_SA < -999, _MISSING_	1	0.806
-999 <= Prn_cash_C_AMT_12m_SA < 0	2	0.619
0 <= Prn_cash_C_AMT_12m_SA < 0.01	3	0.332
0.01 <= Prn_cash_C_AMT_12m_SA < 0.1	4	0.624
0.1 <= Prn_cash_C_AMT_12m_SA < 0.25	5	0.884
0.25 <= Prn_cash_C_AMT_12m_SA < 0.5	6	1.391
0.5 <= Prn_cash_C_AMT_12m_SA	7	1.85

Also the group event rate is significantly different i.e. there is good enough difference between adjacent values and are not too close. Group3 has 0.332 and group 0.624 which is almost double. Such kind of difference between the values is said to be acceptable, if the value would have changed from 0.332 to 0.360 then it is better to keep them into same group.

The nominal variables are in binned in different manner. This requires a lot of insights from business and operational teams, as to how to bin the variables so that they can make business sense.



In the above image the interacted variable of tier and employment type is given. There are two employment types namely self-employed and salaried and three tiers based on the portfolio size of the product which is tier 1, tier 2 and tier3. Hence, we can bin the salaried applicants from tier 1 in one group and applicants from tier 2 and tier 3 into another group due to close enough bad rates. Also, for self-employed applicants from tier 1 is kept in one group and applicants from tier 2 and tier 3 are kept in another group due to close enough bad rates.



The above image shows the variable with an illogical trend. The hypothesis for this variable is that amount credited through the Crediting machine in last 6 months should be more. But in this case the applicants who have credited amount less than 25000 through this medium has lesser bad rate than who have credited amount more than 25000, which rejects our hypothesis. Hence, the variable has trend which is opposite to the business and operational understanding and such variable cannot be included in the model development process.

The whole process of interactive grouping is a very important step of the modelling which creates the base for the model development. The variables which go into the model are the WOE's of the variables which are calculated for every group. Hence the scores assigned to every applicant after the implementation of the model will be dependent on the WOE's. After implementation for the applicants coming in, it is being seen that in which group the raw data of that applicant falls in and the respective calculated WOE is assigned. This is a kind of iterative process which requires a lot of statistical, business and operational knowledge insights to take the decision to finalize the variables depending on their business importance and statistical measures.

4.6 PRELIMINARY SCORECARD

The earlier stage of initial characteristic analysis helps in finalizing the set of strong variables which can be considered as independent variable into the model. In the preliminary scorecard stage, various predictive modelling techniques such as logistic regression, decision trees, various boosting algorithms and neural networks are widely used in the industry. In the risk scorecard space, majorly logistic regression or any other boosting algorithms using logistic regression such as Logit boost is used, reason being the regulatory constraints. The modelling techniques used in the risk space should not be a black box, i.e. it should provide information of all the variables, coefficients and their relation with the dependent variables in the model. These kinds of models are easily interpretable, hence when an applicant gets a low score it is easy to find out which variable has caused the low score. Where as in the modelling techniques such as neural network it is a kind of black box and becomes difficult to interpret. It is also a regulatory condition that the model should be transparent and easily interpretable. In general, the model should consist of variables between 8 and 15. It is always better to too

have more number of variables in the model, as with few variables the model is susceptible to minor changes and may degrade faster.

Irrespective of the modelling techniques used, the process of scorecards development should consist of optimal combination of variables, considering certain issues as:

- Correlation between variables
- Final statistical strength of scorecard
- Interpretability of variables
- Implementability
- Transparency of methodology for regulatory requirements

4.6.1 Risk Profile Concept

The development of scorecard is done for various objectives such as maximum statistical measures, efficiency (i.e. using the best variables), and so on. In real business terms, scorecard should be developed to mimic the task of the seasoned and effective underwriter or risk analyst. A good underwriter will not come to the final decision, by just looking at four or five characteristics of an applicant such as the application form, bureau data, account history, etc. Underwriter will look at all the information regarding the applicant which may lead to increase in risk which might be in a set of permutations and combinations.

The objective of the scorecard is to create a comprehensive risk profile of the customer to predict the probability of default by scorecard more efficiently. Such risk profile should consist of some demographic information like age, employment type, years of employment, obligation, income, region, etc. ; some credit bureau information as number of enquiries, vintage on bureau, number of trade lines opened and closed, etc. ; some liability information like transaction level information i.e. credits and debits in terms of amount and number,

average balance of different period, channel level transactions like NEFT, mobile banking , internet banking, etc. and some ratio variables as well.

The risk concept also helps in housekeeping of the model; it mainly consists of model monitoring by calculating various matrices over time to check the performance of the model. Most risk analyst run a report of "system stability" or "population stability" on monthly and quarterly basis to confirm the validity of scorecard on current applicant or account populations. This model monitoring exercise is done to ensure the performance of the model, in some cases when there are very few variables or some debatable variables which may lose its trend sooner after its implementation this process helps to gauge these problems beforehand so that necessary actions can be taken.

The later section will deal with the modelling techniques which can be used.

4.6.2 Logistic Regression

In most of the financial industry applications the modelling technique used is logistic regression, where the predicted variable is categorical. The purpose of the logistic regression is to predict whether the applicant will good or bad (i.e. applicant will default or not).

Binary logistic regression estimates the probability that a characteristic is present (e.g. estimate probability of "Bad");

$$\pi = Pr(Y = 1|X = x)$$

Variables:

- Let Y be binary response variable

  $Y_i = 1$ if the applicant defaults on his credit ("bad" customer)

  $Y_i = 0$ if the applicant does not default on his credit ("good" customer)

- X = (X_1, X_2,…, X_k) are the WOE calculated of all the strong variables which are selected from earlier exercises.

Model:

$$\pi_i = P\,r(Y_i = 1|X_i = x_i) = \frac{exp(\beta_0 + \beta_1 x_i)}{1 + exp(\beta_0 + \beta_1 x_i)}$$

Or,

$$logit(\pi_i) = log(\frac{\pi_i}{1-\pi_i})$$
$$= \beta_0 + \beta_1 x_i$$
$$= \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}$$

Parameter Estimation:

The *maximum likelihood estimator* (MLE) for $((\beta_0, \beta_1)$ is obtained by finding $((\widehat{\beta_0}, \widehat{\beta_1})$ that maximizes:

$$L(\beta_0, \beta_1) = \prod_{i=1}^{N} {\pi_i}^{y_i} (1-\pi_i)^{\eta_i - y_i} = \prod_{i=1}^{N} \frac{\exp\{y_i(\beta_0 + \beta_1 x_i)\}}{1 + \exp(\beta_0 + \beta_1 x_i)}$$

In general, there are no closed-form solutions, so the maximum likelihood estimates are obtained by using iterative algorithms such as *Newton-Raphson* (NR) or *Iteratively re-weighted least squares* (IRWLS).

Interpretation of Parameter Estimates:

- $exp(\beta_0) =$ Odds that the characteristic is present in an observation *i* when $X_i = 0$, i.e. at baseline.
- $exp(\beta_1) =$ for every unit increase in X_{i1}, the odds that the characteristic is present is multiplied by $exp(\beta_1)$ .This is similar to simple linear regression but instead of additive change it is a multiplicative change in rate. The estimated odd ratio is:

$$\frac{\exp(\beta_0 + \beta_1(x_{i1} + 1)}{\exp(\beta_0 + \beta_1 x_{i1})} = exp(\beta_1)$$

The logistic model stipulates that the effect of a covariate on the chance of "default" is linear on the log-odds scale, or multiplicative on the odds scale.

- If $\beta_j > 0$, then $exp(\beta_j) > 1$, and the odds increase.
- If $\beta_j < 0$, then $exp(\beta_j) < 1$, and the odds decrease.

The regression is run to find out the best possible model using all option available, which is commonly known as "all possible" regression technique. This technique is extensively used when there are a lot of independent input variables. The most commonly used three types of stepwise logistic regression techniques:

- Forward Selection

  In this technique, a variable is added at each step. At every step the variables which are in the model are tested, and one which has significant P-value and when added to the model R-square and adjusted r-square of the model increases is selected to be in the model. Hence we get a set of strong predictors which are used in the model. This is a process where in every variable is tried and tested sequentially in the model only once.

- Backward Elimination

  This technique is exactly opposite of forward selection. In this technique, all variables are first added to the model and then at every step the variable with high P-value and variable causing decrease in the R-square and adjusted R-square is removed from the model. This process also has sequential elimination where in variables with lower significance are eliminated and ones with higher significance which are good predictors are selected.

- Stepwise

  This technique is a combination of above two techniques. This involves dynamically adding and removing of the characteristics at each step of the model based on its significance and the goodness of fit. Hence, finally when all the significant characteristics are selected the model can be run on these selected characteristics.

4.6.3 Boosting Algorithms

The procedure in which many "weak" classifiers are combined to produce a powerful "committee" is called as "boosting".

Basic framework of a boosting algorithm:

We have training data $(x_1\,,\ y_1)\,,\ \ldots\,,(x_N\,,\ y_N)$ with x_i a vector values feature,

$$t_i = -1\ or\ 1$$

And, the prediction is y_i

The Loss Function is defined as

$$L(t_i\,,\ y_i)$$

We have M possible iterations i.e.

$$m = 1\ to\ \ M$$

And, the final prediction is

$$Y = \sum_{m=1}^{M} \alpha_m f_m(x)$$

- For $m = 1$ i.e. for the first iteration:

  Find $f_1(x)$ which minimizes

  $$\sum_{i=1}^{N} w_i^1 L(t_i, f_1(x))$$

  Where,

  $$w_i^1 = 1/N$$

- For $m > 1$ i.e. for the further iterations:

  We define constants and weights as

  $$\{\alpha_{m-1}, w_i^m\}$$

  Based on the error from previous iteration $(m-1)$

  $$E_{m-1} = \frac{\sum_{i=1}^{N} w_i^{m-1} \varepsilon_i(\, t_i, \sum_{m'=1}^{m-2} \alpha_{m'} f_{m'}(x) + \alpha_{m-1} f_{m-1}(x))\,)}{\sum_{i=1}^{N} w_i^{m-1}}$$

  Find $f_m(x)$ which minimizes

$$\sum_{i=1}^{N} w_i^m L(t_i, f_m(x))$$

Final prediction:

$$Y = \sum_{m=1}^{M} \alpha_m f_m(x)$$

4.6.4 Logit Boost Algorithm

The Discrete and Real AdaBoost are unnecessarily complicated; a much simpler way to fit an additive model would be to minimize the squared-error loss $E(y - \sum f_m(x)^2$ in a forward stagewise manner. At the m[th] stage we fix $f_1(x) \ldots f_{m-1}(x)$ and minimize the squared error to obtain

$$f_m(x) = E(y - \sum_1^{m-1} f_j(x)|x)$$

This similar to fitting of residuals and it is commonly used in linear regression and additive modelling. However squared error is not a good choice for classification and hence in this case fitting of residuals does not work well. The AdaBoost fits an additive model using a better loss function which is similar to but not same as binomial log-likelihood for additive logistic regression model.

If $p_m(x)$ are the probability of classes, then an additive logistic regression approximates $log\ p_m(x)$ / (1- $p_m(x)$) by an additive function $\sum_m f_m(x)$.Hence, we further go and derive a new boosting procedure "LogitBoost", which directly optimizes the binomial log-likelihood.

In this article, we concentrate on the algorithm of LogitBoost which given below:

The overall error/loss function using the binomial-likelihood optimization is

$$E = \sum_{i=1}^{N} \log[1 + \exp\{-2t_i F(x_i)\}]$$

Where, the overall prediction is

$$Y = sign\{F(x)\} = sign\{\sum_{m=1}^{M} f_m(x)\}$$

- For $m = 1$ i.e. for the first iteration:

  We define weight for first observation is

  $$w_i^1 = 1/N$$

  We initially set

  $$F_1(x) = f_1(x) = 0$$

  And, probability estimator as

  $$p_1(x_i) = \Pr(t_i = 1|x) = 1/2$$

- For $m > 1$ i.e. for the further iterations:

  We define

  $$e_i^m = \frac{t_i^* - p_{m-1}(x_i)}{p_{m-1}(x_i)(1-p_{m-1}(x_i))}$$

  Where,

  $$t_i^* = \frac{1 + t_i}{2}$$

  We compute weights for further observations as

  $$w_i^m = p_{m-1}(x_i)(1 - p_{m-1}(x_i))$$

  We fit $f_m(x)$ by regressing e_i^m on x_i with weight w_i^m.

  Here, we use linear regression with weighted least-square and minimize

  $$\sum_{i=1}^{N} w_i^m \ (e_i^m - f_m(x_i))^2$$

We update $F_m(x)$ as

$$F_m(x) = F_{m-1}(x) + \frac{1}{2} f_m(x)$$

We update $p_m(x)$ as

$$p_m(x) = \frac{\exp(F_m(x))}{\exp\big(F_m(x)\big) + \exp(-F_m(x))}$$

The final prediction will be:

$$F(x) = F_M(x), \quad p(x) = p_M(x)$$

4.6.5 Ensemble Modelling Algorithm

The ensemble modelling can be broadly classified into three categories

1. **Bagging**

   It stands for Bootstrap Aggregation; it is a way to decrease variance of prediction by generating additional data for training from the original dataset using combinations with repetitions to produce multisets of the same cardinality/size as the original data. By increasing the size of training set you can't improve the model predictive force, but just decrease the variance, narrowly tuning the prediction to expected outcome.

2. **Boosting**

   This a particularly a two-step approach, where first one uses subsets of the original data to produce a series of averagely performing models and then "boosts" their performance by combining them together using a particular cost function. Unlike bagging, in the classical boosting the subset creation is not random and depends upon the performance of the previous models: every new subsets contains the elements that were (likely to be) misclassified by previous models.

3. **Stacking**

   This technique is similar to boosting where we can apply various models to the original data. The difference here is, however, that you don't have just an empirical formula for your weight function, rather you introduce a meta-level and use another model/approach to estimate the input together with outputs of every model to estimate the weights or, in other words, to determine what models perform well and what badly given these input data.

4.6.6 Multi-layer Logistic Regression or Stacking Algorithm

This technique involves training a learning algorithm to combine predictions of several other learning algorithms. Firstly, all the learning algorithms are used to train the available data, then a combiner algorithm is trained over the earlier predictions with some extra inputs and the final prediction is calculated. If any arbitrary combiner model is used, then stacking can theoretically represent any of the ensemble techniques described earlier. But when we use single-layer logistic regression model as combiner it represents the stacking. It is observed that stacking yields better performance than any one of the single models trained.

The data used in the application scorecard development mainly consists of three pool of data, namely bureau data (i.e. information obtained from credit report), liability data (i.e. transaction level data of existing current and savings account customers) and demographic data (i.e. application level data of an individual applicant). Whenever we fit a single-layer logistic model after the dimensionality reduction and removing the problem of multicollinearity the number of variables we get after fitting the model are sometimes very few and they do not have strong representation from all pools of data. Sometimes, few variables from different pool of data which can come-up in the model get suppressed by

another variable from different pool. Hence, in this article, we have tried to fit the multi-layer logistic regression to overcome the above problem. These pools will have heterogeneity amongst themselves and homogeneity within them.

Stacking algorithm:

- Step1

  For implementation of this algorithm, we first segment the whole base into pools of data which represent the same characteristics or have the same source. Hence, all variables which are obtained from credit report or are calculated from the same are placed in bureau data, all transaction level variables and the calculated variables from the same are placed in liability data and lastly the application level data or the demographic variables and it's calculated fields are placed in the demographic data.

- Step2

  Here we first fit a single-layer logistic regression on the whole of bureau data only. All the bureau variables are again run by PROC VARCLUS and the reduced strong variables based on IV and 1-R**2 ratio are selected.

  A = (a_1, a_2,…, a_k) are the WOE calculated of all the strong bureau variables which are selected from earlier exercises.

$$p_{cibil}(a_i) = Pr(Y_i = 1 | A_i = a_i) = \frac{exp(\beta_0 + \beta_1 a_i)}{1 + exp(\beta_0 + \beta_1 a_i)}$$

Or,

$$logit(p_{cibil}(a_i)) = log\left(\frac{p_{cibil}(a_i)}{1 - p_{cibil}(a_i)}\right)$$

$$= \beta_0 + \beta_1 a_i$$

$$= \beta_0 + \beta_1 a_{i1} + \ldots + \beta_k a_{ik}$$

Parallely, in this step another single-layer logistic regression is fit on the whole liability data only. All the liability variables are again run by PROC VARCLUS and the reduced strong variables based on IV and 1-R**2 ratio are selected.

B = (b_1, b_2,…, b_k) are the WOE calculated of all the strong liability variables which are selected from earlier exercises.

$$p_{liability}(b_i) = Pr(Y_i = 1 | B_i = b_i) = \frac{exp(\beta_0 + \beta_1 b_i)}{1 + exp(\beta_0 + \beta_1 b_i)}$$

Or,

$$logit\left(p_{liability}(b_i)\right) = log\left(\frac{p_{liability}(b_i)}{1 - p_{liability}(b_i)}\right)$$

$$= \beta_0 + \beta_1 b_i$$

$$= \beta_0 + \beta_1 b_{i1} + \ldots + \beta_k b_{ik}$$

In the second step, the predictions $p_{cibil}(a_i)$ and $p_{liability}(b_i)$ from the earlier logistic model is used as an input i.e. independent variables with other demographic variables to the new multi-layer logistic regression with other demographic variables. All demographic variables are checked for multicollinearity and the strong demographic variables are selected, which are given as inputs to the multi-layer logistic model with earlier predictions.

C = (c_1, c_2,…, c_k) are the WOE calculated of all the strong liability variables which are selected from earlier exercises.

$$p_{demog}(c_i) = Pr(Y_i = 1 | C_i = c_i)$$

$$= \frac{exp(\beta_0 + \beta_1 c_i + \beta_2 p_{cibil}(a_i) + \beta_3 p_{liability}(b_i))}{1 + exp(\beta_0 + \beta_1 c_i + \beta_2 p_{cibil}(a_i) + \beta_3 p_{liability}(b_i))}$$

Or,

$$logit(p_{demog}(b_i)) = log\left(\frac{p_{demog}(b_i)}{1 - p_{demog}(b_i)}\right)$$

$$= \beta_0 + \beta_1 b_i + \beta_2 p_{cibil}(a_i) + \beta_3 p_{liability}(b_i)$$

At every step the variables in the logistic regression model are checked for VIF and the strong representation of variables from all three pools is obtained.

4.6.7 Designing a Scorecard

While it is possible to build a scorecard by putting all the characteristics into the regression model and generating a statistically optimal output, but this might not produce operationally ideal outcome. The scorecard developers typically rely on some statistical measures such as Chi-Square, R-Square, *p*-value and other measures to obtain the quality of the outcome. However, score card development becomes a trickier task due to considerations of some business and operational goals.

The initial goal is to choose the group of best characteristics to build the most comprehensive risk profile. The concept of risk profile is discussed in the earlier stage. Ideally, the risk profile should be built using as many independent characteristics as possible.

The base is mainly categorized into three sources i.e. bureau data, liability data and demographic data. The bureau data we get is mainly from CIBIL which consists of products namely:

1. Auto Loan
2. Home Loan
3. Personal Loan
4. Credit Cards
5. Secured Core Loan (i.e. Home loan, Auto Loan, etc.)
6. Unsecured Core Loan (i.e. Personal Loan, Credit Cards, etc.)
7. Secured Others Loan (i.e. Gold Loan, Loan against Bonds, etc.)
8. Unsecured Others Loan (i.e. Educational Loan, Overdraft, etc.)

For all the above products we get then credit report for all applicants and their performance on bureau, from which we calculate following variables:

- **Last account closed**

  It has the number account closed latest.

- **Latest utilization**

  It shows the amount utilized over the credit given (e.g. in case of credit card it gives the utilized amount).

- **Number of delinquent 90 plus trade lines**

  It shows total number of trade lines where the applicant has missed three payments i.e. he is 90 days past due in a specific time frame (i.e. in 3, 6, 12, 24 months.).

- **Number of delinquent trade lines**

  It shows the total number of delinquent trade lines i.e. number of trade lines where the applicant may have missed even one payment in a specific time frame (i.e. in 3, 6, 12, 24 months).

- **Maximum delinquency**

  It shows the maximum times applicant going delinquent in specified time (i.e. in 3, 6, 12, 24 months.)

- **Months since delinquency**

  It shows the number of months since last delinquency has taken place for an applicant.

- **Number of total trade lines**

  It shows the total number of trade lines an applicant has had with bureau as of date of application.

- **Number of closed trade lines**

  It shows number of trade lines closed in a specific time frame i.e. total number of trade lines closed in last 3, 6, 12, 24 months.

- **Number of opened trade lines**

  It shows number of trade lines opened in a specific time frame i.e. total number of trade lines opened in last 3, 6, 12, 24 months.

- **Number of open trade lines**

  It shows total number of open trade lines of the applicant with bureau as of time of application.

- **Number of close trade lines**

  It shows total number of close trade lines of the applicant with bureau as of time of application.

- **Times going delinquent**

  It shows the frequency of an applicant of missing payments in a specific time frame (i.e. in 3, 6, 12, 24 months).

- **Number of Enquiries**

  It shows total number of enquiries as of date of application as well as in specified time frame (i.e. in 3, 6, 12, 24 months).

- **Vintage on Bureau**

  It shows how long the information about an applicant is there with the bureau i.e. is the time of history of the applicant with bureau.

All the above variables are calculated for individual products as well as for all products together. So finally we can get variables such as Maximum delinquency for Home Loan trade lines in last 12months, Number of Personal Loan trade lines opened in last 6 months, Number of enquiries in last 3 months etc.

The Liability i.e. transaction data for the existing current and savings account customers, which is obtained from bank itself is used for different channels namely:

1. ATM
2. Branch transactions
3. Cash
4. Cheque
5. Mobile Banking (i.e. through mobile application)
6. NEFT and RTGS
7. Electronic Clearing Service (ECS)
8. Internet Banking (i.e. through website)
9. Point of Sale transaction (POS)

For all the above channels we calculate the ratio variables and balance variables as:

- **Credit amount**

  It shows the amount credited by a particular channel to account in specific time frame (i.e. 1, 3, 6, 9, 12 months).

- **Credit number**

  It shows the number of credits made by a particular channel to account in specific time frame (i.e. 1, 3, 6, 9, 12 months).

- **Debit amount**

  It shows the amount debited by a particular channel to account in specific time frame (i.e. 1, 3, 6, 9, 12 months).

- **Debit number**

  It shows the number of debits made by a particular channel to account in specific time frame (i.e. 1, 3, 6, 9, 12 months).

- **Average balance**

  It shows average balance in the account in a specific time frame (i.e. 1, 3, 6, 9, 12 months).

- **Proportion of average balance**

  It shows ratio of average balance in last 3 months to average balance in last 6 months.

- **Transaction amount number greater than certain amount**

  It shows number of transactions made greater than 10k or 20k 0r 5k in specific time frame (i.e. 1, 3, 6, 9, 12 months).

- **Proportion variables**

  Proportion of amount credited or debited through any particular channel to total amount credited or debited respectively in a specific time frame (i.e. 1, 3, 6, 9, 12 months).

Proportion of number of credits or debits made through any particular channel to total number of credits or debits respectively in a specific time frame (i.e. 1, 3, 6, 9, 12 months).

Proportion of debit amount to average balance in a specific time frame (i.e. 1, 3, 6, 9, 12 months).

Proportion of credit amount to average balance in a specific time frame (i.e. 1, 3, 6, 9, 12 months).

So finally after calculation of these variables we can get variables such as Proportion of ATM credits to total credits in last 6 months, Number of transactions with amount greater than 10k in last 9 months, Cash credit amount in last 12 months, etc.

For demographic variables the source would be the information collected from the application form and the documents collected at the time of application. These documents are then digitized and can be used in the modelling. The variables obtained from the demographic data pool are verified and non- verified as well, so it becomes very important to first confirm the authenticity of the variables from the operations and business teams, as only the verified data should be used in modelling base to avoid ad discrepancies in the model. The model can have variables such as:

- Age
- Employment type
- Type of organization applicant is working in
- Work experience in an organization
- Annual income
- Other obligations
- Co-applicant’s demographic data

- Co-applicant's income
- Property level information
- Region, state and pin code

  etc.

The final set of strong variables is selected from earlier stage, wherein we have checked all variables for multi-collinearity by using PROC VARCLUS. These variables are again checked manually just to ensure that we don't miss any important variable from business sense and we set covers all pool of data we have discussed earlier. Hence, finally we have shortlisted some 24 variables which are selected form the PROC VARCLUS clusters obtained, demographic nominal variable on which variable clustering is not performed and some other variables which are of operational and business importance. Then the WOE for every variable is calculated using the Interactive Grouping node in SAS Enterprise Miner.

PROC LOGISTIC

This is a logistic procedure statement which is used in SAS to fit a logistic regression model. The input dataset is given to the logistic model to get the output predictions. There are many options which are used in the modelling. These options are:

- **DATA**

  It provides the name of the input dataset on which the model is to be fitted.
- **NAMELEN**

  It helps to predefine the length of the variable in the output. As the output usually comes with truncated variable names in case of variables with long names.
- **MODEL**

  It is used to fit a model i.e. it is used to write the basic logistic regression with independent variables (input) and dependent variable (target or bad flag).

