University of Brighton

“Building Behavioural Scorecard for Default Prediction and Identifying key Feature for Delinquency”

Roshan Kachwal

MSc Data Analytics

Student: 22851496

School of Computing, Engineering and Mathematics

Mithras House, Lewes Rd, Brighton, East Sussex BN2 4AT, UK

Submitted: 19th September 2023

Dissertation word count: 10,000

REFERENCE CITATIONS

I declare that no part of the work in this report has been submitted in support of an application for another degree or qualification at this or any other institute of learning.

I give my consent that hard-copy and soft-copy of this dissertation can be made available in full for subsequent. students taking this module, to external examiners and to available accreditation bodies.

Signed: Date: 24/09/23

ABSTRACT

This study aims to improve credit risk management in the banking sector by building predictive models for identifying loan defaults and delinquencies. Our study provides significant insights by leveraging data from a financial institution including 1750 loan accounts with up to 40 monthly snapshots per account. Our findings highlight the critical impact of overdue payments, gross income, MoB and age range in predicting defaults, with missed payments showing as a particularly strong predictor. Furthermore, we find behavioural scores, application scores, MOB, Monthly Income, and age range as critical indicators of delinquencies.

To accomplish this, the research conducted a thorough analysis using Excel, Python, and SAS. Using data from the final 6 months of the observation period, we developed a static behavioural scorecard for predicting defaults. In addition, we use the powerful Random Forest model to assess the strength of particular characteristics as default indications. Using a variety of validation methodologies, our scorecard exhibits strong discriminating strength, which improves its precision in default estimates. Three unique scorecard models have been created, each using a different attribute to predict the likelihood of an account defaulting.

Simultaneously, our study extends to the development of numerous prediction models targeted at identifying key features indicating delinquency using account behavioural data. These models were carefully tested using the 5-fold cross-validation process. It's important to highlight that the dataset we started with was unbalanced, resulting in the use of many statistical approaches to balance and assess the data properly in order to achieve desired accuracy.

Our study presents a thorough behavioural scorecard that allows for accurate default prediction by assigning scores. Furthermore, the prediction models and feature relevance ranking enable financial institutions to proactively manage delinquency risk with an accuracy rate of up to 97%, supporting efficient credit risk assessment practises.

Keywords: Credit Risk Management, Default Prediction, Delinquency Forecasting, Predictive Modelling, Feature Importance, Behavioural Scorecard.

Contents

ACKNOWLEDGMENTS

This project has been made possible by those I would like to thank for their time, attention and encouragement: my supervisor Dr Alexey Chernov for initial feedback on this analysis and dissertation, course leader Dr Sonia Inacio for pushing SAS to demonstrate the Forecasting & Credit Risk Analysis.

1. BUSINESS UNDERSTANDING & METHODOLOGY

1.1. INTRODUCTION - Building Behavioural Scorecard for Default Prediction and Assessing Important Feature for Delinquency

In the changing field of risk evaluation and predicting defaults, in the industry credit risk assessment plays a crucial role. It is essential for determining whether individuals or entities are eligible to obtain credit facilities. As financial institutions get number of loan application it becomes vital to assess the probability of borrowers failing to meet their credit obligations. This assessment, known as credit risk, helps in making decisions about lending and protecting the well-being of institutions. This report goes deep into the world of credit risk and explore scorecard models in detail and identifying key indicators that predict the delinquency stage and create a model using those key indicators that form a strong and predictive framework, for managing credit risks (Peterdy, 2023).

Understanding the risk associated with lending money involves evaluating the likelihood of borrowers failing to make their payments and considering how it might impact the stability of the lender. To effectively handle this risk, it is crucial to have an understanding of the factors that contribute to accounts getting default or goes into delinquency stage. This includes assessing the borrower’s stability, credit history, employment status, demographic data, and other relevant aspects. Additionally, being able to differentiate between types of credit risk models, such, as application scorecard models and behavioural scorecard models is essential, for developing an informed and strong credit risk management strategy (Fibe, n.d.).

A scorecard model is a quantitative technique used in credit risk assessment that makes it possible for the assignment of a credit score or rating to borrowers depending on their creditworthiness. These models provide an objective and consistent framework for making credit judgements by using historical data and advanced statistical approaches. There are two most important and popular scorecard model. Firstly, an application scorecard which examines creditworthiness at the time of application, considering criteria such as credit ratings, financial information, monthly income, residential status etc. supplied by the applicant. Other, Behavioural Scorecard, this approach evaluates borrowers' creditworthiness based on their prior financial conduct, typically relying on historical data such as repayment trends and delinquency histories (Fibe, n.d.).

There are two kinds of Behavioural Scorecards: static and dynamic. Static scorecards assess credit risk based on historical data, without considering changes in borrower behaviour over time. Dynamic Behavioural Scorecard: Dynamic scorecards continuously update credit risk evaluations, taking into account changing borrower habits and payment patterns. The decision between static and dynamic scorecards has ramifications for credit risk management. Dynamic scorecards provide real-time insights on borrower behaviour, but they might be more difficult to deploy. Static scorecards give a simpler model but may lag in reflecting developing borrower behaviour. Another essential component in a scorecard is the observation period, which has a significant influence on the outcome of the credit risk model. In general, most financial regulatory authorities expect a minimum of 12 to 16 months of observation period to effectively examine each account (Finalyse, n.d.).

Delinquency is critical because it provides as an early warning system for potential defaults. Delinquency is described as the inability of a borrower to repay a credit obligation on time, demonstrating a deviation from the predetermined repayment schedule. It is a vital and reliable credit risk indicator. Spotting delinquent accounts early allows lenders to take proactive steps to limit possible losses, collaborate with borrowers to identify solutions, and preserve the overall health of their credit portfolios. Identifying reliable indicators of delinquency and default is critical for improving credit risk-control practises (LendFoundry.com, 2021). It helps lenders to enhance their risk assessment models, reduce their exposure to high-risk borrowers, and make informed lending decisions that protect their financial stability.

1.2. OBJECTIVES & HYPOTHESIS

The primary goals of this project are to harness the power of customer behavioural data gathered over the previous six months in order to create a relevant behavioural scorecard. Alongside, we aim to use a wide range of prediction models, such as logistic regression, random forest, and decision trees, to identify important indicators of borrower's accounts going into delinquency. In-depth discussion of these goals will follow:

I. Creating a Behavioural Scorecard

- **Understanding Behavioural Scorecard**: A behavioural scorecard is a powerful instrument in credit risk assessment, meant for assessing a borrower's creditworthiness based on past financial behaviour and activities. This scorecard combines an array of static and dynamic criteria to create an in-depth understanding of the borrower's risk profile.
- **Data Selection and Preprocessing**: During this step, we select and preprocess the most recent six months of customers behavioural data to ensure its relevance are up to date. This stage comprises of data cleansing, dealing with missing values, and organising the data in a format suitable for analysis (Credit Scoring Series Part Two: Credit Scorecard Modeling Methodology, 2022).
- **Feature Engineering**: This stage involves transforming the data into a more suitable format that can be utilised as input for our models. The process will also include aggregating data over time periods, normalisation of numerical data, and balancing the output variable to eliminate bias.
- **Model Development**: We will start the process of creating the behavioural scorecard using this pre-processed dataset. This stage comprises of applying logistic regression model to identify which variable have strong predictive power to measure default and based on the predictive power of each variable award credit scores or ratings to borrowers' accounts. The scorecard gives lenders an accurate and measurable tool for making lending choices (Finalyse, n.d.).

II. Identifying Strong Indicators of Delinquency

- **Utilizing the Power of Machine Learning**: We will use a broad array of prediction models, including logistic regression, random forest, and decision trees, to gain a better understanding the prominent signs for customers who may become delinquent (LendFoundry.com, 2021).
- **Feature Importance Analysis**: Each of these models offers a distinct perspective on the data. We will use feature importance analysis to determine which factors are most important in predicting delinquency. This procedure highlights the primary causes of delinquency.
- **Model Comparison:** We will examine the performance of several models in terms of predictive accuracy as well as their capacity to detect significant signs. By comparing its

strengths and errors, we gather a complete understanding of the data (LendFoundry.com, 2021)

- **Interpreting the Results:** We will investigate the insights received from the outcomes of these models. Evaluating those that were identified as strong indicators and interpreting their importance in the context of credit risk management (LendFoundry.com, 2021).
- **Recommendations:** Based on the findings of our investigation, we will give suggestions to help guide credit risk management techniques. These suggestions might include changing loan criteria, improving risk assessment models, or adopting proactive actions to reduce possible delinquency (Credit Scoring Series Part Two: Credit Scorecard Modeling Methodology, 2022).

Our major goal is to supply lending institutions with an effective behavioural scorecard that provides a thorough evaluation of borrower's creditworthiness. Simultaneously, we want to find crucial signs that might indicate possible delinquency, allowing lenders to make well-informed and proactive risk management decisions. We want to improve the precision and efficiency of credit risk assessment by using the combined capabilities of data analytics, machine learning methods, and domain expertise. This will protect the financial stability of lending institutions.

2. LITERATURE REVIEW

Credit scoring is the application of statistical models to predict the chance of a potential borrower defaulting on a loan. Credit scoring can support and assist in maximising the projected profit from their customers for financial institutions by minimising the probability of a customer defaulting, which predicts customer risk. Credit scoring models are commonly used to assess business, real estate, bank, and consumer loans' (Gup and Kolari, 2005: 508). Credit scoring is essentially the "application of statistical models to transform relevant data into numerical measures that guide credit decisions." It is the industrialization of trust; the inevitable next step in the evolution of subjective credit ratings (Beynon, 2005).

The classification of good and bad credit is critical, and it is the goal of a credit scoring model (Lee et al., 2002; Lim and Sohn, 2007). As a result, the necessity for a suitable categorisation approach is essential. But knowing what factors influence a new applicant's classification? According to a literature study, factors such as gender, age, marital status, dependants, educational level, employment, loan amount, loan term, monthly income, and having a credit history are commonly employed in the development of scoring models (Orgler, 1971). In financial applications, scoring models have been built using three variables (Fletcher and Goss, 1993; Pendharkar, 2005) to over 20 variables (Tam and Kiang, 1992; Desai et al., 1996; Jo et al., 1997). Meanwhile, others have used more variables in their analysis.

Credit scoring classification models are used to classify new applicants in two categories to either approve or reject. Classification techniques may also be divided into two types: traditional approaches and modern statistical techniques. Weight of evidence, multiple linear regression, discriminant analysis, probit analysis, and logistic regression are examples of the traditional type. The modern approaches and methods, comprise of fuzzy algorithms, genetic algorithms, expert systems and neural networks (Hand and Henley, 1997). On the one hand, one of the most important methods to credit scoring applications is still the use of just two categories of customer credit, either 'good' or 'bad' (Orgler, 1971; Boyes et al., 1989; Banasik et al., 2001; Lee et al., 2002; Kim and Sohn, 2004). The weight-of-evidence approach is being used for these credit scoring applications. While a handful of research have looked at the weight-of-evidence measure (Bailey, 2001; Banasik et al., 2003; Siddiqi, 2006; Abdou, 2009b).

The selection of the sample size is one of the most difficult aspects of developing a credit scoring model. It is considered that the higher the sample size, the more accurate the scoring model. These decisions mostly depend on the availability of the data, the market's structure, and the degree to which this specific dataset will represent the community. (Malhotra and Malhotra (2003), Kim and Sohn (2004), Lee and Chen (2005), and Sustersic et al. (2009)) applied their analysis based on data sets of less than 1100 observations in various personal loan applications. From studies it has been highlighted there was a sample selection bias particularly pertaining to the analysis of only applicants that have been accepted (Greene, 1998; Banasik et al., 2003; Banasik and Crook, 2005, 2007; Verstraeten and Van Den Poel, 2005).

According to Al Amari (2002: 41), while several credit scoring models have been employed in the field, the following essential questions have yet to be resolved conclusively: What is the best way to evaluate customers? What factors should a credit analyst consider while evaluating applications? What type of information is required to improve and assist decision-making? What is the best measure to predict the loan quality (whether a customer will default or not)? To what extent can a customer be classified as good or bad?

In general, two common processes for credit evaluation are loan officer's subjective judgement and credit scoring (Crook, 1996). According to Sullivan (1981) and Bailey (2004), in a judgmental approach evaluation, each credit application contains information that must be examined independently by a decision-maker 'creditor'. In contrast, in a credit scoring model, analysts often utilise their prior experience with debtors to develop a quantitative model for categorising acceptable and unacceptable loan applications. Financial organisations, particularly banks, utilise credit scoring algorithms for granting credit to eligible applicants and distinguish between good and bad credit. Credit scoring can lower the cost of the credit process and the predicted danger of a poor loan, while also improving credit decisions and saving time and effort (Lee et al., 2002; Ong et al., 2005).

The ability and a need for banking and private lending organisations to view the future and predict the success or failure of a loan application to satisfy their obligation are substantial. Borrowers who appear to be ideal candidates for loan origination may exhibit unpredictable payment and financial behaviour once their loan is granted. This is something that underwriters may not be able to anticipate at the time of loan origination. However, this activity significantly raises the danger. Delinquency often occurs when one or more planned payments are missed, resulting in a breach of the agreed-upon payment conditions.

Predictive models are critical in detecting delinquency in numerous kinds of businesses, including financial services, lending, and others. Some machine learning methods, including logistic regression, decision trees, and neural networks, play a crucial role in assessing the risk of delinquency based on a variety of features and variables. Researchers discovered crucial factors for predicting delinquency. Credit score, payment history, debt-to-income ratio, work status, loan amount, and demographic characteristics are often listed as important predictors of delinquency (Chen, S., Guo, Z., & Zhao, X. (2021)).

Predictive models are widely used by banks and credit card businesses to assess credit risk and control delinquency rates. Financial institutions use predictive models to forecast the likelihood of delinquency for existing accounts. Using this approach, they can identify accounts at risk of becoming delinquent and take proactive measures to reduce the risks associated with loan defaults and delinquent accounts.

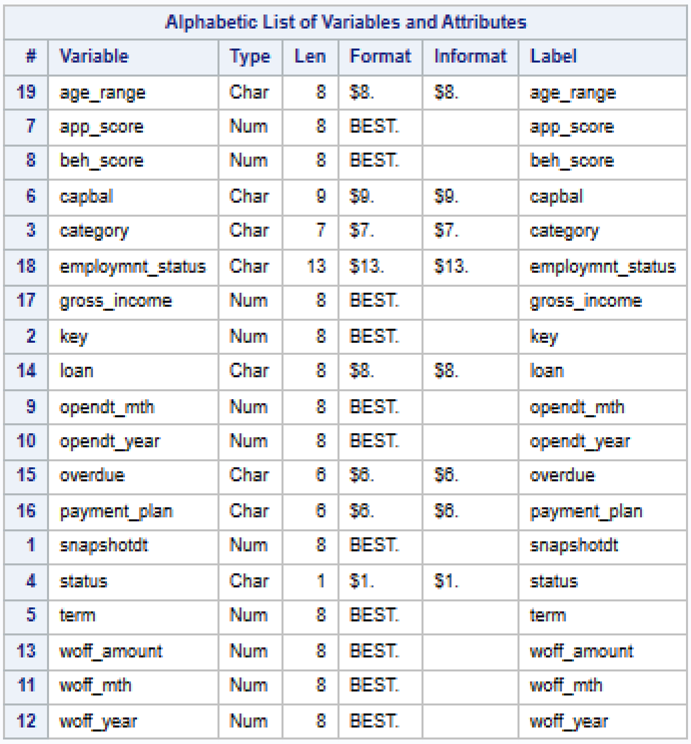
3. DATA UNDERSTANDING

3.1. DATA DESCRIPTION

The dataset utilised in this study was obtained from a financial institution in the form of an Excel file. It is a comprehensive collection of data taken from clients who acquired loans from this organisation. The dataset contains a total of 1,750 unique accounts, each with a maximum of 40 associated snapshots. The dataset was originally provided as an Excel file, making it easy to analyse. It is important to highlight that all relevant data has been aggregated into a single dataset. Furthermore, the data is organised in a monthly snapshot format, allowing for an analysis of customer behaviour of historical financial transaction, credit history loan-related features and etc.

3.1.1. Data Dictionary

This dataset contains 25036 records and 19 variables of both numerical and categorical type. The status variable is the target variable for creating Behavioural scorecard whereas category is the target variable for predicting the strong indicator of delinquency.



Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
19	age_range	Char	8	$8.	$8.	age_range
7	app_score	Num	8	BEST.		app_score
8	beh_score	Num	8	BEST.		beh_score
6	capbal	Char	9	$9.	$9.	capbal
3	category	Char	7	$7.	$7.	category
18	employmnt_status	Char	13	$13.	$13.	employmnt_status
17	gross_income	Num	8	BEST.		gross_income
2	key	Num	8	BEST.		key
14	loan	Char	8	$8.	$8.	loan
9	opendt_mth	Num	8	BEST.		opendt_mth
10	opendt_year	Num	8	BEST.		opendt_year
15	overdue	Char	6	$6.	$6.	overdue
16	payment_plan	Char	6	$6.	$6.	payment_plan
1	snapshotdt	Num	8	BEST.		snapshotdt
4	status	Char	1	$1.	$1.	status
5	term	Num	8	BEST.		term
13	woff_amount	Num	8	BEST.		woff_amount
11	woff_mth	Num	8	BEST.		woff_mth
12	woff_year	Num	8	BEST.		woff_year

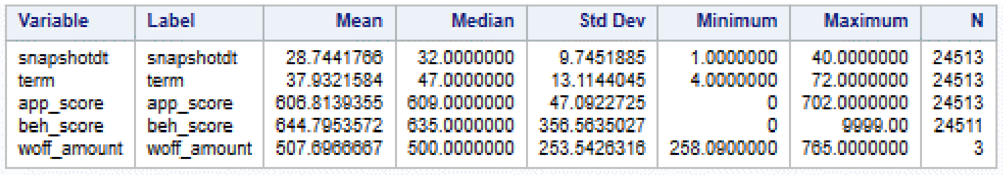
Table 1

Below figure is the explanation of all the variables contained in the table 1.

Variable	Definition	Type	Format	Note

snapshotdt	reporting period	dynamic	numeric	Month end reporting period
key	Key	static	character	Unique at account level
category	delinquency stage	dynamic	character	Level of the account position (Up-To-Date to 10+)
status	level of quality ('G'ood, 'B'ad and 'C'losed)	dynamic	character	
term	contract length	static	numeric	
capbal	current balance	dynamic	numeric	
app_score	score captured at application stage	static	numeric	Generated for account at application
beh_score	monthly gauge score of accounts on the book	dynamic	numeric	Generated at each month end
opendt_mth	month when deal was written	static	numeric	Month when account was open
opendt_year	year when deal was written	static	numeric	Year when account was open
woff_mth	month of write off event	static	numeric	
woff_year	year of write off event	static	numeric	
woff_amount	write off amount	static	numeric	
gross_income	bands of income at origination	static	numeric	
employmnt_status	bands of employment status at origination	static	text	
age_range	bands of age at origination	static	character	
loan	loan Amount	static	character	
overdue	Amount customer failed to pay in that month	static	character	
Monthly_income	Income every month	Static	character	
MOB	Month on Books	Dynamic	Numeric	
payment_plan	Monthly payment plan set by the customer	static	character	

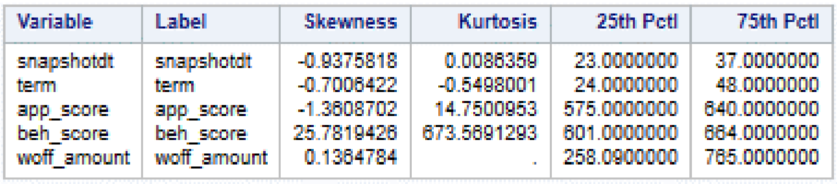
3.1.2. Summary Statistics



Variable	Label	Mean	Median	Std Dev	Minimum	Maximum	N
snapshotdt	snapshotdt	28.7441766	32.0000000	9.7451885	1.0000000	40.0000000	24513
term	term	37.9321584	47.0000000	13.1144045	4.0000000	72.0000000	24513
app_score	app_score	606.8139355	609.0000000	47.0922725	0	702.0000000	24513
beh_score	beh_score	644.7953572	635.0000000	356.5635027	0	9999.00	24511
woff_amount	woff_amount	507.6966667	500.0000000	253.5426316	258.0900000	765.0000000	3

Table 2

The Table 2 shows the mean, median and mode which give numerical characteristics distribution and dispersion of the values of the variables.

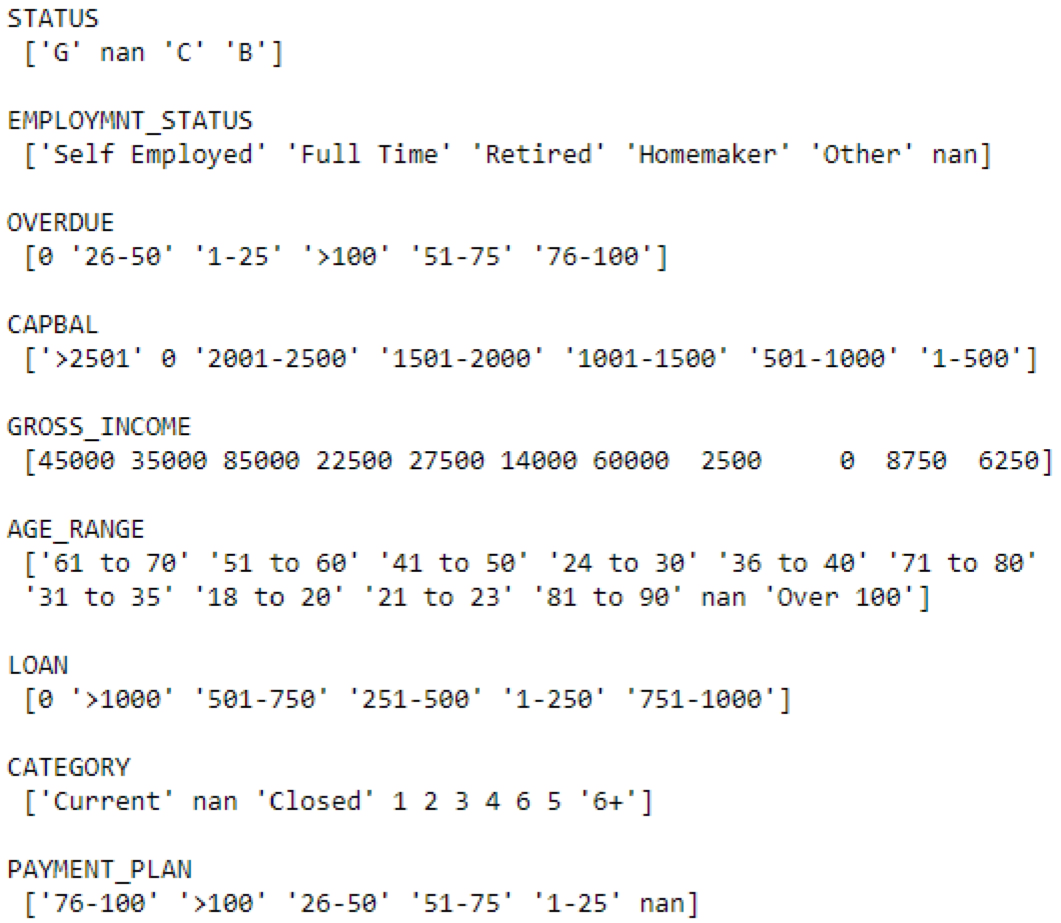


Variable	Label	Skewness	Kurtosis	25th Pctl	75th Pctl
snapshotdt	snapshotdt	-0.9375818	0.0086359	23.0000000	37.0000000
term	term	-0.7006422	-0.5498001	24.0000000	48.0000000
app_score	app_score	-1.3608702	14.7500953	575.0000000	640.0000000
beh_score	beh_score	25.7819426	673.5691293	601.0000000	664.0000000
woff_amount	woff_amount	0.1364784	.	258.0900000	765.0000000

Table 3

The Table 3 further shows the dispersion of the values of the variables.

3.1.3. Unique variables



```
STATUS
 ['G' nan 'C' 'B']

EMPLOYMNT_STATUS
 ['Self Employed' 'Full Time' 'Retired' 'Homemaker' 'Other' nan]

OVERDUE
 [0 '26-50' '1-25' '>100' '51-75' '76-100']

CAPBAL
 ['>2501' 0 '2001-2500' '1501-2000' '1001-1500' '501-1000' '1-500']

GROSS_INCOME
 [45000 35000 85000 22500 27500 14000 60000  2500     0  8750  6250]

AGE_RANGE
 ['61 to 70' '51 to 60' '41 to 50' '24 to 30' '36 to 40' '71 to 80'
 '31 to 35' '18 to 20' '21 to 23' '81 to 90' nan 'Over 100']

LOAN
 [0 '>1000' '501-750' '251-500' '1-250' '751-1000']

CATEGORY
 ['Current' nan 'Closed' 1 2 3 4 6 5 '6+']

PAYMENT_PLAN
 ['76-100' '>100' '26-50' '51-75' '1-25' nan]
```

Table 4

Snapshot Count variable

Apart from Above variable, a new snapshot count variable will be created which will contains number of snapshots consist in each key which is a unique identifier of each account.

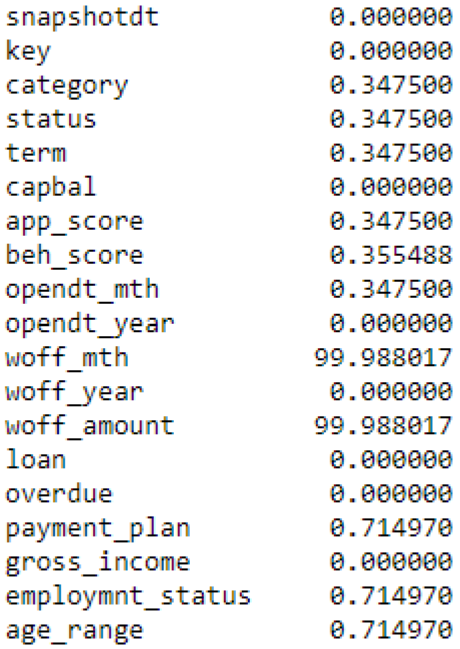
3.2. DATA QUALITY

3.2.1. Duplicate Data

This dataset contains approximately 364 duplicates records out of 25036. After removing all the duplicates, this dataset contains approximately 24672 records and 19 variables (Credit Scoring Series Part Three: Data Preparation and Exploratory Data Analysis, 2022).

3.2.2. Missing Values and Variable Exclusion

Missing Values (in percentage)



```
snapshotdt           0.000000
key                  0.000000
category             0.347500
status               0.347500
term                 0.347500
capbal               0.000000
app_score            0.347500
beh_score            0.355488
opendt_mth           0.347500
opendt_year          0.000000
woff_mth            99.988017
woff_year            0.000000
woff_amount         99.988017
loan                 0.000000
overdue              0.000000
payment_plan         0.714970
gross_income         0.000000
employmnt_status     0.714970
age_range            0.714970
```

Table 5

From Table 5, It is clear that variables such as Write of amount are 99% empty, indicating that the bank has a low-risk exposure. Hence, there is not much analysis that can be done on the variable like write of amount and all the variable which is related to it. Therefore, will exclude this variable before going to the modelling stage. Also, no other variables contain missing values more than 5%, so we can drop this records for predicting important feature for delinquency (Credit Scoring Series Part Three: Data Preparation and Exploratory Data Analysis, 2022).