- **EVENT**

  It is used to specify the event with which the model is to be fitted i.e. in this case the model is to be fitted to find out probability of default.

- **SELECTION**

  It is used to select the technique with which the variables are selected. In earlier section we have seen that the variable can be selected in forward, backward or stepwise manner.

- **SLSTAY**

  It is the p-value threshold which is set for the variable to be included in the model. The variables with p-value greater than SLSTAY would be dropped and less than threshold will stay in the model. The SLSTAY value is generally 0.01 (for 99% confidence interval) or 0.05 (for 95% confidence interval).

- **OUTPUT**

  It will output to a particular dataset after certain calculation.

- **OUT**

  It specifies the destination dataset name in which the output data is to be stored.

- **PRED**

  It is used to specify the variable name in which the prediction of the logistic regression is to be stored after calculation.

- **SCORE**

  It is used to score data over other data sets i.e. when a model is trained over the train dataset it needs to be tested or validated over the test data, for this purpose SCORE option is used. In this option we specify the DATA which has the test data on which the model is to be tested and OUT is the option which is used to output data with prediction.

The model is fit in SAS Enterprise Guide using the following procedure:

```
PROC LOGISTIC DATA= BASE NAMELEN=50;

MODEL Bad_Flag(EVENT = '1')=

Variable1

Variable2

Variable3

Variable4

Variable5

Variable6

Variable7

Variable8

Variable9

Variable10

/SELECTION=STEPWISE SLSTAY=0.001;

OUTPUT OUT=TRAIN PRED=P_1;

SCORE DATA= VALIDATION OUT= VALIDATION;

SCORE DATA= OOT OUT= OOT;

RUN;
```

The variable list above is of 24 variables which are used in fitting model, but the code only consists of representation of actual variables. In this article, we use the stepwise method for selection of variable in logistic regression, as the stepwise selection is best suited for this project. In stepwise selection method, a particular variable is added in the model and is then tested for the Adjusted R-square, if it has an incremental effect it is kept in the model or else it is dropped in the model.

Stepwise Logistic Regression

Summary of Stepwise Selection							
	Effect			Number	Score	Wald	
Step	Entered	Removed	DF	In	Chi-Square	Chi-Square	Pr > ChiSq
1	Variable1		1	2	220.4056		<.0001
2	Variable2		1	3	127.5468		<.0001
3	Variable3		1	4	125.3094		<.0001
4	Variable4		1	5	79.9646		<.0001
5	Variable5		1	7	61.1625		<.0001
6	Variable6		1	9	53.4862		<.0001
7	Variable7		1	10	31.4418		<.0001
8	Variable8		1	11	28.526		<.0001
9	Variable9		1	12	20.9599		<.0001
10	Variable10		1	13	43.6042		<.0001
11	Variable11		1	15	21.333		<.0001
12	Variable12		1	16	21.1893		<.0001
13	Variable13		1	17	17.2866		<.0001
14	Variable14		1	18	13.5088		0.0002
15	Variable15		1	19	30.2396		<.0001
16	Variable16		1	20	18.4402		<.0001
17	Variable17		1	21	13.4492		0.0002
18	Variable18		1	22	13.2887		0.0003
19	Variable19		1	23	11.9224		0.0006
20	Variable20		1	25	8.6989		0.0032
21	Variable21		1	26	11.2161		0.0008
22	Variable22		1	27	12.174		0.0005
23	Variable23		1	28	9.9621		0.0016
24	Variable24		1	29	6.2589		0.0124
25		Variable24	1	28		6.2487	0.0124

The above table shows the output of stepwise selection of logistic regression. After implementation of the stepwise logistic regression of the 24 variables we get the final output, where one variable is removed from the model in the stepwise selection and finally we have 23 variables in the model.

Estimates

Analysis of Maximum Likelihood Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**Wald Chi-Square**	**Pr > ChiSq**
Intercept	1	-4.6244	0.0331	19481.8055	<.0001
Variable1	1	-0.6594	0.0915	51.9431	<.0001
Variable2	1	-0.808	0.1087	55.285	<.0001
Variable3	1	-1.0729	0.1299	68.2364	<.0001
Variable4	1	-0.7717	0.1522	25.7191	<.0001
Variable5	1	-0.6249	0.1209	26.7265	<.0001
Variable6	1	-0.5949	0.1017	34.2234	<.0001
Variable7	1	-0.7686	0.1328	33.5039	<.0001
Variable8	1	-0.8662	0.127	46.5192	<.0001
Variable9	1	-0.6139	0.102	36.2218	<.0001
Variable10	1	4.1617	0.4703	78.3167	<.0001
Variable11	1	-0.7869	0.1619	23.6156	<.0001
Variable12	1	-1.7812	0.3077	33.5023	<.0001
Variable13	1	-0.7769	0.2051	14.3534	0.0002
Variable14	1	-0.9557	0.1647	33.6808	<.0001
Variable15	1	1.8601	0.2732	46.346	<.0001
Variable16	1	-0.6541	0.1612	16.471	<.0001
Variable17	1	-0.3991	0.1413	7.9739	0.0047
Variable18	1	-1.3777	0.3014	20.8902	<.0001
Variable19	1	-0.9018	0.1954	21.2959	<.0001
Variable20	1	-0.5513	0.1477	13.9284	0.0002
Variable21	1	2.0655	0.3518	34.4646	<.0001
Variable22	1	-1.3933	0.3073	20.5558	<.0001
Variable23	1	-1.1836	0.3756	9.9286	0.0016

The above table shows the output of Maximum likelihood estimates which are calculated after fitting the model. The rule of thumb for the selection of the variables for next step of logistic regression is:

- **Estimated coefficients should be negative**

  The coefficient estimates obtained from the maximum likelihood estimation should be negative this is because we use the WOE versions of the raw variables in the model. The reason can be explained as:

  The logistic regression equation is

$$logit(p(x_i)) = log\left(\frac{p(x_i)}{1-p(x_i)}\right)$$

$$= \beta_0 + \beta_1 x_i$$

  Hence,

$$logit\left(\frac{Pr(Y_i = 1|X_i = x_i)}{Pr(Y_i = 0|X_i = x_i)}\right) = \beta_0 + \beta_1 x_i$$

  Where, x_i is the WOE version of the raw variable i.e. it has the calculated WOE of the group in which the raw variable falls.

$$x_i = ln\left(\frac{Distribution\ of\ Good_i}{Distribution\ of\ Bad_i}\right)$$

  Hence,

$$log\left(\frac{Pr(Y_i = 1|X_i = x_i)}{Pr(Y_i = 0|X_i = x_i)}\right) = \beta_0 + \beta_1 x_i$$

  In the above equation to the left hand side there is logarithm value of probability of default to probability of good whereas on the right hand side the x_i has WOE value of the particular group in which the raw variable for particular observation falls in. This WOE value is calculated as logarithm of probability of good to probability of default. The left hand side has probability of bad to probability of good and right hand side

has probability of good to probability of bad, which is inverse. Hence we expect the coefficients to be negative.

- **P-Value**

  The output table above shows the p-value in the last column which shows the significance of the variable in the model. The p-value depends on the confidence interval which is considered earlier. For example, if the confidence interval is set to be 99% the variable to be significant should have p-value below 0.01. But in this case since we have used the SLSTAY option of SAS PORC LOGISTIC statement we don't need to worry as it will keep only those variables which have p-value less than the applied threshold. Variable17 and Variable 23 has the p-value slightly higher i.e. 0.0047 and 0.0016 respectively which is still less than 0.01 which our required threshold but greater than SLSTAY.

- **Wald Chi-Square**

  We generally get the wald chi-square in the output of SAS PROC LOGISTIC. This is a test of hypothesis used for deciding the significance of variable. The variables in the output are arranged according to the wald chi-square value in descending order. Basically, it shows the importance of the variable in the model, it is a test in which we come to know how well the target can be predicted if only that variable is used in the model i.e. regressing dependent variable with only that particular independent variable. In case of disparity amongst the stake holders over the use of particular variable, the wald chi-square makes it easy to decide whether that particular variable can be dropped or not from the model. As the variable with high wald chi-square is

important variable and cannot be dropped were as one with low wald chi-square can be dropped if needed.

- **Variance Inflation Factor (VIF)**

  The variables present in the logistic regression are checked for the multicollinearity in the initial stages, but we usually select two or more variables from each uncorrelated group depending on various parameters discussed earlier. Hence, it becomes important to check for multicollinearity between the variables after running the logistic regression. This is then again checked by variance inflation factor (VIF). The VIF is calculated in following manner:

  Step1:

  First we run the ordinary least square regression with Yas the dependent variable and $X_1, X_2, X_3, \ldots, X_k$ as independent variable. The equation will be

$$Y = \beta_0 + \sum_{i=1}^{i=k} \beta_i X_i$$

  Step2:

  We calculate the VIF factor for $\widehat{\beta_i}$ with formula

$$VIF_i = \frac{1}{1 - {R^2}_i}$$

  Where ${R^2}_i$ is the coefficient of determination, calculated by regressing one of the independent variable with all other independent variables. Hence, we can get VIF for all the variables by regressing that particular variable with other independent variables.

  Step3:

The magnitude of multicollinearity is analysed y considering the size of $VIF(\widehat{\beta}_\iota)$. The rule of thumb is that if $VIF(\widehat{\beta}_\iota) > 2$ then the multicollinearity is high. Hence, from the formula we can see that the permissible R^2 value should be 0.5.

VIF

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	1	0.01578	0.000315	50.02	<.0001	0
Variable1	1	-0.00387	0.000855	-4.53	<.0001	1.82262
Variable2	1	-0.01045	0.00129	-8.12	<.0001	1.39507
Variable3	1	-0.00911	0.00125	-7.32	<.0001	2.67414
Variable4	1	-0.01206	0.00145	-8.34	<.0001	1.06776
Variable5	1	-0.00524	0.00126	-4.14	<.0001	1.80305
Variable6	1	-0.00442	0.00115	-3.86	0.0001	1.55375
Variable7	1	-0.00839	0.00133	-6.29	<.0001	1.39722
Variable8	1	-0.00438	0.00104	-4.2	<.0001	1.59729
Variable9	1	-0.00546	0.000854	-6.39	<.0001	1.79581
Variable10	1	0.04041	0.00462	8.75	<.0001	19.63063
Variable11	1	-0.0098	0.00184	-5.31	<.0001	2.02579
Variable12	1	-0.01723	0.00303	-5.68	<.0001	1.30139
Variable13	1	-0.0117	0.00245	-4.77	<.0001	5.68579
Variable14	1	-0.00625	0.00136	-4.61	<.0001	2.31171
Variable15	1	0.01613	0.00276	5.84	<.0001	3.15344
Variable16	1	-0.00398	0.00138	-2.88	0.004	1.81189
Variable17	1	-0.00324	0.00144	-2.25	0.0244	3.25111
Variable18	1	-0.01103	0.0027	-4.09	<.0001	7.57697
Variable19	1	-0.00883	0.00189	-4.66	<.0001	4.93148
Variable20	1	-0.00557	0.00169	-3.3	0.001	1.64844
Variable21	1	0.0312	0.00382	8.16	<.0001	15.81984
Variable22	1	-0.02385	0.00324	-7.35	<.0001	13.00702
Variable23	1	-0.00774	0.00371	-2.09	0.0368	3.82985

In the above table we can see the VIFs of the variables. The variables which are marked yellow are the ones which have VIF >2 and these variables are not be used in next logistic regression. These variables should be checked from the business point, as

some variables might represent the same information, in that case anyone variable which is most important should be kept in the model. For example we might have variables such as Home loan enquiries in last 12 months (1), Secured Core enquiries in last 12 months (2) and Home loan enquiries in last 12 months (3), in this case the first variable and the third variable are from the same pool which are just a time variant so if we want to choose amongst them we should choose the first as Home Loan is a high vintage product and we should consider long time variant option. Amongst the first and third if there is correlation we should consider the second variable as it has high information, because secured core consists of many products such as home loan, auto loan, etc. so this variable would have much higher data representation.

The fitting of the logistic regression model is an iterative process, in which various variables are added and removed from the model. These variables are selected on some statistical measures and business insights. It is very easy to select the variables from the statistical metrics but becomes difficult to validate and prove the importance of the variable in the model. The key points to remember while selection of variable and improvise over the variables appearing in the model:

- **For bureau variables**

  The products are the main concern in this source of data. The products like home loan, auto loan, secured core loan, etc. are high ticket size and shows that an applicant may have high amount of obligation but at same time it reduces the chance of credit hungriness of the applicant, as these products are not bought frequently. Secured core loan secured others may also consist of the products in which the loan is taken against mortgage. Hence, these products are kind of safe for bank's portfolio due to mortgage

with bank from which the loan can be recovered. While the products such as credit cards, personal loan, unsecured core, unsecured others, etc. are the products which show the credit hungriness of the applicant as these products have higher rate of interest and are usually avoided by individuals. But nowadays credit cards are taken by everyone and the performance of an individual over this product is very important to gauge the characteristic of an individual. There are various variables of these products and their time variants which we had discussed earlier. The vintage variables of all the products show the availability of performance data with bureau and hence become very important variables over all. The delinquency and enquiry variables are the second most important variables after vintage. The number of closed or opened trade lines variables are the third important variables which show the count of current obligation of the individual. Utilization and balance variables show the magnitude of the obligation in some form of an individual. There are time variants such as 3, 6, 12, 24 months. For the products of high ticket size the variable variants with higher time frame such as 12 or 24 months are preferred over rest. It is very important that the variables in every stage of the model should cover every pool of data such as enquiry, utilization, delinquency and vintage, and the efforts should be made to include representation of all pool of data in the final model. Further if the model has more than two variables from the same pool, then the business insight and intuitive knowledge is required for final selection of strong representative over other variables.

- **For liability variables**

  The variables in the liability space are obtained from various channels of transactions such as average balance, branch, cash, cheque, mobile banking, NEFT/RTGS, internet banking, point of sale, ATM, etc. There are further credit and debit variants describing the number and amount of transactions for all the channels. There are also

time variants of these variables such as 1, 3, 6, 9, 12 months. These variables are raw variables, but it is always preferred to use ratio variables (e.g. proportion of debit amount from ATM to total debit amount) over the raw variables. Since the proportion variables seem to be more stable and are representative of larger pool (as in the above example the variable describes debit amount using ATM and total debit amount from account).Those balance variable should be considered which have a higher time span, i.e. variables which have considered more number of months for calculation of average balance. The balance variable shows upto certain extent individuals ability to repay, his income and his pattern of expenditure. Considering the balance of every variable will not be a good option because the balance will not be stable for every month. But considering average of balance over months will smoothen the effect of outlier and will provide a stable balance representation. It is very important that the variables in every stage of the model should cover every pool of data such as proportion variables, balance, cash and ATM, and the efforts should be made to include representation of all pool of data in the final model.

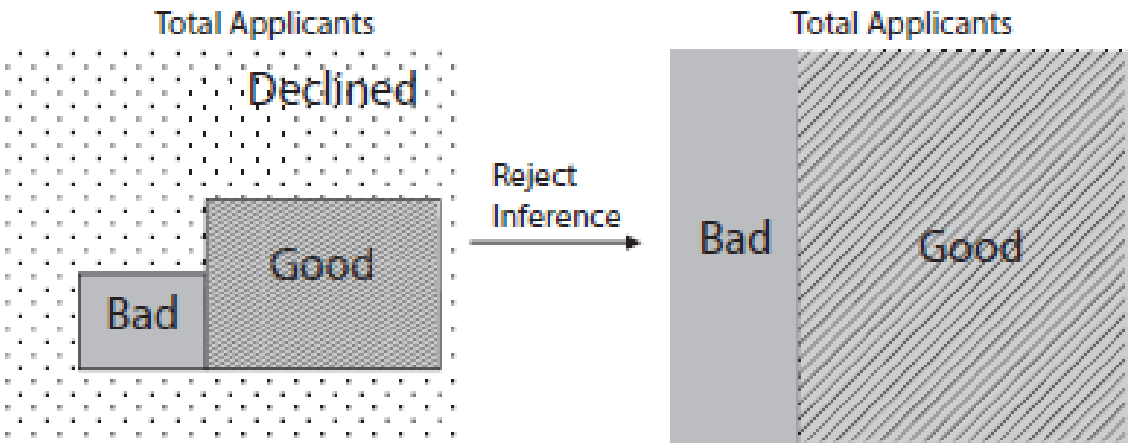
- **For demographic variables**

  The demographic space mostly consists of nominal variables such as employment type, age, experience in job, residential status, region, area etc. The variables as interacted with each other so that a single variable can capture more information and be more stable. Since variable reduction is not performed over the demographic variables, utmost care is to be taken to check for multicollinearity problem while final inclusion of the variable in the model. The demographic variables are mostly included with some earlier experience and business knowledge. The insight from all the stake holders acts important role in final selection of variables in the model.

4.7 REJECT INFERENCE

The model development analyses performed before this point was simply on the accounts whose performance was known with us. These samples are commonly called as "Known Good/Bad Sample". But in the real time the model or the application scorecard is implemented to predict the behaviour of all the applicants who are good (i.e. having low probability of default) and bad (i.e. having high probability of default). Hence if the model is developed on only on the previously approved applicants, its prediction would be inaccurate and biased ("sample bias"). This is true when previously accept/decline decisions were made systematically and not randomly; that is the accept population is a biased sample and has no representation of the rejects. A method is required in which the cases whose performance is missing can be handled.

Reject inference is a process in which the performance of the previously rejected application can be analysed to find out the behaviour (i.e. to assign the target class). Just like some bads in the population that is approved, there will be some goods that have been rejected. This process gives relevance to the scorecard development process by recreating the population performance for a 100% approval rate (i.e., obtaining the "population odds").



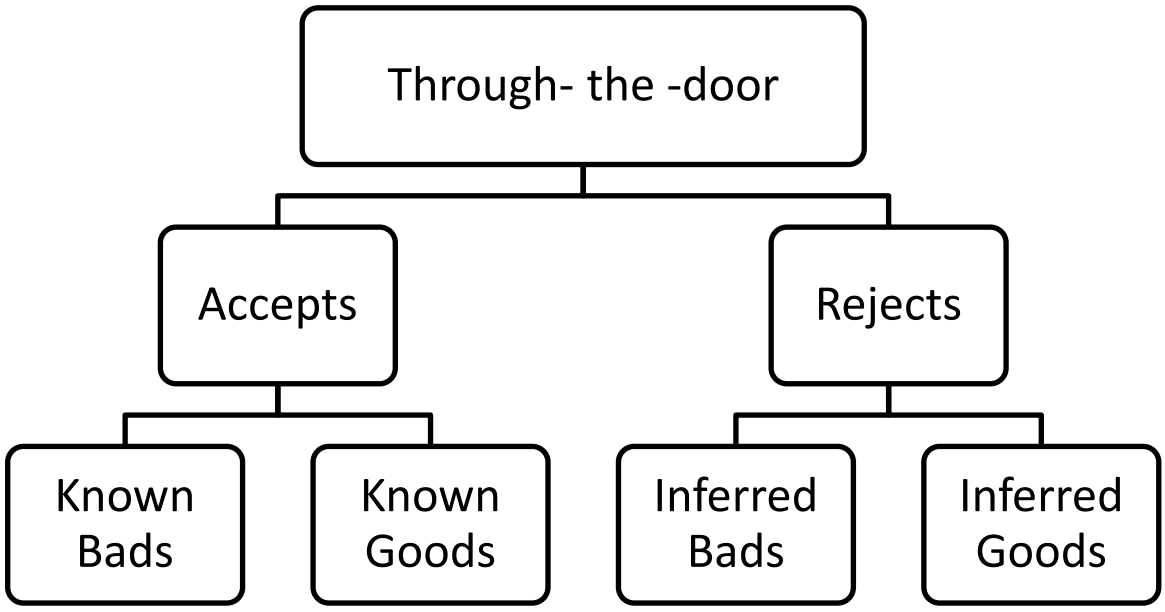
The picture above shows how the population performance is recreated. The left side shows the picture before inference, with known goods, bads and declined applications. In order to develop a score card for the total applicants, all the applicants should be tagged as good or bad as shown right side of the picture.

4.7.1 Reasons for Reject Inference

The first reason of performing reject inferencing is that the final scorecard is run on the total population (i.e. accepts, rejects and account not opened), but the model is only trained on the accept base which are considered in the development sample. Hence, the issue of sample bias will arise as mentioned earlier.

The reject inferencing considers the influence of the past decision into the scorecard development process. The reject inferencing considers through the door population in the development base, hence when the model is developed it is trained over all the through the door population and not just accepts. Since the model has seen all the characteristics of accept and reject population, it is better equipped to predict the accurate performance of the applicant coming in. For example, consider a bank that traditionally used to accept the applications whose score is greater than 200 and below this threshold would be rejected. In case of "Known Good/Bad Sample" we would have only used these accepts for the development and the model would have never experienced the characteristics of the applicants with score less than 200. Also further, if the bank thinks it is very conservative, losing its potential customers and is ready to increase the risk, it starts accepting the applicants with score between 170 and 200. But the bank has never accepted such kind of customers in past and hence will not have any idea about the incremental level of risk involved in lowering the cutoff. But reject inference, by allowing them to estimate the bad rates by score of those who were previously rejected, will help them make this decision. It

also creates the opportunity to gauge the future performance using the "Swap Set" analysis, which we will see in the further chapter. The through the door population can be shown as:



4.7.2 When Reject Inference should be used

The importance and impact of the reject inferencing is dependent on the rate of acceptance of the applications and the level of confidence in previous credit granting policy. A high level of confidence with very high approval rate totally creates an assumption that "all reject are bad" and very low level of confidence leads to an assumption that there is nearly random adjudication. This reduces the need for reject inferencing. In sub-prime lending, medium approval rate may allow the assumption of "all reject are bad", if adjudication made with high level of confidence.

In case of high approval rate and correspondingly high bad rate, since the approved population is fairly close to the through the door population, there is no need of reject inferencing. The same is applicable in institutions wherein the adjudication is done randomly

or using inaccurate adjudication tool. In case of high approval rate and low bad rate it can be safely be assumed that "all reject are bad", hence there is no need of reject inferencing.

In environments where there is low or medium approval rates and low bad rates, reject inference helps in identifying potential to increase the market share with risk-adjusted strategies. In case where the institute thinks that they are currently rejecting the creditworthy applicants, reject inference will help in identifying these applicants. Reject inference can also be used wherein there is high confidence in adjudication, but it is believed that the bad rates can be reduced through better selection.

4.7.3 Reject Inference Techniques

There are many techniques used for reject inferencing in the industry. Here are few of them discussed below.

1. Assign all rejects to be bads.
2. Assign rejects in the same proportion of goods to bads as in the accepts base.
3. Ignore all the rejects, firstly select all the accept accounts score them and then select those accounts who are below specified cutoff.
4. Approve all applications for a specific period of the base enough to generate a sample, then approve all the applications above specified cutoff but only randomly select below cutoff (upto 10 or 20 points) and randomly sampling the rest.
5. Similar In-House or Bureau data based method, i.e. considering the performance on any other product with the bank of the rejected application or to consider the performance of the same product with any other lender who has accepted the application.
6. Augmentation in Historically Judgemental decision-Making Environment
7. Simple Augmentation
8. Augmentation

9. Parcelling
10. Fuzzy Augmentation
11. Iterative Reclassification
12. Nearest Neighbour (Clustering)
13. Memory-Based Reasoning

In this project there are three such reject inferencing methods which are used from the above list. These methods are explained elaborately below:

4.7.3.a Fuzzy Augmentation

This method is the most scientific and feasible method. The "through the door" population usually consists of accepts, rejects and account not opened. The bank generally has the performance of the accepts and hence we can classify the accepts as good or bad depending on the bad definition. But in case of the rejects and account not opened, since these applicants are either rejected or have not opened the account we do not have their performance and hence cannot classify them as good or bad. The accounts not opened are the applicants who have applied for the loan and have got accepted but have not taken up the loan from bank. This shows that though we do not have the performance of the rejects and account not opened, they are of different characteristics and should be treated differently while classification. In this method we assign each rejects as partial "good" and partial "bad" class. The full process involves classification and then augmentation using following steps:

Step1 Classification

- Score rejects with "Known Good/ Bad model" with each reject application considered twice, firstly classified as good and secondly classified as bad. Also, score account not opened applications with "Known Good/ Bad model" only once.

- Determine probability of good(i.e. $p(good)$)and probability of bad (i.e. $p(bad)$) for each reject application, and probability of good(i.e. $p(good)$) for each accept but account not opened.
- Assign each reject as a partial good and a partial bad, by creating two weighted cases from each reject.
- Weigh each rejected application with $p(good)$ and $p(bad)$. The rejected application firstly weighted by the $p(good)$ multiplied by a weight 0.5 and then the $p(bad)$ multiplied by a weight 0.5. The accept but account not opened applications are weighted by the $p(good)$ multiplied by a weight 1.

Step2 Augmentation

- Combine rejects and account not opened with accepts by adjusting for approval rate, $p(approve)$.