3.2.3. Variable Exclusion

For, creating behavioural scorecard we won't be needing variable like application score and behavioural score but for identifying the important feature for the delinquency we will be using variable such as application score and behavioural score. We are going to create few models which won't contain variable overdue because of its very strong relationship with the response variable (Credit Scoring Series Part Three: Data Preparation and Exploratory Data Analysis, 2022).

3.2.4. Distribution and Outliers

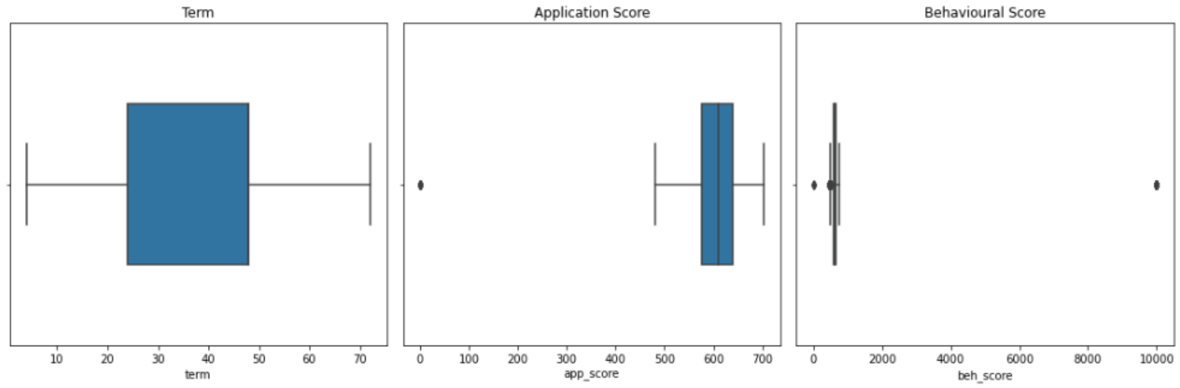


Figure 1

From figure 1, we can observe that Term doesn't contain any outliers whereas the application score and behavioural score contains outliers.

For variable app_score & beh_score, minimum whisker is 481 & 507 respectively and maximum whisker is 702 & 743 respectively and any records containing value beyond this are outliers. And these records will be excluded before predicting the important feature for the delinquency.

3.2.5. Consistency, Accuracy & Timeliness

After analysing the dataset, it has come to notice that variables like open date month and open date year are not in proper date format. This could create an issue to analyse the behaviour of the account overtime. But this issue will be verified in the Exploratory data analysis which would then become evident.

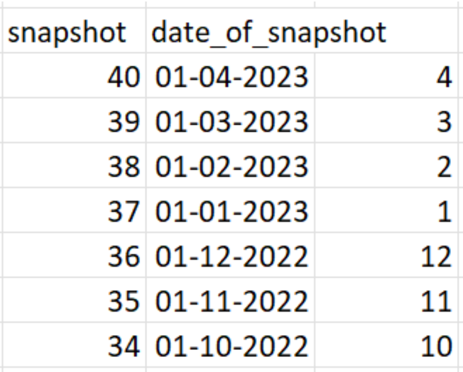
4. DATA PREPARATION

4.1. DATA PREPARATION FOR BEHAVIOURAL SCORECARD

4.1.1. Selecting The Observation Period

The selection of the observation period is a vital stage in the process of preparing data for the behavioural scorecard. Each unique key in our dataset acts as an identification for individual accounts, and we have around 40 snapshots inside each key, with the 40th snapshot being the most recent. There are two reasons for selecting the past six months of data for study. For starters, a considerable amount of accounts default over this time period, which is a major event we want to capture and examine in our scorecard. This tendency is visible in the contingency tables investigated during the Exploratory Data Analysis (EDA) phase.

To determine our observation window, we used particular criteria. First, we ensured that each account in the dataset had at least 6 months of observable data. Second, the observation window was limited to the 34th and 40th snapshots. Now that we are developing a static behavioural scorecard, we are effectively deciding on a particular moment in time to judge the creditworthiness of each account. All independent variable values for that account are obtained from the 34th snapshot. The response variable, which represents the default status, is obtained from the 40th snapshot (Credit Scoring Series Part Two: Credit Scorecard Modeling Methodology, 2022).



snapshot	date_of_snapshot	
40	01-04-2023	4
39	01-03-2023	3
38	01-02-2023	2
37	01-01-2023	1
36	01-12-2022	12
35	01-11-2022	11
34	01-10-2022	10

Figure 2

This strategy is legitimate because once an account reaches a default state, it typically remains in that state. As a result, by capturing the status of the account at the 40th snapshot, we gain a meaningful and representative snapshot of its credit behaviour across the selected observation window.

After this procedure, this dataset will have about 1269 unique records. And the proportion of output variables is 99.07% Good and 0.93% Bad.

4.1.2. Exclusion Criteria

Exclusion criteria are constraints or limits given to a dataset to identify and reject individual accounts or data points that are inappropriate for inclusion in the research. These criteria help to ensure the quality and relevance of the data used to generate the behavioural scorecard. In accordance with the exclusion criteria, we will exclude certain accounts from our analysis for the following reasons:

1. **Closed Default Status:** Any accounts that are marked as having a default status of ‘C’ will be excluded from the analysis. This exclusion is logical since accounts that have already been closed are no longer relevant to assessing current credit behaviour.
2. **Delinquency Category (Closed):** Similarly, accounts with a delinquency category of 'Closed' which indicates a closed account, will also be excluded. These closed delinquency records do not contribute to the assessment of current risk evaluation.
3. **Loan Amount Equals 0:** Accounts with zero loan balances will be excluded from the study. This is because such accounts don't include any active credit or debt, making them irrelevant to creditworthiness evaluation.
4. **Delinquency Stage 2:** Accounts in the second stage of delinquency will be eliminated from the analysis. While these accounts are not yet classed as defaults, they do qualify a "grey area" in which credit risk is missed last payments but not defaulted. Excluding these helps to balance the data and keeps our attention on clearer default indicators.

By employing these exclusion criteria, we want to improve our dataset, focusing on accounts that are most relevant for analysing current credit conduct and enhancing the quality and robustness of our behavioural scorecard analysis.

After this procedure, this dataset will have about 704 unique records. And the proportion of output variables is 97.59% Good and 2.41% Bad.

4.1.3. Defining The Default

When preparing data for the behavioural scorecard, it is critical to develop a clear and well-defined criterion for what constitutes a "default." This definition will serve as the foundation for our study. As a result, we've amended our definition of default to include the following:

Definition of Default:

For this analysis, we classify an account as being in a state of default if it meets any of the following criteria:

- **Category Stage 3+ (60 Days Past Due)**: An account is said to be in default if it is assigned a delinquency stage 3 or above, which often means it is 60 days or more past due (Team, 2023).
- **Status "Bad":** Additionally, if an account's status is labelled as "bad," it is classified as being in a state of default.

Conversely, accounts that fall into the category stage 1 or lower are considered not to be in default based on this definition.

It is critical to highlight that the treatment of accounts in category stage 2 is distinct and will be handled individually in the context of the exclusion criteria, as explained in the next parts of this report.

After this procedure, this dataset will have about 699 records. And the proportion of output variables is 97.28% Good and 2.72% Bad.

4.1.4. Data Transformation

During the data transformation process, we make the necessary modifications to the dataset in order to ensure that it is in a suitable format for analysis. This phase comprises encoding categorical data into numerical values to enhance model compatibility. Specifically, we will perform the following transformations:

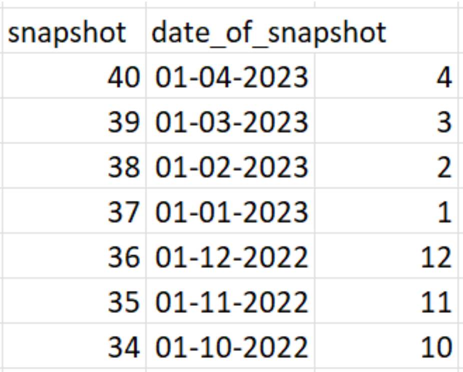
1. **Delinquency Category Variable:** We will convert the delinquency category variable, which is currently categorical, into a numerical format. This transformation involves assigning numerical values to each category, providing a clear representation of delinquency stages. The transformation will be as follows:
    - "Current" will be encoded as 0.
    - "6+" will be encoded as 7.
    - All other categories will retain their existing numerical values.
2. **Default Status Variable:** The default status variable, currently labelled as "G" and "B" will be transformed into binary format for ease of analysis. This transformation assigns numerical values to represent default and non-default status:
    - "G" will be encoded as 0.
    - "B" will be encoded as 1.

These data transformations guarantee that our dataset is standard and ready for modelling. It facilitates the analysis and modelling of credit behaviour by simplifying the description of crucial variables. My final target variable distribution is 97.28% Good and 2.72% Bad.

4.2. DATA PREPARATION FOR PREDICTING IMPORTANT FEATURE FOR DELIENQUENCY

4.2.1. Selecting The Observation Period

Like the data preparation for the behavioural scorecard, the process of selecting the observation period is a crucial step in preparing data for predicting important features related to delinquency. The main distinction here is that, unlike the behavioural scorecard, we will be utilizing data from all snapshots between the 34th and 40th month for each unique key (Credit Scoring Series Part Two: Credit Scorecard Modeling Methodology, 2022).



snapshot	date_of_snapshot	
40	01-04-2023	4
39	01-03-2023	3
38	01-02-2023	2
37	01-01-2023	1
36	01-12-2022	12
35	01-11-2022	11
34	01-10-2022	10

Figure 3

To summarize, we will use all snapshots from the 34th to the 40th month for each unique key, allowing us to capture a comprehensive view of credit behaviour leading up to potential delinquency events. This wide data window will serve as the foundation for our study, which will anticipate the key factors associated with delinquency (Credit Scoring Series Part Two: Credit Scorecard Modeling Methodology, 2022).

After this procedure, this dataset will have about 8726 records. And the proportion of output variables is 99.07% Good and 0.93% Bad.

4.2.2. Exclusion

To ensure the quality and relevance of the dataset, exclusion criteria must be established during data preparation for predicting important features associated to delinquency. The following are the exclusion criteria for this specific analysis:

1. **Loan Amount Equals 0**: Accounts with zero loan balances will be eliminated from the study. Accounts with no related loan amount are not relevant to the evaluation of delinquency risk since there is no credit exposure to consider.

2. **Outliers in Application Score and Behavioural Score:** Accounts that exhibit extreme outlier values in both application score and behavioural score variables will also be excluded from the analysis. Outliers have the potential to skew statistical analysis and predictive models. As a result, eliminating these extreme data points contributes to the integrity of our study.

By implementing these exclusion criteria, we aim to ensure that the dataset used for predicting important delinquency-related features is both meaningful and statistically robust. This process of refining allows us to focus on the most important data points for our research and modelling efforts.

After this procedure, this dataset will have about 4852 records. And the proportion of output variables is 98.58% Good and 1.42% Bad.

4.2.3. Data Transformation

We will categorise accounts depending on their delinquency stage as part of the data transformation process for predicting important variables associated with delinquency. This transformation will be used as follows to simplify the representation of delinquent status:

1. **Delinquency Stage Recategorization:** We will reclassify accounts into two distinct categories based on their delinquency stage:
    - **Delinquent Accounts:** Accounts with a delinquency stage of "2" or "2+" will be categorized as delinquent. This consolidation simplifies the representation of delinquent accounts.
    - **Non-Delinquent Accounts**: Accounts with a delinquency stage of "1" or any value less than "1" will be categorized as non-delinquent. This category encompasses accounts that are not in a delinquent state (LendFoundry.com, 2021).

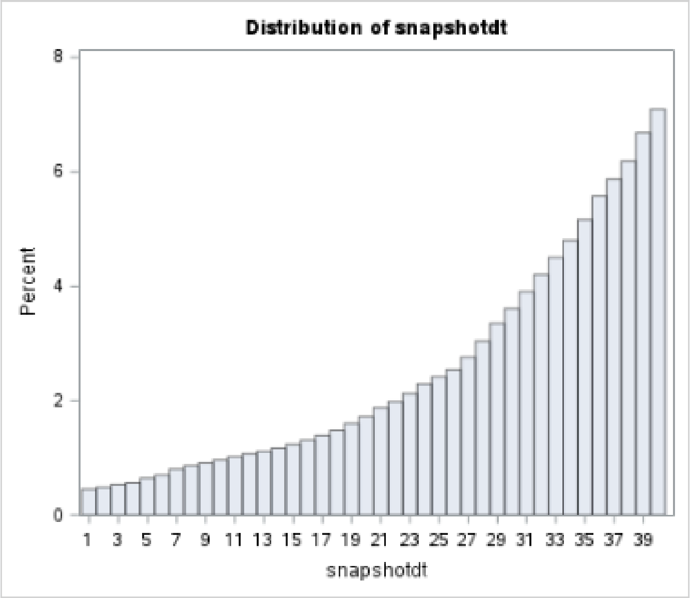
We generate a binary representation of delinquent status by recategorizing accounts in this manner, which will serve as an essential component in our research and modelling efforts. This change allows us to focus on the contrast between delinquent and non-delinquent accounts, making it easier to predict significant delinquency traits (LendFoundry.com, 2021). Final Target variable distribution is 96.54% Good and 3.46% Default.

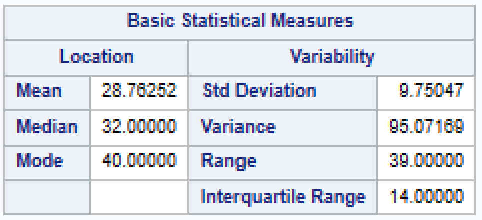
5. DATA EXPLORATION

5.1. EXPLORATORY DATA ANALYSIS

5.1.1. Univariate Distribution

1. **Snapshot:**

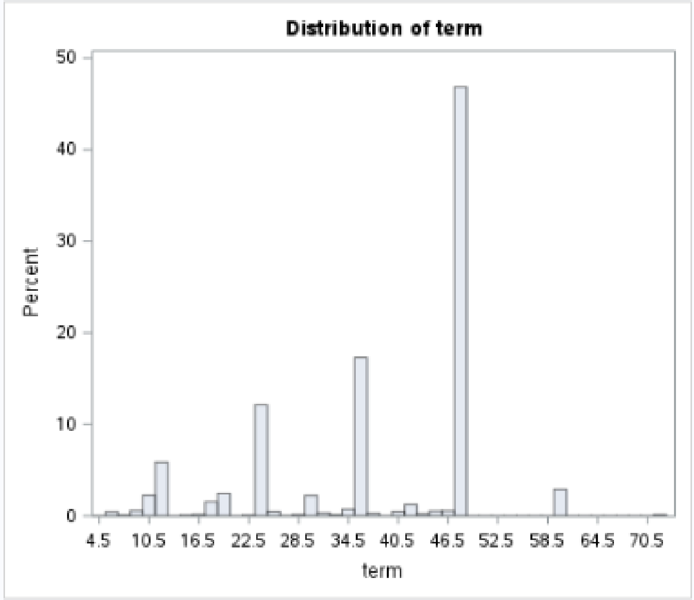


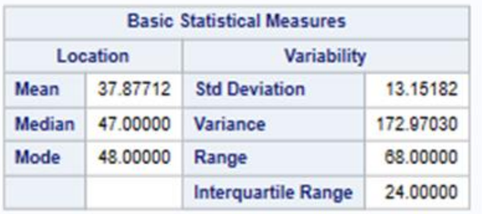


Basic Statistical Measures			
Location		Variability	
Mean	28.76252	Std Deviation	9.75047
Median	32.00000	Variance	95.07169
Mode	40.00000	Range	39.00000
		Interquartile Range	14.00000

The mean being less than the median and the mode being greater than both, hence we can conclude that snapshotdt variable distribution is left-skewed.

2. **Term:**

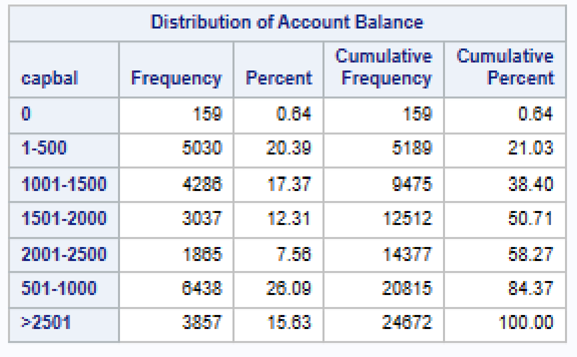




Basic Statistical Measures			
Location		Variability	
Mean	37.87712	Std Deviation	13.15182
Median	47.00000	Variance	172.97030
Mode	48.00000	Range	68.00000
		Interquartile Range	24.00000

The data dispersion appears to have some indications of right-skewness, primarily because the mode is higher than both the mean and median.

3. **Account Balance**

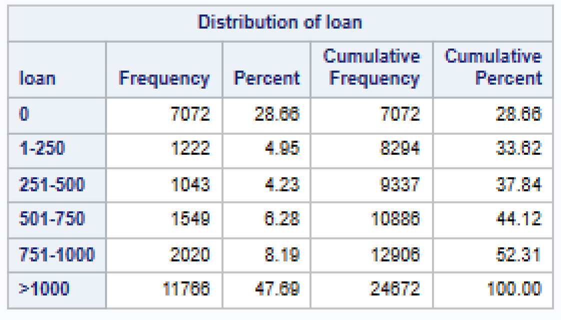


Distribution of Account Balance				
capbal	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	159	0.64	159	0.64
1-500	5030	20.39	5189	21.03
1001-1500	4286	17.37	9475	38.40
1501-2000	3037	12.31	12512	50.71
2001-2500	1865	7.56	14377	58.27
501-1000	6438	26.09	20815	84.37
>2501	3857	15.63	24672	100.00

Table 6

Most of the data falls into the "501-1000" category, which accounts for about 26.09% of the observations. Relatively smaller proportions are found in the "1-500" and ">2501" categories, contributing 20.39% and 15.63%, respectively.

4. **Loan**

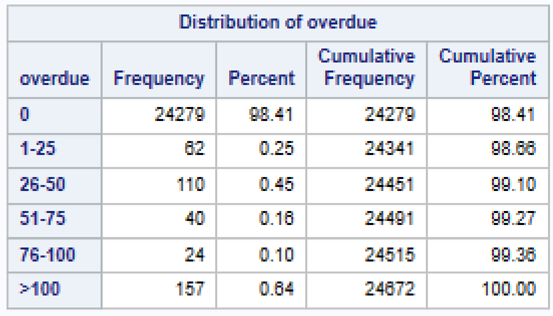


Distribution of loan				
loan	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	7072	28.66	7072	28.66
1-250	1222	4.95	8294	33.62
251-500	1043	4.23	9337	37.84
501-750	1549	6.28	10886	44.12
751-1000	2020	8.19	12906	52.31
>1000	11766	47.69	24672	100.00

Table 7

Most of the data falls into the ">1000" & “0” category, representing a significant portion of approximately 47.69% & 28.66% respectively of the observations. This distribution suggests that a significant portion of the dataset has no loans, and as the loan amount increases, the number of observations decreases.

5. **Overdue**

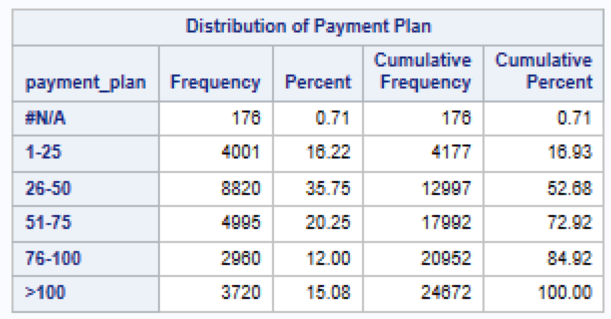


Distribution of overdue				
overdue	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	24279	98.41	24279	98.41
1-25	62	0.25	24341	98.66
26-50	110	0.45	24451	99.10
51-75	40	0.16	24491	99.27
76-100	24	0.10	24515	99.36
>100	157	0.64	24672	100.00

Table 8

Overall, this distribution reveals a highly skewed pattern, with the majority of accounts having no overdue amounts.

6. **Payment Plan**



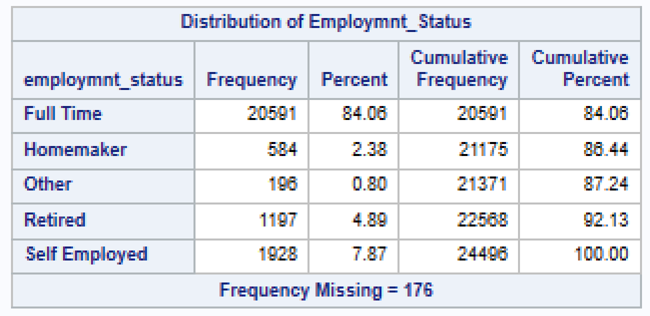
Distribution of Payment Plan

payment_plan	Frequency	Percent	Cumulative Frequency	Cumulative Percent
#N/A	176	0.71	176	0.71
1-25	4001	16.22	4177	16.93
26-50	8820	35.75	12997	52.68
51-75	4995	20.25	17992	72.92
76-100	2960	12.00	20952	84.92
>100	3720	15.08	24672	100.00

Table 9

The most prominent category is "26-50," with approximately 35.75% of accounts falling into this range, indicating that a significant portion of accounts have payment plans within this bracket.

7. Employment Status



Distribution of Employmnt_Status

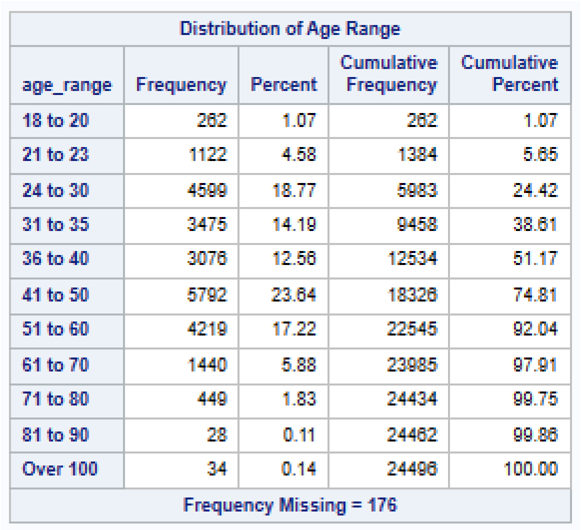
employmnt_status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Full Time	20591	84.06	20591	84.06
Homemaker	584	2.38	21175	86.44
Other	196	0.80	21371	87.24
Retired	1197	4.89	22568	92.13
Self Employed	1928	7.87	24496	100.00

Frequency Missing = 176

Table 10

"Full Time" employment status is the most prevalent, accounting for approximately 84.06% of the dataset. This indicates that a significant majority of individuals are employed full-time.

8. Age Range



Distribution of Age Range

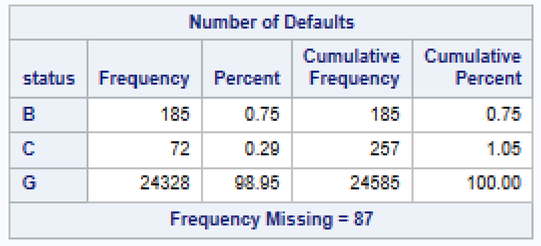
age_range	Frequency	Percent	Cumulative Frequency	Cumulative Percent
18 to 20	262	1.07	262	1.07
21 to 23	1122	4.58	1384	5.65
24 to 30	4599	18.77	5983	24.42
31 to 35	3475	14.19	9458	38.61
36 to 40	3076	12.56	12534	51.17
41 to 50	5792	23.64	18326	74.81
51 to 60	4219	17.22	22545	92.04
61 to 70	1440	5.88	23985	97.91
71 to 80	449	1.83	24434	99.75
81 to 90	28	0.11	24462	99.86
Over 100	34	0.14	24496	100.00

Frequency Missing = 176

Table 11

Age range "41 to 50" is the most prevalent, constituting approximately 23.64% of the dataset. This suggests a substantial representation of individuals in their forties.

9. Target Variable

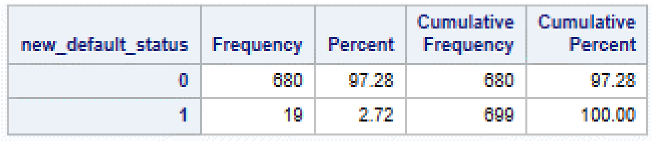


Number of Defaults				
status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
B	185	0.75	185	0.75
C	72	0.29	257	1.05
G	24328	98.95	24585	100.00
Frequency Missing = 87				

Table 12

This distribution suggests that most accounts in the dataset are in good category. And there are 87 missing values present in the Status variable which needs to be excluded.

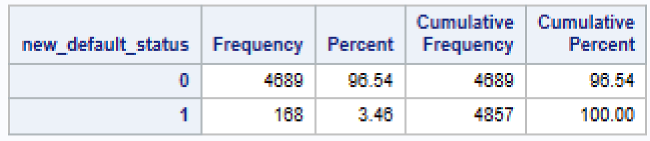
10. Target variable distribution for behavioural scorecard model



new_default_status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	680	97.28	680	97.28
1	19	2.72	699	100.00

Table 13

11. Target variable distribution for predicting Important feature of Delinquency



new_default_status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	4689	96.54	4689	96.54
1	168	3.46	4857	100.00

Table 14

5.1.2. Bivariate Distribution

1. Snapshot Distribution with respect to Target Variable

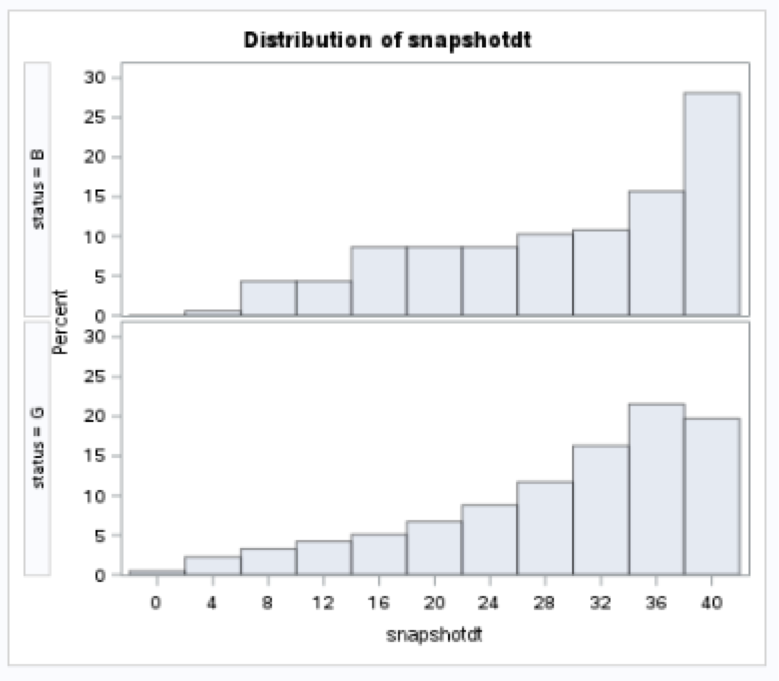


Figure 4

Figure 2 shows that the majority of persons who defaulted did so between the months of 36 and 40.