Since combining accept and reject on one-to-one basis would imply that both have equal chance of being in the dataset, the additional weight factor is introduced. This method not only considers the probability of reject being bad but also its probability of getting accepted. This is a better approach as it assigns measure of importance to a reject in final sample. Using partial classification makes it better than those who use arbitrary measures to do so.

4.7.3.b Similar In-house or Bureau Data Based Method

In this method, the performance data for applicants rejected and account not taken up for one product but approved and have taken up for similar kind of product with same lender. The another method is to consider the performance with the credit bureau of the applicant rejected

and account not taken up by one creditor but approved and account taken up for similar product with another lender.

For example, a bank rejects an applicant but the same applicant is accepted for same kind of loan by another lender. Hence, for such kind of applicant the bank may not have his performance in-house but his performance with another lender will be there with the bureau. This can be used to classify the applicant as good or bad. The other case is when the applicant is accepted but due to some reason does not take up the loan, but he may have taken loan from another lender. In that case the performance from bureau data can be obtained.

This method approximates the actual performance, but is has few drawbacks. Firstly there would be regulatory hurdles, because once the applicant is rejected or has not open account, the bank has no right to track the applicant's performance. Hence in some jurisdiction the bureau data for such applications won't be available. In that case if the bank is multiproduct bank and the same applicant has applied for any other product in the latest time period, we can get the delinquency report for the same applicant and use and use to classify good or bad. The bureau information for this kind of application can only be used if he has obtained similar credit from another lender in similar time frame (i.e. soon after getting declined). Further, the "bad" definition of the Known Good/ Bad model is calculated for the rejects and account not opened cases from the delinquencies obtained from credit reports or in-house data.

4.7.3.c Combination of fuzzy augmentation and bureau data based method.

In the bureau data based method, it is always most likely that the applicant who is rejected by one bank is rejected by other lenders; hence there would be very few applicants who would be accepted by other vendors, leading to reducing the potential sample size. Whereas, if we use just the fuzzy augmentation method we would be looking at only partial probability of the applicant going good or bad; hence turning blind eye towards the actual performance of the

applicant. But combining the two methods helps to use the whole sample size without any reduction and also use the actual bureau performance of applicant if present. In this method the applicants whose bureau performance is present, the bureau delinquency on the similar products as applied for will be considered for the bad definition, and for those applicants whose information delinquency is not present the fuzzy augmentation method is used to classify good or bad.

The algorithm for the combined method is shown below:

Step1 Fuzzy logic method

1. Consider Reject application twice. Firstly by assigning it to be bad and secondly by assigning it as good.
2. Consider Accept but account not opened twice. Firstly by assigning it to be bad and secondly by assigning it as good.
3. Scoring the Known Good/ Bad model on this data, we will get probability of good and probability of bad for each application.
4. Weigh each application by :

   For reject application tagged as good

   Fuzzy Weight=$p(good)$

   For reject application tagged as bad

   Fuzzy Weight=$p(bad)$ X 2

   For account not opened application tagged as good

   Fuzzy Weight=$p(good)$

   For account not opened application tagged as bad

   Fuzzy Weight=$p(bad)$

Step2 Bureau Bad Method

1. Calculate the "bad" definition from the bureau data based on the similar product as applied by the applicant, as per the Known Good/ Bad model.
2. Weigh each application by :

   Bureau Bad Weight = 0.5

Step3 Augmentation

1. Combine accepts with the rejects and account not opened.
2. The rejected applicants who have bureau bad will appear three times in the data. Firstly observation tagged with bureau bad, secondly tagged as partial bad and thirdly tagged as partial good from the fuzzy step.

   Weigh each application by:

   New Weight= Fuzzy Weight (for fuzzy observation)

   New Weight= 1 (for bureau bad observation)

   Finally,

   Final Weight= New Weight X 0.5

   Where, 0.5 is Bureau Bad Weight.
3. The rejected applicants who do not have bureau bad will appear two times in the data. Firstly observation tagged as partial bad with probability of bad and secondly tagged as partial good with probability of good from the fuzzy step.

   Weigh each application by:

   New Weight= Fuzzy Weight (for fuzzy observation)

   Finally,

   Final Weight= New Weight
4. The account not opened applicants who have bureau bad will appear three times in the data. Firstly observation tagged with bureau bad, secondly tagged as partial bad and thirdly tagged as partial good from the fuzzy step.

Weigh each application by:

New Weight= Fuzzy Weight (for fuzzy observation)

New Weight= 1 (for bureau bad observation)

Finally,

Final Weight= New Weight X 0.5

Where, 0.5 is Bureau Bad Weight.

5. The account not opened applicants who do not have bureau bad will appear two times in the data. Firstly observation tagged as partial bad with probability of bad and secondly tagged as partial good with probability of good from the fuzzy step.

   Weigh each application by:

   New Weight= Fuzzy Weight (for fuzzy observation)

   Finally,

   Final Weight= New Weight

6. The accounts which are accepted are weighed as :
7. Final Weight= 1

Hence, after combination of all these above datasets we get the bureau bad and fuzzy inferred data which can be used further for All Good/ Bad model development. Henceforth for either for model fitting, for any other frequency counting procedure or for calculating of WOE, etc. it is very important to consider the Final Weight related to every observation.

The logistic model can be fitted in SAS Enterprise Guide as:

```
PROC LOGISTIC DATA= BASE NAMELEN=50;
MODEL Bad_Flag(EVENT = '1')=
Variable1
Variable2
Variable3
Variable4
Variable5
Variable6
Variable7
Variable8
Variable9
Variable10
/SELECTION=STEPWISE SLSTAY=0.001;
WEIGHT FINAL_WEIGHT;
OUTPUT OUT=TRAIN PRED=P_1;
SCORE DATA= VALIDATION OUT= VALIDATION;
SCORE DATA= OOT OUT= OOT;
RUN;
```

4.7.4 Verification

After successful completion of reject inference process, simple verification exercise is done to check on the results. This includes:

- Compare bad rates of the inferred base with the approved base and applying industry rule of thumb as discussed earlier. For example, if the previous model was good and the approval rate was high, then the inferred bad rate should be three of four times that of the approved base. A medium approval rate may have twice bad as in the inferred base.
- Compare weight of evidence or bad rates of grouped attributes for pre-inferred and post-inferred datasets.

Different reject inference techniques and parameters can also be tested using "fake" rejects. This involves splitting the approved population into arbitrary accepts and rejects, for example a 70/30 split. A model developed on the 70% "approved" sample can then be used for inferring the performance of the remaining 30%. Since the actual performance of the 30% "rejects" is already known, misclassification can be used to gauge the performance of each reject inference method. Once reject inference is completed, the combined dataset (of approves and inferred rejects) is created and used for the next phase of scorecard development. Now that sample bias is resolved, this final scorecard is applicable to the entire "through the door" population.

4.8 FINAL SCORECARD PRODUCTION

After creation of the inferred dataset, the final scorecards are produced by running the initial characteristic analysis and statistical algorithm to fit the model (e.g. regression). The

statistical algorithm to be used is the one which is finalized in the Known Good/ Bad benchmark model. The variables which were finalized in the earlier model were based on the accepts, but then after getting the inferred base we need to perform the whole exercise of model development i.e. including variable reduction, variable selection, creation of new required variables, fitting model etc.

The next few things which are to be addressed further are scaling of score, validity of points of allocation, misclassification and scorecard strength.

4.8.1 Scaling

The scaling refers to range and format of scores in a scorecard and the rate of change in odds for increases in score. Scorecard scores can take several forms with decimal or discrete number scores:

- Where the score is the good/bad odd or probability of bad (e.g., score of 6 means a 6:1 odd or 6% chance of default)
- With some defined numerical minimum/maximum scale (e.g. –1,0–1000, 150–350) with a specified odds ratio at a certain point (e.g., odds of 5:1 at 500) and specified rate of change of odds (e.g., double every 30 points)

The choice of scaling does not affect the predictive strength of the scorecard. It is an operational decision based on considerations such as:

- Implementability of the scorecard into application processing software.
- Ease of understanding by staff (e.g., discrete numbers are easier to work with).
- Continuity with existing scorecards or other scorecards in the company. This avoids retraining on scorecard usage and interpretation of scores.

There are various scales used in the industry. One of the scales used most commonly in the industry is discrete scores scaled logarithmically, with odds doubling at every 30 points. Example of such scaling is shown below:

4.8.2 Scaling calculation

The relationship between odds and scores can be represented as linear transformation:

$$Score = offset + Factor\ ln(odds)$$

Where the scorecard is being developed using specified odds at a score and specified "points to double the odds"

(pdo), the factor and offset can easily be calculated by using the following simultaneous equations:

$$Score = offset + Factor * \ ln(odds)$$

$$Score = offset + Factor * \ ln(2 * odds)$$

Solving above equation for *pdo*

$$pdo = \ Factor * \ ln(2)$$

$$Factor = pdo\ /\ln(2)$$

$$Offset = Score - \ Factor * \ ln(odds)$$

For example, if a scorecard were being scaled where the user wanted odds of 50:1 at 600 points and wanted the odds to double every 20 points (i.e., pdo = 10), the factor and offset would be:

$$Factor = 30/\ln(2) = 43.281$$

$$Offset = 600 - \{14.427 * l\,n(50)\} = 543.39$$

The corresponding score can be calculated as:

$$Score = 543.39 + 43.281 * ln(odds)$$

This formula can be used to generate scores for any model where the probability of bad or odds is generated. These models will include any other modelling techniques including regression as explained in previous section. Since scorecard here is developed using weight of evidence as input, the previous relationship can be modified as:

$$Score = ln(odds) * Factor + Offset$$

$$= -\left(\sum_{j,i=1}^{k,n} (WOE_j * \beta_i) + a\right) * factor + offset$$

$$= -\left(\sum_{j,i=1}^{k,n} (WOE_j * \beta_i) + \frac{a}{n}\right) * factor + offset$$

$$= \sum_{j,i=1}^{k,n} \left(-\left(WOE_j * \beta_i\right) + \frac{a}{n}\right) * factor + \frac{offset}{n}\right)$$

Where,

WOE = weight of evidence for each grouped attribute

β = regression coefficient for each characteristic

a = intercept term from logistic regression

n = number of characteristics

k = number of groups (of attributes) in each characteristic

The above formula will help to calculate the scores assigned to the grouped attribute for every characteristic in the scorecard developed, and summing al the scored for each attribute we can get the final score. At this point it is worth noting that the trend and difference

between weights of evidence in the grouped attributes will affect the points assigned using this approach. This underscores the emphasis placed on both maintaining a logical trend of WOE and trying to maximize the differences in the WOE of successive groups.

4.8.3 Adverse Codes

In some jurisdictions, it is necessary for the lenders to give the reason of decline to the borrowers. This is called as adverse codes. The adverse scores are generated by obtaining "neutral score". The neutral score is obtained by putting WOE as 0, once we obtain factor and offset in the above equation. The equation of neutral score hence will be:

$$-\left(\frac{a}{n}\right) * factor + \frac{offset}{n}$$

For any applicant whose score is below the neutral score for any characteristic in the scorecard, infers that the probability of going bad on this characteristic is more than 50% (since, if WOE=0 then probability of being good or bad is 50%).

4.9 CHOOSING A SCORECARD

The scorecard development is an iterative process in which a scorecard developer creates various scorecards which are then compared for various metrics to select the best one amongst them. The best scorecard is selected based on the statistical metrics and business measures.

- **Misclassification**

  The scorecard is developed to predict the probability of being good or bad. Most importantly, predictive models are used for differentiating between good and bad cases. The misclassification statistics provides good help to find out right scorecard. For the operational and business purpose the companies decide a cut-off above which the cases are accepted and below it the cases are declined. Hence, the actual good case

may be tagged as bad and may be declined, and vice versa. Hence, the final scorecard needs to be chosen such that the level of misclassification is minimized.

There are several metrics to gauge the level of misclassification and compare different scorecards. These measures compare the number of true goods and bads with the number of predicted goods and bads for a certain cut-off. "Goods" and "Bads" here refer to cases above and below the proposed cut-off. These measures are based on confusion or misclassification matrix as given below.

		Predicted	
		Good	**Bad**
Actual	**Good**	True Positive	False Negative
	Bad	False Positive	True Negative

A better scorecard will have maximum "true" cases and minimum "false" cases. The four main measures used to gauge misclassification:

- Accuracy: (true positives and negatives) / (total cases)
- Error rate: (false positives and negatives) / (total cases)
- Sensitivity: (true positives) / (total actual positives)
- Specificity: (true negatives) / (total actual negatives)

These statistics are interpreted as:

- False Positive—Acceptance of bads
- True Positive—Acceptance of goods
- False negative—Decline goods
- True Negative—Decline bads

Based on this metrics the bank can decide the acceptance and rejection strategy. If the bank decides to choose scorecard to reduce losses, then the bank will choose the scorecard with maximum specificity. Whereas if bank decides to compromise on risk

and to increase the market share, then will maximize the sensitivity my minimizing rejection of good and approving some bads.

Methods used to compare scorecard predictive power:

- **AIC (Akaike's Information Criterion)**

  Penalizes for adding parameters to the model. Small values of AIC are preferred.

- **c-statistic**

  This is the most powerful nonparametric two-sample test, and the measure is equivalent to the area under the Receiver Operating Characteristic (ROC) curve, Gini coefficient, and the Wilcoxon-Mann-Whitney test. It measures classifier performance across all score ranges and is a better measure of overall scorecard strength. The c-statistic measures the area under the Sensitivity vs. (1 −Specificity) curve for the entire score range.

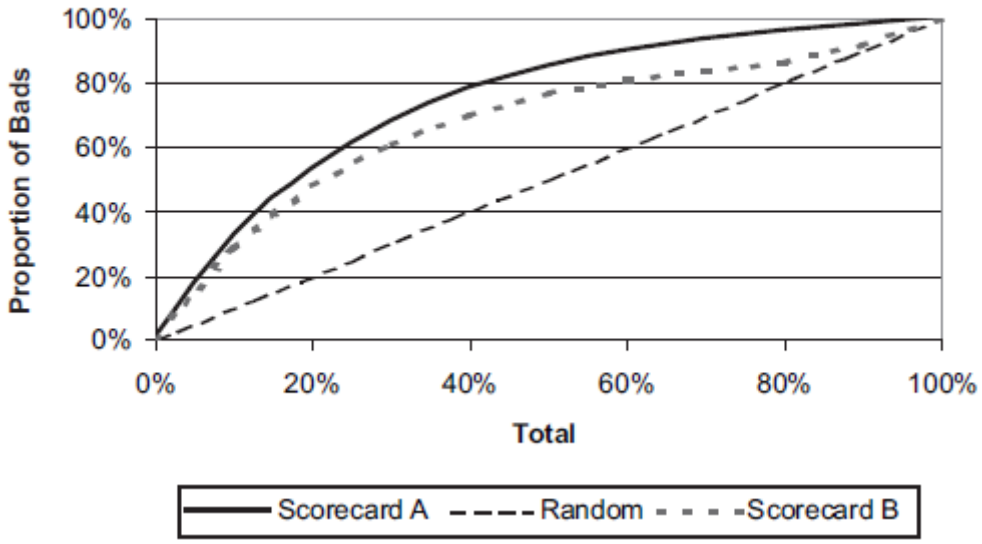
- **Kolmogorov-Smirnov (KS)**

  This measures the maximum vertical separation (deviation) between the cumulative distributions of goods and bads. The weakness of this method is that the separation is measured only at the one point (which may not be around the expected cutoff point), and not on the entire score range. If the intended scorecard cutoff is at the upper or lower range of scores, this method may not provide a good indication of scorecard comparison. In such cases, it might be better to compare the deviation at the intended cutoff, since that is where maximum separation is most required.

- **Lorenz Curve**

  A measure similar to the ROC curve used in the industry to compare models is to plot the distribution of "bad" cases and total cases by deciles across all score ranges. This is referred to as the Lorenz curve, and measures how well a scorecard isolates the

bads and goods into selected deciles. An example of a Lorenz curve is shown in diagram.



In diagram, for the bottom 60% of the total sample, Scorecard "A" isolates about 90% of all bads, whereas scorecard "B" only isolates about 80%. Therefore, scorecard "A" displays stronger performance. Note that the ratio of the area between a scorecard's Lorenz curve and the 45 degree line, to the entire triangular area under the 45 degree line, is also equivalent to the Gini index.

It is important here to compare scorecard performance in operationally logical deciles, meaning that if the expected approval rate is about 60%, then performance should be compared at the 60% percentile mark. Comparing performance at the lowest 10% is irrelevant when what is needed at implementation is best performance, at 60% in this case. However, when dealing with scorecards such as bankruptcy or response, making comparisons at the lowest percentiles does make sense and should be done—since in these cases the objective is to isolate the worst/best few performers for action.

- **Gains Chart**

  Cumulative positive predicted value vs. distribution of predicted positives (depth).

- **Somers' D, Gamma, Tau-a**

  Based on the numbers of concordant and discordant pairs. These measures are related to the c-statistic.

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	75	**Somers' D**	0.552
Percent Discordant	19.8	**Gamma**	0.582
Percent Tied	5.2	**Tau-a**	0.011
Pairs	1.5E+08	**c**	0.776

The above table shows the output of the SAS PROC LOGISTIC to determine the performance of the model

Chapter 5

Results and Validation

This project was about developing the application scorecard for the Mortgage products of the bank. This involved a lot of business understanding and thorough knowledge of the product from various stake holders. The scorecard development process in general is a very systematic process which includes many stages as seen from the diagram in earlier chapter. The initial stage consists of developing a model only on accepts base. After the basic exploratory analysis and data cleaning the next most important stage is of the variable selection. Once the final set of strong variables is selected we move forward to fit a model using the machine learning techniques.

5.1 MODEL 1

5.1.1 KGB Model is developed on

1. Sample window : January 2012 to June 2014 (only accepts)
2. Performance window : Rolling window of 2 years starting from date of application
3. Bad definition : Ever 90 plus days past due in 24 months
4. Good definition : Current and 1 to30 days past due
5. Indeterminate : 30 to 90 days past due
6. Exclusions : Non Individuals
7. Development Base : January 2012 to March 2014
8. The development base is split into training and validation randomly as 70% and 30% respectively.

9. Out of Time Base : April 2014 to June 2014
10. The final base for KGB consists only of the accept applications from the through the door population. It has all the bureau, liability, demographic and other computed variables.

In process of creation of the KGB benchmark model we created many intermediate models and then finalised one model based on various statistical metrics.

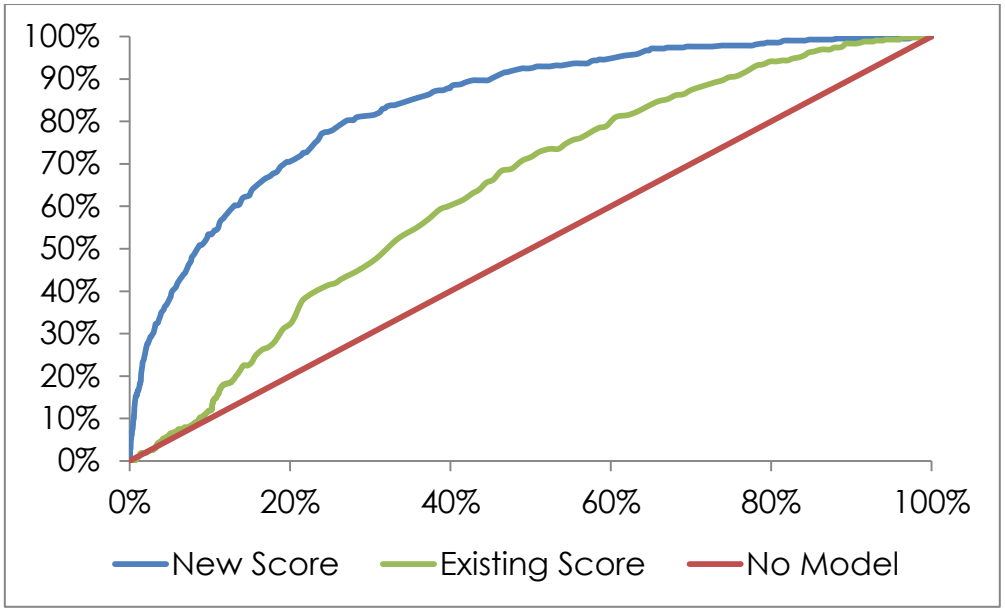
No.	**Variable Description**	**Source/ Category**	**IV**
1	Variable 1	Demographic	0.76
2	Variable 2	CIBIL	0.23
3	Variable 3	CIBIL	0.22
4	Variable 4	Demographic	0.21
5	Variable 5	CIBIL	0.21
6	Variable 6	CIBIL	0.14
7	Variable 7	Liability	0.13
8	Variable 8	CIBIL	0.12
9	Variable 9	Liability	0.12
10	Variable 10	Liability	0.11

The final KGB Model consists of 10 variables which are arranged on the basis of IV in the table above. The model seems to be good amongst all since it covers up whole pool of data as discussed earlier. It has strong representation of liability, bureau and demographic variables with strong IV.

5.1.2 Metrics for deciding the best model

5.1.2.a Lift Chart and GINI

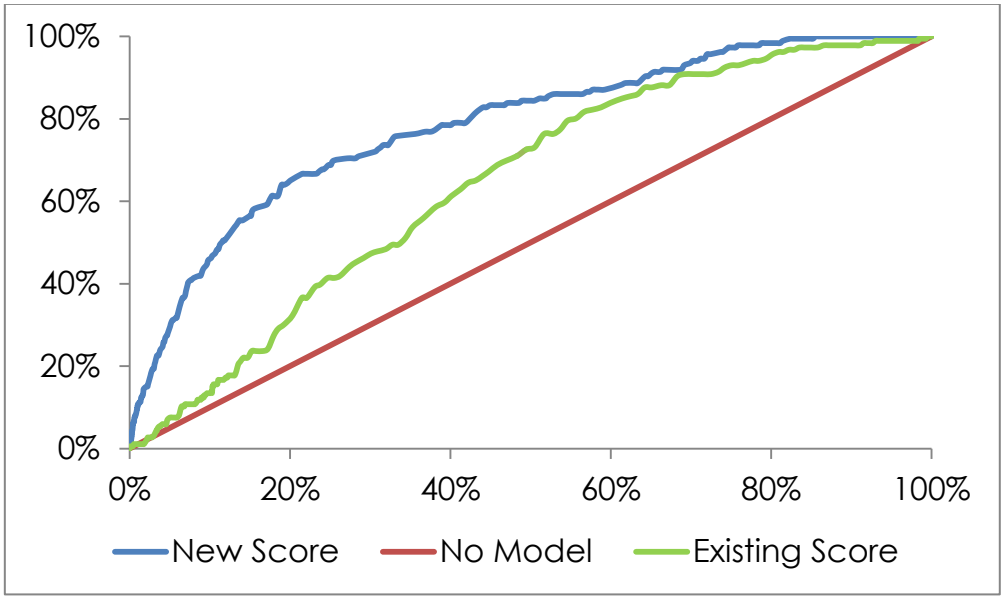
1. **Lift chart for the training base:**



Form the above lift chart we can infer the performance of the model. The approval rate for the mortgage product portfolio is around 20%; hence we will consider the cut-off at bottom 20% which will reject first two deciles of the score distribution. The Lorenz curve shows that the new model isolates around 65% of bad accounts in bottom 20% of the population, whereas the existing model isolates around 30% of bad accounts in bottom 20% of the population. This shows that there is a need for a new

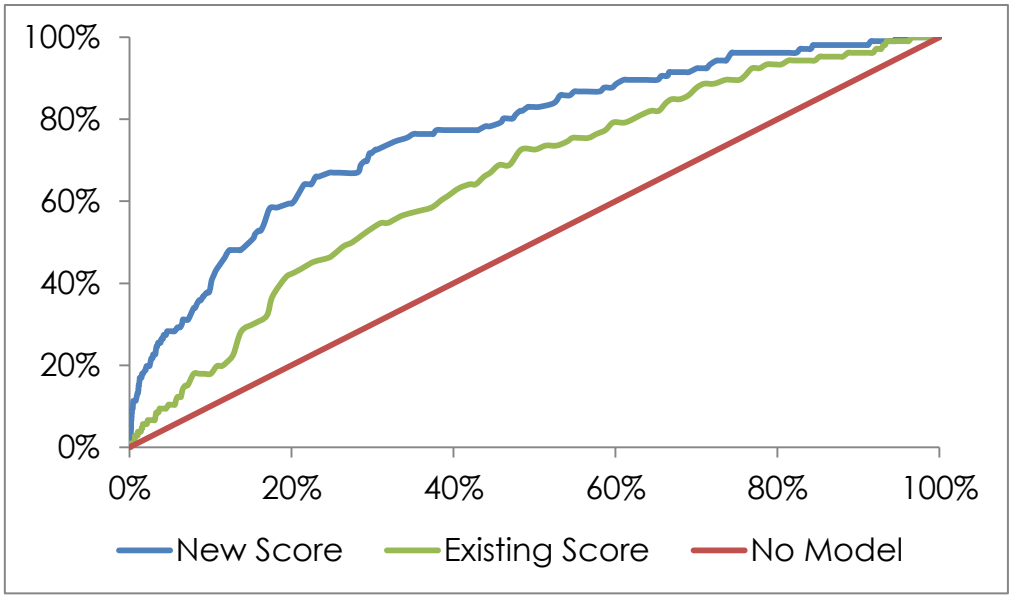
model and the new model has a better performance over the old model in the training base.