2. Overdue Distribution with respect to Target Variable

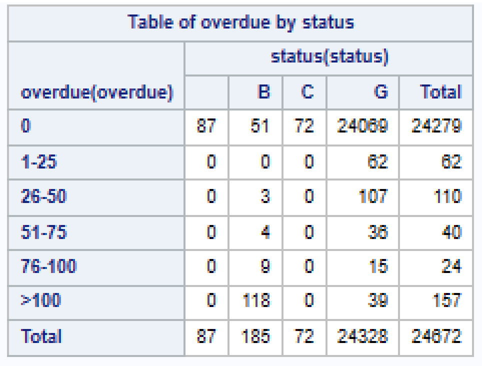


Table of overdue by status					
	status(status)				
overdue(overdue)		B	C	G	Total
0	87	51	72	24069	24279
1-25	0	0	0	62	62
26-50	0	3	0	107	110
51-75	0	4	0	36	40
76-100	0	9	0	15	24
>100	0	118	0	39	157
Total	87	185	72	24328	24672

Table 15

Overall, this trend supports the idea that accounts with a "Good" status are less likely to have late balances, but accounts with a "Closed" or "Bad" status are more likely to have overdue amounts, particularly in the higher overdue categories.

3. Distribution of Delinquency stage with respect to Target Variable

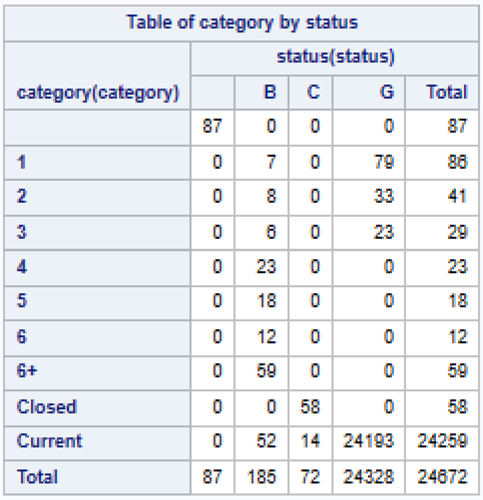


Table of category by status					
	status(status)				
category(category)		B	C	G	Total
	87	0	0	0	87
1	0	7	0	79	86
2	0	8	0	33	41
3	0	6	0	23	29
4	0	23	0	0	23
5	0	18	0	0	18
6	0	12	0	0	12
6+	0	59	0	0	59
Closed	0	0	58	0	58
Current	0	52	14	24193	24259
Total	87	185	72	24328	24672

Table 16

Overall, the trend implies that as the delinquency stage grows, so does the risk of an account entering into default. Accounts at the "4" delinquency stage and higher do not have any "Good" status accounts. This implies that accounts in these phases of delinquency are more likely to default.

6. DATA PRE-PROCESSING

6.1. Label Encoding Technique

The label encoding approach is a critical step in preparing our data for analysis. This approach involves transforming categorical data into numerical format, with each category assigned a unique numerical label. The significance and implications of label encoding are summarised below:

Importance of Label Encoding:

- **Compatibility with Algorithms**: Many machine learning techniques require numerical input data. Label encoding converts categorical data into a format that is compatible with a wide range of algorithms, allowing us to successfully use statistical and machine learning methods (Abdou & Pointon, 2011).
- **Preservation of Ordinality**: Label encoding is especially beneficial when the categorical variable has an intrinsic ordinal connection between its categories. By assigning numerical labels, we retain the ordinality of the data, ensuring that algorithms can capture the relative order of categories (Abdou & Pointon, 2011).
- **Balancing Data for SMOTE**: An essential reason for implementing label encoding in this context is its role in preparing the data for the Synthetic Minority Over-sampling Technique (SMOTE) (Brownlee, 2021).

By applying label encoding to selected categorical variables such as 'capbal', 'loan', 'overdue', 'payment_plan', 'employmnt_status', 'age_range', and 'gross_income', we convert these variables into numerical representations. This transformation not only enhances the compatibility of the data with various analytical techniques but also plays a crucial role in achieving a balanced dataset, a vital aspect of our data preparation process.

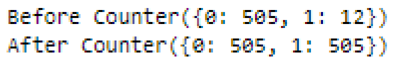
6.2. SMOTE Technique

The Synthetic Minority Over-Sampling Technique (SMOTE) is to address the issue of class imbalance within our dataset (Brownlee, 2021).

Minority Synthetic Over-sampling Technique (SMOTE) is a resampling approach built primarily to address class imbalance in a dataset. In the case of behavioural scorecard, class imbalance arises when one class ('default') is greatly underrepresented compared to another ('non-default'), but in the case of delinquency prediction, class (delinquent) is significantly underrepresented compared to another ('non-delinquent'). Such imbalance can have a considerable influence on predictive model performance, resulting in biased findings (Brownlee, 2021).

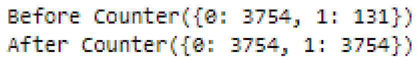
Overfitting: SMOTE has the ability to add noise into the dataset by producing synthetic samples that are highly close to existing ones. Overfitting occurs when the model fits too closely to the training data, limiting generalisation performance on unknown data (Brownlee, 2021).

Behavioural Scorecard Target Variable Distribution:



```
Before Counter({0: 505, 1: 12})
After Counter({0: 505, 1: 505})
```

Delinquency Prediction Target Variable Distribution:



```
Before Counter({0: 3754, 1: 131})
After Counter({0: 3754, 1: 3754})
```

6.3. Weight of Evidence (WOE) & Information Value (IV)

Weight of Evidence (WoE) and Information Value (IV) are essential techniques in the development of a Behavioural Scorecard (Siddiqi N. 2006). In the context of the Behavioural Scorecard, WoE and IV serve specific purposes:

Weight of Evidence (WoE):

- Weight of Evidence (WoE) is a statistical technique used in credit risk modelling to assess the strength of the relationship between a predictor variable and the likelihood of a binary outcome, such as defaulting on a loan (Siddiqi N. 2006).
- It measures how well a variable separates the observations into distinct categories (in this context, non-default and default clients) (Siddiqi N. 2006).
- WoE is calculated by taking the natural logarithm (ln) of the ratio of the percentage of non-default clients to the percentage of default clients within each category or bin (Siddiqi N. 2006).
- The WoE values provide insights into the predictive power of a variable. Positive WoE values indicate that the category is associated with a higher likelihood of default, while negative WoE values suggest a lower likelihood of default (Siddiqi N. 2006).
- WoE values are used to convert categorical and continuous predictor variables into a numerical form that is more informative. These modified variables can then be fed into prediction models as inputs (Siddiqi N. 2006).

Information Value (IV):

- Information Value (IV) is a statistical measure used to assess the predictive power of a variable or feature in a binary classification problem.
- IV is derived by calculating the difference in WoE values between each category or bin for a particular variable. It takes into account both the predictive power of the variable and the separation between the target classes (Siddiqi N. 2006).
- IV helps prioritize variables based on their ability to distinguish between the positive (default) and negative (non-default) cases.
- The IV values provide a clear understanding of which variables are the most influential in predicting the outcome. It helps in feature selection and model building.

The interpretation of IV values is as follows:

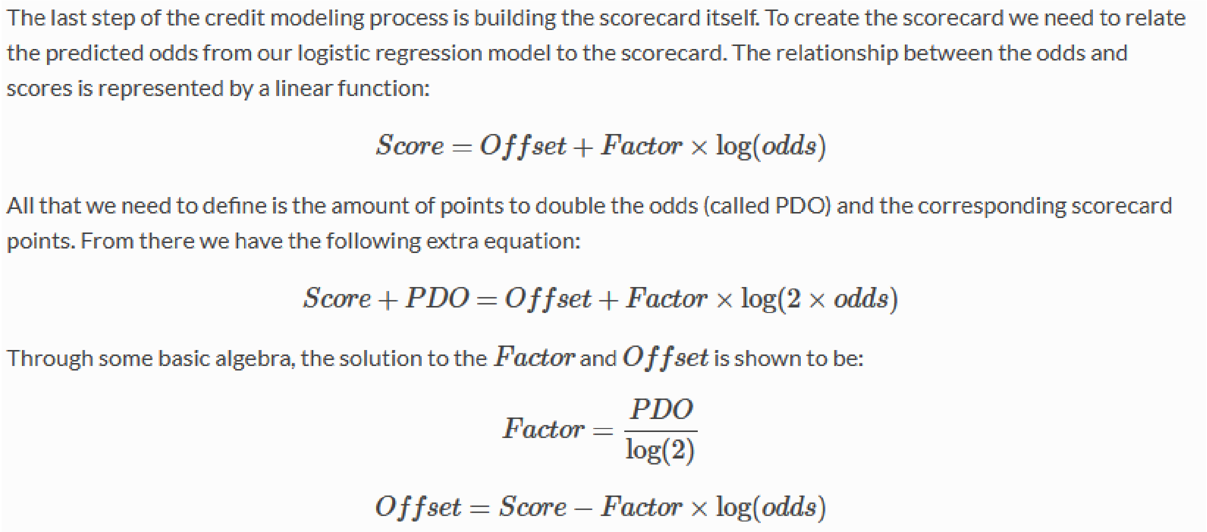
- IV < 0.02: Very weak relationship
- 0.02 ≤ IV < 0.1: Weak relationship
- 0.1 ≤ IV < 0.3: Medium strength relationship
- IV ≥ 0.3: Strong relationship

IV values should be summed across all bins or categories for a variable. Typically, a variable with a total IV between 0.02 and 0.5 is considered valuable for modelling (Siddiqi N. 2006).

Steps in WoE and IV Calculation:

- WoE Calculation: For each bin or category within a variable, calculate the percentage of good (non-default) clients and the percentage of bad (default) clients. Then, calculate the WoE for each bin using the formula: WoE = ln(Percentage of Good / Percentage of Bad) (Siddiqi N. 2006).
- WoE Visualization: Develop a line graph to show the WoE values for each bin. Inspect to see if the WoE numbers have a logical trend, either increasing or decreasing. It should be noted that if there are missing values, the WoE values might change (Siddiqi N. 2006).
- IV Calculation: Calculate the Information Value (IV) for each bin by subtracting the percentage of bad clients from the percentage of good clients and multiplying the result by the WoE of the bin: IV = (Percentage of Good - Percentage of Bad) * WoE (Siddiqi N. 2006).
- Interpretation of IV: Determine the strength of the association between the variable and the goal by evaluating the IV values. Each variable's IV value indicates its predictive strength, with larger values suggesting more predictive skill (Siddiqi N. 2006).
- IV Thresholds: Apply thresholds to IV values to assess the relationship's strength. For example, IV < 0.02 may indicate a very weak relationship, while IV ≥ 0.3 suggests a strong relationship (Siddiqi N. 2006).
- Summation of IV: Ensure that the total IV for all bins of a variable falls within the range of 0.02 to 0.5. Variables with IV values outside this range may be less informative (Siddiqi N. 2006).

Scaling The Scorecard:



The last step of the credit modeling process is building the scorecard itself. To create the scorecard we need to relate the predicted odds from our logistic regression model to the scorecard. The relationship between the odds and scores is represented by a linear function:

$$Score = Offset + Factor \times \log(odds)$$

All that we need to define is the amount of points to double the odds (called PDO) and the corresponding scorecard points. From there we have the following extra equation:

$$Score + PDO = Offset + Factor \times \log(2 \times odds)$$

Through some basic algebra, the solution to the $Factor$ and $Offset$ is shown to be:

$$Factor = \frac{PDO}{\log(2)}$$

$$Offset = Score - Factor \times \log(odds)$$

Equation 1 (*Credit Score Modelling*, 2021)

The value for Offset and factor we got are 87.1228762 & 28.85390082 respectively.

Note: For the Figures, please refer appendix B

6.4. Normalisation Technique

Normalisation, also known as standard scaling, is a data preparation process used to rescale numerical features to have a mean of 0 and a standard deviation of 1. To guarantee that machine learning models perform properly, all numerical variables must be on the same scale. Standard scaling is significant because it ensures that all numerical properties are on the same scale, preventing one variable from dominating the others during model training.

The Normalisation Technique is quite useful for predicting feature significance for delinquency. Scaling is used on numerical variables such as 'Term,' 'app_score,' and 'Beh_score.' By scaling these variables, we ensure that all factors have equal effect when calculating feature relevance. Machine learning algorithms that identify feature significance, such as decision trees or random forests, benefit from standardised input since it allows them to make meaningful comparisons across features.

6.5. Splitting Data in Train & Test

The "Split and Train" method is widely used in machine learning and predictive modelling. It entails dividing a dataset into two or more subgroups and then utilising these subsets to train and test machine learning models. This approach is essential for evaluating model performance, such as logistic regression, random forest, and decision tree models, as well as predicting the feature importance of delinquency (Ali et al., 2021).

Here's how the "Split and Train" technique works and its importance for different models:

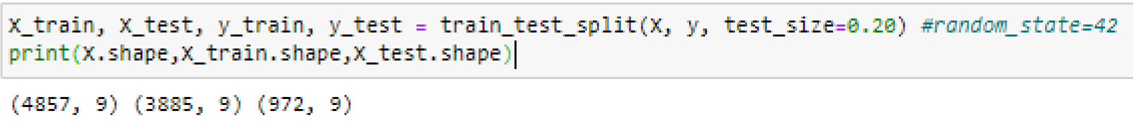
1. **Data Splitting:** The original dataset is divided into two or more subsets: typically, a training set and a testing set (Ali et al., 2021).

2. **Training Set:** This subset, often the larger portion of the data (e.g., 70-80%), is used to train the machine learning model. The model learns patterns and relationships in the data (Ali et al., 2021).
3. **Testing Set:** The remaining portion (e.g., 20-30%) is used to evaluate the model's performance. It serves as an unseen dataset to assess how well the model generalizes to new, unseen data (Ali et al., 2021).

Importance of Split and Train" technique for Predicting the feature importance of Delinquency:

1. **Model Evaluation:** The split and train method are critical for assessing the performance of machine learning models. It allows us to evaluate the capabilities of models like as logistic regression, random forest, and decision trees to generate accurate predictions on new data (Ali et al., 2021).
2. **Feature Importance:** When projecting feature significance, it is critical to split the data. In a random forest, for example, one could analyse each feature's contribution to model predictions through assessing how much they influence the model's performance on the testing set.

Delinquency Prediction Train & Test Split:



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20) #random_state=42
print(X.shape,X_train.shape,X_test.shape)
```

```
(4857, 9) (3885, 9) (972, 9)
```

6.5.1. Correlation Coefficient Strengths

Correlation between the different variables

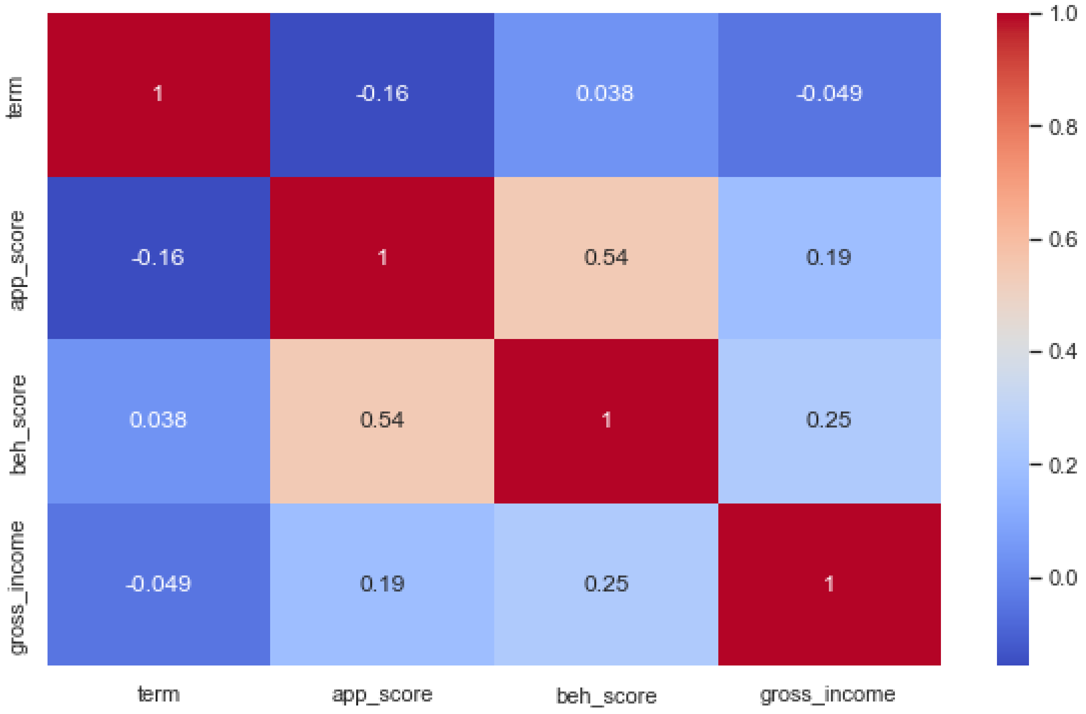
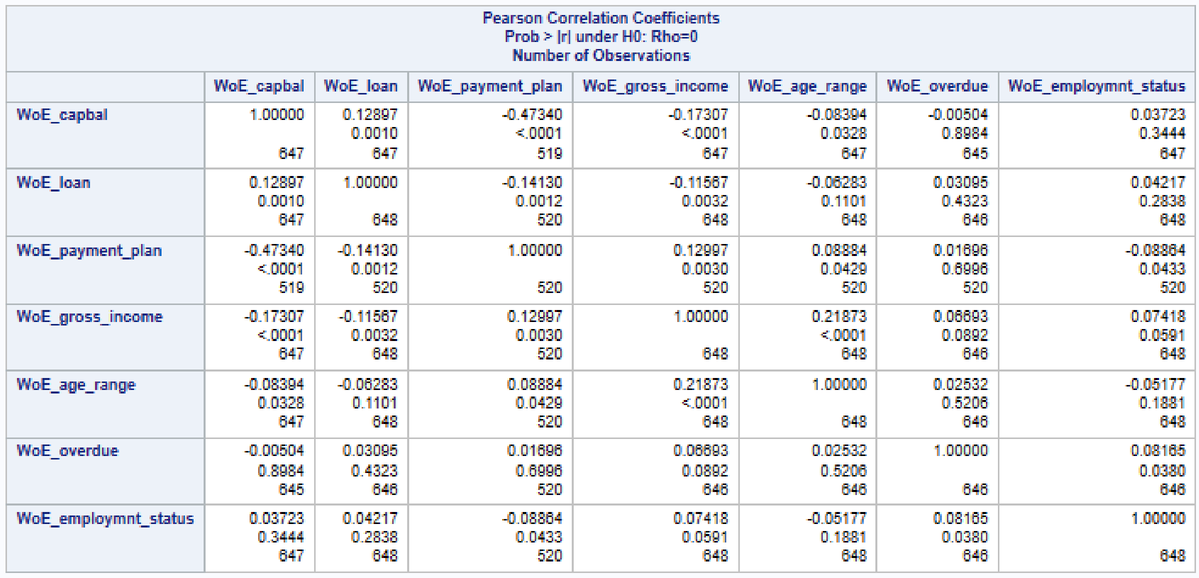


Figure 5

From the Figure 5, It is clearly visible that there are no variables highly correlated with each other. The threshold for highly correlated variables is considered as 0.7. Hence, we won't face any problem of multicollinearity.

Correlation between the different variables of WOE



Pearson Correlation Coefficients
Prob > |r| under H0: Rho=0
Number of Observations

	WoE_capbal	WoE_loan	WoE_payment_plan	WoE_gross_income	WoE_age_range	WoE_overdue	WoE_employmnt_status
WoE_capbal	1.00000 647	0.12897 0.0010 647	-0.47340 <.0001 519	-0.17307 <.0001 647	-0.08394 0.0328 647	-0.00504 0.8984 645	0.03723 0.3444 647
WoE_loan	0.12897 0.0010 647	1.00000 648	-0.14130 0.0012 520	-0.11567 0.0032 648	-0.06283 0.1101 648	0.03095 0.4323 646	0.04217 0.2838 648
WoE_payment_plan	-0.47340 <.0001 519	-0.14130 0.0012 520	1.00000 520	0.12997 0.0030 520	0.08884 0.0429 520	0.01696 0.6996 520	-0.08864 0.0433 520
WoE_gross_income	-0.17307 <.0001 647	-0.11567 0.0032 648	0.12997 0.0030 520	1.00000 648	0.21873 <.0001 648	0.06693 0.0892 646	0.07418 0.0591 648
WoE_age_range	-0.08394 0.0328 647	-0.06283 0.1101 648	0.08884 0.0429 520	0.21873 <.0001 648	1.00000 648	0.02532 0.5206 646	-0.05177 0.1881 648
WoE_overdue	-0.00504 0.8984 645	0.03095 0.4323 646	0.01696 0.6996 520	0.06693 0.0892 646	0.02532 0.5206 646	1.00000 646	0.08165 0.0380 646
WoE_employmnt_status	0.03723 0.3444 647	0.04217 0.2838 648	-0.08864 0.0433 520	0.07418 0.0591 648	-0.05177 0.1881 648	0.08165 0.0380 646	1.00000 648

Table 17

From table 17, It can be observed that WOE variable for Overdue has a very strong correlations with other variables, hence we are going to create one model excluding Overdue.

7. MODELLING & EVALUATION

7.1. BEHAVIOURAL SCORECARD

7.1.1. Logistic Regression Model for Behavioural Scorecard

Logistic Regression is a fundamental statistical approach that is used in predictive modelling and classification problems. It is ideal for developing a Behavioural Scorecard in credit risk management. The use of Logistic Regression in our modelling approach was driven by several crucial factors:

1. **Interpretability:** Logistic Regression is highly comprehensible, making it easy for people to understand the association between input characteristics and default probability (Abdou & Pointon, 2011).
2. Simplicity: Logistic Regression is a relatively simple yet powerful algorithm, which makes it suitable for modelling credit risk behaviour effectively (Abdou & Pointon, 2011).

3. Scalability: It can handle both small and large datasets efficiently, ensuring robust performance regardless of data size (Abdou & Pointon, 2011).

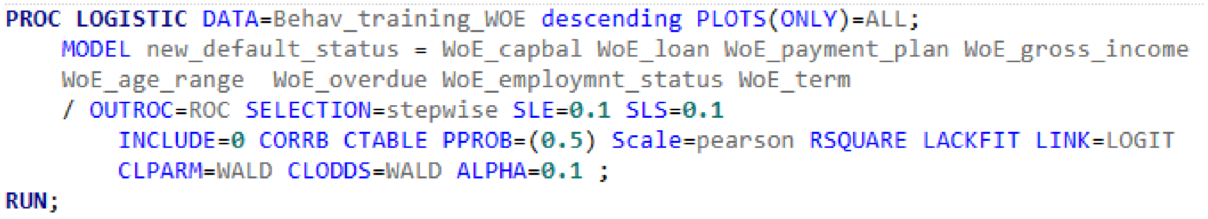
The Logistic Regression model discussed here is especially used in the development of the Behavioural Scorecard, is a critical tool for assessing customer creditworthiness. Based on their behavioural characteristics, it is used to estimate the likelihood of customers moving into a default position (Abdou & Pointon, 2011).

- Generative Model 1:

The model's variable selection procedure was led by a stepwise method. This strategy includes adding and eliminating predictor variables repeatedly based on their statistical significance. Variables that improve the predictive power of the model are included, while those that do not fulfil the chosen criteria are eliminated.

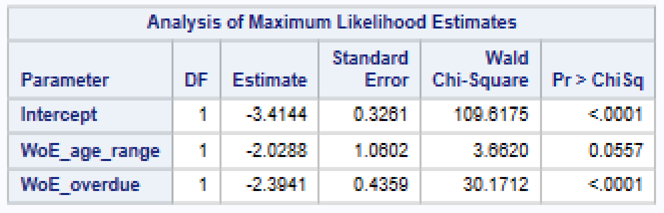
In logistic regression, the significance level, written as "ALPHA=0.1," is an important parameter. It establishes the cutoff for statistical significance when evaluating the contribution of each predictor variable. The logit link function is useful for binary classification applications because it converts the linear combination of predictor variables into probabilities. The model uses a probability threshold of 0.5 for categorization. When the anticipated likelihood of an account defaulting reaches 0.5, it is categorised as a default (1), otherwise as a non-default (0). And the other functions are used to print model validation metrics (Abdou & Pointon, 2011).

SAS Code:



```
PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
    MODEL new_default_status = WoE_capbal WoE_loan WoE_payment_plan WoE_gross_income
    WoE_age_range  WoE_overdue WoE_employmnt_status WoE_term
    / OUTROC=ROC SELECTION=stepwise SLE=0.1 SLS=0.1
        INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
        CLPARM=WALD CLODDS=WALD ALPHA=0.1 ;
RUN;
```

Output:



Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.4144	0.3261	109.6175	<.0001
WoE_age_range	1	-2.0288	1.0602	3.6620	0.0557
WoE_overdue	1	-2.3941	0.4359	30.1712	<.0001

Table 18

The logistic regression result is shown in the Analysis of Maximum Likelihood Estimates table above. The standard error for each parameter represents the uncertainty in the estimate. The Wald chi-square statistic and p-value are used to determine the importance of each parameter. As our threshold alpha value is 10%, we would accept both variables in creating the scorecard. However, if we keep the Threshold alpha value as 5% (0.05). Only the Overdue variable could be accepted, while the Age_range variable has to be rejected (Finalyse, n.d.).

Scorecard Table

Characteristic	Attribute	Scorepoints
overdue	O	72
overdue	1 = < Overdue <100	36

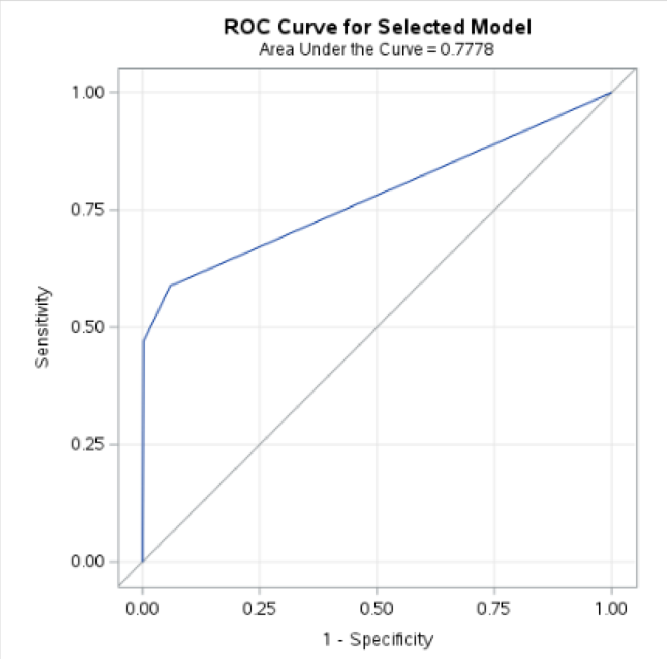
overdue	Neutral	62
Age Range	18 =< Age < 23	54
Age Range	23 <= Age	99
Age Range	Neutral	92

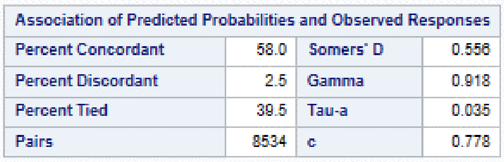
Applicant Profile

Client A		
overdue	0	72
Age Range	28	99
Total		171
Decision	Safe	

Client B		
overdue	100	36
Age Range	34	99
Total		135
Decision	Risky	

Mode Validation





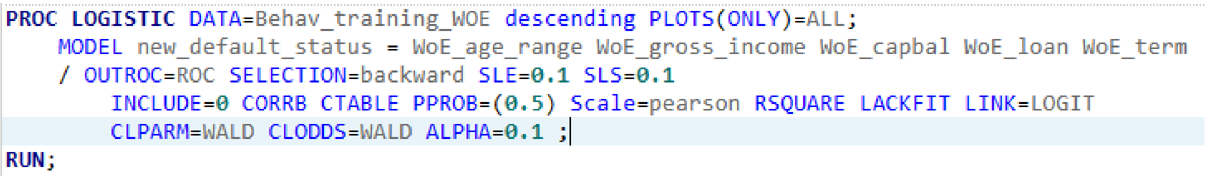
Association of Predicted Probabilities and Observed Responses			
Percent Concordant	58.0	Somers' D	0.556
Percent Discordant	2.5	Gamma	0.918
Percent Tied	39.5	Tau-a	0.035
Pairs	8534	c	0.778

As indicated by the high values of Somers' D (0.556) and Gamma (0.918), the model reveals a good correlation between projected probabilities and observed reactions. This shows that the model's projected probability matches the actual results, which is an important feature for a credit risk assessment model. Furthermore, the AUC, a commonly used indicator of discriminating power, reaches a significant value of 0.7778. This implies that the model successfully differentiates between accounts at risk of default and those that are not, exhibiting strong discrimination skills (Finalyse, n.d.).

- Generative Model 2:

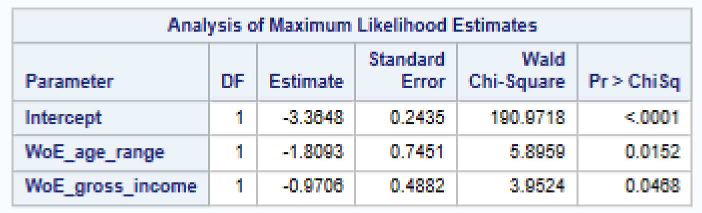
Backward selection is a variable selection method used in logistic regression modelling. Backward selection begins with a model that contains all predictor variables and repeatedly removes the least significant ones, as opposed to forward selection, which starts with an empty model and adds variables (Abdou & Pointon, 2011).

SAS Code:



```
PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
    MODEL new_default_status = WoE_age_range WoE_gross_income WoE_capbal WoE_loan WoE_term
    / OUTROC=ROC SELECTION=backward SLE=0.1 SLS=0.1
        INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
        CLPARM=WALD CLODDS=WALD ALPHA=0.1 ;
RUN;
```

Output:



Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.3648	0.2435	190.9718	<.0001
WoE_age_range	1	-1.8093	0.7451	5.8959	0.0152
WoE_gross_income	1	-0.9706	0.4882	3.9524	0.0468

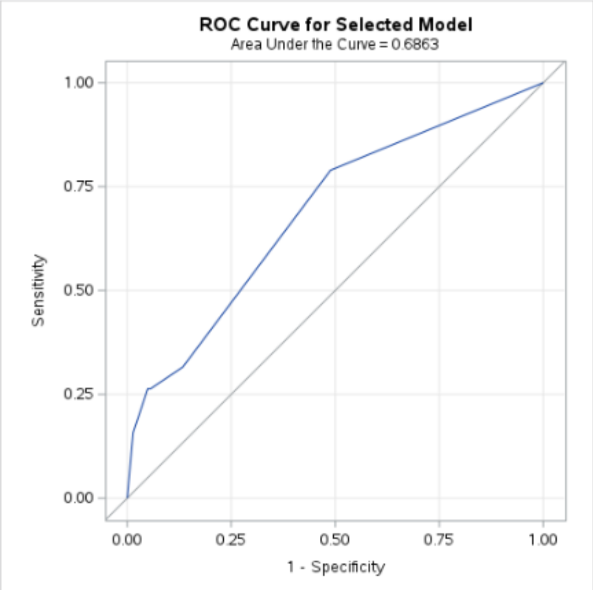
Table 19

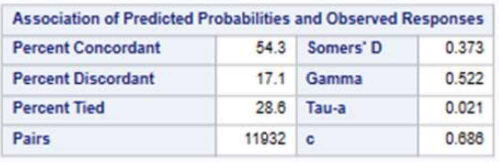
The logistic regression result is shown in the Analysis of Maximum Likelihood Estimates table above. The standard error for each parameter represents the uncertainty in the estimate. The Wald chi-square statistic and p-value are used to determine the importance of each parameter. As our threshold alpha value is 10%, we would accept both variables in creating the scorecard.

Scorecard Table

Characteristic	Attribute	Scorepoints
gross_income	gross_income <= 14000	80
gross_income	14000 < gross_income <= 27500	85
gross_income	27500 < gross_income	118
gross_income	Neutral	94
Age Range	18 =< Age < 23	57
Age Range	23 <= Age	98
Age Range	Neutral	91

Mode Validation





Association of Predicted Probabilities and Observed Responses			
Percent Concordant	54.3	Somers' D	0.373
Percent Discordant	17.1	Gamma	0.522
Percent Tied	28.6	Tau-a	0.021
Pairs	11932	c	0.686

The model exhibits a moderate association between predicted probabilities and observed responses, as indicated by Somers' D (0.373) and Gamma (0.522). The AUC, a vital measure of discriminatory power, registers a commendable value of 0.686. This implies that the model possesses a reasonable ability to differentiate between accounts at risk of default and those not at risk (Finalyse, n.d.).

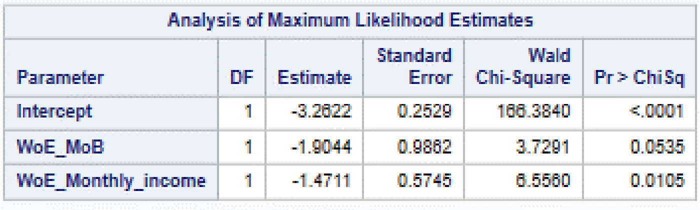
- Generative Model 3:

SAS Code:

```
PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
    MODEL new_default_status = WoE_capbal WoE_loan WoE_payment_plan
    WoE_age_range WoE_employmnt_status WoE_term WoE_MoB WoE_Monthly_income
    / OUTROC=ROC SELECTION=backward SLE=0.1 SLS=0.1
        INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
        CLPARM=WALD CLODDS=WALD ALPHA=0.1 ;
RUN;
```

Figure 6

Output:



Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.2622	0.2529	166.3840	<.0001
WoE_MoB	1	-1.9044	0.9862	3.7291	0.0535
WoE_Monthly_income	1	-1.4711	0.5745	6.5560	0.0105

Table 20

Scorecard Table

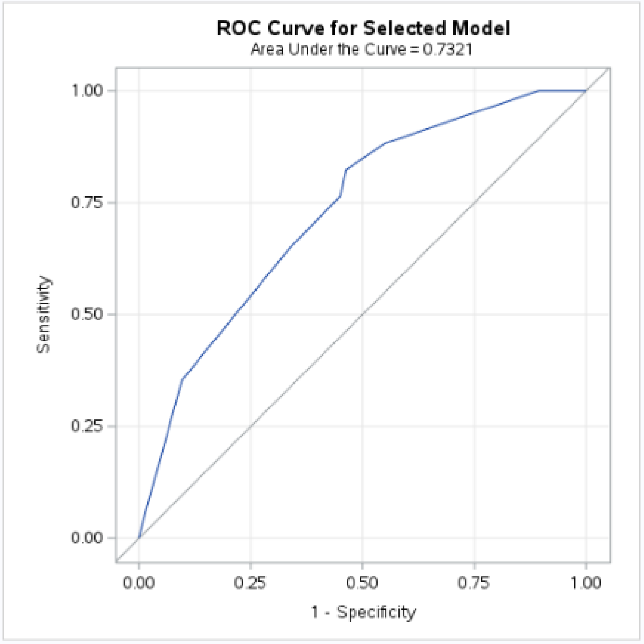
Characteristic	Attribute	Scorepoints
MoB	MoB <10	74
MoB	10 =< MoB <29	94
MoB	MoB >=29	124
MoB	Neutral	92

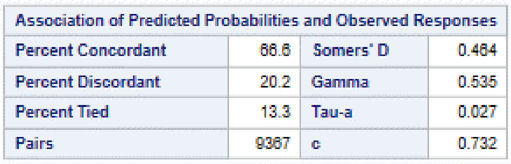
Monthly_income	Monthly_income < 1824.25006	66
Monthly_income	1824.25006 =< Monthly_income < 2758.234721	78
Monthly_income	Monthly_income >= 2758.234721	130
Monthly_income	Neutral	100

Client A		
MoB	9	74
Monthly_income	2000	78
Total		152
Decision	Risky	

Client B		
MoB	12	94
Monthly_income	3000	130
Total		224
Decision	Safe	

Mode Validation





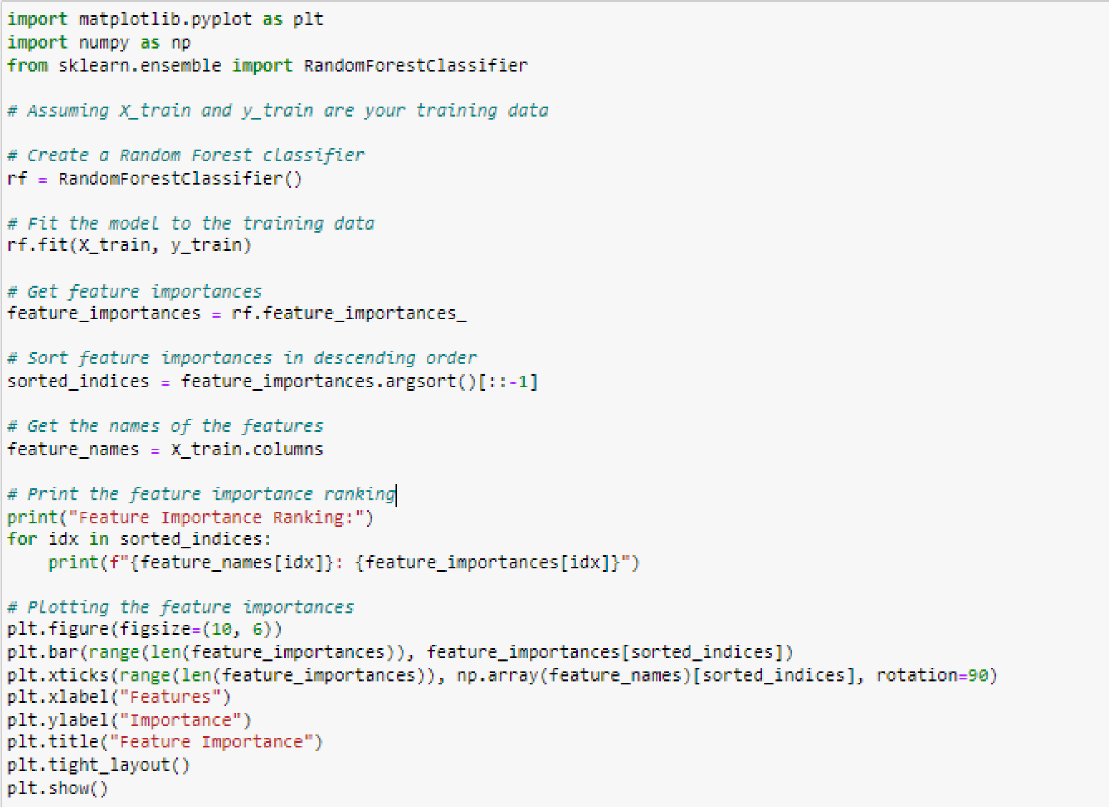
Association of Predicted Probabilities and Observed Responses			
Percent Concordant	66.6	Somers' D	0.464
Percent Discordant	20.2	Gamma	0.535
Percent Tied	13.3	Tau-a	0.027
Pairs	9367	c	0.732

The model demonstrates a moderate association between predicted probabilities and observed responses, as evidenced by Somers' D (0.464) and Gamma (0.535). The AUC, a commonly used indicator of discriminating power, reaches a significant value of 0.7321. This implies that the model successfully differentiates between accounts at risk of default and those that are not, exhibiting strong discrimination skills (Finalyse, n.d.).

7.1.2. Feature Importance Ranking Using Random Forest

The evaluation of feature significance is a vital step in developing a strong Behavioural Scorecard. We use the Random Forest method in this part to estimate the importance of each characteristic in predicting default. This procedure helps us prioritise essential factors and improves the accuracy of our scorecard (Abdou & Pointon, 2011). The use of Random Forest for feature significance is inspired by the following fundamental considerations:

1. Random Forest is a strategy for ensemble learning that incorporates numerous decision trees. This ensemble technique reduces overfitting and data noise.
2. It gives detailed insights on variable significance, allowing us to assess the influence of each feature on our default target variable.
3. Unlike linear models, Random Forest can capture complicated, non-linear interactions between predictors and the default variable, which is critical for correct evaluation.



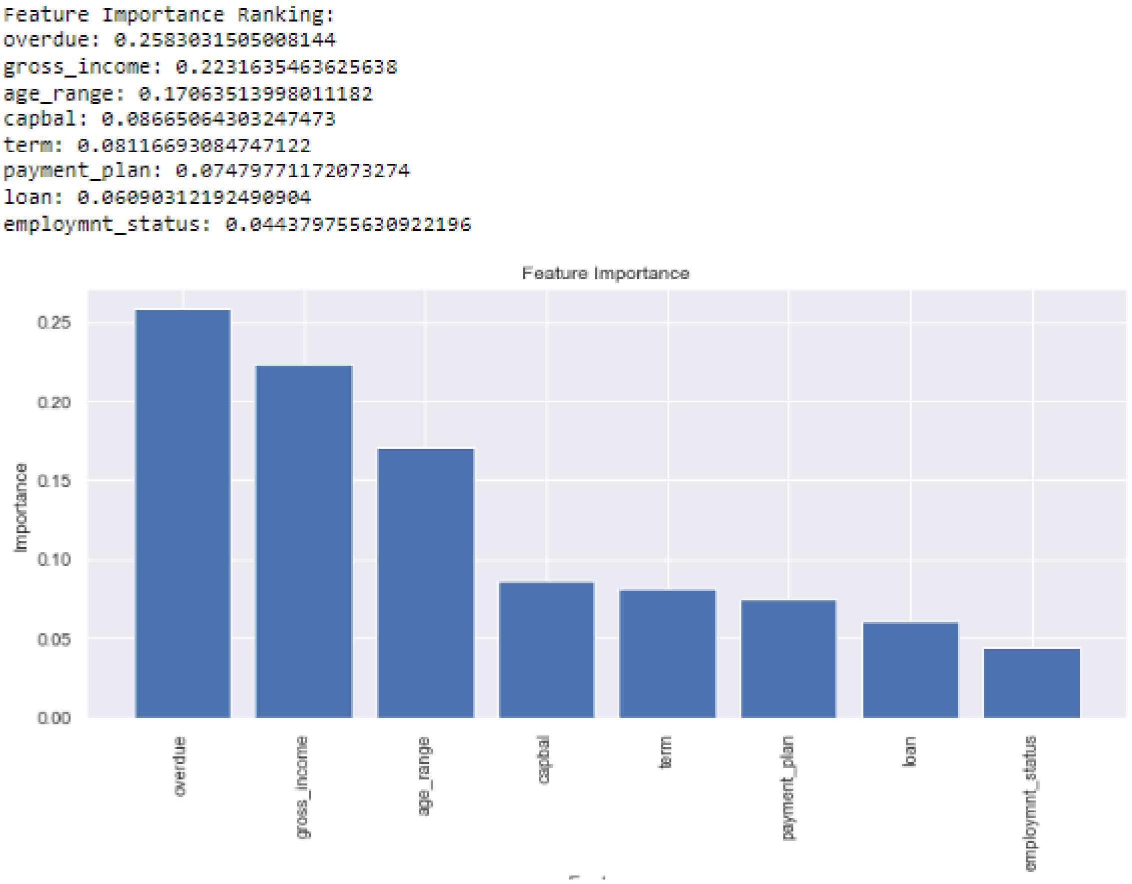
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.ensemble import RandomForestClassifier

# Assuming X_train and y_train are your training data

# Create a Random Forest classifier
rf = RandomForestClassifier()

# Fit the model to the training data
rf.fit(X_train, y_train)

# Get feature importances
feature_importances = rf.feature_importances_

# Sort feature importances in descending order
sorted_indices = feature_importances.argsort()[::-1]

# Get the names of the features
feature_names = X_train.columns

# Print the feature importance ranking
print("Feature Importance Ranking:")
for idx in sorted_indices:
    print(f"{feature_names[idx]}: {feature_importances[idx]}")

# Plotting the feature importances
plt.figure(figsize=(10, 6))
plt.bar(range(len(feature_importances)), feature_importances[sorted_indices])
plt.xticks(range(len(feature_importances)), np.array(feature_names)[sorted_indices], rotation=90)
plt.xlabel("Features")
plt.ylabel("Importance")
plt.title("Feature Importance")
plt.tight_layout()
plt.show()
```

Figure 7

We want to evaluate the efficiency of our Logistic Regression model, which is essential in the development of the Behavioural Scorecard, during this part of our investigation. We use a Random Forest model to create a Feature Importance Ranking.

Understanding the importance of numerous elements impacting customers behaviour is critical. While logistic regression is a popular approach for creating a behavioural scorecard, we acknowledge that more complex algorithms like Random Forest often outperform it in terms of predictive accuracy. As a result, to improve the reliability and robustness of our logistic regression model, we used a sophisticated validation tool developed from a Random Forest model: Feature Importance Ranking (Finalyse, n.d.).

We can determine the relative significance of each variable in predicting account defaulting using the Random Forest model's: Feature Importance Ranking. By applying this ranking to our logistic regression model, which is the foundation of our behavioural scorecard creation, we gain valuable insights. This procedure not only shows the logistic regression model, but it also assists us in identifying and prioritising the most significant variables causing accounts to default. As we go deeper into the analysis of our behavioural scorecard, this validation process guarantees that our risk assessment stays accurate, up to date, and in line with industry best practises (Finalyse, n.d.).



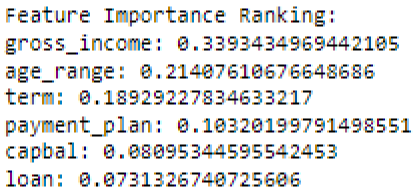
```
Feature Importance Ranking:
overdue: 0.2583031505008144
gross_income: 0.2231635463625638
age_range: 0.17063513998011182
capbal: 0.08665064303247473
term: 0.08116693084747122
payment_plan: 0.07479771172073274
loan: 0.06090312192490904
employmnt_status: 0.044379755630922196
```

Figure 8

As mentioned in EDA the overdue has highly skewed pattern which may be the reason overdue is shown as most Important feature, that is followed by gross_income as the second most important feature closely followed by Age_range at a third spot. In summary, the trend suggests that variables related to income, age, loan amount, loan duration, and repayment plans are key drivers in predicting outcomes.

Each feature importance score indicates the relative significance of a feature in predicting default. Larger values suggest that this feature contributes more to the overall predictive power of the Random Forest model. Lower value might not be as influential as some other features, but it still contributes to the model's performance.

Output 2 (no Overdue or Employment Status)



```
Feature Importance Ranking:
gross_income: 0.3393434969442105
age_range: 0.21407610676648686
term: 0.18929227834633217
payment_plan: 0.10320199791498551
capbal: 0.08095344595542453
loan: 0.0731326740725606
```

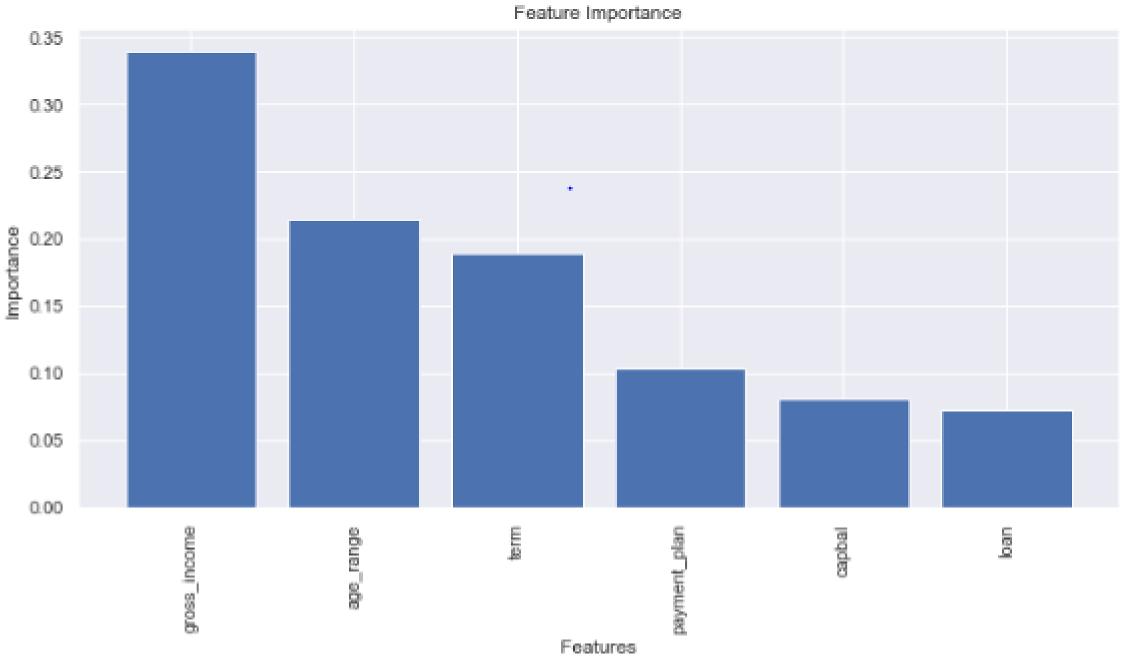
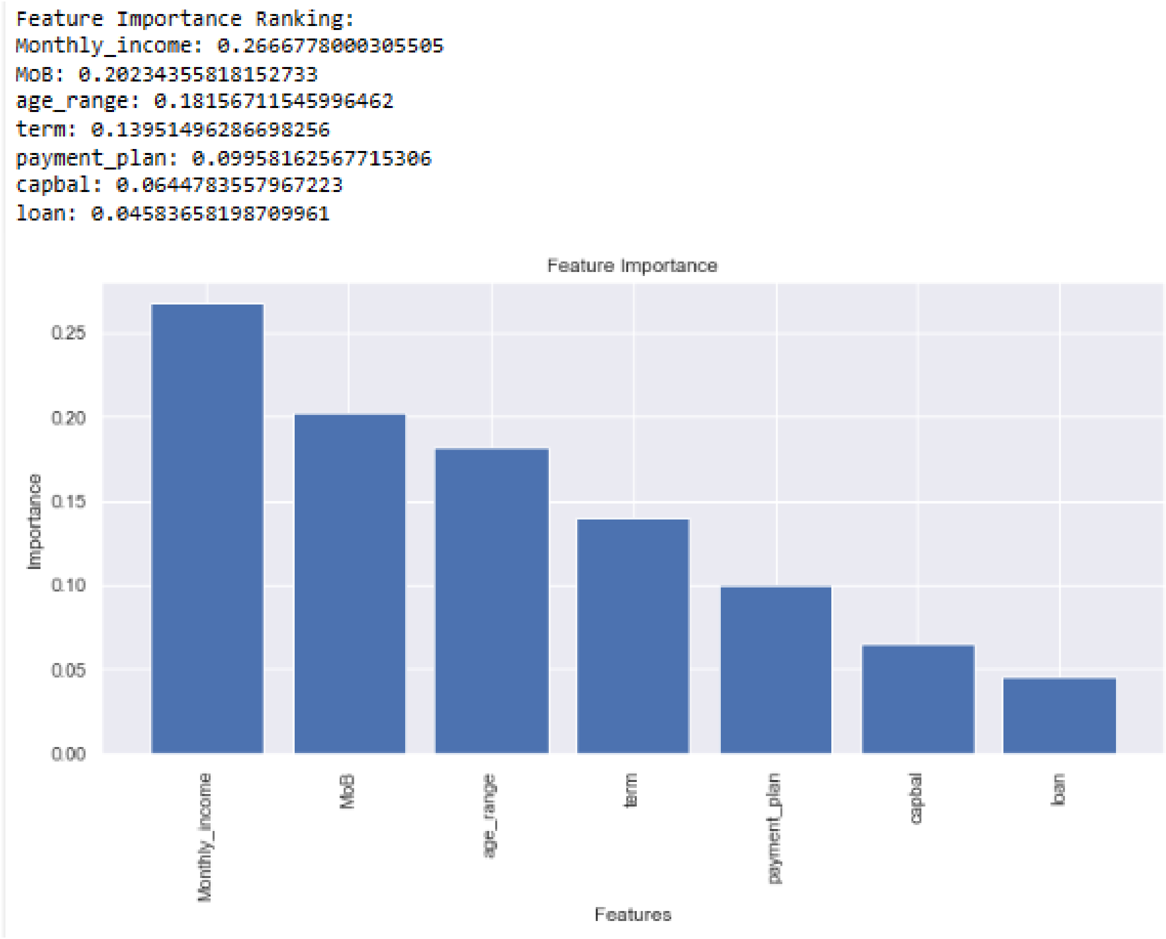


Figure 9

The Figure 7, suggest that the variable "gross_income" has the highest feature importance, indicating that it plays a crucial role in making predictions as it reflects an individual's ability to meet financial obligations. The "age_range" variable comes next in terms of feature importance. Age is a significant factor in credit risk assessment, as younger and older individuals may have different financial behaviours and risk profiles. And it is closely followed by loan duration.

Output 3 (with a new dataset that includes variables such as MOB and Monthly Income but excludes Overdue and Employment Status)



```
Feature Importance Ranking:
Monthly_income: 0.2666778000305505
MoB: 0.2023435581815273З
age_range: 0.18156711545996462
term: 0.13951496286698256
payment_plan: 0.09958162567715306
capbal: 0.0644783557967223
loan: 0.04583658198709961
```

Figure 10

Each feature importance score indicates the relative significance of a feature in predicting default. Larger values suggest that this feature contributes more to the overall predictive power of the Random Forest model. Lower value might not be as influential as some other features, but it still contributes to the model's performance. In this model most important features are Monthly income, MOB, Term and age range. In summary, the feature importance scores indicate the relative contribution of each feature to the model's ability to make accurate predictions. This model attained a remarkable mean accuracy of 96%.