2. **Lift chart for the validation base:**



The Lorenz curve shows that the new model isolates around 55% of bad accounts in bottom 20% of the population, whereas the existing model isolates around 29% of bad accounts in bottom 20% of the population. This shows that there is a need for a new model and the new model has a better performance over the old model in the training base. This shows that there is a drop in Gini in the validation base as compared to the validation base. This drop is considerate and the model performance is stable and in acceptable range.

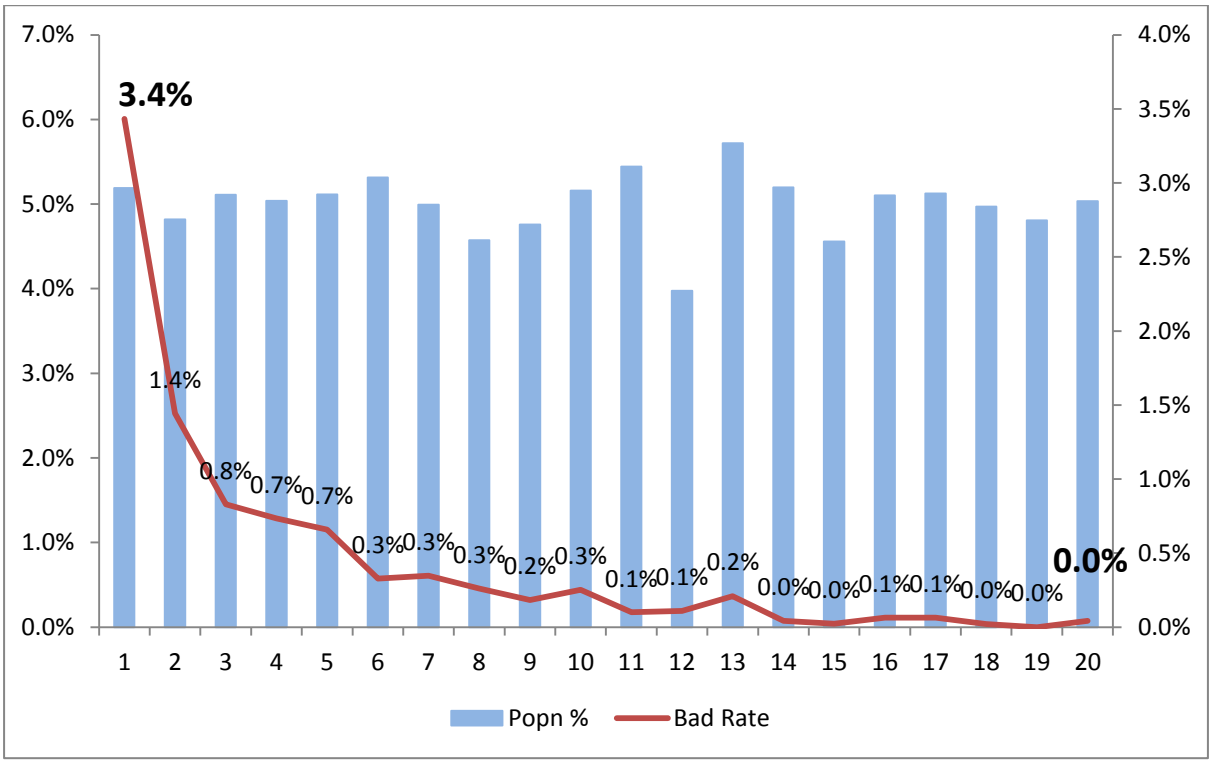
3. **Lift chart for the out of time base:**



The Lorenz curve shows that the new model isolates around 52% of bad accounts in bottom 20% of the population, whereas the existing model isolates around 30% of bad accounts in bottom 20% of the population. This shows that there is a need for a new model and the new model has a better performance over the old model in the training base. This shows that there is a drop in Gini in the validation base as compared to the out of time base. This drop in Gini in the out of time base is very close to the validation base and is in the acceptable range.

5.1.2.b Risk Ranking charts

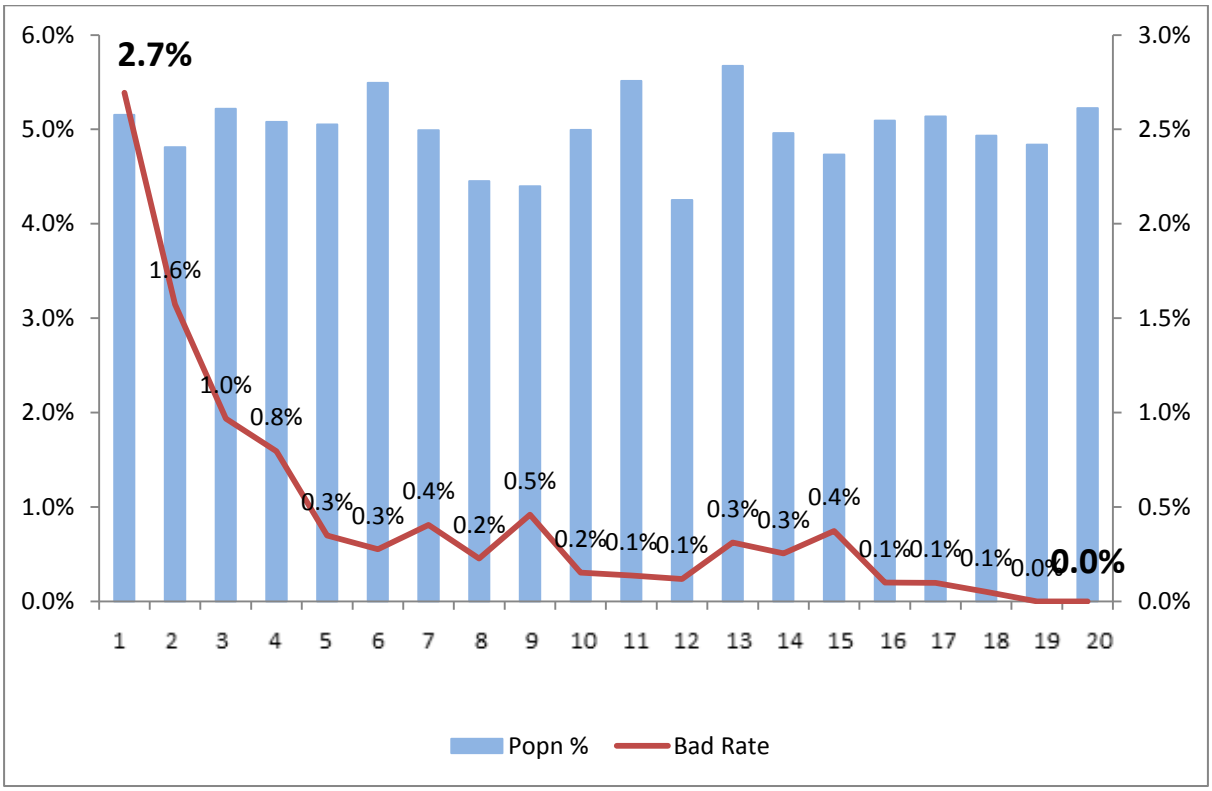
1. **Risk Ranking for the training base:**



The above chart shows the risk ranking of the training base. The scored population is distributed into twentiles i.e. twenty parts where each bin has approximately 5% of the total population based on the rank ordering of the score. Hence the lower twentiles will contain the applicants with lower score (i.e. "bad" accounts) and the higher twentiles will contain the applicants with higher score (i.e. "good" accounts).The bad rate in the lower twentiles is higher and since the approval rate is 80% we reject first 4 twentiles resulting in rejection of bottom 20% of the population. The first bin has bad

rate of 3.4% which is the highest, the second bin has bad rate of 1.4 %, the third bin has bad rate of 0.8 % and followed by the forth bin which has bad rate of 0.7%.Here we can see that the bad rate is gradually decreasing except some marginal peaks in the 10^{th} and 11^{th} bin and higher bins i.e. 18, 19 and 20 has 0% bad rate. Hence we can infer that the bad rate curve should be like a hockey stick, starting from a high point in the lower bins and flattening out towards the higher bins. When this kind of bad rate curve is observed we can say that the model is performing well.

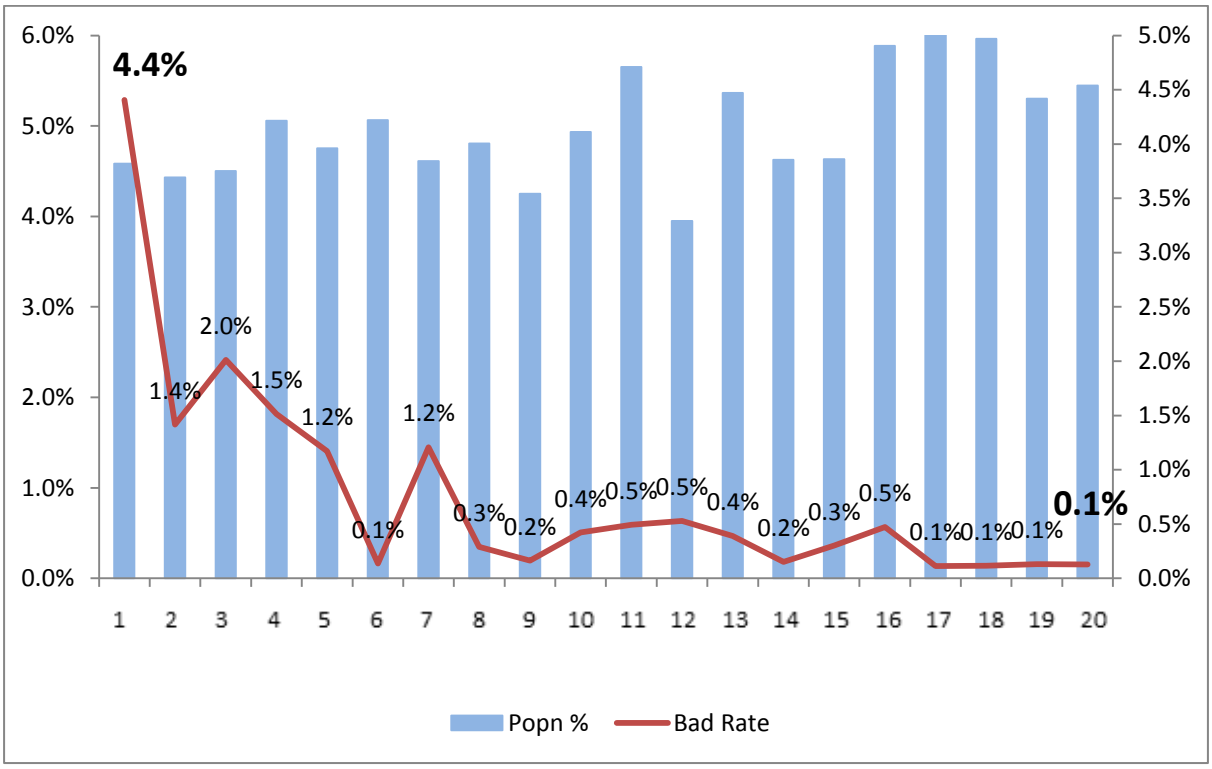
2. Risk Ranking for the validation base:



The first bin has bad rate of 2.7% which is the highest, the second bin has bad rate of 1.6%, the third bin has bad rate of 1% and followed by the forth bin which has bad rate of 0.8%.Here we can see that the bad rate is gradually decreasing except some

marginal peaks in the 9th and 15th bin and higher bins i.e. 19 and 20 has 0% bad rate. Hence we can see a sort of hockey stick curve of bad rate in the validation base risk ranking.

3. Risk Ranking for the out of time base:



The first bin has bad rate of 4.4% which is the highest, the second bin has bad rate of 1.4%, the third bin has bad rate of 2% and followed by the forth bin which has bad rate of 1.5%.Here we can see that the bad rate has drastically dropped from bin 1 to bin 2 which symbolises that the risk is very finely captured by first bin and hence lowering down for the further bins. There are many peaks which are observed over the spread of distribution, because the risk ranking is done on the out of time scored base and the model develop was on the train base. Hence there would be some amount

of difference between the actual development base risk ranking and test base risk ranking.

These are the metrics and the validations done on the model just to confirm the robustness of the model. Then this KGB benchmark model is finalised. This KGB benchmark model is presented to all the stake holders, who after having look at the Gini index risk rankings, lift charts, univariates of all the variables appearing in the model decide whether to go further or perform any changes like creation segmented model, extension of sample window, machine learning technique to be used for classification, etc.

In this project the above benchmark model was considered good and the recommendations were given by the stake holders to extend the sample window time frame, i.e. start the sample window from January 2012 to January 2011. Hence to increase one year of data so that model could see characteristics of few more number of bad accounts, since in mortgage products generally across the industry the bad rate is slight low as compared to other products. Stake holders also recommended trying new techniques like boosting algorithm for model development as discussed in earlier chapters. They also recommended to change the bad definition and to see a longer time frame than the rolling window to define the indeterminate and good accounts. The following changes were performed and the new updated KGB model was prepared.

5.2 Model 2

5.2.1 KGB Model 2 is developed on

1. Sample window : January 2011 to June 2014 (only accepts)
2. Performance window : Rolling window of 2 years starting from date of application
3. Bad definition : Ever 90 plus days past due in 24 months
4. Good definition : Current and 1 to30 days past due in first 24 months and less than 89 days past due post 24 months
5. Indeterminate : 30 to 89 days past due in first 24 months or (0 plus days past due post 24 months
6. Exclusions : Non Individuals
7. Development Base : January 2011 to March 2014
8. The development base is split into training and validation randomly as 70% and 30% respectively.
9. Out of Time Base : April 2014 to June 2014
10. The final base for KGB consists only of the accept applications from the through the door population. It has all the bureau, liability, demographic and other computed variables.

In this model we are trying three new techniques namely Logistic Regression, Logit Boost and Multi-layer Logistic Regression. Before going to fitting model we again do all kinds of cleansing, exploratory analysis and variable reduction, and then run model on the strong variables finally selected.

No.	Variable Name	IV	Logistic Regression	Logit Boost	Ensemble
1	Variable 1	0.32	-	-	Y
2	Variable 2	0.30	Y	Y	-
3	Variable 3	0.29	Y	Y	Y
4	Variable 4	0.26	Y	Y	-
5	Variable 5	0.26	Y	Y	Y
6	Variable 6	0.25	Y	Y	Y
7	Variable 7	0.24	Y	Y	Y
8	Variable 8	0.23	Y	-	Y
9	Variable 9	0.23	-	Y	-
10	Variable 10	0.22	Y	Y	Y
11	Variable 11	0.21	Y	Y	Y
12	Variable 12	0.17	-	-	Y
13	Variable 13	0.16	-	-	Y
14	Variable 14	0.16	Y	Y	-
15	Variable 15	0.14	Y	Y	Y
16	Variable 16	0.09	-	-	Y
17	Variable 17	0.09	Y	Y	Y
18	Variable 18	0.09	Y	Y	Y
19	Variable 19	0.08	Y	Y	-

The above table shows all the variables coming up in all three modelling techniques. The variables are arranged in the descending order of the IV and the indicator showing in which technique the variable is coming up.

5.2.2 Logistic Regression

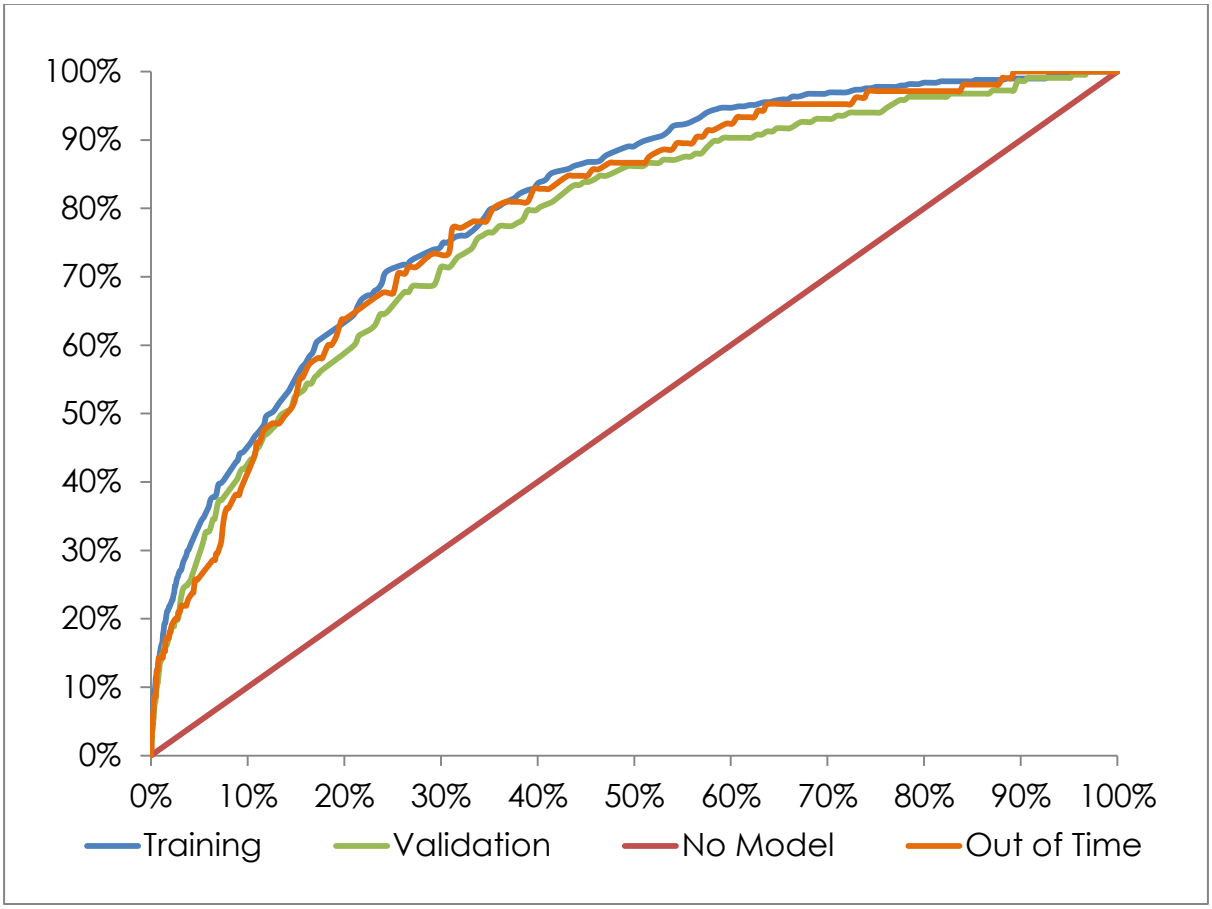
In this technique we will perform simple logistic regression as discussed in earlier chapter.

No.	Variable Name	Source	IV	Wald Chi-square
1	Variable 2	Demographic	0.30	67.88
2	Variable 3	CIBIL	0.29	18.74
3	Variable 4	Company Level	0.26	11.76
4	Variable 5	CIBIL	0.26	55.72
5	Variable 6	CIBIL	0.25	17.10
6	Variable 7	CIBIL	0.24	88.92
7	Variable 8	CIBIL	0.23	11.38
8	Variable 10	CIBIL	0.22	16.17
9	Variable 11	Demographic	0.21	20.47
10	Variable 14	CIBIL	0.16	17.40
11	Variable 15	CIBIL	0.14	16.04
12	Variable 17	Liability	0.09	21.27
13	Variable 18	Liability	0.09	16.87
14	Variable 19	Liability	0.08	11.49

The above table shows the variables coming up in the model arranged in order of the IV which is strength of the variable, Wald chi-square which shows the importance of the variable in the model and source of the variable. Logistic regression has 14 variables which are in the model and has strong representation of all the pool of data.

5.2.2.a Metrics for deciding the best model

1. **Lift Chart and GINI**

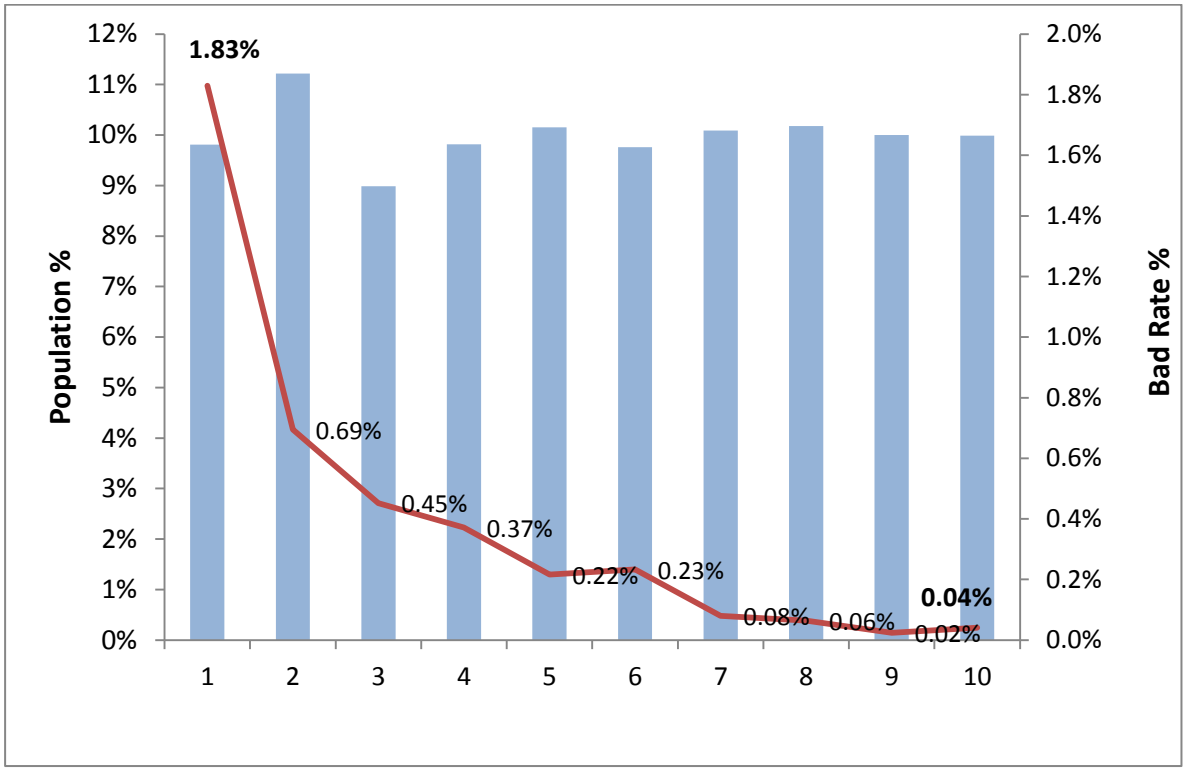


Form the above lift chart we can infer the performance of the model. The approval rate for the mortgage product portfolio is around 20%; hence we will consider the cut-off at bottom 20% which will reject first two deciles of the score distribution. The Lorenz curve shows that the new model on the training base isolates around 61% of bad accounts in bottom 20% of the population which is slightly less than earlier model. The Lorenz curve for the validation base shows that it isolates 55% of the bad accounts in bottom 20% of the population which is same as the earlier model. The

Lorenz curve for the out of time base shows that it isolates 58% of the bad accounts in bottom 20% of the population which is far better than the earlier model. Usually the capture rate of bad account drops in the test dataset than in validation dataset but in this model it is other way round. This makes the model better off than the earlier one.

2. Risk Ranking charts

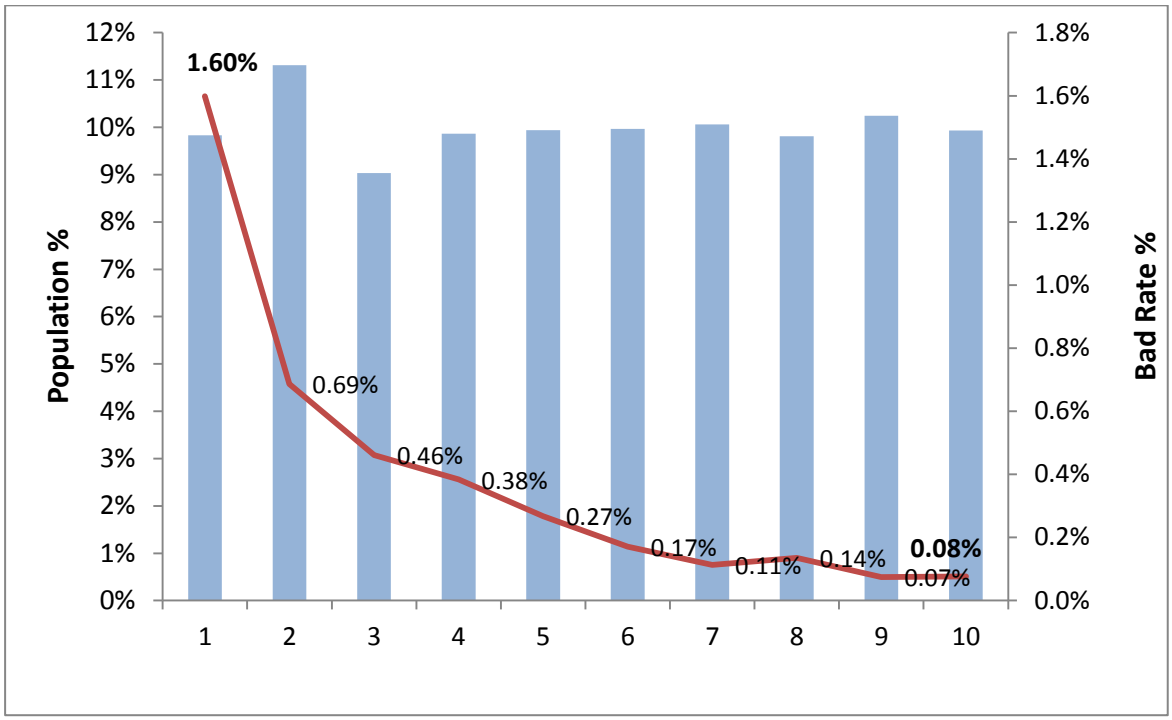
a. Risk Ranking for the training base:



The above chart shows the risk ranking of the training base. The scored population is distributed into deciles i.e. ten parts where each bin has approximately 10% of the total population based on the rank ordering of the score. Hence the lower deciles will contain the applicants with lower score

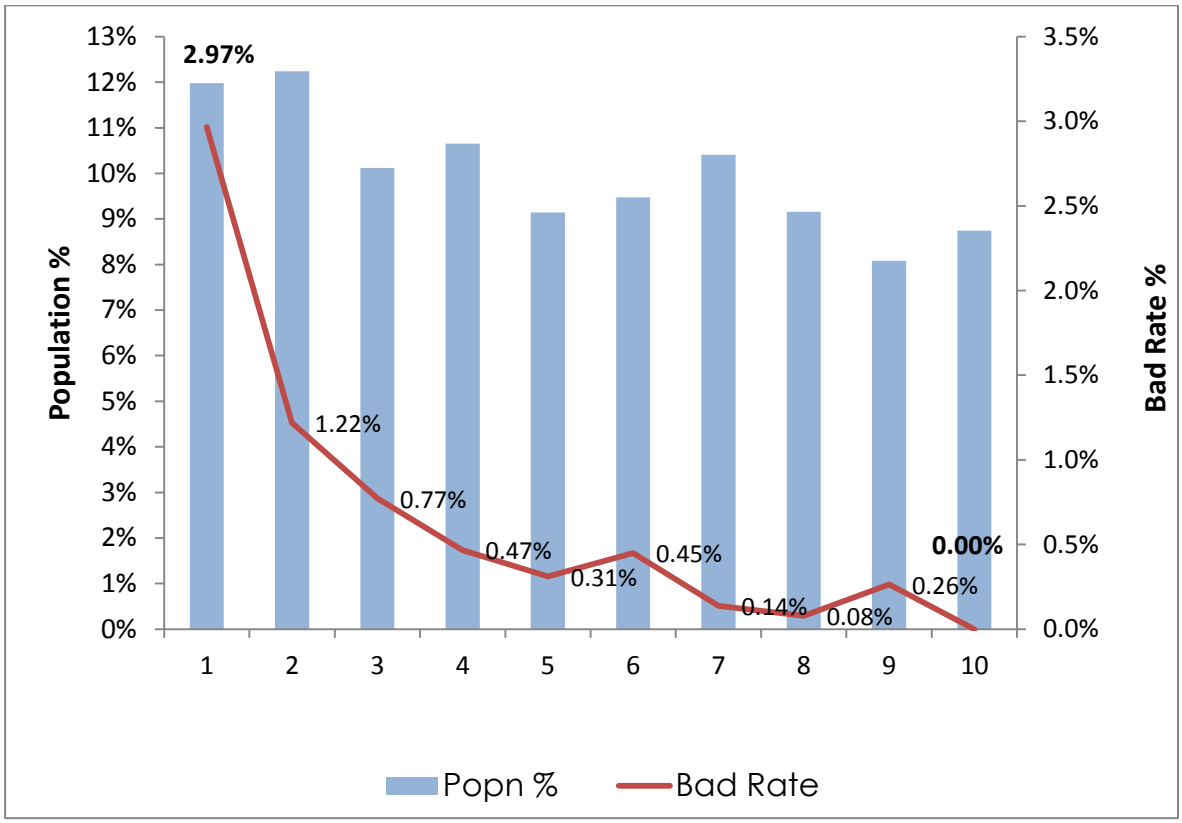
(i.e. “bad” accounts) and the higher deciles will contain the applicants with higher score (i.e. “good” accounts).The bad rate in the lower deciles is higher and since the approval rate is 80% we reject first 2 deciles resulting in rejection of bottom 20% of the population. The first bin has bad rate of 1.83% which is the highest and the second bin has bad rate of 0.69%.Here we can see that the bad rate in second bin drastically falls as compared to first bin and further gradually decreasing except some marginal peaks in higher bins. Hence we can infer that the bad rate curve should be like a hockey stick, starting from a high point in the lower bins and flattening out towards the higher bins. When this kind of bad rate curve is observed we can say that the model is performing well.

b. Risk Ranking for the validation base:



The first bin has bad rate of 1.6% which is the highest and the second bin has bad rate of 0.69%. Here we can see that the bad rate is gradually decreasing except some marginal peaks. Hence we can see a sort of hockey stick curve of bad rate in the validation base risk ranking. The bad rate curve is much better than the earlier model we had seen in the validation base with almost no peaks.

c. Risk Ranking for the out of time base:



The first bin has bad rate of 2.97% which is the highest and the second bin has bad rate of 1.22%.Here we can see that the bad rate has drastically dropped from bin 1 to bin 2 which symbolises that the risk is very finely captured by first bin and hence lowering down for the further bins. There are few peaks as compared to earlier model seen, which are observed over the spread of distribution, because the risk ranking is done on the out of time scored base and the model develop was on the train base. Hence there would be some amount of difference between the actual development base risk ranking and test base risk ranking.

Hence from the metrics, charts and univariate analysis we can say that the new Logistic Regression model is better than the earlier one because there are 14variable coming up in the model which is greater than he earlier count. The risk ranking is also much smoother than the earlier model. Also, the risk ranking is better on the validation and out of time bases. The also drawback is the drop in Gini index in the training base, but further the Gini index becomes stable on the validation and out of time base. This shows that the model is doing better on the validation and out of time base. The performance on the test dataset i.e. out of time base is slightly less than the train data which means that the model has an outstanding performance.

5.2.3 Logit Boost

In this technique we perform the boosting algorithm using logistic regression. Initially we fit logistic model and then have multiple iterations to model over the error term to reduce the error of misclassification as much as possible. It is similar to the concept of boosting discussed in earlier chapter.

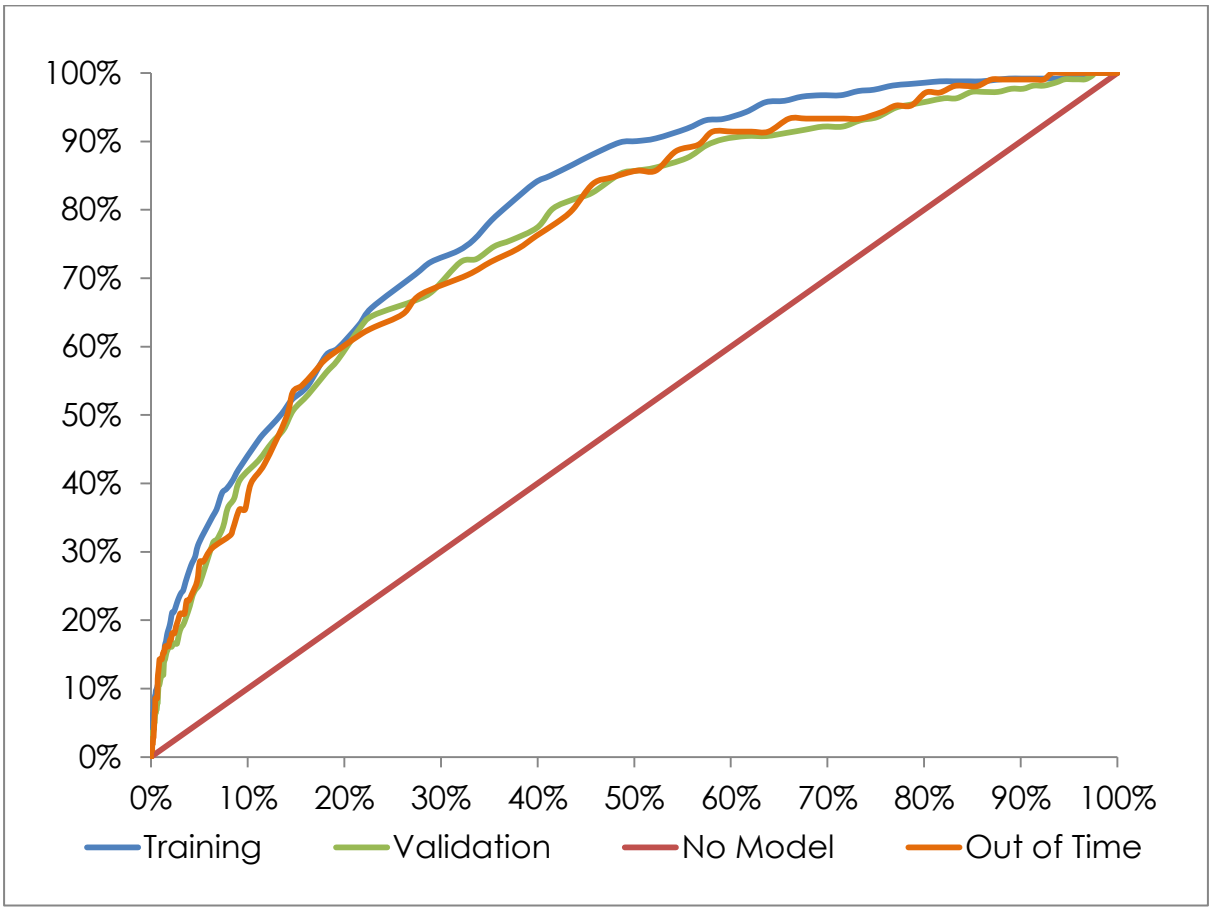
No.	Variable Name	Source	IV	Final Model Coeff
1	Variable 2	Demographic	0.30	-0.22
2	Variable 3	CIBIL	0.29	-0.21
3	Variable 4	Company Category	0.26	-0.11
4	Variable 5	CIBIL	0.26	-0.37
5	Variable 6	CIBIL	0.25	-0.12
6	Variable 7	CIBIL	0.24	-0.37
7	Variable 9	Company Category	0.23	-0.01
8	Variable 10	CIBIL	0.22	-0.16
9	Variable 11	Demographic	0.21	-0.22
10	Variable 14	CIBIL	0.16	-0.07
11	Variable 15	CIBIL	0.14	-0.21
12	Variable 17	Liability	0.09	-0.13
13	Variable 18	Liability	0.09	-0.15
14	Variable 19	Liability	0.08	-0.05

The above table show the list of all variables coming up in the last iteration of the model. The variables are rank ordered on the basis of the IV i.e. strength of variable to predict. It also

shows the final model coefficients which are all negative and hence significant to be in the model.

5.2.3.a Metrics for deciding the best model

1. **Lift Chart and GINI**

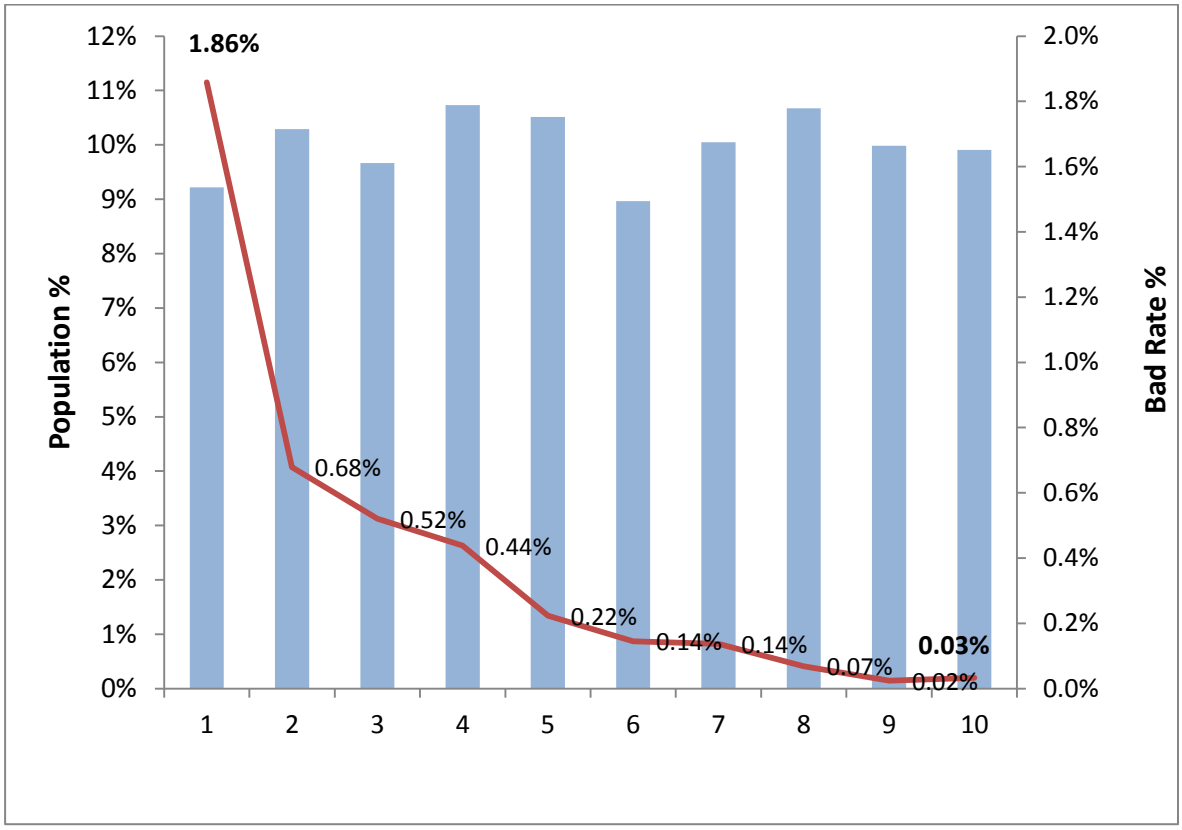


Form the above lift chart we can infer the performance of the model. The approval rate for the mortgage product portfolio is around 20%; hence we will consider the cut-off at bottom 20% which will reject first two deciles of the score distribution. The Lorenz curve shows that the new logit boost model on the training base isolates around 59% of bad accounts in bottom 20% of the population which is slightly less

than earlier logistic model. The Lorenz curve for the validation base shows that it isolates 53% of the bad accounts in bottom 20% of the population which is same as the earlier model. The Lorenz curve for the out of time base shows that it isolates 53% of the bad accounts in bottom 20% of the population which is far better than the earlier model. Here we can see that the Gini index in the training base is high and it fall down in the validation and out of time base which is usual. But the peculiar thing about this model is that the Gini index and the capture rate in the validation and the out of time base is similar and there is no drop. This shows that the model is very stable over the test datasets which enhances its performance.

2. Risk Ranking charts

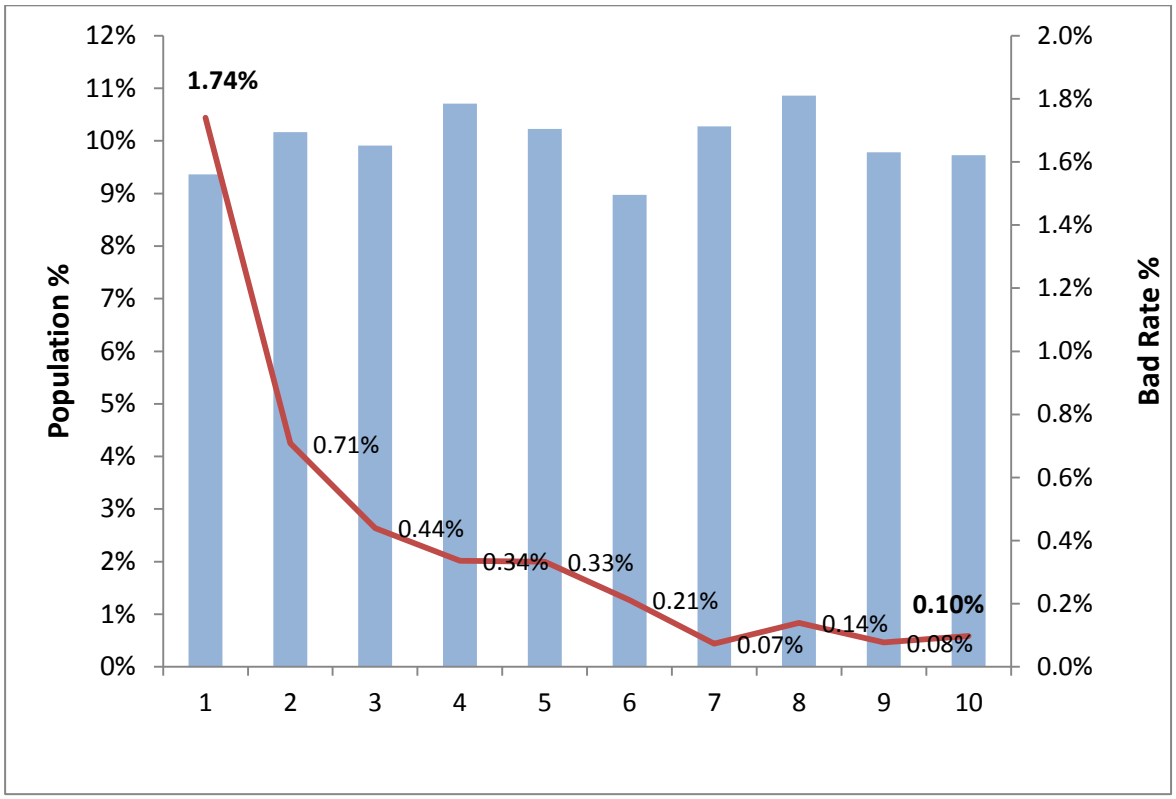
a. Risk Ranking for the training base:



The above chart shows the risk ranking of the training base. The scored population is distributed into deciles i.e. ten parts where each bin has approximately 10% of the total population based on the rank ordering of the score. Hence the lower deciles will contain the applicants with lower score (i.e. "bad" accounts) and the higher deciles will contain the applicants with higher score (i.e. "good" accounts).The bad rate in the lower deciles is higher and since the approval rate is 80% we reject first 2 deciles resulting in rejection of bottom 20% of the population. The first bin has bad rate of 1.86% which is the highest and the second bin has bad rate of 0.68%.Here we can see

that the bad rate in second bin drastically falls as compared to first bin and further gradually decreasing except some marginal peaks in higher bins. This is a special feature of the Logit Boost that in the risk ranking there is sharp drop in the bad rate after the first bin. Hence we can infer that the bad rate curve is like a hockey stick, starting from a high point in the lower bins and flattening out towards the higher bins, and hence we can say that the model is performing well.

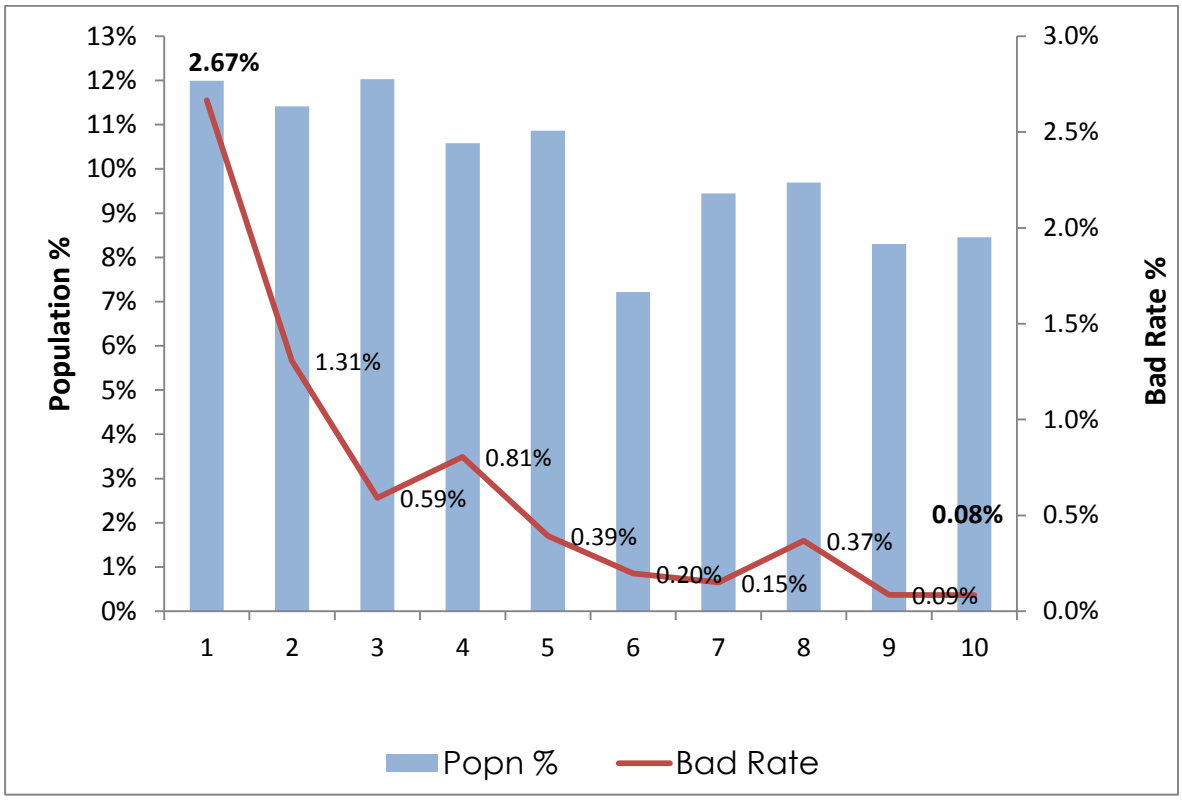
b. Risk Ranking for the validation base:



The first bin has bad rate of 1.74% which is the highest and the second bin has bad rate of 0.71%. Here we can see that the bad rate is gradually decreasing except some marginal peaks. Hence we can see a sort of hockey stick curve of

bad rate in the validation base risk ranking. The bad rate curve is much smoother than the earlier logistic model in the validation base with almost no peaks.

c. Risk Ranking for the out of time base:



The first bin has bad rate of 2.67% which is the highest and the second bin has bad rate of 1.31%.Here we can see that the bad rate has drastically dropped from bin 1 to bin 2 which symbolises that the risk is very finely captured by first bin and hence lowering down for the further bins. There are slightly more peaks as compared to earlier logistic model , which are observed over the spread of distribution, because the risk ranking is done on the out of time scored base and the model develop was on the train base. Hence there would

be some amount of difference between the actual development base risk ranking and test base risk ranking.

Hence, we can conclude that the Logit Boost model is seen to be stable with respect to Gini index in the validation and out of time base as compared to the earlier logistic model. The risk ranking is also smooth towards the higher bins and with a sharp drop in bad rate in first 2 bins. This helps us set a clean cut-off for the first 2 bins with higher confidence to accept the population lying in rest of the higher bins.

5.2.4 Multi-layer Logistic Regression

This technique is kind of an ensemble modelling which we have discussed in earlier chapter. Here we run logistic regression in two levels on different pool of data. In the first level we run logistic regression on the bureau and liability data separately. In the second level we provide the prediction of first level as input with demographic variables to another logistic regression.

1st Level Model

No.	Variable Name	IV	Wald Chi-square
1	Variable 3	0.29	23.86
2	Variable 5	0.26	50.82
3	Variable 6	0.25	17.96
4	Variable 7	0.24	72.89
5	Variable 8	0.23	27.45
6	Variable 10	0.22	18.28
7	Variable 12	0.17	25.83
8	Variable 13	0.16	19.42
9	Variable 15	0.14	20.58

The above table shows the list of variable coming up in the first level of logistic regression when run on the bureau variables. The variables are rank ordered on the basis of IV i.e. strength of prediction of variable. The table also has the Wald Chi-Square values which shows the importance of that particular variable in the model.

No.	Variable Name	IV	Wald Chi-square
1	Variable 16	0.099	26.99
2	Variable 18	0.096	29.69
3	Variable 17	0.092	17.96

The above table shows the list of variable coming up in the first level of logistic regression when run on the liability variables. The variables are rank ordered on the basis of IV i.e. strength of prediction of variable. The table also has the Wald Chi-Square value which shows the importance of that particular variable in the model.