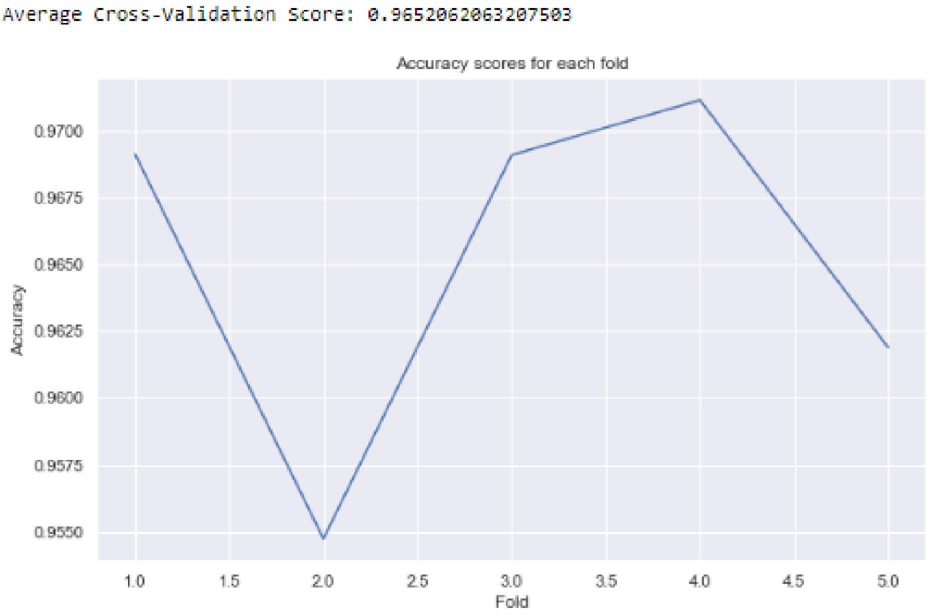
7.2. PREDICTING FEATURE IMPORTANCE OF DELINQUENCY

7.2.1. Logistic Regression Model

To predict the feature importance of delinquency, the Logistic Regression model is used. It allows us to determine which factors have the most impact on the probability of an account being overdue. These insights are critical in developing proactive credit risk management techniques that allow us to predict and address potential delinquencies (LendFoundry.com, 2021).

We used a 5-Fold Cross Validation approach with the Logistic Regression model, which is intended to predict account going into delinquency stage. This validation approach divides the dataset into five equally sized subgroups, or "folds." Following that, the model is trained and tested five times, with each fold acting as a validation set once and the remaining folds serving as training folds. We can comprehensively evaluate the model's performance across numerous data divisions thanks to this iterative procedure (Brownlee, 2020).

Output

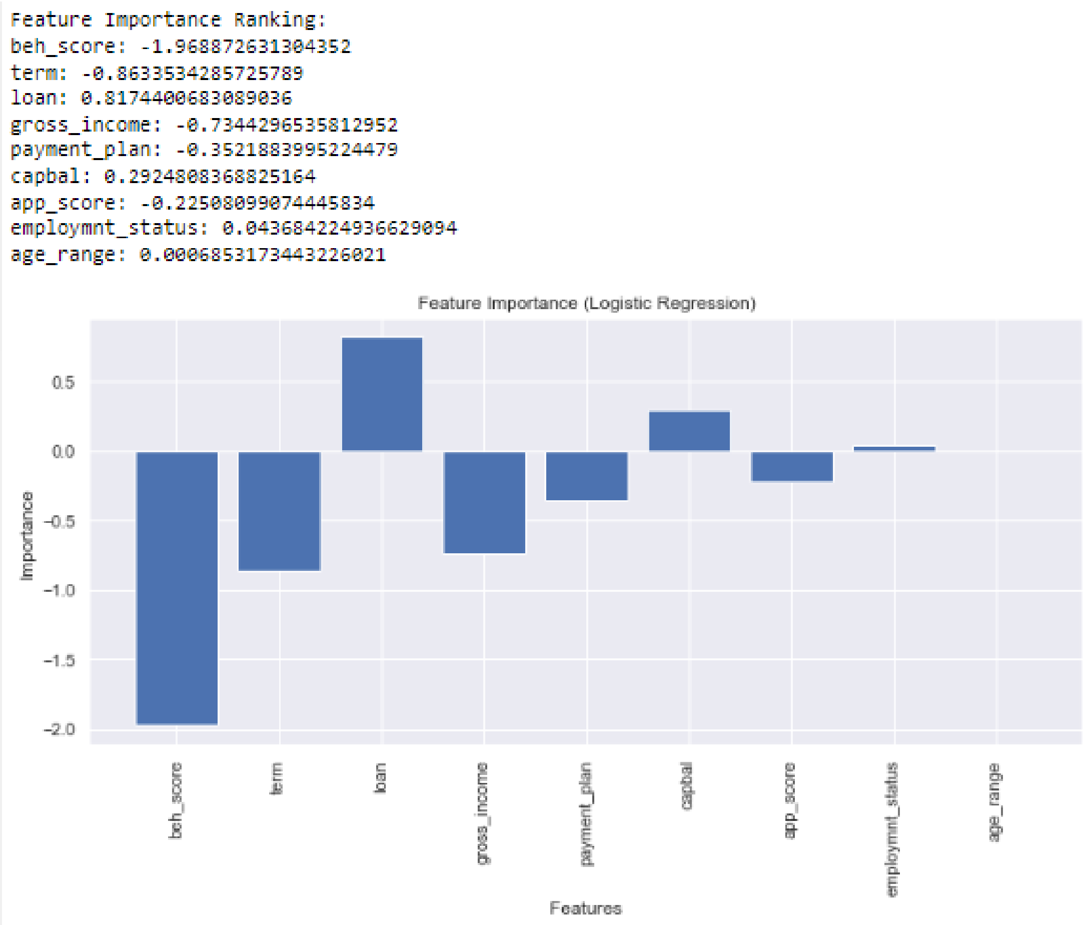


```
Average Cross-Validation Score: 0.9652062063207503
```

Figure 11

The notion of 5-fold cross-validation is crucial in verifying the first logistic regression model developed for predicting delinquency. The 5-fold cross-validation method divides the dataset into 'K' equally sized subgroups, trains the model on 'K-1' of these subsets, then validates it on the remaining subset. This technique is done '5' times to ensure that each subset is used as training and validation data at least once. This approach improves in analysing the model's performance robustly, reducing the risk of overfitting, and offering a more reliable assessment of its accuracy (Brownlee, 2020).

The model attained a remarkable mean accuracy of 96%, suggesting its great predictive powers in identifying delinquency. The predictor variables 'beh_score,' 'term,' 'loan amount,' and 'gross income' all contributed significantly to the model's success. However, there is a difference in the graph between the logistic regression and the other models. This disparity might be traced to the different techniques and algorithms employed by each model.



```
Feature Importance Ranking:
beh_score: -1.968872631304352
term: -0.8633534285725789
loan: 0.8174400683089036
gross_income: -0.7344296535812952
payment_plan: -0.3521883995224479
capbal: 0.2924808368825164
app_score: -0.2250809907445834
employmnt_status: 0.043684224936629094
age_range: 0.0006853173443226021
```

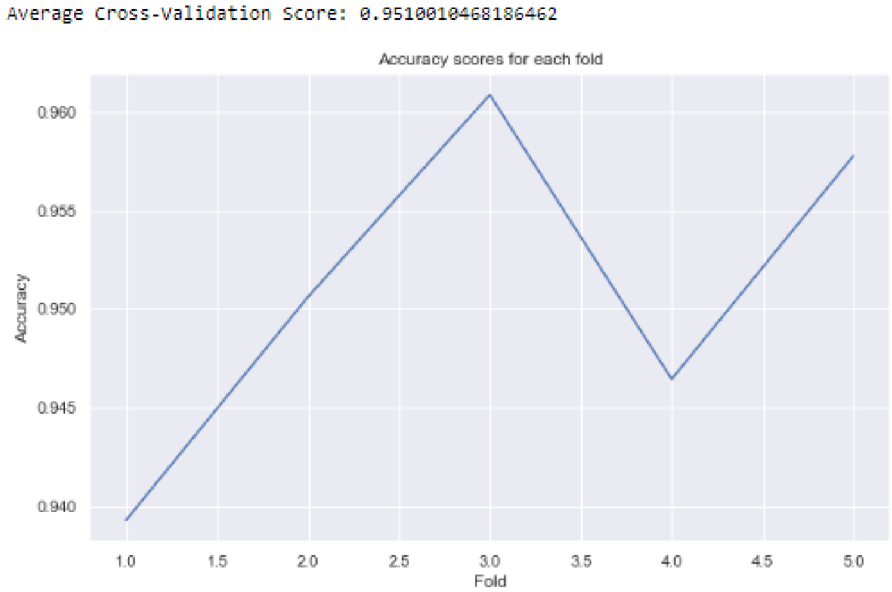
Figure 12

Each feature importance score indicates the relative significance of a feature in predicting delinquency. The lower the behaviour score, the greater the chance of delinquency, and a value closer to -2 suggests that it is a highly influential factor. Term and Gross income in negative values show that the lower the term and gross income, the greater the odds of delinquency, and values more than 0.5 suggest that it is still a very important feature. In the case of a loan, it implies that the more the loan amount, the greater the likelihood of delinquency. Finally, factors such as payment plan, capbal, app score, employment status, and age range have a much smaller effect on delinquency prediction. In summary, the feature importance scores indicate the direction and strength of the relationship between each feature and the likelihood of delinquency.

7.2.2. Decision Tree Model

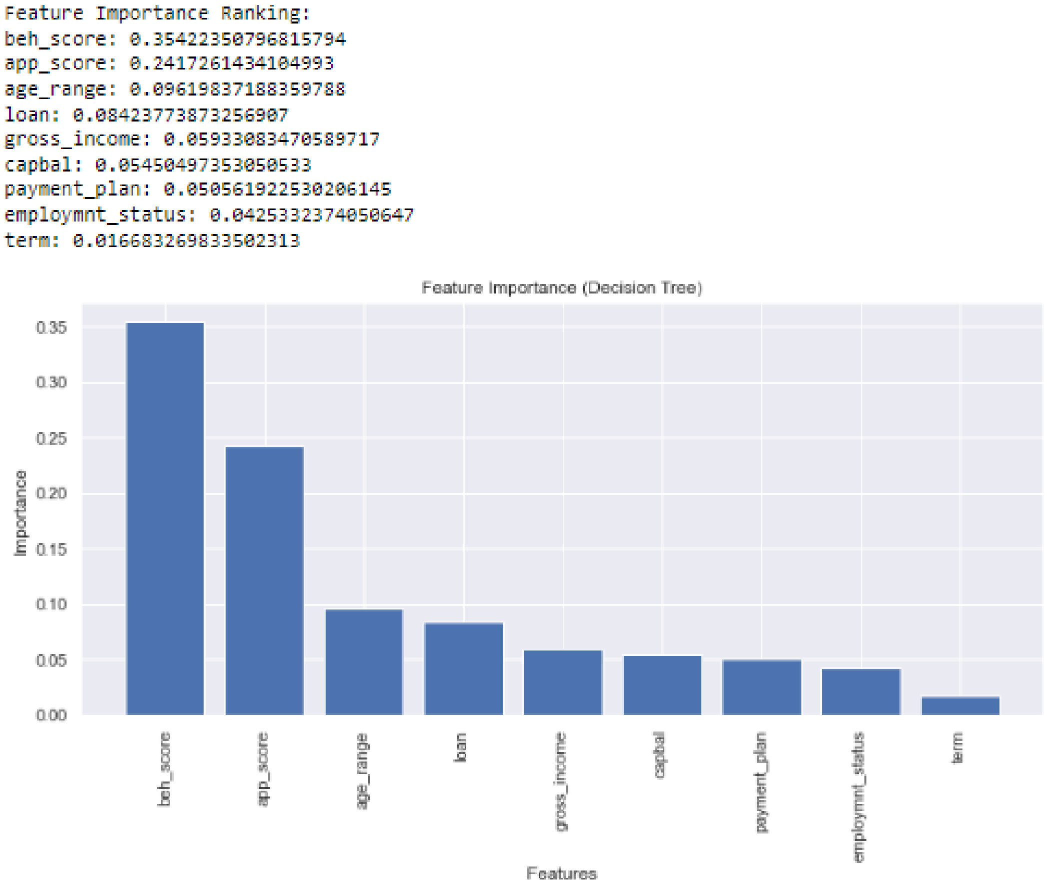
We used 5-Fold Cross Validation to create the Decision Tree model, which was designed to predict account delinquency. This validation method divides our dataset into five equal subgroups, or "folds." After that, the model is trained and tested five times, with each fold acting as a validation set once and the remaining folds used for model training. This rigorous, iterative technique enables us to thoroughly examine the model's performance across various data divisions (LendFoundry.com, 2021).

Our validation efforts yielded an excellent average accuracy rate of 95%. This result demonstrates the model's ability to correctly identify accounts as delinquent or non-delinquent. The Decision Tree model repeatedly proves its ability to make accurate predictions, which is critical in credit risk management. Its consistent performance across different data subsets instils trust in its capacity to detect delinquency effectively, allowing for proactive risk reduction strategies (LendFoundry.com, 2021).



```
Average Cross-Validation Score: 0.9510010468186462
```

Figure 13



```
Feature Importance Ranking:
beh_score: 0.35422350796815794
app_score: 0.2417261434104993
age_range: 0.09619837188359788
loan: 0.08423773873256907
gross_income: 0.05933083470589717
capbal: 0.05450497353050533
payment_plan: 0.050561922530206145
employmnt_status: 0.0425332374050647
term: 0.016683269833502313
```

Figure 14

Each feature importance score indicates the relative significance of a feature in predicting delinquency. Larger values suggest that this feature contributes more to the overall predictive power of the decision Tree model. Lower value might not be as influential as some other features, but it still contributes to the model's performance. In this model most important features are behavioural score, application score, loan amount and age range. . In summary, the feature importance scores indicate the relative contribution of each feature to the model's ability to make accurate predictions.

7.2.3. Random Forest Model

We used a rigorous 5-Fold Cross Validation approach on the Random Forest model developed to predict account delinquency. This approach divides our dataset into five equal-sized subgroups called "folds" (Brownlee, 2020). The model is then trained and tested five times, with each fold acting as a validation set once and contributing to model training the remaining folds. This methodical technique allows us to comprehensively examine the model's performance across several data subsets, offering vital insights into its consistency and generalizability (LendFoundry.com, 2021).

Our validation approach produced a fantastic average accuracy rate of 97%. This outstanding result demonstrates the Random Forest model's ability to reliably identify accounts as delinquent or non-delinquent. The model's capacity to produce correct predictions across diverse data divisions is continuously demonstrated, validating its usefulness in credit risk management. The Random Forest model is a reliable tool for identifying possible delinquencies and assisting financial institutions in developing proactive risk mitigation strategies due to its strong performance (LendFoundry.com, 2021).Top of Form

```
Average Cross-Validation Score: 0.9742643662085244
```

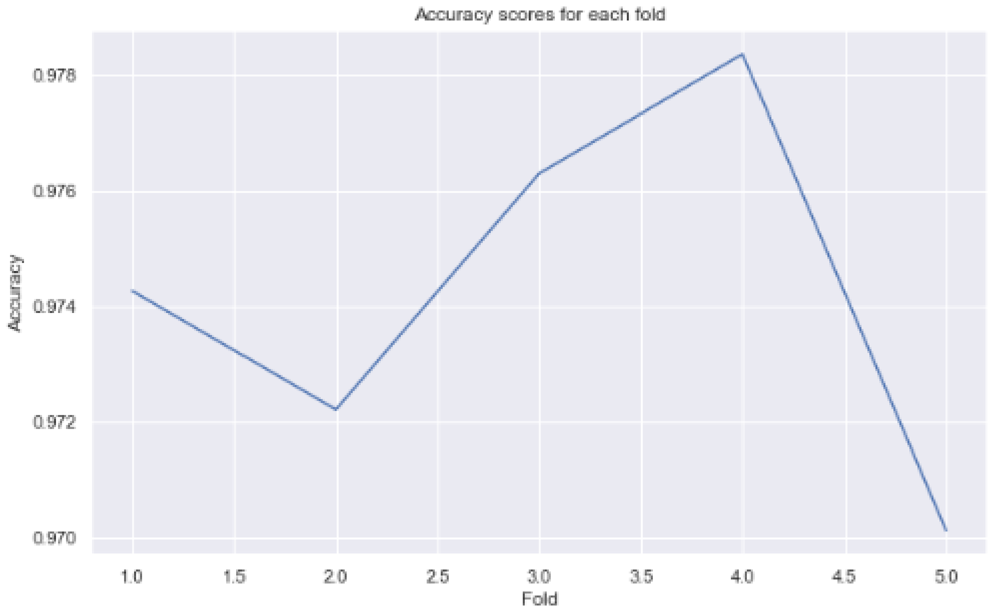
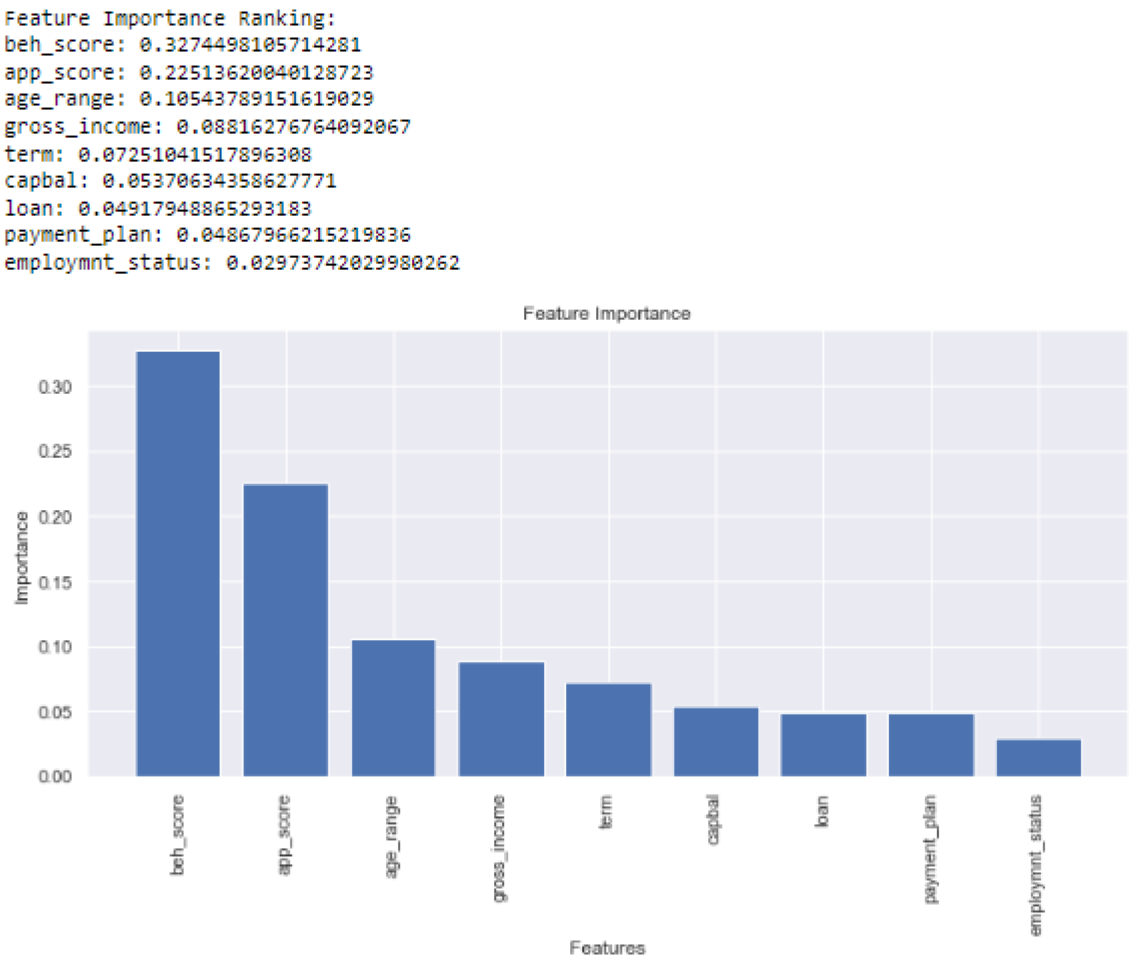


Figure 15



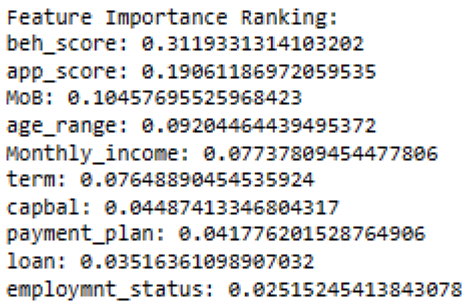
```
Feature Importance Ranking:
beh_score: 0.3274498105714281
app_score: 0.22513620040128723
age_range: 0.10543789151619029
gross_income: 0.08816276764092067
term: 0.07251041517896308
capbal: 0.05370634358627771
loan: 0.04917948865293183
payment_plan: 0.04867966215219836
employmnt_status: 0.02973742029980262
```

Figure 16

Each feature importance score indicates the relative significance of a feature in predicting delinquency. Larger values suggest that this feature contributes more to the overall predictive power of the Random Forest model. Lower value might not be as influential as some other features, but it still contributes to the model's performance. In this model most important features are behavioural score, application score, gross Income and age range. In summary, the feature importance scores indicate the relative contribution of each feature to the model's ability to make accurate predictions.

Output 2 (with a new dataset that includes variables such as MOB and Monthly Income but excludes Overdue)

Note: For other models using new dataset please refer Appendix C



```
Feature Importance Ranking:
beh_score: 0.3119331314103202
app_score: 0.19061186972059535
MoB: 0.10457695525968423
age_range: 0.09204464439495372
Monthly_income: 0.07737809454477806
term: 0.07648890454535924
capbal: 0.04487413346804317
payment_plan: 0.04177620152876490б
loan: 0.03516361098907032
employmnt_status: 0.02515245413843078
```

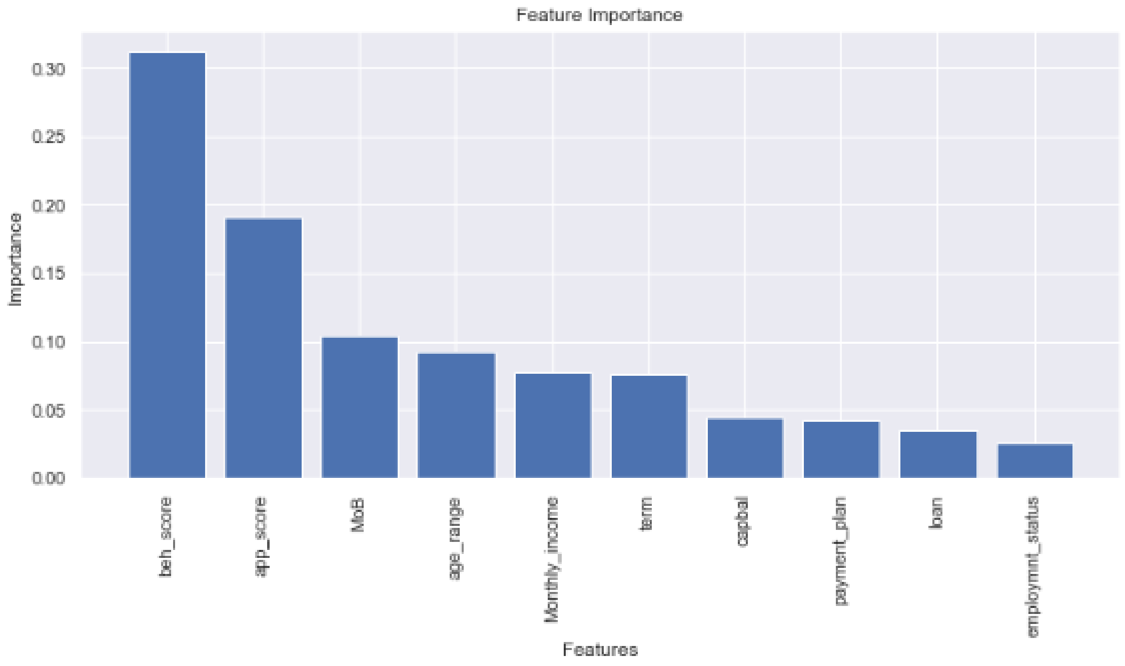
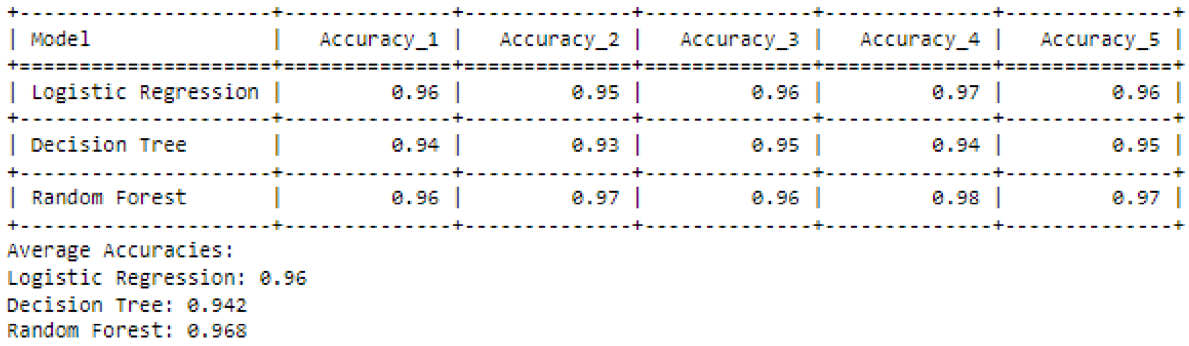


Figure 17

Each feature importance score indicates the relative significance of a feature in predicting delinquency. Larger values suggest that this feature contributes more to the overall predictive power of the Random Forest model. Lower value might not be as influential as some other features, but it still contributes to the model's performance. In this model most important features are behavioural score, application score, MOB, Monthly income, and age range. In summary, the feature importance scores indicate the relative contribution of each feature to the model's ability to make accurate predictions. This model attained a remarkable mean accuracy of 96%.

Note: please refer the appendix C for Accuracy score of Latest Dataset which contains MOB & Monthly Income. Only 2 models are created using the latest dataset. Whereas the whole analysis and modelling is done using the first dataset.

7.2.4. Model Comparison



```
+---------------------+-------------+-------------+-------------+-------------+-------------+
| Model               |  Accuracy_1 |  Accuracy_2 |  Accuracy_3 |  Accuracy_4 |  Accuracy_5 |
+=====================+=============+=============+=============+=============+=============+
| Logistic Regression |        0.96 |        0.95 |        0.96 |        0.97 |        0.96 |
+---------------------+-------------+-------------+-------------+-------------+-------------+
| Decision Tree       |        0.94 |        0.93 |        0.95 |        0.94 |        0.95 |
+---------------------+-------------+-------------+-------------+-------------+-------------+
| Random Forest       |        0.96 |        0.97 |        0.96 |        0.98 |        0.97 |
+---------------------+-------------+-------------+-------------+-------------+-------------+
Average Accuracies:
Logistic Regression: 0.96
Decision Tree: 0.942
Random Forest: 0.968
```

Figure 18

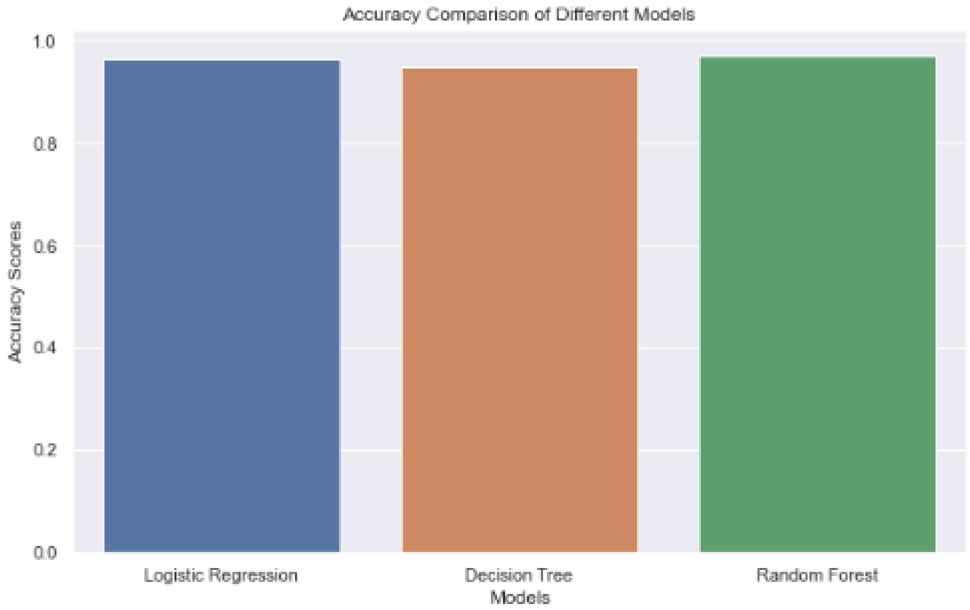


Figure 19

8. BUSINESS EVALUATION & RECOMMENDATIONS

Our findings have significant results for predicting defaults and delinquencies. The timing of defaults revealed an interesting trend. Most of those who defaulted done so over the past six months, between the 36th and 40th snapshot range. This historical pattern indicates that accounts are more vulnerable to default during this time period. The investigation also found that accounts with a "Good" designation had fewer outstanding amounts. Accounts defined as "Closed" or "Bad" on the other hand were more likely to have overdue amounts, particularly in the higher overdue categories. This pattern emphasises the importance of account status in predicting defaults and delinquencies. In terms of delinquency phases, an unusual pattern emerged. The likelihood of an account going into default grew significantly as the delinquency stage progressed. Notably, no "Good" status accounts were included in accounts at the "4" delinquency level or higher. This suggests that accounts in these advanced phases of delinquency are at a much-increased risk of default.

Our research resulted in several notable achievements:

Behavioural Scorecard Development: We successfully created a behavioural scorecard that assigned scores based on the estimated likelihood of defaults. S-test verified the scorecard's powerful discriminating powers, emphasising its dependability in distinguishing between high-risk and low-risk accounts. By adding these models into risk assessment procedures, businesses may greatly improve their capacity to detect high-risk accounts. This results into more targeted and effective risk management techniques, lowering the possibility of default losses (Fibe, n.d.).

Delinquency Prediction Models: We built three separate models to predict delinquency: a logistic regression model, a decision tree, and a random forest. We confirmed the dependability of these models via rigorous 5-fold cross-validation (Brownlee, 2020). The random forest model has the best accuracy of 97% among them, suggesting its excellent predictive potential. Businesses can better deploy resources, such as collection efforts and risk mitigation methods, by identifying high-risk accounts early on. This avoids wasteful expenditure on low-risk accounts while prioritising measures for those with a higher possibility of delinquency (LendFoundry.com, 2021).

Feature Importance Ranking: The application of feature importance ranking, generated from the logistic regression, decision tree, and random forest models, unveiled the most critical features for predicting delinquency. With a 96% accuracy rate, logistic regression identified Beh_score, term, and loan as the top three features. With 95% accuracy, the decision tree prioritised Beh_score, app_score, and age_range. The random forest model, which had the best accuracy of 97%, highlighted Beh_score, app_score, MOB, Monthly income, and age_range as significant features. These findings highlighted the robustness of Beh_score, Monthly income and age_range across all models. A company may streamline its data-driven decision-making processes by concentrating on five important elements. This enables firms to give credit to low-risk consumers while taking required safeguards for high-risk ones, allowing for a more precise evaluation of customer creditworthiness.

9. LIMITATIONS

Some of our findings lack strong empirical evidence due to dataset limitations or sample size constraints. To address these gaps, we recommend the following for future research:

1. **Longer Observation Period:** Extending the observation period beyond six months, as recommended by IFRS guidelines, can provide a more robust dataset for behavioural scorecard creation. A more extended time frame may capture nuanced changes in borrower behaviour.
2. **Dynamic Scorecard:** Shifting from a static to a dynamic behavioural scorecard can capture real-time changes in borrower behaviour, reducing the likelihood of missed information.
3. **Imbalanced Data Handling**: Further exploration of techniques to handle imbalanced datasets, such as Synthetic Minority Over-sampling Technique (SMOTE) or data augmentation, can mitigate potential biases introduced by oversampling.
4. **Feature Engineering:** Investigate advanced feature engineering techniques to extract additional information from existing variables, enhancing model performance.
5. **Missing Data Handling:** Implement strategies for imputing or handling missing data effectively to prevent information loss.

6. **Variable Formatting:** Standardize variable formats consistently across the dataset to avoid numerical variables in range formats.

10. CONCLUSION

The dataset utilised in this research comes from a financial institution and contains information about 1750 clients, as well as 19 numerical and categorical characteristics. The descriptive statistical analysis was performed using Python, SAS, and Excel to assess the dataset's quality and structure, as well as the features of its attributes and the connection of the variables to the target variable (default & Delinquency). The research further recommended data transformation and purification procedures to increase the dataset's quality, as well as input for a variety of data mining and machine learning algorithms. We made tremendous progress in understanding and predicting defaults and delinquency in our comprehensive research on credit risk management, eventually contributing to more effective risk assessment and mitigation measures. our findings give useful tools and insights for credit risk management. While there are limitations, but our models and behavioural scorecard provide substantial benefits in enhancing risk assessment and minimising financial exposure, eventually protecting lending institutions' financial health.

11. REFERENCES

Abdou, H. A., & Pointon, J. (2011). CREDIT SCORING, STATISTICAL TECHNIQUES AND EVALUATION CRITERIA: A REVIEW OF THE LITERATURE. *International Journal of Intelligent Systems in Accounting,* Finance *& Management*, *18*(2–3), 59–88. https://doi.org/10.1002/isaf.325

Al Amari A. 2002. The credit evaluation process and the role of credit scoring: a case study of Qatar. PhD thesis, University College Dublin.

Ali, S. E. A., Rizvi, S. S. H., Lai, F., Ali, R. F., & Jan, A. A. (2021). Predicting Delinquency on Mortgage Loans: An exhaustive parametric comparison of machine learning techniques. *International Journal of Industrial Engineering and Management*, *Volume 12*(Issue 1), 1–13. https://doi.org/10.24867/ijiem-2021-1-272

Anderson R. 2007. *The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation*. Oxford University Press: New York.

Brownlee, J. (2021). SMOTE for Imbalanced Classification with Python. *MachineLearningMastery.com*. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Brownlee, J. (2020). A Gentle Introduction to k-fold Cross-Validation. *MachineLearningMastery.com*. https://machinelearningmastery.com/k-fold-cross-validation/

Chen, S., Guo, Z., & Zhao, X. (2021). Predicting mortgage early delinquency with machine learning methods. *European Journal of Operational Research*, *290*(1), 358–372. https://doi.org/10.1016/j.ejor.2020.07.058

Credit Scoring Series Part Three: Data Preparation and Exploratory Data analysis. (2022, May 24). Default. https://altair.com/newsroom/articles/credit-scoring-series-part-three-data-preparation-and-exploratory-data-analysis?_ga=2.239042432.1439808571.1695496046-999283193.1695496045

Credit Scoring Series Part Two: Credit Scorecard Modeling Methodology. (2022, May 18). Default. https://altair.com/newsroom/articles/credit-scoring-series-part-two-credit-scorecard-modeling-methodology

Credit Score Modelling. (2021). Ariclabarr. Retrieved September 24, 2023, from https://www.ariclabarr.com/credit-modeling.html

Fibe. (n.d.). *Difference between credit score and credit risk assessment | FIBE*. fibe.in. https://www.fibe.in/blogs/credit-score-vs-credit-risk-assessment-whats-the-difference/

Finalyse. (n.d.). *Behavioral Scorecard with Machine Learning Components*. Finalyse - www.finalyse.com. https://www.finalyse.com/blog/behavioral-scorecard-with-machine-learning-components

Gup BE, Kolari JW. 2005. *Commercial Banking: The Management of Risk*. John Wiley and Sons, Inc.: Alabama.

Kyeong, S., & Shin, J. (2022). Two-stage credit scoring using Bayesian approach. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00665-5

LendFoundry.com. (2021, December 7). Delinquency Prediction For A Loan In Service Using Analytics Data. *Medium*. https://medium.com/@lendfoundry/delinquency-prediction-for-a-loan-in-service-using-analytics-data-feae749aa8f8

Malhotra R, Malhotra DK. 2003. Evaluating consumer loans using neural networks *Omega – The International Journal of Management Science* **31**(2): 83–96.

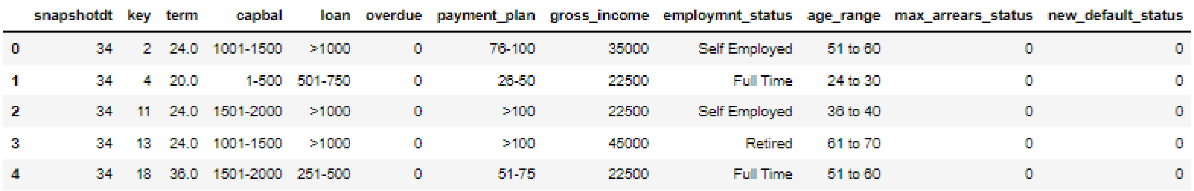
Peterdy, K. (2023). Credit risk. Corporate Finance Institute: https://corporatefinanceinstitute.com/resources/commercial-lending/credit-risk/#:~:text=Credit%20risk%20is%20a%20specific%20financial%20risk%20borne,loss%20in%20the%20event%20a%20default%20does%20occur

Siddiqi N. 2006. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. John Wiley and Sons, Inc.: Hoboken, NJ.

Team, C. (2023). Credit risk analysis models. *Corporate Finance Institute*. https://corporatefinanceinstitute.com/resources/commercial-lending/credit-risk-analysis-models/

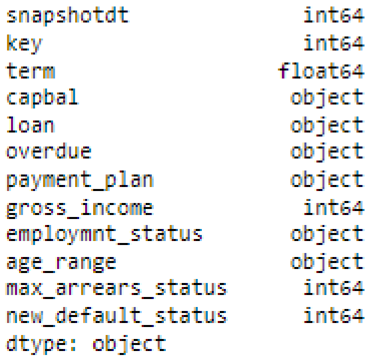
12. APPENDIX A

(EDA)



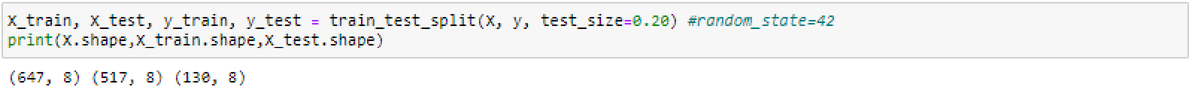
	snapshotdt	key	term	capbal	loan	overdue	payment_plan	gross_income	employmnt_status	age_range	max_arrears_status	new_default_status
0	34	2	24.0	1001-1500	>1000	0	76-100	35000	Self Employed	51 to 60	0	0
1	34	4	20.0	1-500	501-750	0	26-50	22500	Full Time	24 to 30	0	0
2	34	11	24.0	1501-2000	>1000	0	>100	22500	Self Employed	36 to 40	0	0
3	34	13	24.0	1001-1500	>1000	0	>100	45000	Retired	61 to 70	0	0
4	34	18	36.0	1501-2000	251-500	0	51-75	22500	Full Time	51 to 60	0	0

Figure 20



```
snapshotdt              int64
key                     int64
term                  float64
capbal                 object
loan                   object
overdue                object
payment_plan           object
gross_income            int64
employmnt_status       object
age_range              object
max_arrears_status      int64
new_default_status      int64
dtype: object
```

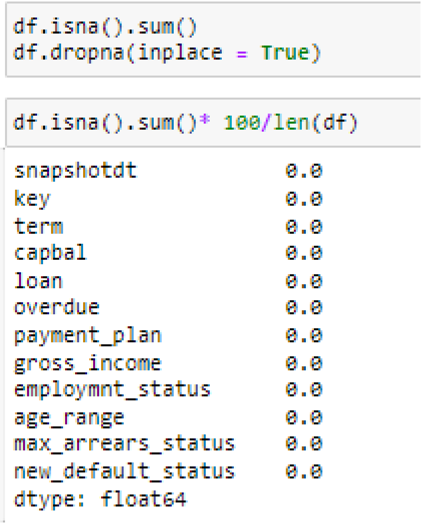
Figure 21



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20) #random_state=42
print(X.shape,X_train.shape,X_test.shape)
```

```
(647, 8) (517, 8) (130, 8)
```

Figure 22

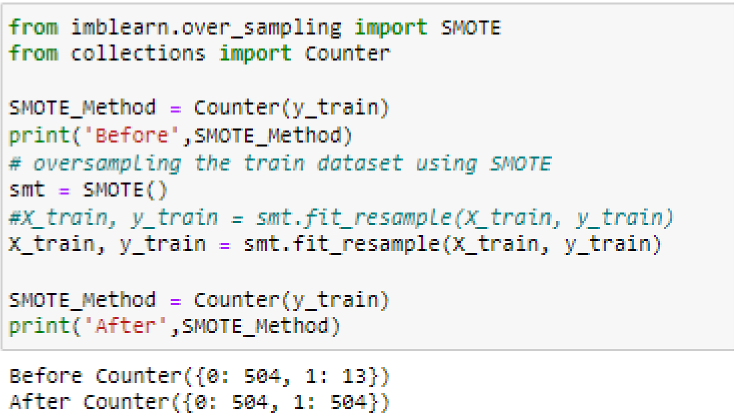


```
df.isna().sum()
df.dropna(inplace = True)
```

```
df.isna().sum()* 100/len(df)
```

```
snapshotdt            0.0
key                   0.0
term                  0.0
capbal                0.0
loan                  0.0
overdue               0.0
payment_plan          0.0
gross_income          0.0
employmnt_status      0.0
age_range             0.0
max_arrears_status    0.0
new_default_status    0.0
dtype: float64
```

Figure 23

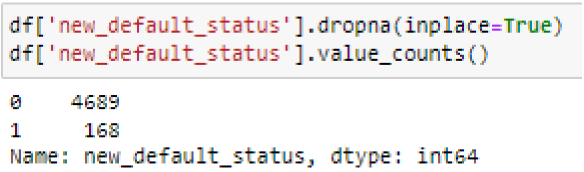


```
from imblearn.over_sampling import SMOTE
from collections import Counter

SMOTE_Method = Counter(y_train)
print('Before',SMOTE_Method)
# oversampling the train dataset using SMOTE
smt = SMOTE()
#X_train, y_train = smt.fit_resample(X_train, y_train)
X_train, y_train = smt.fit_resample(X_train, y_train)

SMOTE_Method = Counter(y_train)
print('After',SMOTE_Method)
```

```
Before Counter({0: 504, 1: 13})
After Counter({0: 504, 1: 504})
```

Figure 24

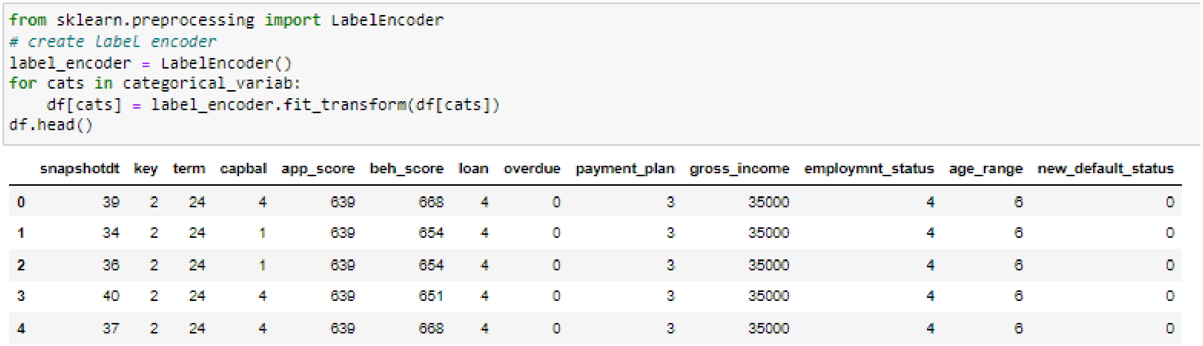
Delinquency dataset After Data Preprocessing



```
df['new_default_status'].dropna(inplace=True)
df['new_default_status'].value_counts()
```

```
0    4689
1     168
Name: new_default_status, dtype: int64
```

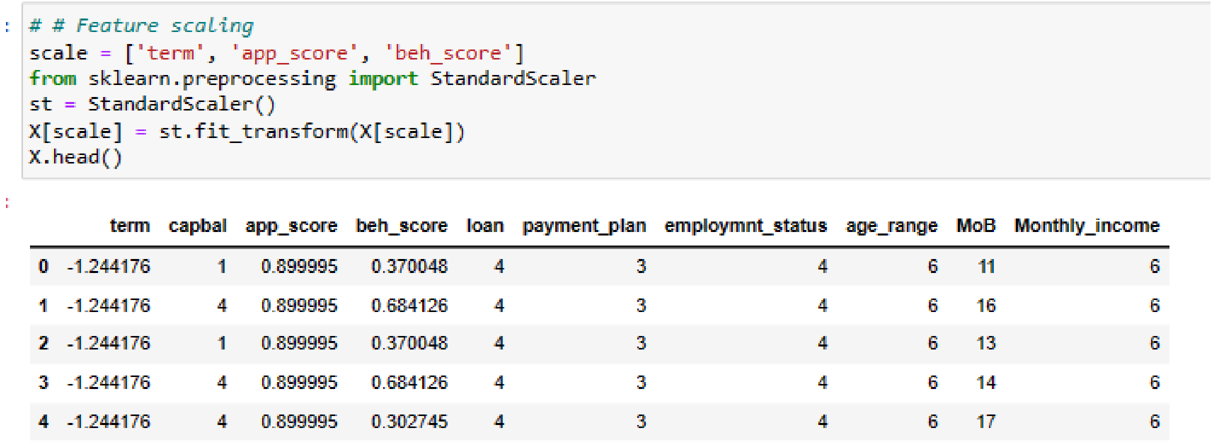
Figure 25



```
from sklearn.preprocessing import LabelEncoder
# create label encoder
label_encoder = LabelEncoder()
for cats in categorical_variab:
    df[cats] = label_encoder.fit_transform(df[cats])
df.head()
```

	snapshotdt	key	term	capbal	app_score	beh_score	loan	overdue	payment_plan	gross_income	employmnt_status	age_range	new_default_status
0	39	2	24	4	639	668	4	0	3	35000	4	6	0
1	34	2	24	1	639	654	4	0	3	35000	4	6	0
2	36	2	24	1	639	654	4	0	3	35000	4	6	0
3	40	2	24	4	639	651	4	0	3	35000	4	6	0
4	37	2	24	4	639	668	4	0	3	35000	4	6	0

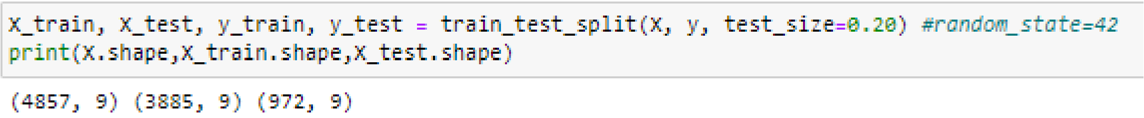
Figure 26



```
# # Feature scaling
scale = ['term', 'app_score', 'beh_score']
from sklearn.preprocessing import StandardScaler
st = StandardScaler()
X[scale] = st.fit_transform(X[scale])
X.head()
```

	term	capbal	app_score	beh_score	loan	payment_plan	employmnt_status	age_range	MoB	Monthly_income
0	-1.244176	1	0.899995	0.370048	4	3	4	6	11	6
1	-1.244176	4	0.899995	0.684126	4	3	4	6	16	6
2	-1.244176	1	0.899995	0.370048	4	3	4	6	13	6
3	-1.244176	4	0.899995	0.684126	4	3	4	6	14	6
4	-1.244176	4	0.899995	0.302745	4	3	4	6	17	6

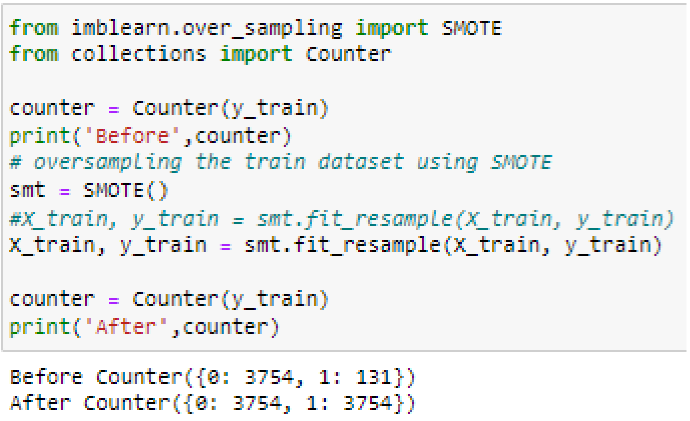
Figure 27



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20) #random_state=42
print(X.shape,X_train.shape,X_test.shape)
```

```
(4857, 9) (3885, 9) (972, 9)
```

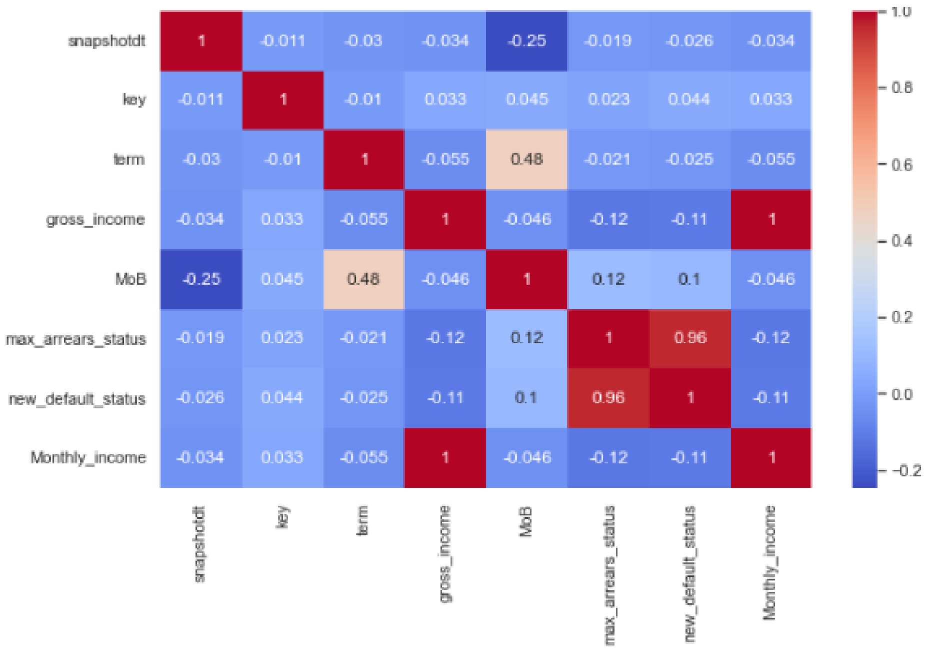
Figure 28



```
from imblearn.over_sampling import SMOTE
from collections import Counter

counter = Counter(y_train)
print('Before',counter)
# oversampling the train dataset using SMOTE
smt = SMOTE()
#X_train, y_train = smt.fit_resample(X_train, y_train)
X_train, y_train = smt.fit_resample(X_train, y_train)

counter = Counter(y_train)
print('After',counter)
```

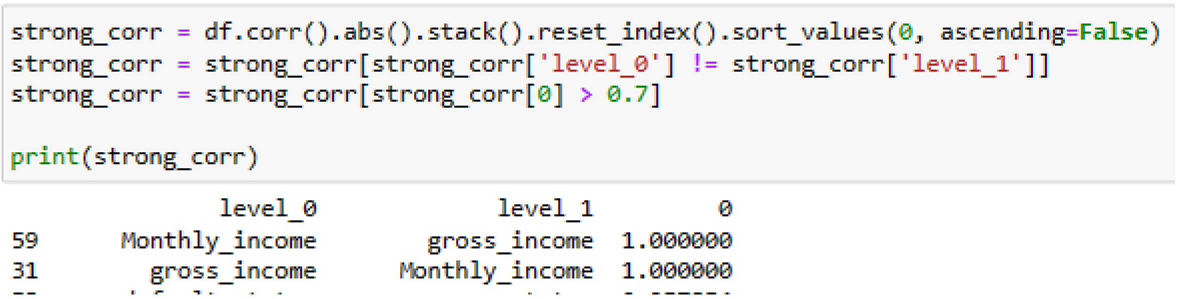
```
Before Counter({0: 3754, 1: 131})
After Counter({0: 3754, 1: 3754})
```

Figure 29



	snapshotdt	key	term	gross_income	MoB	max_arrears_status	new_default_status	Monthly_income
snapshotdt	1	-0.011	-0.03	-0.034	-0.25	-0.019	-0.026	-0.034
key	-0.011	1	-0.01	0.033	0.045	0.023	0.044	0.033
term	-0.03	-0.01	1	-0.055	0.48	-0.021	-0.025	-0.055
gross_income	-0.034	0.033	-0.055	1	-0.046	-0.12	-0.11	1
MoB	-0.25	0.045	0.48	-0.046	1	0.12	0.1	-0.046
max_arrears_status	-0.019	0.023	-0.021	-0.12	0.12	1	0.96	-0.12
new_default_status	-0.026	0.044	-0.025	-0.11	0.1	0.96	1	-0.11
Monthly_income	-0.034	0.033	-0.055	1	-0.046	-0.12	-0.11	1

Figure 30



```
strong_corr = df.corr().abs().stack().reset_index().sort_values(0, ascending=False)
strong_corr = strong_corr[strong_corr['level_0'] != strong_corr['level_1']]
strong_corr = strong_corr[strong_corr[0] > 0.7]

print(strong_corr)
```