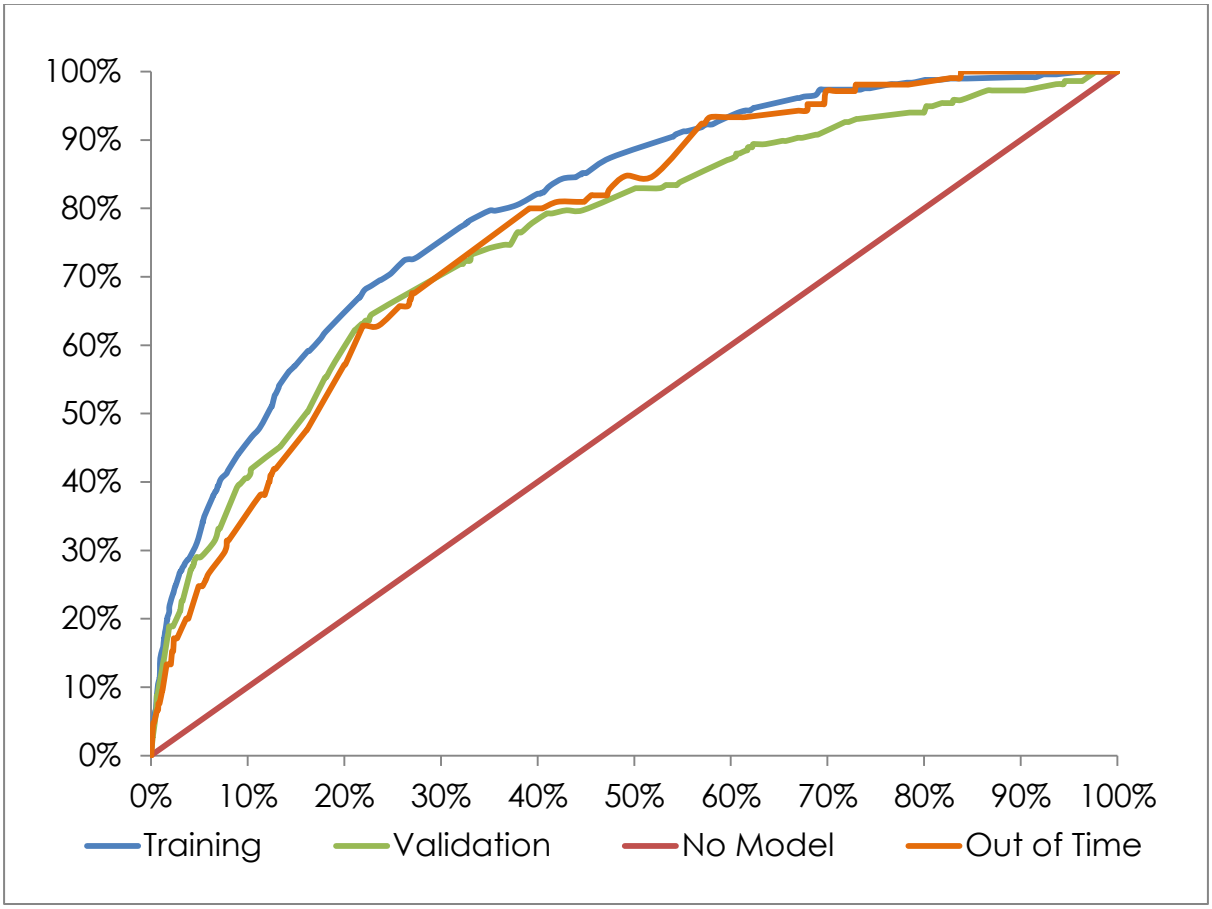
2nd Level Model

Variable Name	IV	Wald Chi-square
Log Odds from Bureau Only Model	-	395.41
Variable 1	0.32	71.14
Log Odds from Liability Only Model	-	56.29
Variable 11	0.21	37.55

The above table shows that there are 2 fixed inputs to the model which are the prediction of first level. The log odds from the bureau model and the liability model with addition to the demographic variables are given as input to the second level model. There are only 2 variables coming up in the model from the demographic pool.

5.2.4.a Metrics for deciding the best model

1. **Lift Chart and GINI**

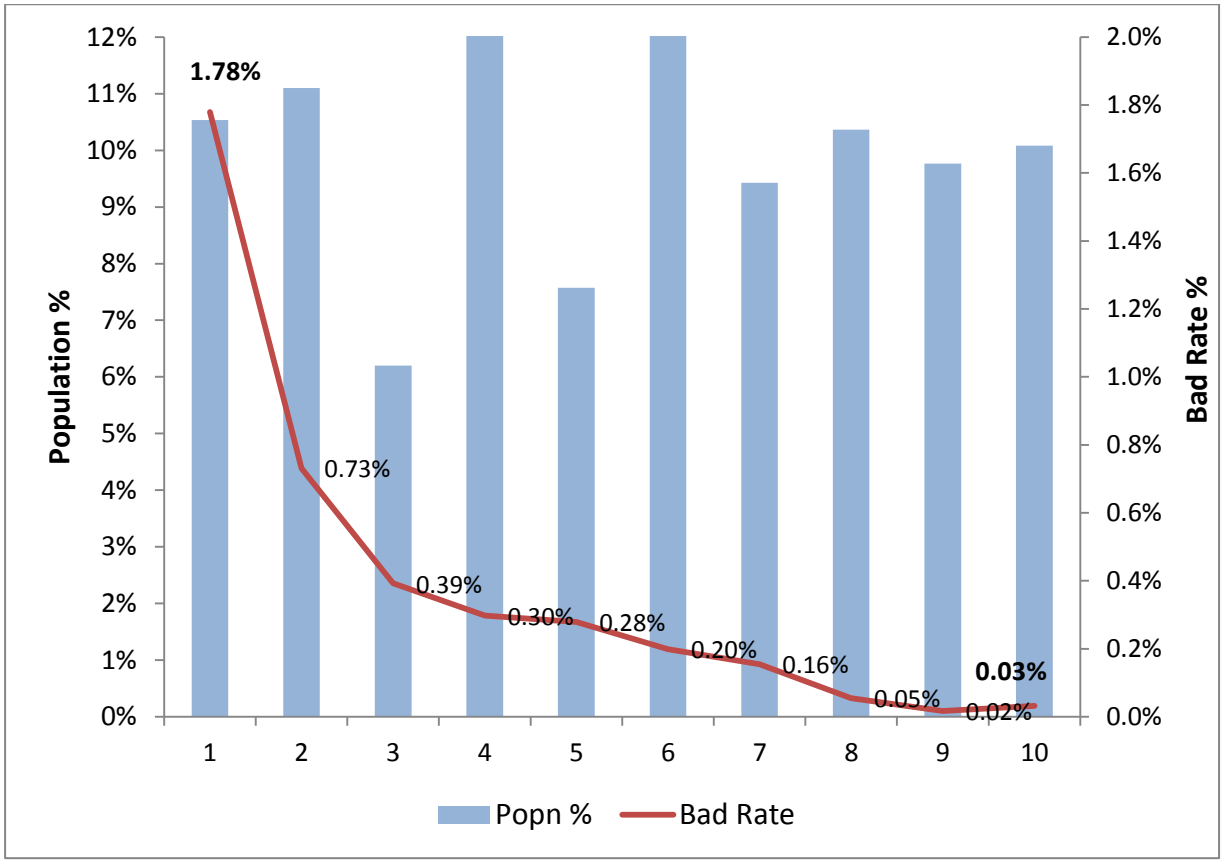


Form the above lift chart we can infer the performance of the model. The approval rate for the mortgage product portfolio is around 20%; hence we will consider the cut-off at bottom 20% which will reject first two deciles of the score distribution. The Lorenz curve shows that the new ensemble model on the training base isolates around 61% of bad accounts in bottom 20% of the population which is slightly less than earlier logistic model. The Lorenz curve for the validation base shows that it isolates 51% of the bad accounts in bottom 20% of the population which is same as the earlier

model. The Lorenz curve for the out of time base shows that it isolates 54% of the bad accounts in bottom 20% of the population which is far better than the earlier model. Here we can see that the Gini index in the training base is high and it fall down in the validation which is usual. But it is comparatively high in the out of time base which is quite peculiar. This shows that the model is very stable over the test datasets which enhances its performance.

2. **Risk Ranking charts**

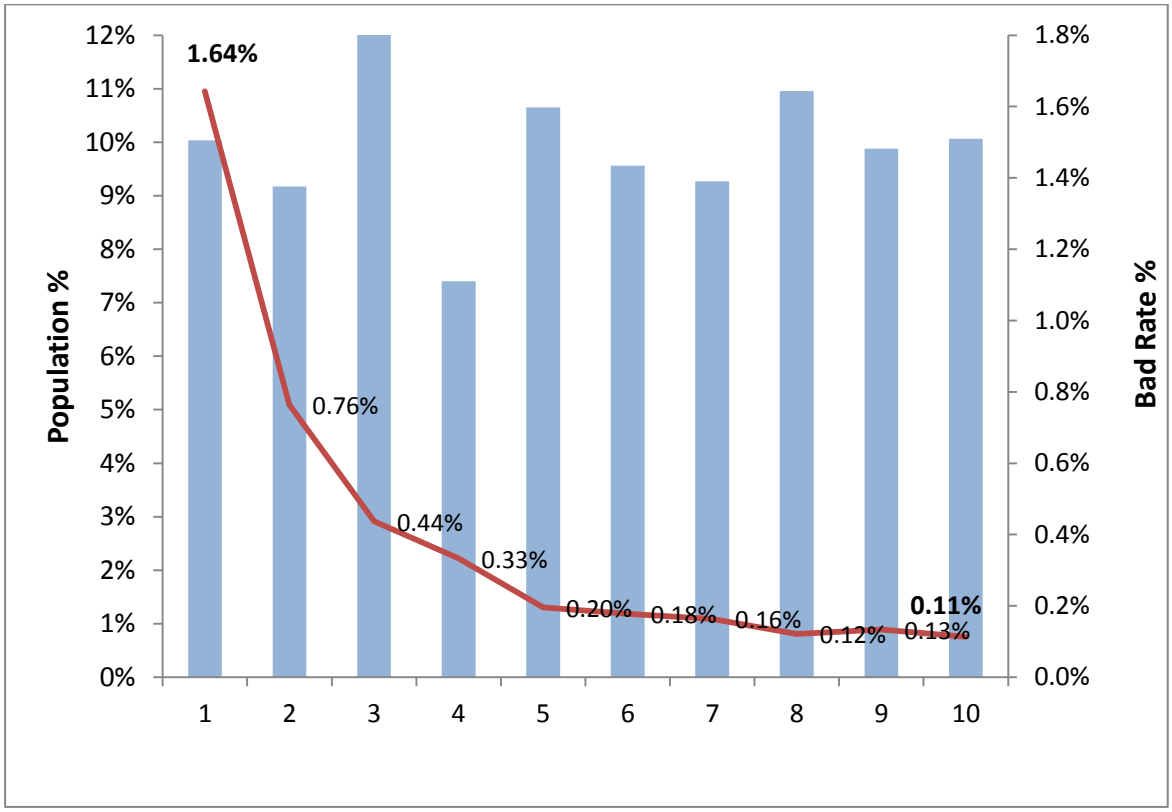
a. Risk Ranking for the training base:



The above chart shows the risk ranking of the training base. The scored population is distributed into deciles i.e. ten parts where each bin has approximately 10% of the total population based on the rank ordering of the score. Hence the lower deciles will contain the applicants with lower score (i.e. "bad" accounts) and the higher deciles will contain the applicants with higher score (i.e. "good" accounts).The bad rate in the lower deciles is higher and since the approval rate is 80% we reject first 2 deciles resulting in rejection of bottom 20% of the

population. The first bin has bad rate of 1.78% which is the highest and the second bin has bad rate of 0.73.Here we can see that the bad rate in second bin drastically falls as compared to first bin and further gradually decreasing except some marginal peaks in higher bins. Hence we can infer that the bad rate curve should be like a hockey stick, starting from a high point in the lower bins and flattening out towards the higher bins. When this kind of bad rate curve is observed we can say that the model is performing well.

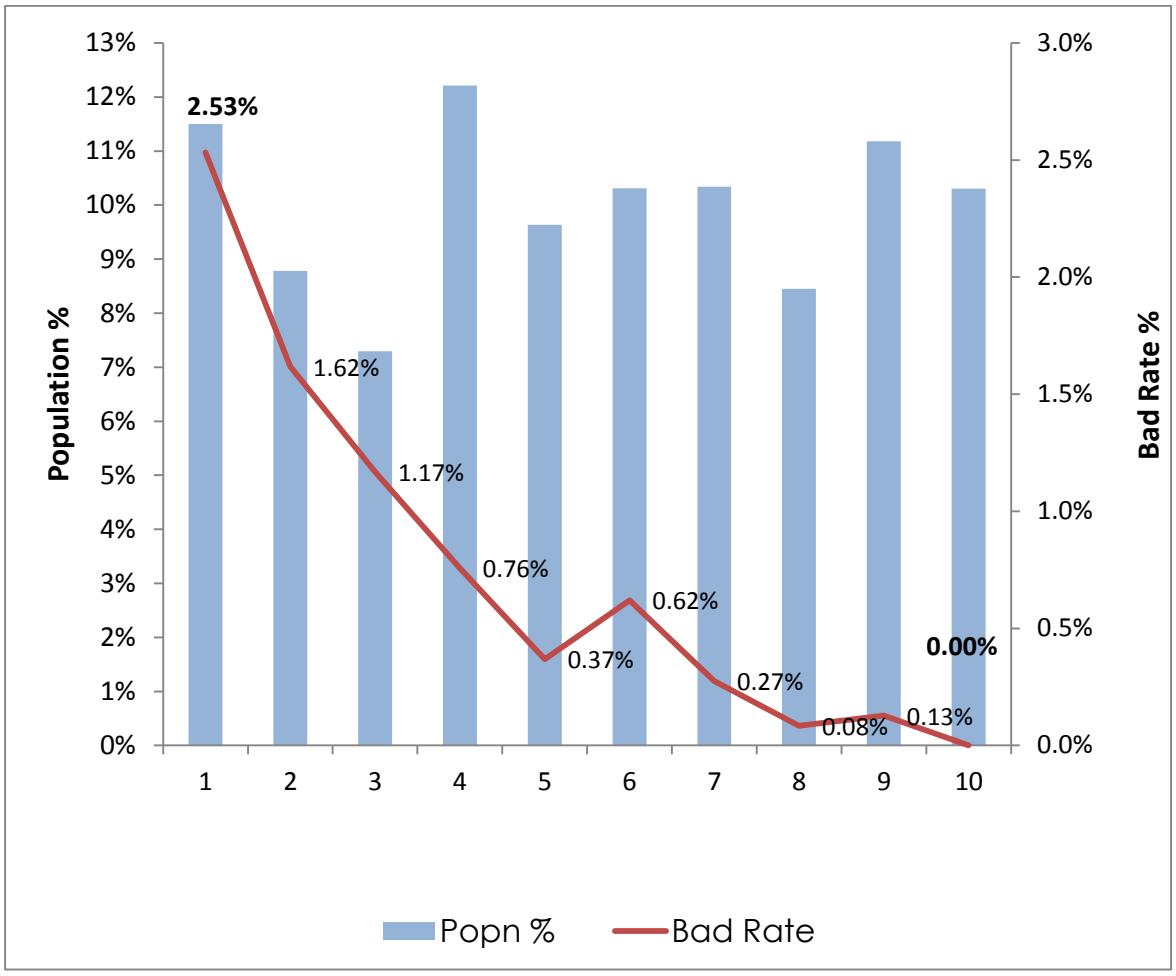
b. Risk Ranking for the validation base:



The first bin has bad rate of 1.64% which is the highest and the second bin has bad rate of 0.76%. Here we can see that the bad rate is gradually decreasing except some marginal peaks. Hence we can see a sort of hockey stick curve of

bad rate in the validation base risk ranking. The bad rate curve is much smoother than the earlier Logit Boost model in the validation base with almost no peaks.

c. Risk Ranking for the out of time base:



The first bin has bad rate of 2.53% which is the highest and the second bin has bad rate of 1.62%.Here we can see that the bad rate has drastically dropped from bin 1 to bin 2 which symbolises that the risk is very finely captured by first bin and hence lowering down for the further bins. There are few peaks as compared to earlier Logit Boost model which are observed over the spread of

distribution, because the risk ranking is done on the out of time scored base and the model develop was on the train base. Hence there would be some amount of difference between the actual development base risk ranking and test base risk ranking.

Hence, we can conclude that the Ensemble model is seen to be stable with respect to Gini index in the validation and out of time base as compared to the earlier Logit Boost model. We can also see that the risk ranking and Gini is outperforming in the out of time base. The risk ranking is also smooth towards the higher bins and with a sharp drop in bad rate in first 2 bins. This helps us set a clean cut-off for the first 2 bins with higher confidence to accept the population lying in rest of the higher bins.

Finally after comparing all three modelling techniques we can conclude that:

Technique	Details	Expected Benefits
Logistic Regression	Usual Method	
Boosting (Logit Boost)	Fitting model iteratively – Each iteration is attempted at minimizing the error of the previous iteration	Pocket information specific variables will get an opportunity to appear in the model
	Final Model is an aggregate of all the iterations	Final model is expected to have a sharper risk ranking and higher capture rate in lower deciles
Ensemble Modeling (Two stage modeling)	Stage 1 – 'Bureau only' & 'Liability only' models are developed	Bureau or Liability will not overpower each other, thereby allowing adequate representation from both
	Stage 2 – Log odds computed from stage1, along with demographic variables become input to stage2 model	Final model can be expected to be more robust

Logistic Regression seems to provide most robust performance over all the three modelling techniques. Hence after presentation of the model to all the stake holders it was decided to go forward with the Logistic Regression modelling technique. The scorecard developed by this

method is considered to be the KGB benchmark model. Thus further any models prepared should have their performance better than the KGB benchmark model.

5.3 Model 3

In this model we will develop "All Good/Bad" model in which we consider whole through the door population which consists of accept, reject and account not taken up cases. We have the on us performance with the bank only for the accept accounts and the performance of the reject and account not open cases is further inferred using different techniques as discussed in earlier chapter. We have used fuzzy augmentation and combination of fuzzy augmentation and bureau bad.

5.3.1 AGB Model is developed on:

1. Sample window : January 2011 to June 2014 (through the door)
2. Performance window : Rolling window of 2 years starting from date of application
3. Bad definition : Ever 90 plus days past due in 24 months
4. Good definition : Current and 1 to30 days past due
5. Indeterminate : 30 to 90 days past due
6. Exclusions : Non Individuals and Indeterminates
7. Development Base : January 2011 to March 2014
8. The development base is split into training and validation randomly as 70% and 30% respectively.
9. Out of Time Base : April 2014 to June 2014
10. The final base for AGB consists only of the through the door population. It has all the bureau, liability, demographic and other computed variables.
11. Amongst the reject and account not open cases, those who have the performance of the similar product with bureau we consider the performance with some weights assigned. Whereas for the cases whose performance is not available with the bureau we consider fuzzy augmentation for assigning their performance with some weights.

In process of creation of the AGB model we created many intermediate models and then finalised one model based on various statistical metrics.

No.	Variable Name	Source	IV	Wald Chi-square
1	Variable 1	CIBIL	0.39	114.16
2	Variable 2	CIBIL	0.35	78.35
3	Variable 3	CIBIL	0.33	128.75
4	Variable 4	CIBIL	0.31	160.07
5	Variable 5	Demographic	0.29	131.79
6	Variable 6	Demographic	0.29	147.03
7	Variable 7	CIBIL	0.27	90.93
8	Variable 8	CIBIL	0.24	15.26
9	Variable 9	CIBIL	0.24	73.39
10	Variable 10	Liability	0.13	50.36
11	Variable 11	Liability	0.11	17.94
12	Variable 12	Liability	0.10	52.32
13	Variable 13	CIBIL	0.10	27.69
14	Variable 14	CIBIL	0.06	26.90
15	Variable 15	CIBIL	0.06	18.77

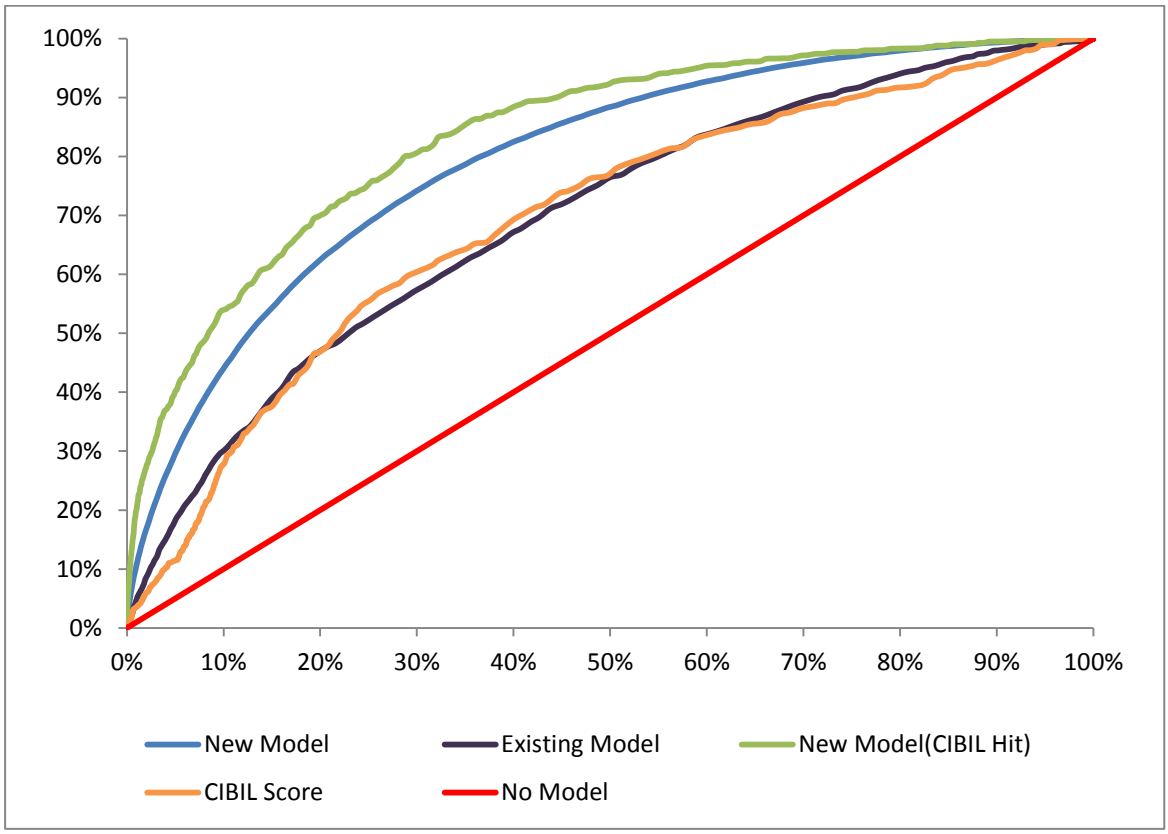
The final AGB Model consists of 15 variables which are arranged on the basis of IV in the table above. The model seems to be good amongst all since it covers up whole pool of data as discussed earlier. It has strong representation of liability, bureau and demographic variables

with strong IV. The variables are rank ordered on the basis of the IV i.e. strength of variable. It also shows the Wald Chi-Square value which implies the importance of the variable in the model.