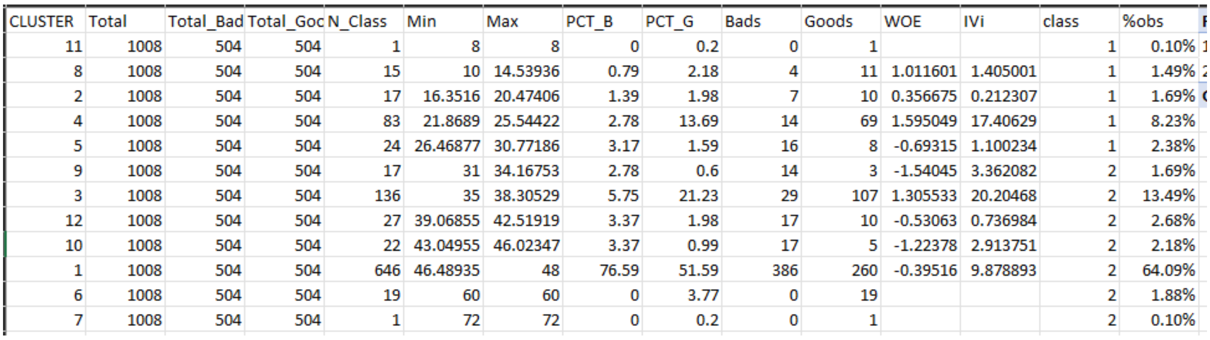
```
             level_0          level_1         0
59    Monthly_income     gross_income  1.000000
31      gross_income   Monthly_income  1.000000
```

Figure 31

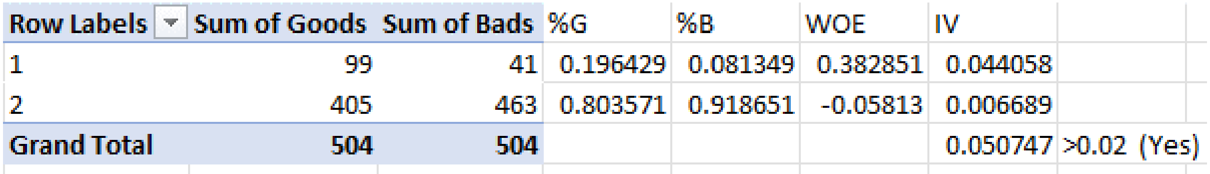
13. APPENDIX B

(Behavioural Scorecard)

Term



CLUSTER	Total	Total_Bad	Total_Goc	N_Class	Min	Max	PCT_B	PCT_G	Bads	Goods	WOE	IVi	class	%obs	F
11	1008	504	504	1	8	8	0	0.2	0	1			1	0.10%	1
8	1008	504	504	15	10	14.53936	0.79	2.18	4	11	1.011601	1.405001	1	1.49%	2
2	1008	504	504	17	16.3516	20.47406	1.39	1.98	7	10	0.356675	0.212307	1	1.69%	(
4	1008	504	504	83	21.8689	25.54422	2.78	13.69	14	69	1.595049	17.40629	1	8.23%	
5	1008	504	504	24	26.46877	30.77186	3.17	1.59	16	8	-0.69315	1.100234	1	2.38%	
9	1008	504	504	17	31	34.16753	2.78	0.6	14	3	-1.54045	3.362082	2	1.69%	
3	1008	504	504	136	35	38.30529	5.75	21.23	29	107	1.305533	20.20468	2	13.49%	
12	1008	504	504	27	39.06855	42.51919	3.37	1.98	17	10	-0.53063	0.736984	2	2.68%	
10	1008	504	504	22	43.04955	46.02347	3.37	0.99	17	5	-1.22378	2.913751	2	2.18%	
1	1008	504	504	646	46.48935	48	76.59	51.59	386	260	-0.39516	9.878893	2	64.09%	
6	1008	504	504	19	60	60	0	3.77	0	19			2	1.88%	
7	1008	504	504	1	72	72	0	0.2	0	1			2	0.10%	



Row Labels	Sum of Goods	Sum of Bads	%G	%B	WOE	IV	
1	99	41	0.196429	0.081349	0.382851	0.044058	
2	405	463	0.803571	0.918651	-0.05813	0.006689	
Grand Total	504	504				0.050747	>0.02 (Yes)

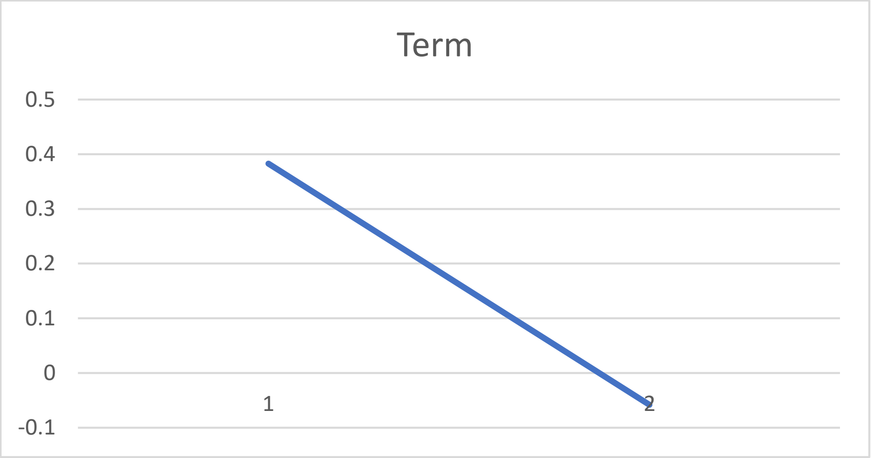
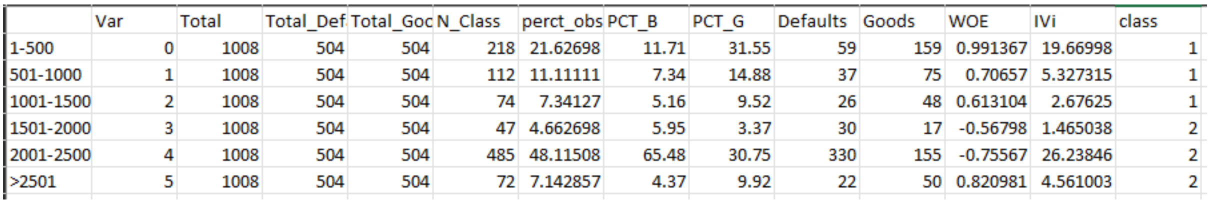
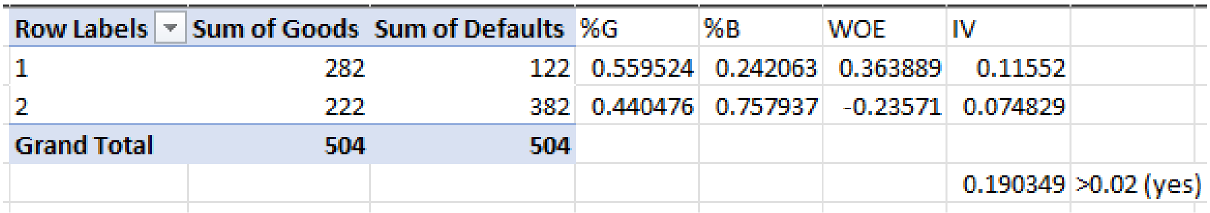


Figure 32

Capbal



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
1-500	0	1008	504	504	218	21.62698	11.71	31.55	59	159	0.991367	19.66998	1
501-1000	1	1008	504	504	112	11.11111	7.34	14.88	37	75	0.70657	5.327315	1
1001-1500	2	1008	504	504	74	7.34127	5.16	9.52	26	48	0.613104	2.67625	1
1501-2000	3	1008	504	504	47	4.662698	5.95	3.37	30	17	-0.56798	1.465038	2
2001-2500	4	1008	504	504	485	48.11508	65.48	30.75	330	155	-0.75567	26.23846	2
>2501	5	1008	504	504	72	7.142857	4.37	9.92	22	50	0.820981	4.561003	2



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
1	282	122	0.559524	0.242063	0.363889	0.11552	
2	222	382	0.440476	0.757937	-0.23571	0.074829	
Grand Total	504	504					
						0.190349	>0.02 (yes)

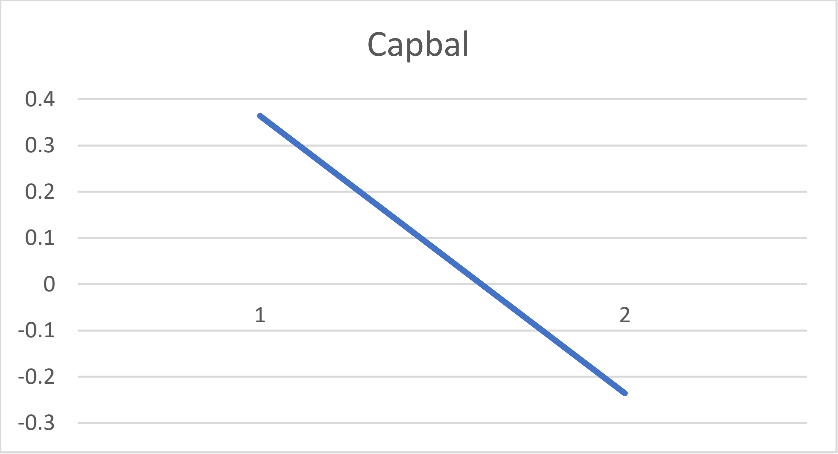
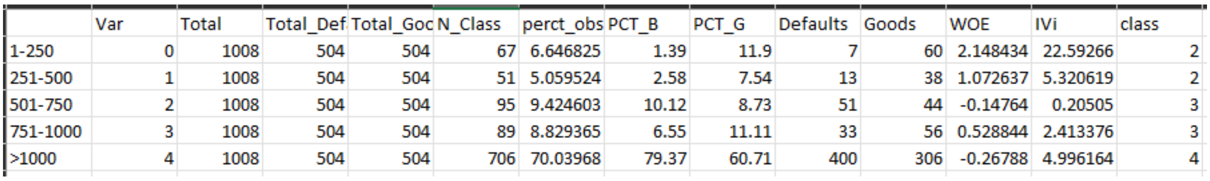
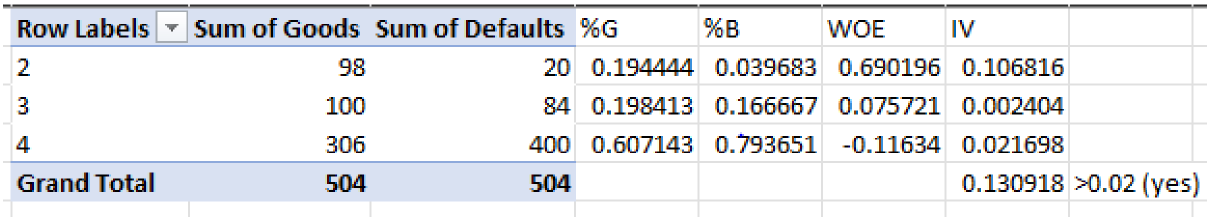


Figure 33

Loan



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
1-250	0	1008	504	504	67	6.646825	1.39	11.9	7	60	2.148434	22.59266	2
251-500	1	1008	504	504	51	5.059524	2.58	7.54	13	38	1.072637	5.320619	2
501-750	2	1008	504	504	95	9.424603	10.12	8.73	51	44	-0.14764	0.20505	3
751-1000	3	1008	504	504	89	8.829365	6.55	11.11	33	56	0.528844	2.413376	3
>1000	4	1008	504	504	706	70.03968	79.37	60.71	400	306	-0.26788	4.996164	4



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
2	98	20	0.194444	0.039683	0.690196	0.106816	
3	100	84	0.198413	0.166667	0.075721	0.002404	
4	306	400	0.607143	0.793651	-0.11634	0.021698	
Grand Total	504	504				0.130918	>0.02 (yes)

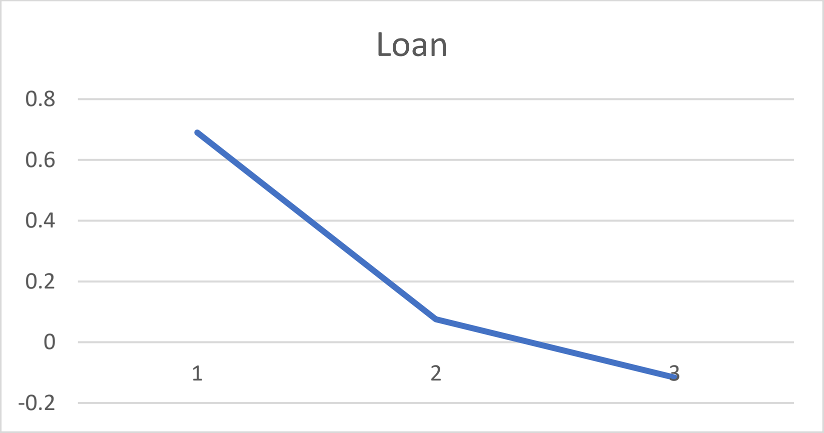
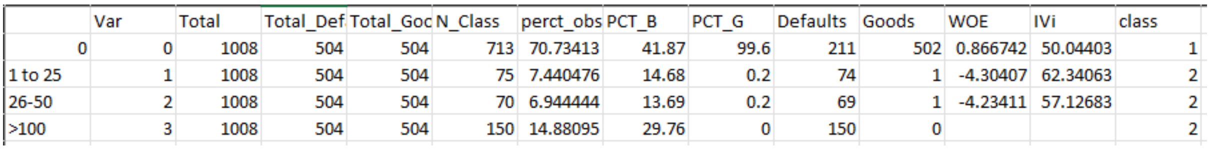
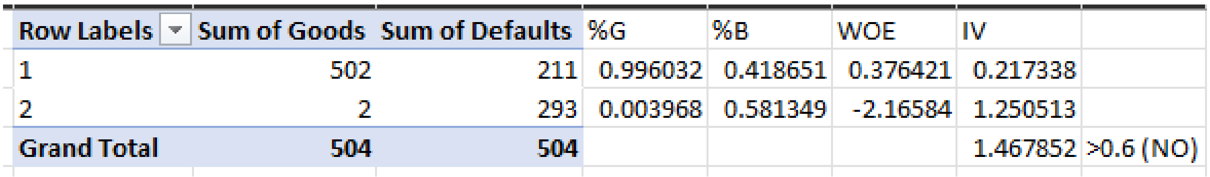


Figure 34

Overdue



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
0	0	1008	504	504	713	70.73413	41.87	99.6	211	502	0.866742	50.04403	1
1 to 25	1	1008	504	504	75	7.440476	14.68	0.2	74	1	-4.30407	62.34063	2
26-50	2	1008	504	504	70	6.944444	13.69	0.2	69	1	-4.23411	57.12683	2
>100	3	1008	504	504	150	14.88095	29.76	0	150	0			2



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
1	502	211	0.996032	0.418651	0.376421	0.217338	
2	2	293	0.003968	0.581349	-2.16584	1.250513	
Grand Total	504	504				1.467852	>0.6 (NO)

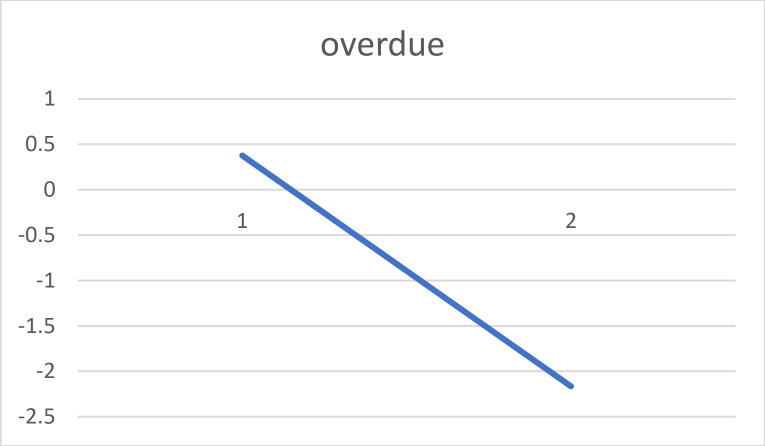
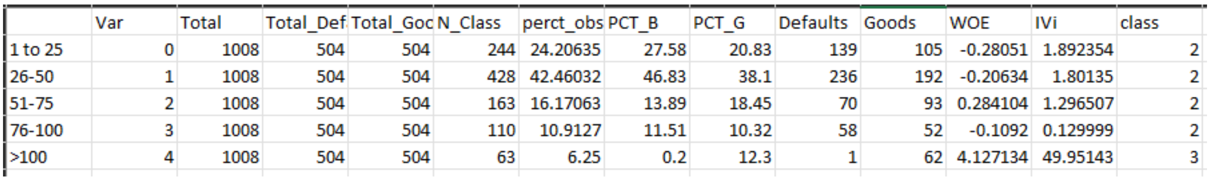
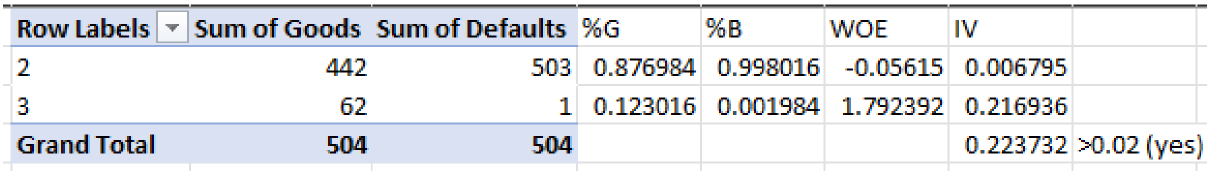


Figure 35

payment_plan



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
1 to 25	0	1008	504	504	244	24.20635	27.58	20.83	139	105	-0.28051	1.892354	2
26-50	1	1008	504	504	428	42.46032	46.83	38.1	236	192	-0.20634	1.80135	2
51-75	2	1008	504	504	163	16.17063	13.89	18.45	70	93	0.284104	1.296507	2
76-100	3	1008	504	504	110	10.9127	11.51	10.32	58	52	-0.1092	0.129999	2
>100	4	1008	504	504	63	6.25	0.2	12.3	1	62	4.127134	49.95143	3



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
2	442	503	0.876984	0.998016	-0.05615	0.006795	
3	62	1	0.123016	0.001984	1.792392	0.216936	
Grand Total	504	504				0.223732	>0.02 (yes)

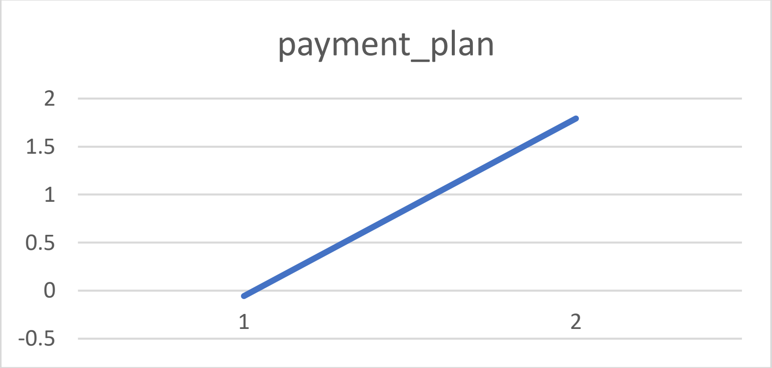
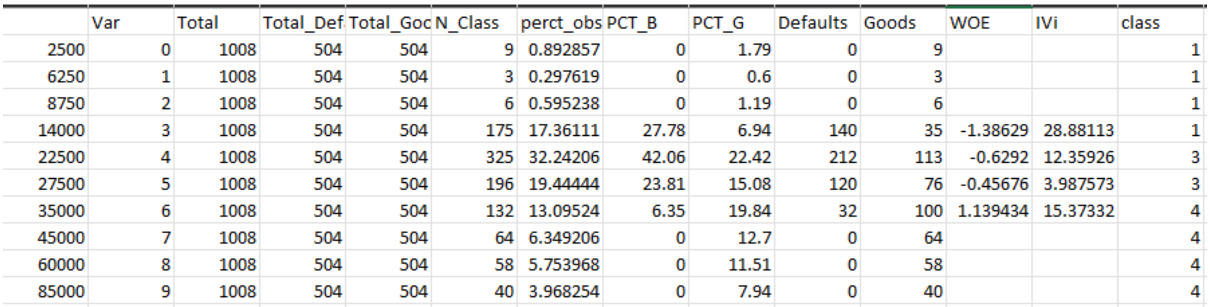
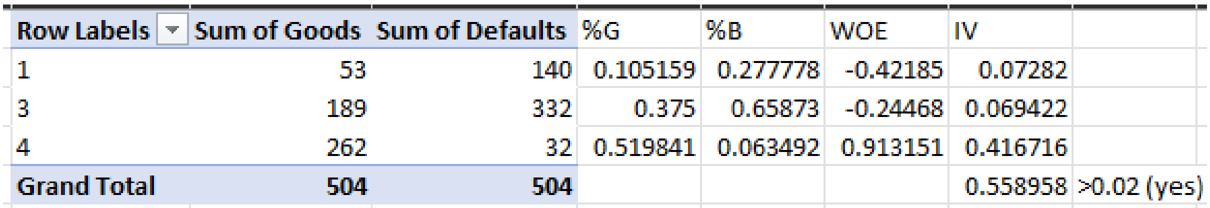


Figure 36

gross_income



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
2500	0	1008	504	504	9	0.892857	0	1.79	0	9			1
6250	1	1008	504	504	3	0.297619	0	0.6	0	3			1
8750	2	1008	504	504	6	0.595238	0	1.19	0	6			1
14000	3	1008	504	504	175	17.36111	27.78	6.94	140	35	-1.38629	28.88113	1
22500	4	1008	504	504	325	32.24206	42.06	22.42	212	113	-0.6292	12.35926	3
27500	5	1008	504	504	196	19.44444	23.81	15.08	120	76	-0.45676	3.987573	3
35000	6	1008	504	504	132	13.09524	6.35	19.84	32	100	1.139434	15.37332	4
45000	7	1008	504	504	64	6.349206	0	12.7	0	64			4
60000	8	1008	504	504	58	5.753968	0	11.51	0	58			4
85000	9	1008	504	504	40	3.968254	0	7.94	0	40			4



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
1	53	140	0.105159	0.277778	-0.42185	0.07282	
3	189	332	0.375	0.65873	-0.24468	0.069422	
4	262	32	0.519841	0.063492	0.913151	0.416716	
Grand Total	**504**	**504**				0.558958	>0.02 (yes)

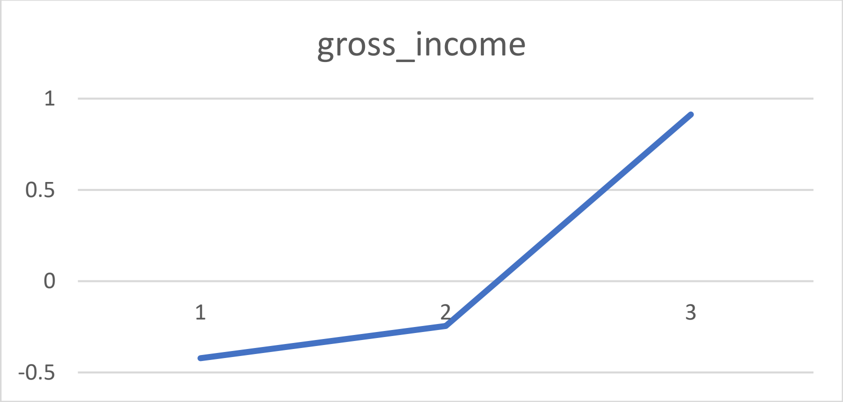
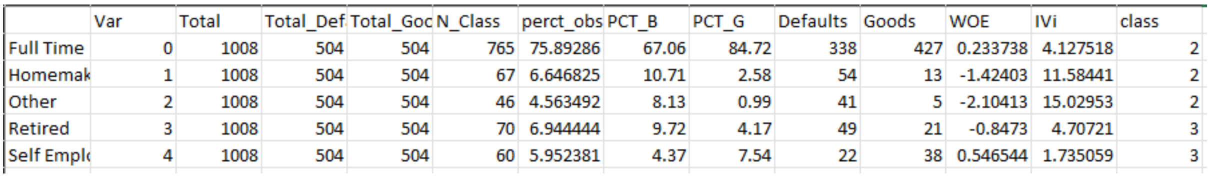
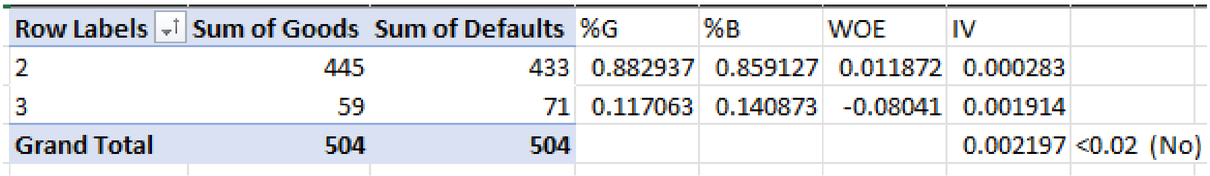


Figure 37

Employment status



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
Full Time	0	1008	504	504	765	75.89286	67.06	84.72	338	427	0.233738	4.127518	2
Homemak	1	1008	504	504	67	6.646825	10.71	2.58	54	13	-1.42403	11.58441	2
Other	2	1008	504	504	46	4.563492	8.13	0.99	41	5	-2.10413	15.02953	2
Retired	3	1008	504	504	70	6.944444	9.72	4.17	49	21	-0.8473	4.70721	3
Self Empl	4	1008	504	504	60	5.952381	4.37	7.54	22	38	0.546544	1.735059	3



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
2	445	433	0.882937	0.859127	0.011872	0.000283	
3	59	71	0.117063	0.140873	-0.08041	0.001914	
Grand Total	**504**	**504**				0.002197	<0.02 (No)