5.3.2 Metrics for deciding the best model

5.3.2.a Lift Chart and GINI

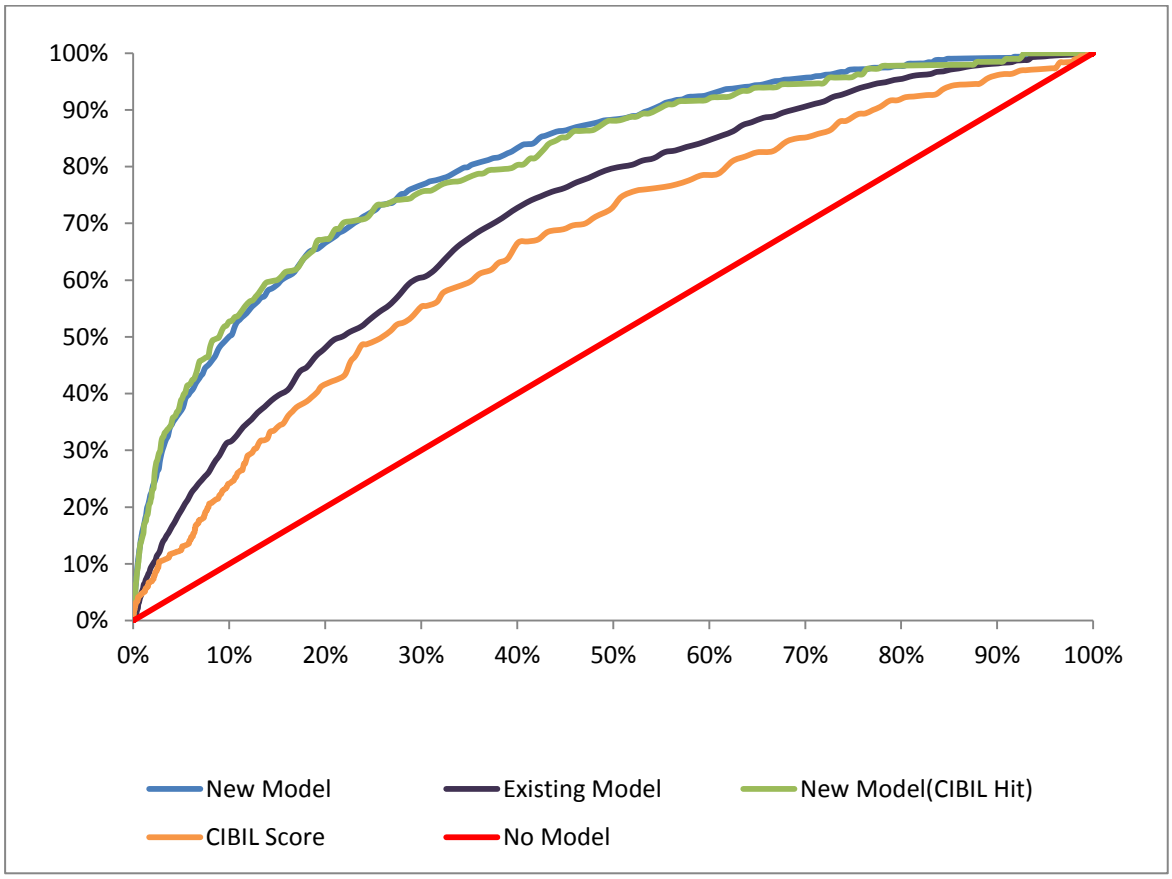
- **Lift chart for the training base:**



Form the above lift chart we can infer the performance of the model. The approval rate for the mortgage product portfolio is around 20%; hence we will consider the cut-off at bottom 20% which will reject first two deciles of the score distribution. The Lorenz curve shows that the new model isolates around 65% of bad accounts in bottom 20% of the population, the existing model isolates around 40% of bad accounts in bottom 20% of the population, the

CIBIL score isolates around 39% of bad accounts in bottom 20% of the population and CIBIL hit model i.e. when the scorecard is run only on the application who have vintage on CIBIL (bureau) model isolates around 67% of bad accounts in bottom 20% of the population on the training base. Since this is the final scorecard we need to consider all the scores and characteristics considered by the underwriter to adjudicate the application. Hence we have seen the performance of the model on different segments of data to validate. The Gini index of the new model has increased as compared to the KGB benchmark model with adding reject inferenced base to the base. We can also see that the Lorenz curve of the existing model and the CIBIL Score model is close enough. Also, there is quite a lot difference between the new model over the whole sample and the one over the bureau hit sample. So we can infer that the performance of the new model is slightly better for the bureau hit sample because there are many bureau variables in the scorecard (i.e. model is bureau specific), but it also classifies well over the whole sample of bureau hit and no hit.

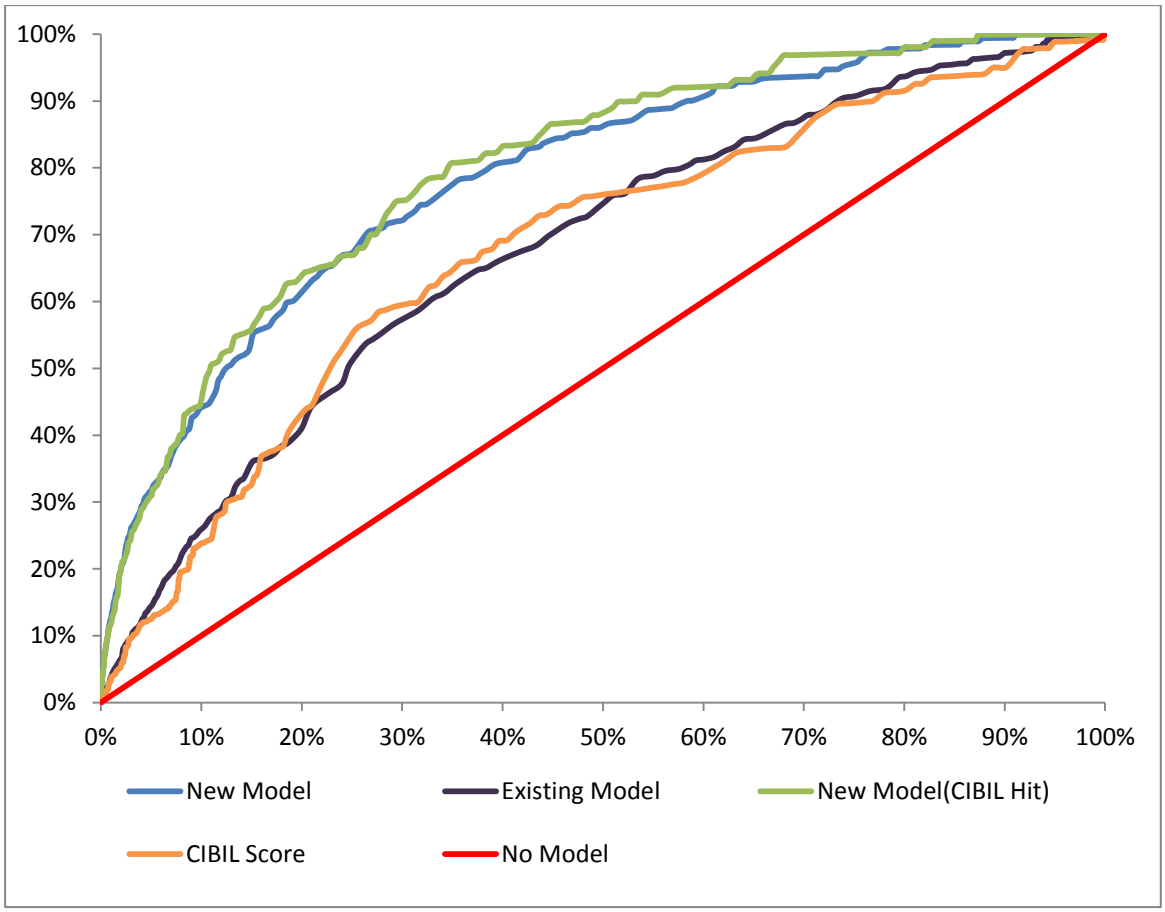
- **Lift chart for the validation base:**



The Lorenz curve shows that the new model isolates around 62% of bad accounts in bottom 20% of the population, the existing model isolates around 42% of bad accounts in bottom 20% of the population, the CIBIL score isolates around 32% of bad accounts in bottom 20% of the population and CIBIL hit model i.e. when the scorecard is run only on the application who have vintage on CIBIL (bureau) model isolates around 61% of bad accounts in bottom 20% of the population on the validation base. Here in the lift chart we can see that the Lorenz curve for the new model on the whole sample and the bureau hit sample is close enough, whereas there is difference between existing model and CIBIL score. Thus we can conclude that the model is

performing well on the whole sample as well as the bureau hit segment which shows that there is no need of separate segmented model for hit and no hit segment.

- **Lift chart for the out of time base:**

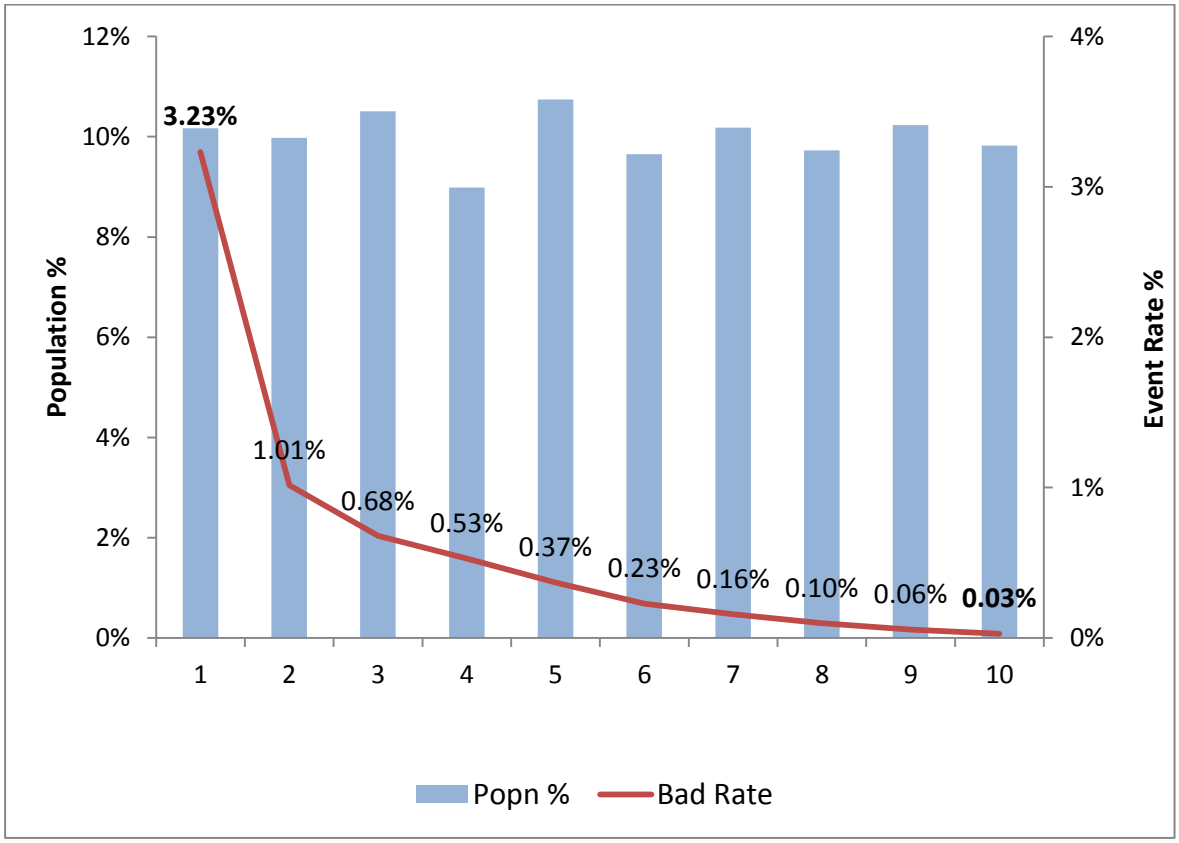


The Lorenz curve shows that the new model isolates around 58% of bad accounts in bottom 20% of the population, the existing model isolates around 36% of bad accounts in bottom 20% of the population, the CIBIL score isolates around 35% of bad accounts in bottom 20% of the population and CIBIL hit model i.e. when the scorecard is run only on the application who

have vintage on CIBIL (bureau) model isolates around 60% of bad accounts in bottom 20% of the population on the out of time base. Here in the lift chart we can see that the Lorenz curve for the new model on the whole sample and the bureau hit sample is close enough, also between existing model and CIBIL score there is no much difference. Thus we can conclude that the model is performing well on the whole sample as well as the bureau hit segment as there is a slight difference between the two which shows that there is no need of separate segmented model for hit and no hit segment.

5.3.2.b Risk Ranking chart

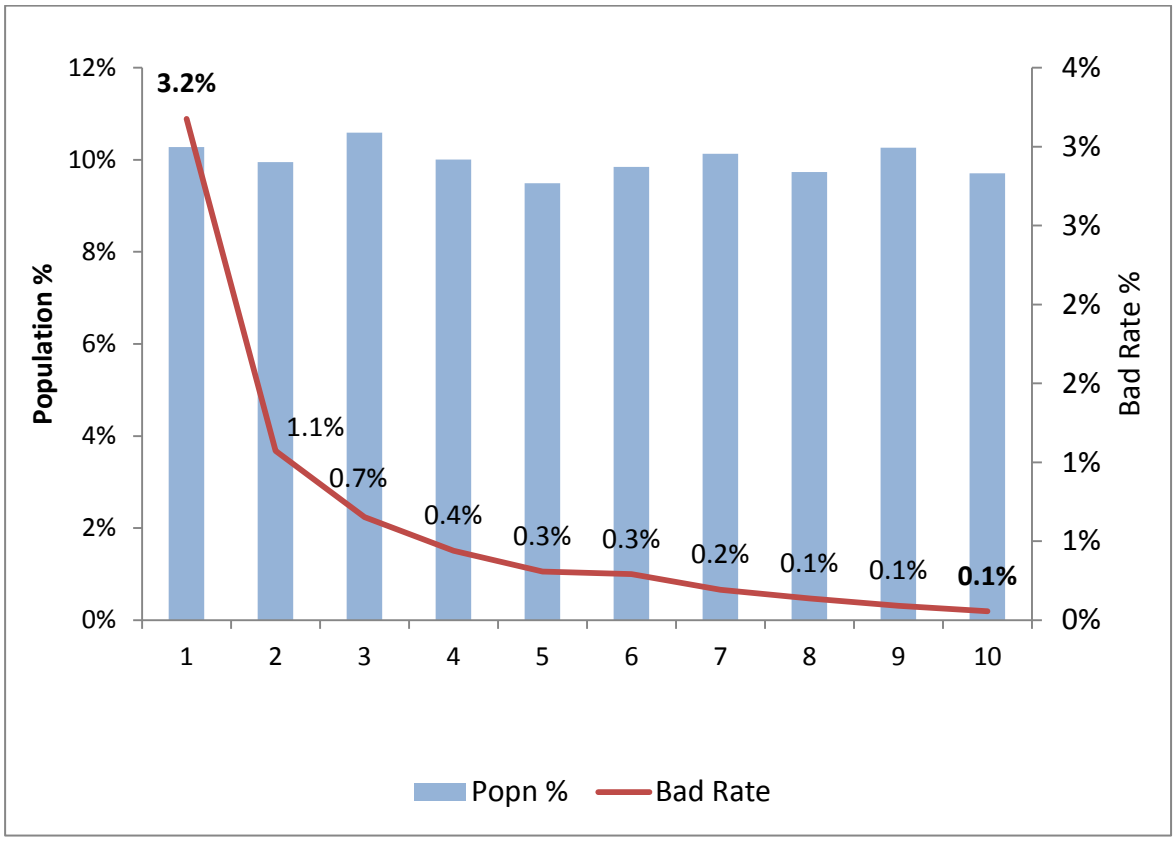
- **Risk Ranking for the training base:**



The above chart shows the risk ranking of the training base. The scored population is distributed into deciles i.e. ten parts where each bin has approximately 10% of the total population based on the rank ordering of the score. Hence the lower deciles will contain the applicants with lower score (i.e. "bad" accounts) and the higher deciles will contain the applicants with higher score (i.e. "good" accounts).The bad rate in the lower deciles is higher and since the approval rate is 80% we reject first 2 deciles resulting in rejection of bottom 20% of the population. The first bin has bad rate of 3.23%

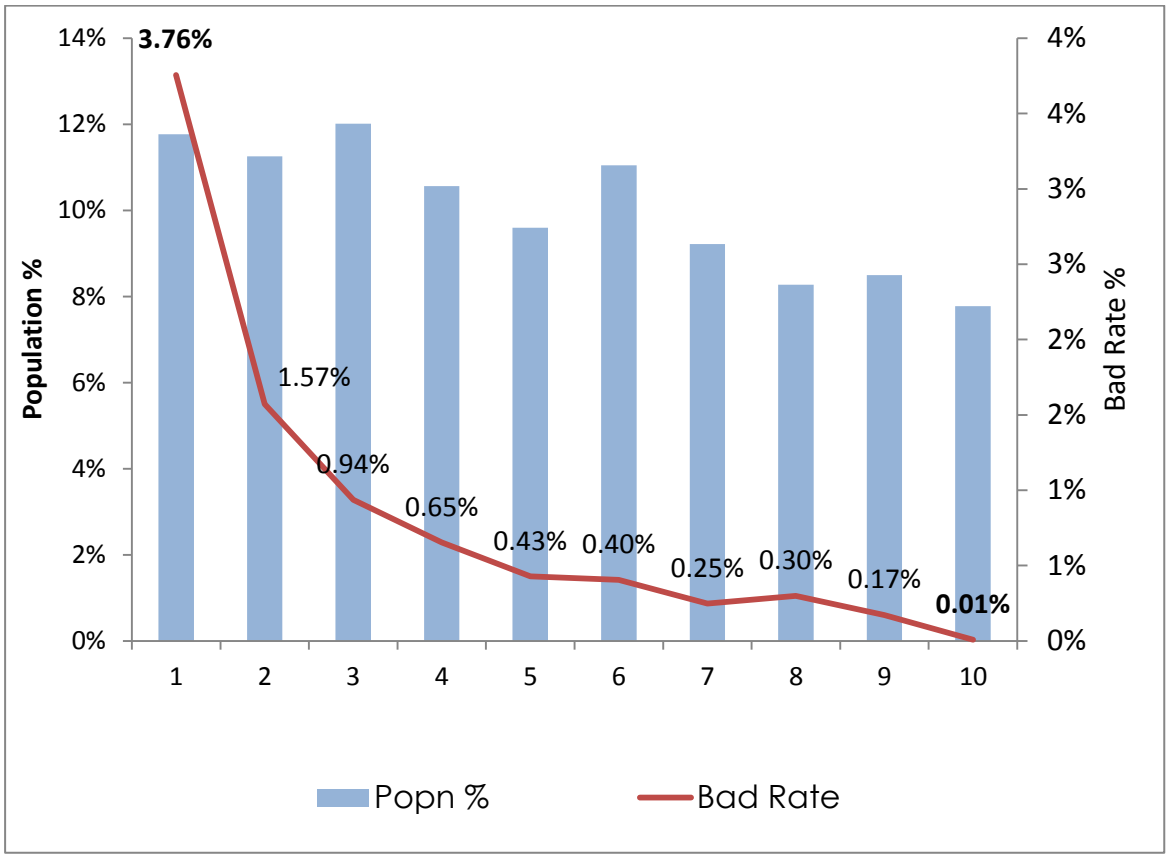
which is the highest and the second bin has bad rate of 1.01%.Here we can see that the bad rate in second bin drastically falls as compared to first bin and further gradually decreasing in higher bins. Hence we can infer that the bad rate curve is like a hockey stick, starting from a high point in the lower bins and flattening out towards the higher bins. When this kind of bad rate curve is observed we can say that the model is performing well. The bad rate curve is very much smooth with literally no peaks which make this model ideal and better than KGB benchmark model.

- **Risk Ranking for the validation base:**



The first bin has bad rate of 3.2% which is the highest and the second bin has bad rate of 1.1%. Here we can see that the bad rate is gradually decreasing except some marginal peaks. Hence we can see a sort of hockey stick curve of bad rate in the validation base risk ranking. The bad rate curve is much smoother than the KGB benchmark model in the validation base. The bad rate curve is very much smooth with literally no peaks which make this model ideal and better than KGB benchmark model.

- **Risk Ranking for the out of time base:**

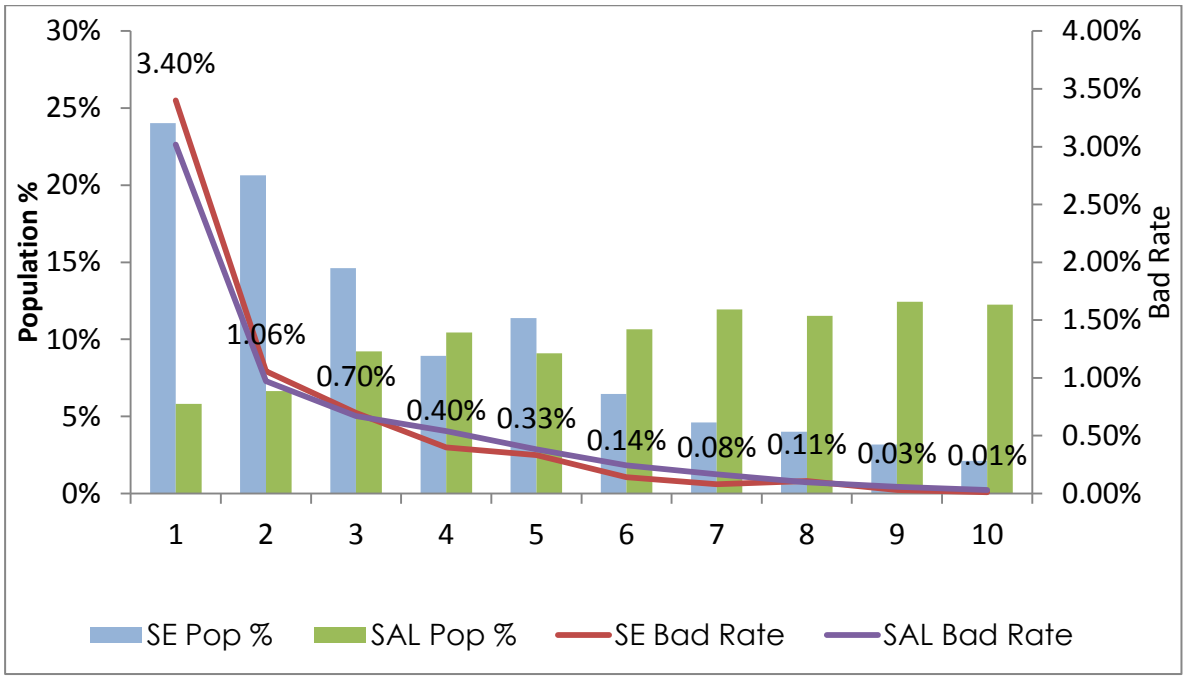


The first bin has bad rate of 3.76% which is the highest and the second bin has bad rate of 1.57%. Here we can see that the bad rate is gradually decreasing except some marginal peaks. Hence we can see a sort of hockey stick curve of bad rate in the out of time base risk ranking. The bad rate curve is much smoother than the KGB benchmark model in the out of time base. The bad rate curve is very much smooth with literally no peaks which make this model ideal and better than KGB benchmark model.

5.4 Validation of AGB Model

5.4.1 Population Distribution and Event Rate by Employment Status

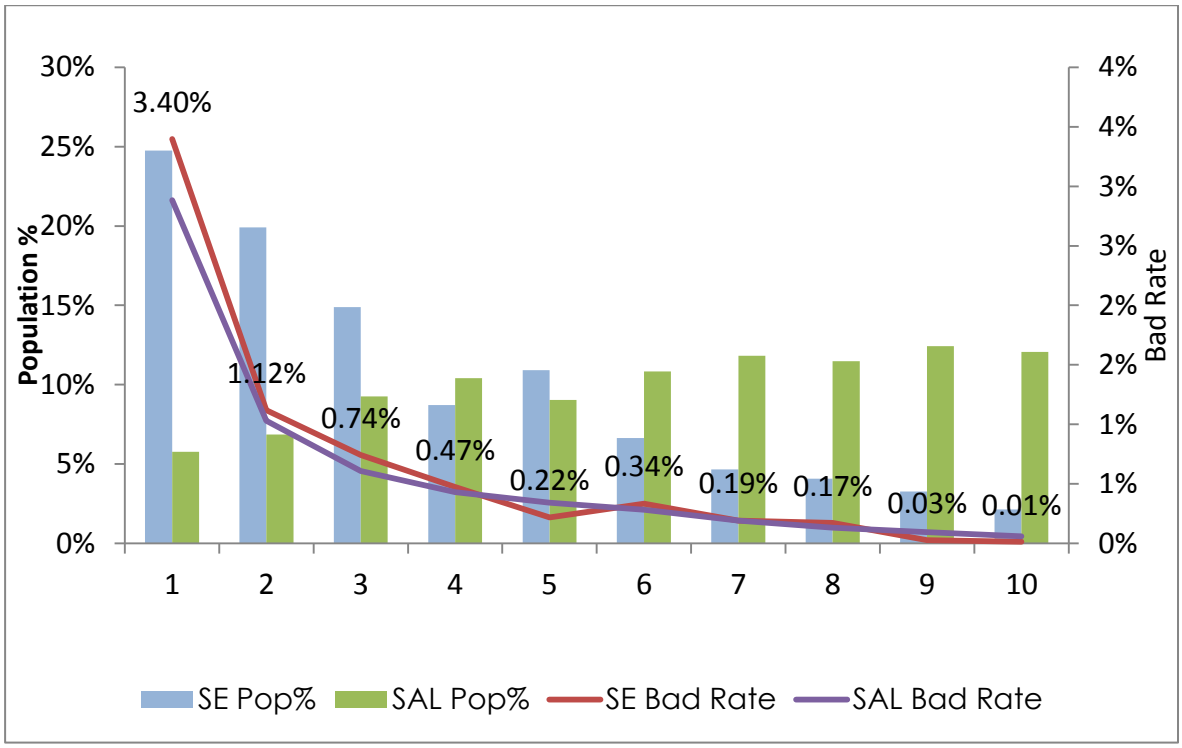
- **On training base**



The above chart shows the population distribution and the event rate by employment status on the training base. We can see from the chart that the bad rate in the first bin is 3.4% which is much higher than in the second bin which is 1.06% and further decrease to flatten off in the higher bins. Population in the lower bins of the self-employed applicants is much higher and it goes on decreasing in the higher bins. As it is usually considered that the self-employed applicants are much riskier than the salaried applicants, and the final scorecard seems to capture 45% of the total self-employed riskier applicants in first 2 bins. Hence the final scorecard is much better since we have set the

cutoff at first 2 bins to reject and accept in the further bins. The populations of salaried applicants in the first 2 bins is 13%, as per our consideration salaried applicants are better and we are rejecting only 13% riskier applications and accepting the further applications. Bad rate of salaried and self-employed applicants is almost moving together with slight level shift in the self-employed than salaried which is expected due to high risk in self-employed segment. Thus we can say that there is no need of segmented model on the basis of the employment type as both the models are performing very well.

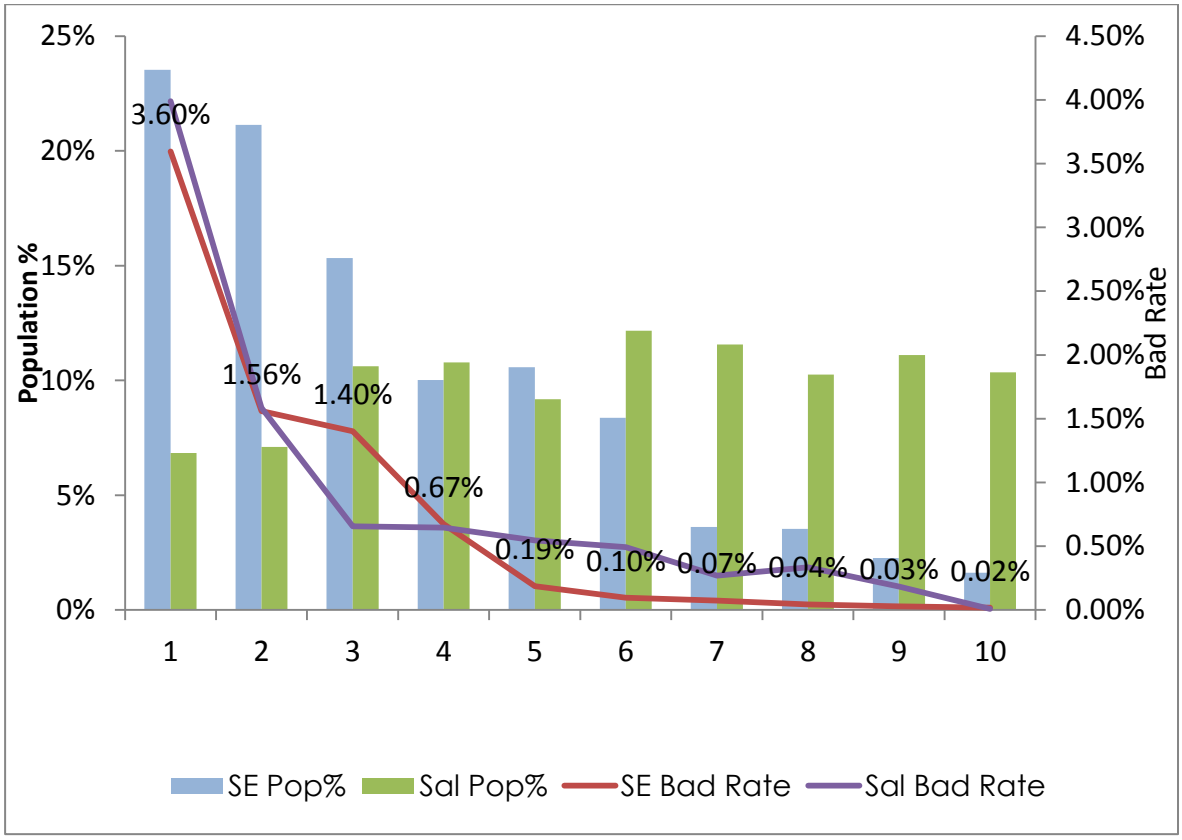
- **On validation base:**



The above chart shows the population distribution and the event rate by employment status on the validation base. We can see from the chart that the bad rate in the first bin is 3.4% which is much higher than in the second bin

which is 1.12% and further decrease to flatten off in the higher bins. Population in the lower bins of the self-employed applicants is much higher and it goes on decreasing in the higher bins. As it is usually considered that the self-employed applicants are much riskier than the salaried applicants, and the final scorecard seems to capture 44% of the total self-employed riskier applicants in first 2 bins. Hence the final scorecard is much better since we have set the cutoff at first 2 bins to reject and accept in the further bins. The populations of salaried applicants in the first 2 bins is 13%, as per our consideration salaried applicants are better and we are rejecting only 13% riskier applications and accepting the further applications. Bad rate of salaried and self-employed applicants is almost moving together with slight level shift in the self-employed than salaried which is expected due to high risk in self-employed segment. Thus we can say that the model is performing very well on the validation base on both the self-employed and salaried segments.

- **On out of time base:**

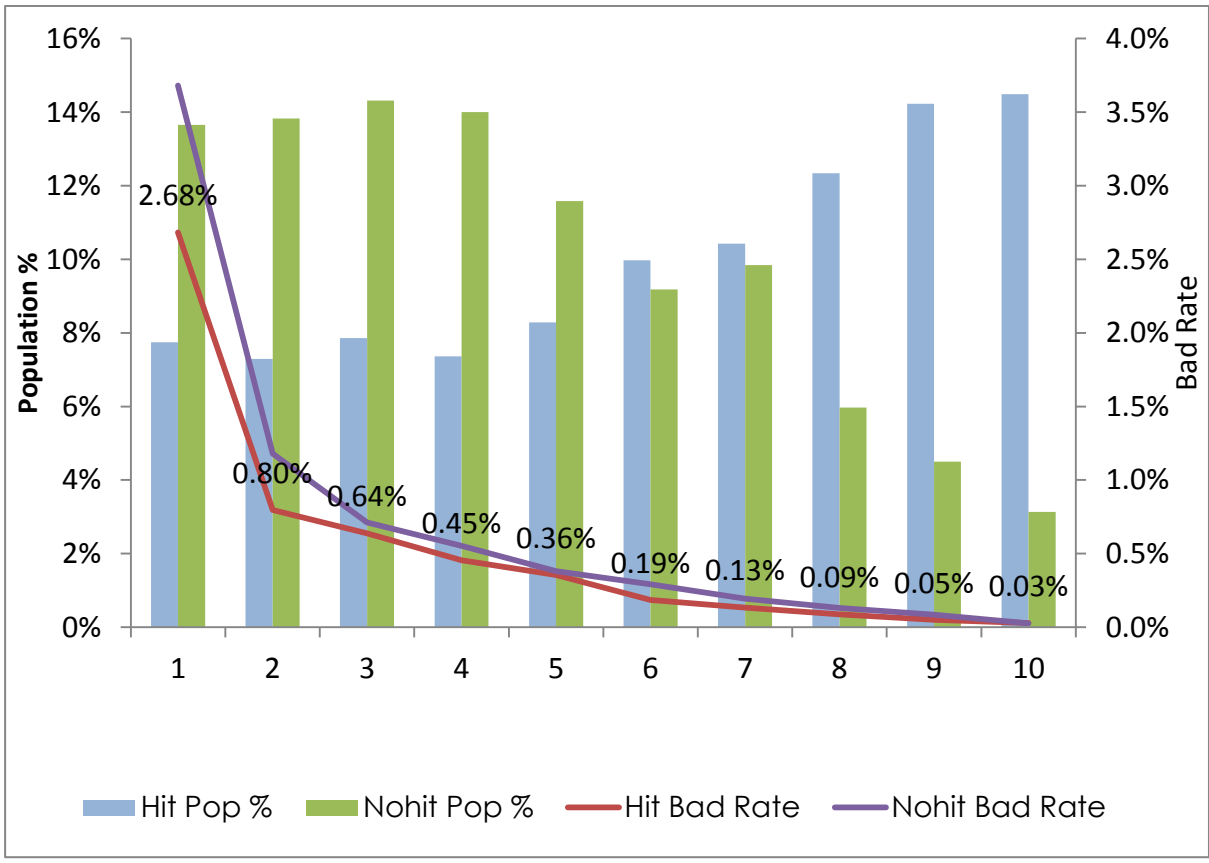


The above chart shows the population distribution and the event rate by employment status on the out of time base. We can see from the chart that the bad rate in the first bin is 3.6% which is much higher than in the second bin which is 1.56% and further decrease to flatten off in the higher bins. Population in the lower bins of the self-employed applicants is much higher and it goes on decreasing in the higher bins. As it is usually considered that the self-employed applicants are much riskier than the salaried applicants, and the final scorecard seems to capture 45% of the total self-employed riskier applicants in first 2 bins. Hence the final scorecard is much better since we

have set the cutoff at first 2 bins to reject and accept in the further bins. The populations of salaried applicants in the first 2 bins is 14%, as per our consideration salaried applicants are better and we are rejecting only 14% riskier applications and accepting the further applications. Bad rate of salaried and self-employed applicants is almost moving together in the first 2 bins but further there is a lateral level shift in the bad rates. Thus we can say the model is capturing risk finely in first 2 bins but further there is some difference in performance of the model.

5.4.2 Population Distribution and Event Rate by CIBIL Hit/Nohit

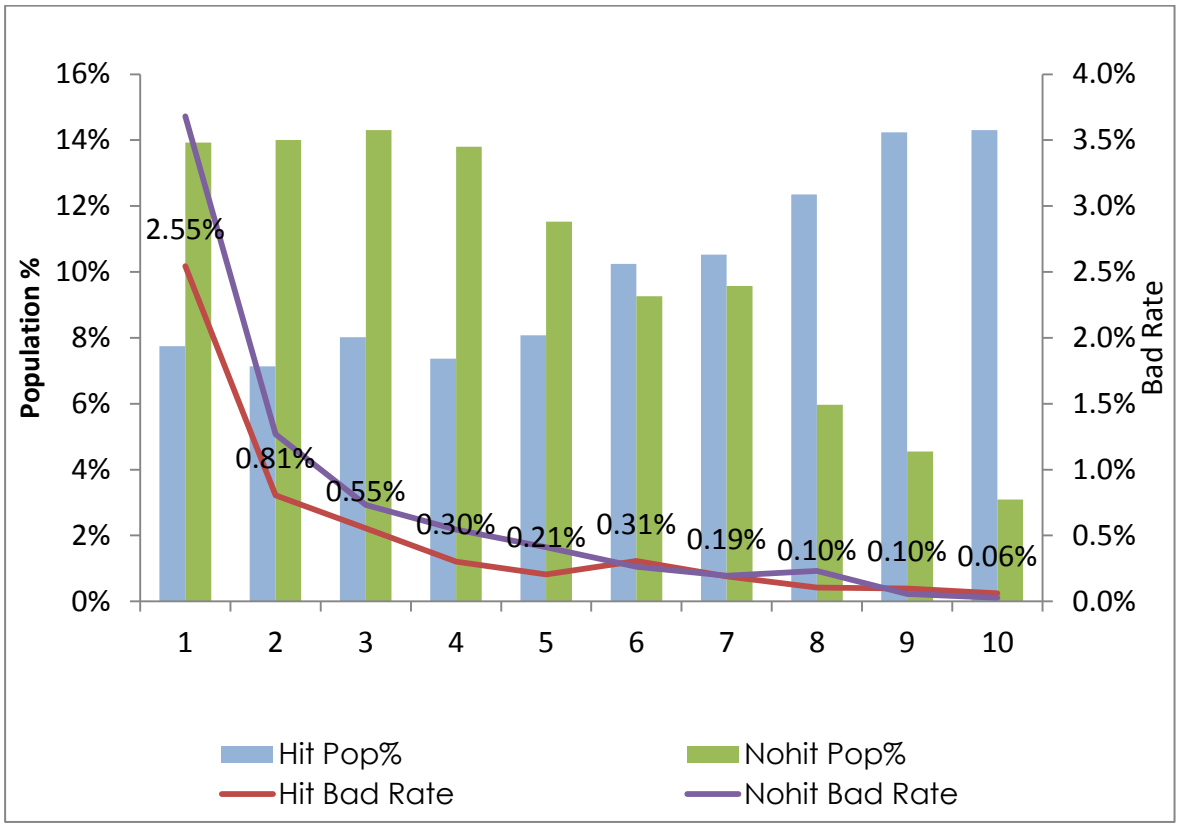
- **On training base:**



The above chart shows the population distribution and the event rate by CIBIL Hit/Nohit on the training base. We can see from the chart that the bad rate in the first bin is 2.68% which is much higher than in the second bin which is 0.80% and further decrease to flatten off in the higher bins. Population in the lower bins of the nohit applicants is much higher and it goes on decreasing in the higher bins. As it is usually considered that the nohit applicants are much riskier than the CIBIL hit applicants, and the final scorecard seems to capture

27% of the total nohit riskier applicants in first 2 bins. Hence the final scorecard is much better since we have set the cutoff at first 2 bins to reject and accept in the further bins. The populations of hit applicants in the first 2 bins is 15%, as per our consideration hit applicants are better and we are rejecting only 15% riskier applications and accepting the further applications. There is a level shift in the bad rate of nohit over the hit which shows that the nohit population is more risky than hit population as expected.

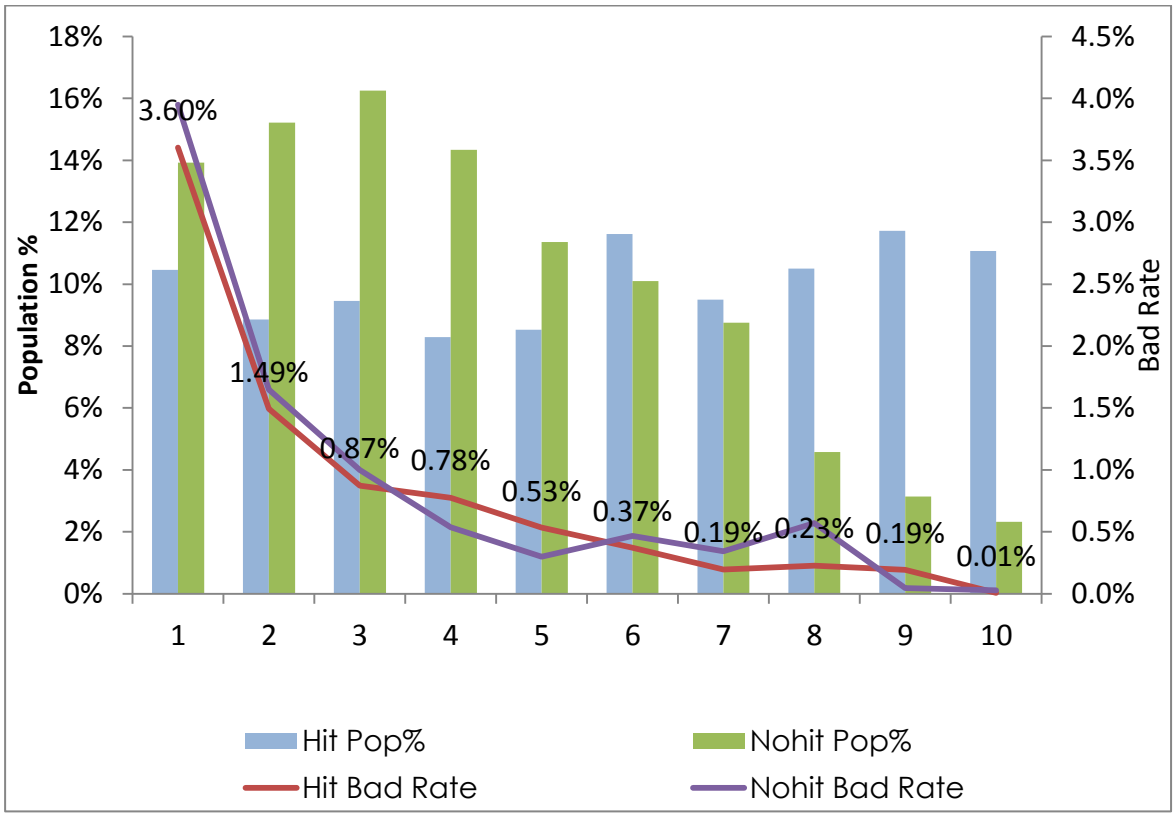
- **On validation base:**



The above chart shows the population distribution and the event rate by CIBIL Hit/Nohit on the validation base. We can see from the chart that the bad rate in the first bin is 2.55% which is much higher than in the second bin which is 0.81% and further decrease to flatten off in the higher bins. Population in the lower bins of the nohit applicants is much higher and it goes on decreasing in the higher bins. As it is usually considered that the nohit applicants are much riskier than the CIBIL hit applicants, and the final scorecard seems to capture 27% of the total nohit riskier applicants in first 2 bins. Hence the final scorecard is much better since we have set the cutoff at first 2 bins to reject and accept in the further bins. The populations of hit applicants in the first 2 bins is 15%, as per our consideration hit applicants are better and we are

rejecting only 15% riskier applications and accepting the further applications. There is a level shift in the bad rate of nohit over the hit which shows that the nohit population is more risky than hit population as expected.

- **On out of time base:**



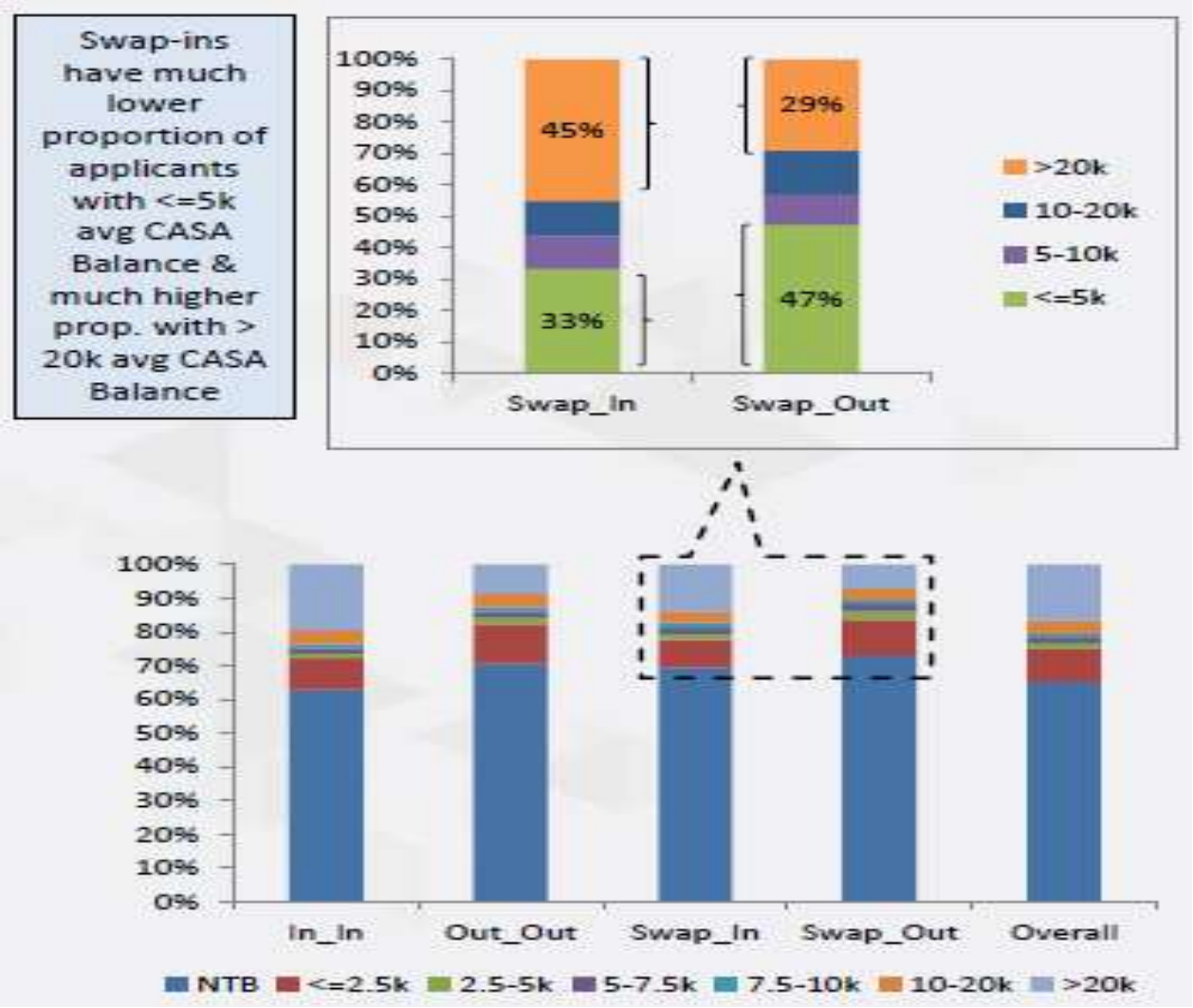
The above chart shows the population distribution and the event rate by CIBIL Hit/Nohit on the validation base. We can see from the chart that the bad rate in the first bin is 3.6% which is much higher than in the second bin which is 1.49% and further decrease to flatten off in the higher bins. Population in the lower bins of the nohit applicants is much higher and it goes on decreasing in the higher bins. As it is usually considered that the nohit applicants are much riskier than the CIBIL hit applicants, and the final scorecard seems to capture 29% of the total nohit riskier applicants in first 2 bins. Hence the final

scorecard is much better since we have set the cutoff at first 2 bins to reject and accept in the further bins. The populations of hit applicants in the first 2 bins is 18%, as per our consideration hit applicants are better and we are rejecting only 18% riskier applications and accepting the further applications. There is a level shift in the bad rate of nohit over the hit which shows that the nohit population is more risky than hit population as expected.

5.5 Swap Set Analysis

The AGB model consists of accept accounts with on us performance and reject accounts with reject inferencing. After the final selection of the scorecard it is very important to find out the efficiency of the scorecard and its robustness before its implementation. To gauge the efficiency of the scorecard it is very important to observe what kind of applicants is the scorecard accepting and rejecting. The swap set analysis is mainly done to identify the profile of customers brought-in by scorecard and thrown-out by the scorecard. It is always of best interest that the scorecard accepts the applications with low delinquencies, low enquiries, high income, high vintage on bureau, high vintage with bank and low utilization. It is also advisable to bring in applicants with less delinquency and having fewer obligations of secured products because they the secured products generally have very high loan obligation as compared to unsecured products on the applicants. Also, it is advisable to bring in applicants with less delinquency and having unsecured products because the unsecured products generally have very low loan obligation as compared to secured products on the applicants.

The applicants which were rejected by the earlier model and are accepted by the new model are called as “Swap-In” whereas applicants which were accepted by the earlier model and is rejected by the new model is called “Swap-Out”. The swap set analysis is performed over the out of time i.e.the test data only. It can be done in two forms either only on the accept base or on the through the door base. Performing swap set analysis on the accept base will only help us infer the scorecard efficiency on the accepts and blind folding over the reject applications. However if we perform swap set analysis on the through the door population it would help us understand the efficiency of the scorecard on the entire universe of the test base.



The chart above shows the swap set analysis of the variable average balance. Swap set analysis is performed on the out of time base with the through the door population. The lower chart shows the swap set analysis of the through the door population with distribution of applicants in bins described above in the form of stack chart. The “In-In” and “Out-Out” applications are those which were earlier accepted and are also accepted by new model and rejected and are also rejected by new model respectively. Final column of the stack chart is the overall distribution of applications in different bins. The above analysis consists of both existing to bank customers and new to bank customers, but it appeared that this analysis is improper and the new to bank applicants should be removed for the analysis to have more granular and clear view of the “Swap-In” and “Swap-Out”. Hence, the chart above shows

only “Swap-In” and “Swap-Out” for the existing to bank customer (i.e. existing current account and savings account customers) .

From the chart we can see that out of total Swap-Ins 45% belong to the higher account balance group whereas out of total Swap-Outs only 29% belong to the higher account balance group. Out of total Swap-Ins only 33% belong to the lower account balance group whereas out of total Swap-Outs only 47% belong to the lower account balance group. Hence we can infer that the Swap-Ins has higher proportion of the high account balance group and lower proportion of low account balance group which is inverse in the Swap-Outs.

Hence we can conclude that the reject inferencing exercise performed on the through the door population has led to a better and more stable model. The Gini index and the risk ranking signify the stability of the model. Also, the efficiency and robustness of the scorecard is confirmed from the swap set analysis. After presenting the AGB model to all the stake holders, the results were approved and the scorecard is further send for implementation.

Bibliography

[1] Siddiqi Naeem. Credit Risk Scorecards. 1st ed. Hoboken, N.J.: Wiley, 2013.

[2] "SAS/STAT(R) 9.2 User's Guide, Second Edition". Support.sas.com. N.p., 2017. Web. 6 May 2017.

[3] Bryan D. Nelson. "Variable Reduction for Modeling using PROC VARCLUS". Fingerhut Companies Incorporated, Minnetonka, MN. http://www2.sas.com/proceedings/sugi26/p261-26.pdf. N.p., 2017. Web. 6 May 2017.

[4] Jerome Friedman, Trevor Hastie, Robert Tribshirani. "Additive Logistic Regression: A Statistical View of Boosting". The Annals of Statistics2000. Vol 28, No.2, 337-407.