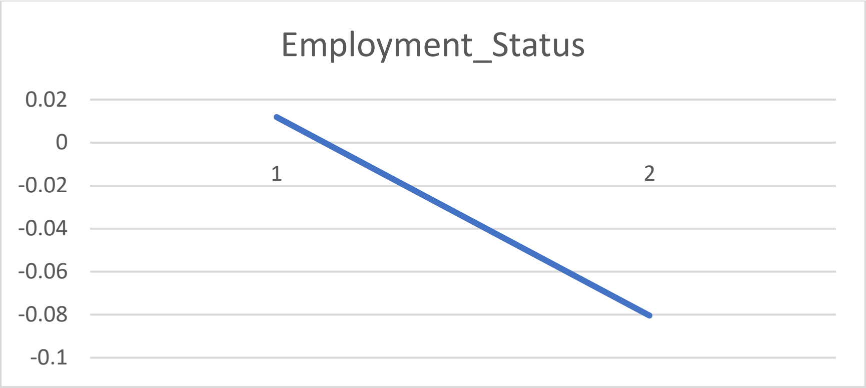
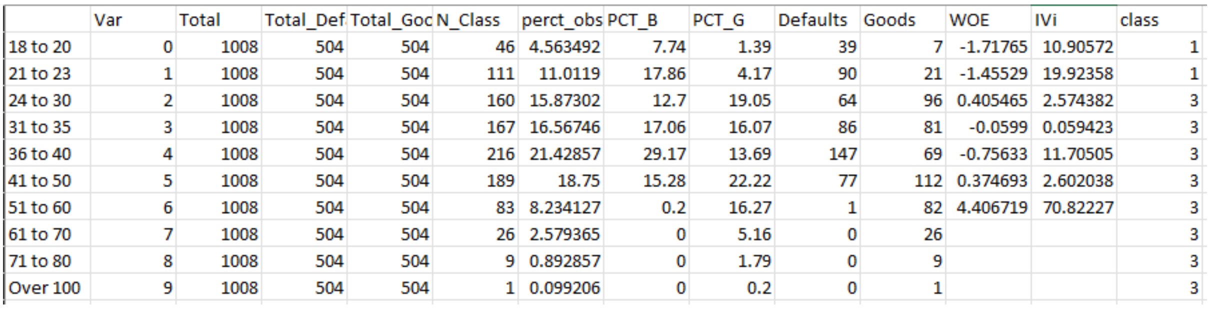
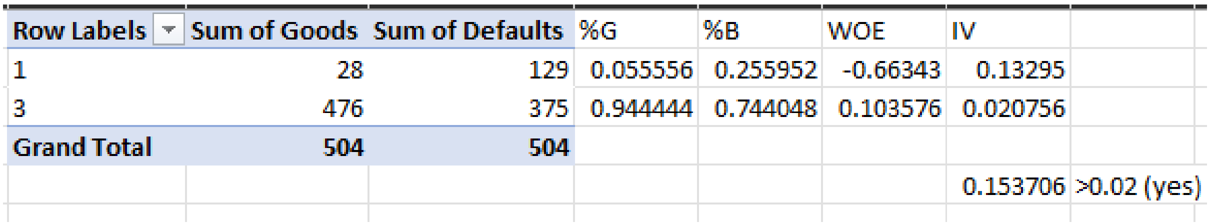


Figure 38

Age Range



	Var	Total	Total_Def	Total_Goc	N_Class	perct_obs	PCT_B	PCT_G	Defaults	Goods	WOE	IVi	class
18 to 20	0	1008	504	504	46	4.563492	7.74	1.39	39	7	-1.71765	10.90572	1
21 to 23	1	1008	504	504	111	11.0119	17.86	4.17	90	21	-1.45529	19.92358	1
24 to 30	2	1008	504	504	160	15.87302	12.7	19.05	64	96	0.405465	2.574382	3
31 to 35	3	1008	504	504	167	16.56746	17.06	16.07	86	81	-0.0599	0.059423	3
36 to 40	4	1008	504	504	216	21.42857	29.17	13.69	147	69	-0.75633	11.70505	3
41 to 50	5	1008	504	504	189	18.75	15.28	22.22	77	112	0.374693	2.602038	3
51 to 60	6	1008	504	504	83	8.234127	0.2	16.27	1	82	4.406719	70.82227	3
61 to 70	7	1008	504	504	26	2.579365	0	5.16	0	26			3
71 to 80	8	1008	504	504	9	0.892857	0	1.79	0	9			3
Over 100	9	1008	504	504	1	0.099206	0	0.2	0	1			3



Row Labels	Sum of Goods	Sum of Defaults	%G	%B	WOE	IV	
1	28	129	0.055556	0.255952	-0.66343	0.13295	
3	476	375	0.944444	0.744048	0.103576	0.020756	
Grand Total	504	504					
						0.153706	>0.02 (yes)

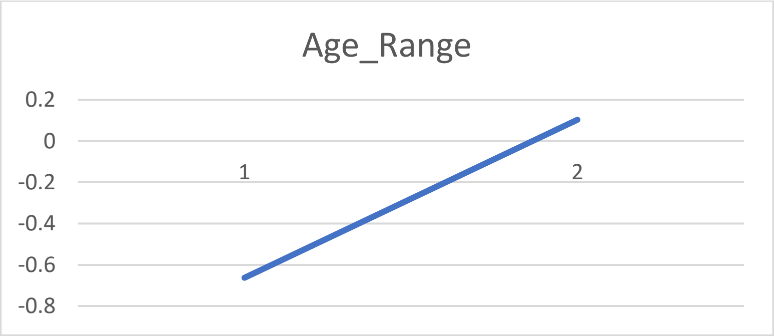
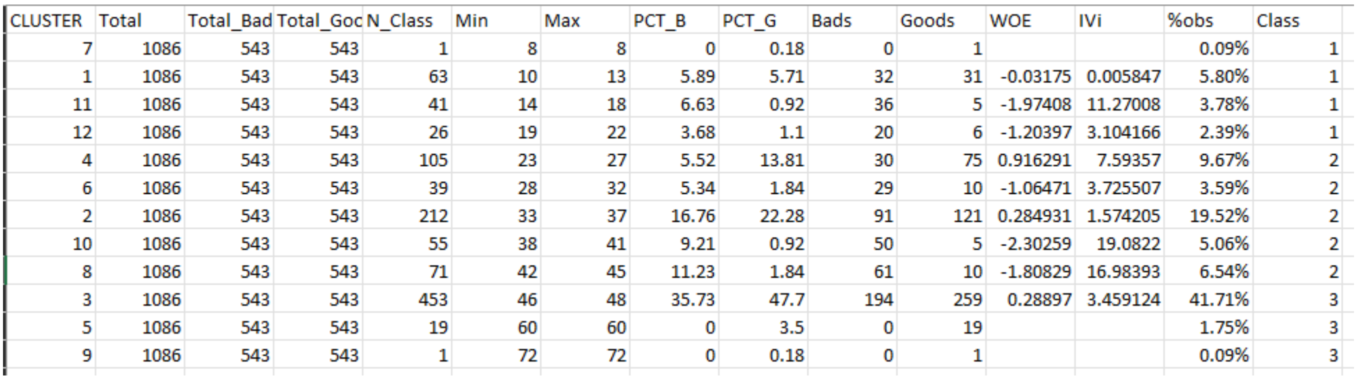
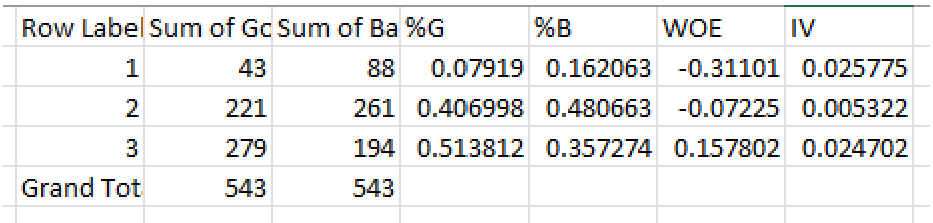


Figure 39

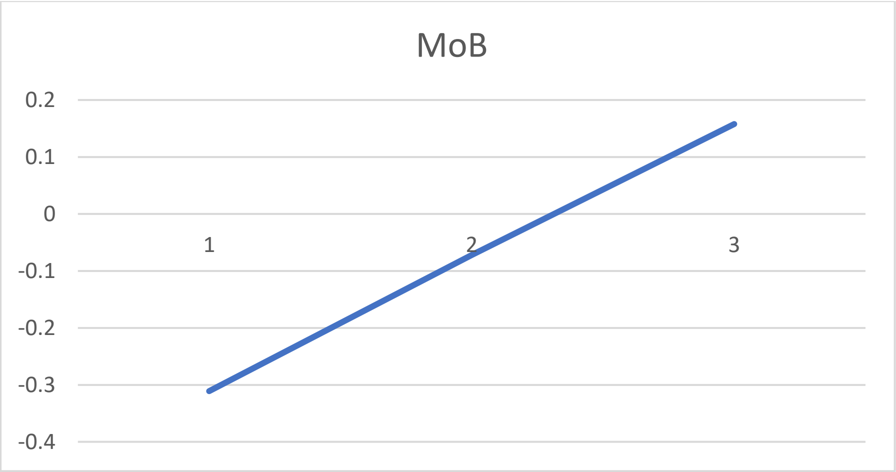
MOB



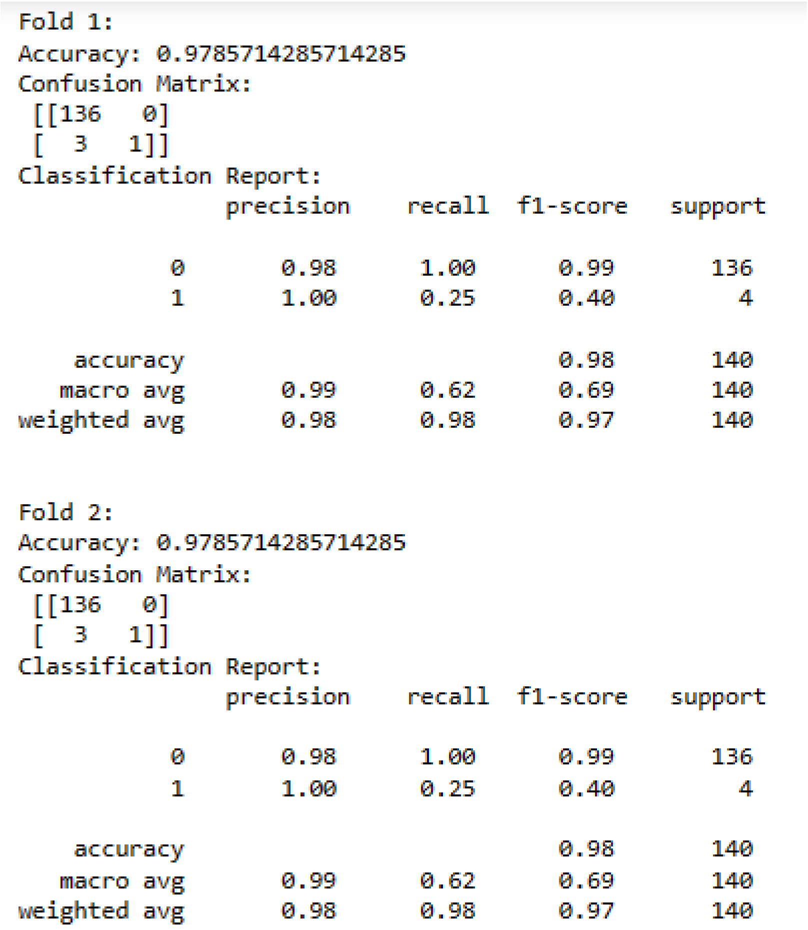
CLUSTER	Total	Total_Bad	Total_Goc	N_Class	Min	Max	PCT_B	PCT_G	Bads	Goods	WOE	IVi	%obs	Class
7	1086	543	543	1	8	8	0	0.18	0	1			0.09%	1
1	1086	543	543	63	10	13	5.89	5.71	32	31	-0.03175	0.005847	5.80%	1
11	1086	543	543	41	14	18	6.63	0.92	36	5	-1.97408	11.27008	3.78%	1
12	1086	543	543	26	19	22	3.68	1.1	20	6	-1.20397	3.104166	2.39%	1
4	1086	543	543	105	23	27	5.52	13.81	30	75	0.916291	7.59357	9.67%	2
6	1086	543	543	39	28	32	5.34	1.84	29	10	-1.06471	3.725507	3.59%	2
2	1086	543	543	212	33	37	16.76	22.28	91	121	0.284931	1.574205	19.52%	2
10	1086	543	543	55	38	41	9.21	0.92	50	5	-2.30259	19.0822	5.06%	2
8	1086	543	543	71	42	45	11.23	1.84	61	10	-1.80829	16.98393	6.54%	2
3	1086	543	543	453	46	48	35.73	47.7	194	259	0.28897	3.459124	41.71%	3
5	1086	543	543	19	60	60	0	3.5	0	19			1.75%	3
9	1086	543	543	1	72	72	0	0.18	0	1			0.09%	3



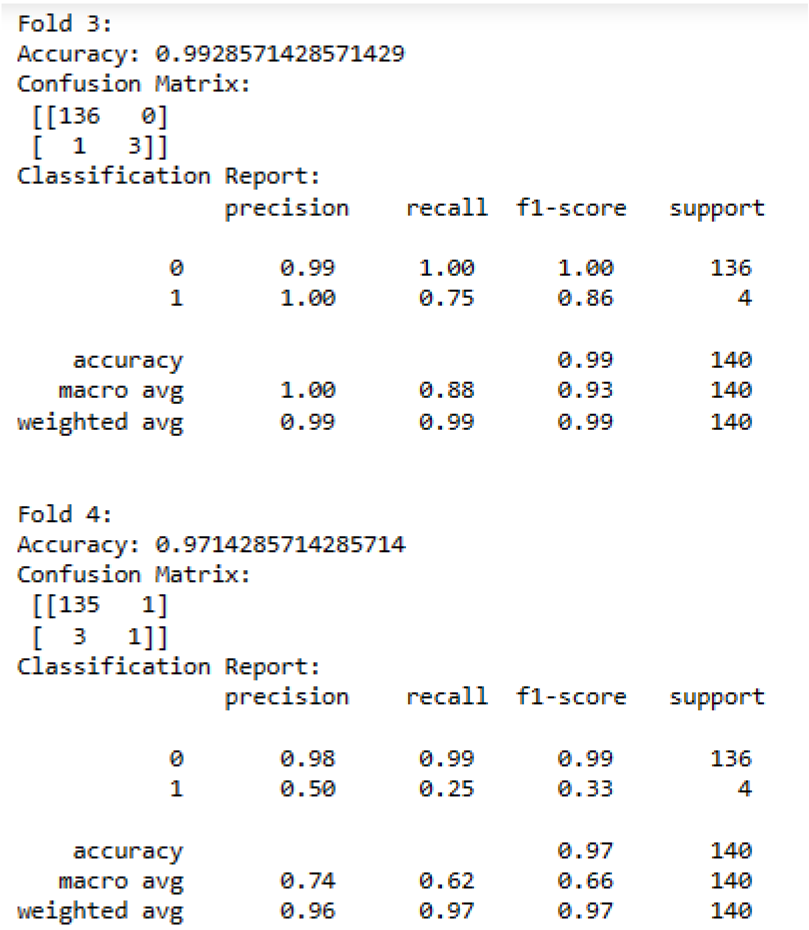
Row Label	Sum of Gc	Sum of Ba	%G	%B	WOE	IV
1	43	88	0.07919	0.162063	-0.31101	0.025775
2	221	261	0.406998	0.480663	-0.07225	0.005322
3	279	194	0.513812	0.357274	0.157802	0.024702
Grand Tot	543	543				



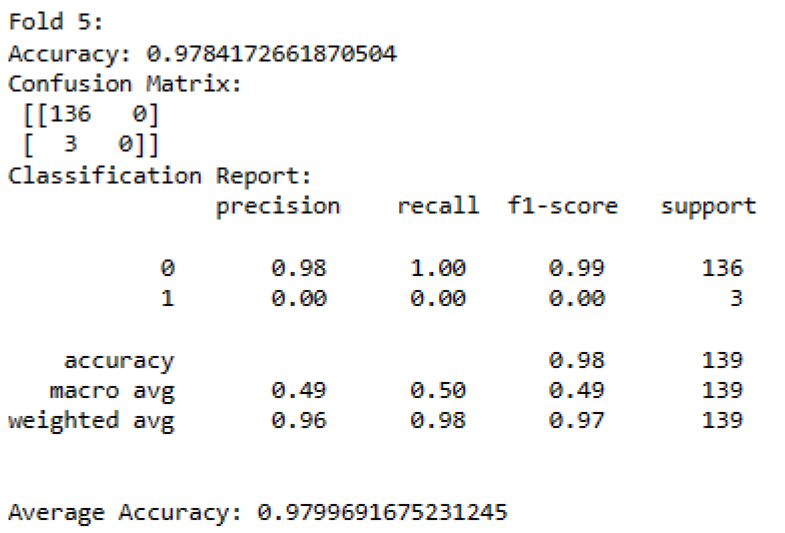
Random forest Accuracy



```
Fold 1:
Accuracy: 0.9785714285714285
Confusion Matrix:
 [[136   0]
 [  3   1]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99       136
           1       1.00      0.25      0.40         4

    accuracy                           0.98       140
   macro avg       0.99      0.62      0.69       140
weighted avg       0.98      0.98      0.97       140


Fold 2:
Accuracy: 0.9785714285714285
Confusion Matrix:
 [[136   0]
 [  3   1]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99       136
           1       1.00      0.25      0.40         4

    accuracy                           0.98       140
   macro avg       0.99      0.62      0.69       140
weighted avg       0.98      0.98      0.97       140
```



```
Fold 3:
Accuracy: 0.9928571428571429
Confusion Matrix:
 [[136   0]
 [  1   3]]
Classification Report:
               precision    recall  f1-score   support

           0       0.99      1.00      1.00       136
           1       1.00      0.75      0.86         4

    accuracy                           0.99       140
   macro avg       1.00      0.88      0.93       140
weighted avg       0.99      0.99      0.99       140


Fold 4:
Accuracy: 0.9714285714285714
Confusion Matrix:
 [[135   1]
 [  3   1]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.99      0.99       136
           1       0.50      0.25      0.33         4

    accuracy                           0.97       140
   macro avg       0.74      0.62      0.66       140
weighted avg       0.96      0.97      0.97       140
```

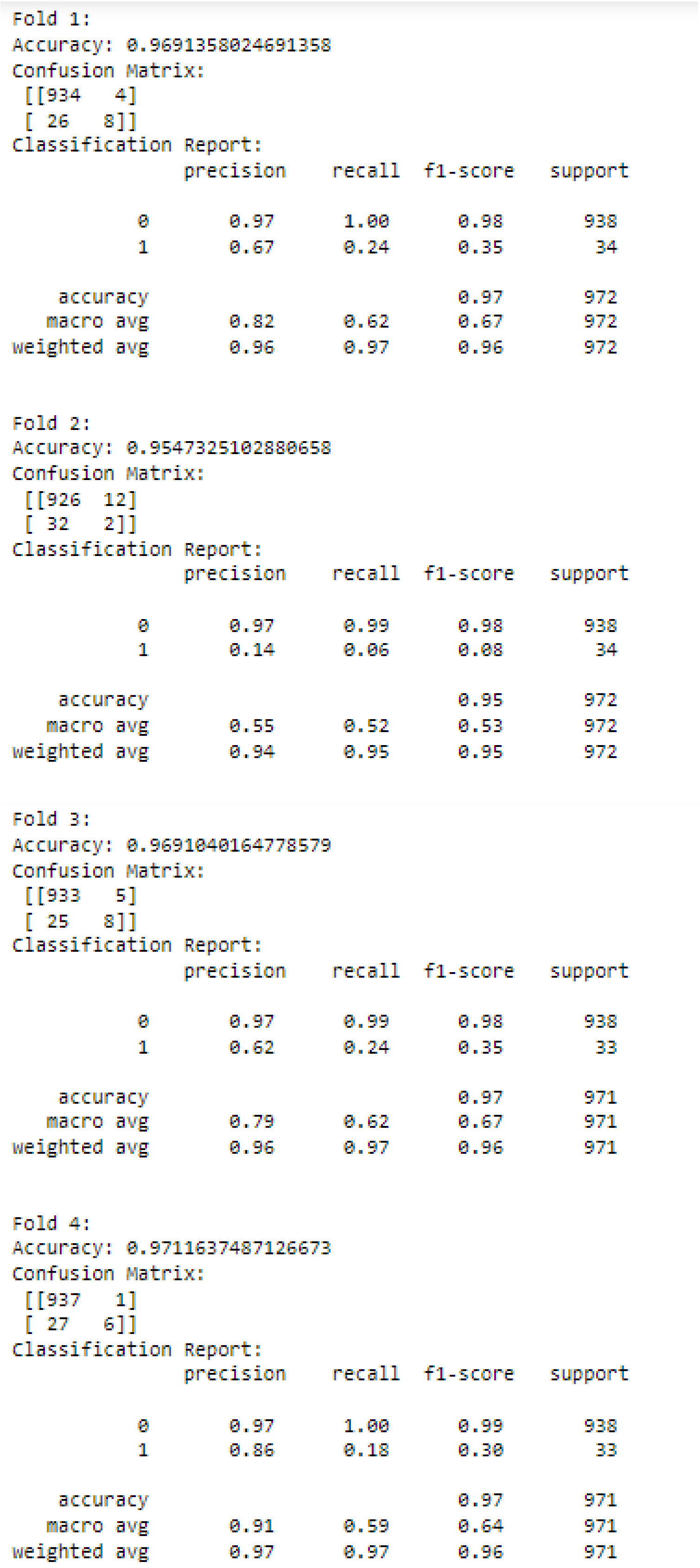


```
Fold 5:
Accuracy: 0.9784172661870504
Confusion Matrix:
 [[136   0]
 [  3   0]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99       136
           1       0.00      0.00      0.00         3

    accuracy                           0.98       139
   macro avg       0.49      0.50      0.49       139
weighted avg       0.96      0.98      0.97       139


Average Accuracy: 0.9799691675231245
```

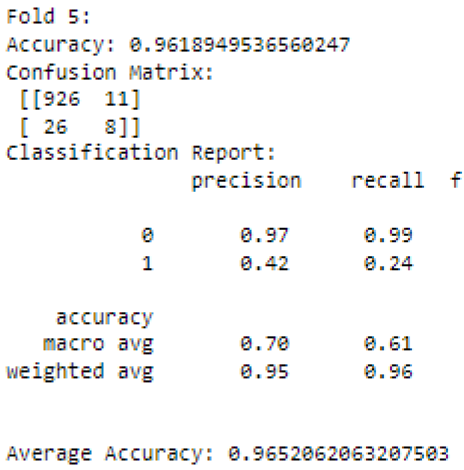
14. APPENDIX C

(Delinquency)

5-Fold Cross validation for Logistic Regression Model

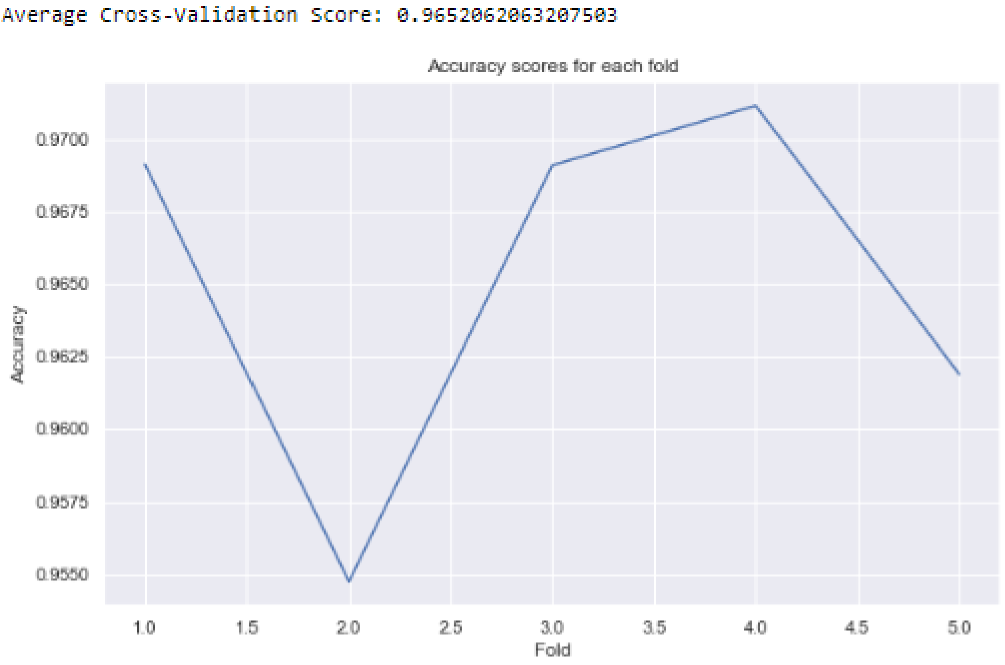


```
Fold 1:
Accuracy: 0.9691358024691358
Confusion Matrix:
 [[934   4]
 [ 26   8]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       938
           1       0.67      0.24      0.35        34

    accuracy                           0.97       972
   macro avg       0.82      0.62      0.67       972
weighted avg       0.96      0.97      0.96       972


Fold 2:
Accuracy: 0.9547325102880658
Confusion Matrix:
 [[926  12]
 [ 32   2]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       938
           1       0.14      0.06      0.08        34

    accuracy                           0.95       972
   macro avg       0.55      0.52      0.53       972
weighted avg       0.94      0.95      0.95       972


Fold 3:
Accuracy: 0.9691040164778579
Confusion Matrix:
 [[933   5]
 [ 25   8]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       938
           1       0.62      0.24      0.35        33

    accuracy                           0.97       971
   macro avg       0.79      0.62      0.67       971
weighted avg       0.96      0.97      0.96       971


Fold 4:
Accuracy: 0.9711637487126673
Confusion Matrix:
 [[937   1]
 [ 27   6]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       938
           1       0.86      0.18      0.30        33

    accuracy                           0.97       971
   macro avg       0.91      0.59      0.64       971
weighted avg       0.97      0.97      0.96       971
```



```
Fold 5:
Accuracy: 0.9618949536560247
Confusion Matrix:
 [[926  11]
 [ 26   8]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       937
           1       0.42      0.24      0.30        34

    accuracy                           0.96       971
   macro avg       0.70      0.61      0.64       971
weighted avg       0.95      0.96      0.96       971


Average Accuracy: 0.9652062063207503
```

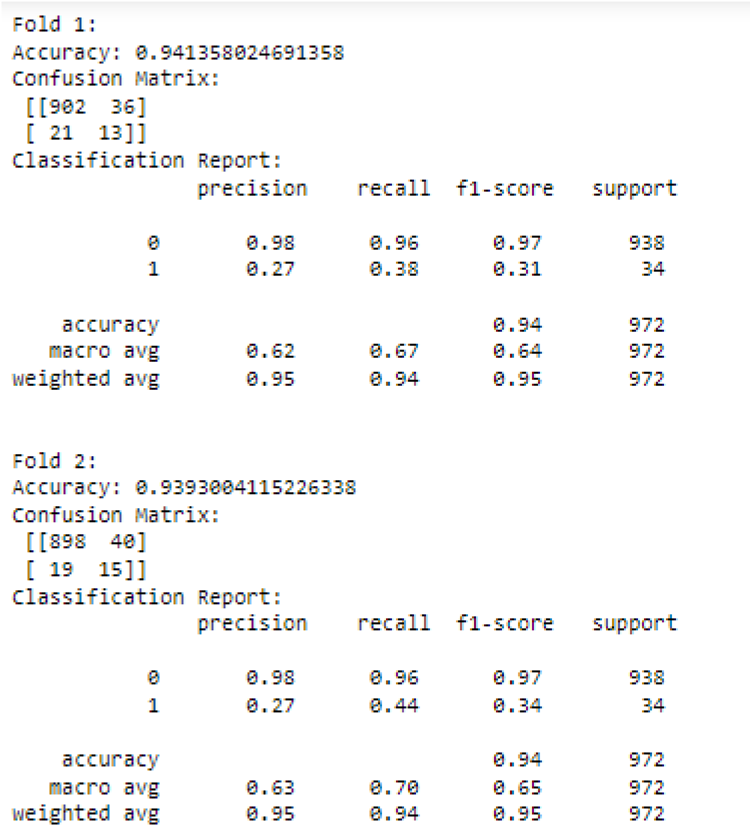
Figure 40



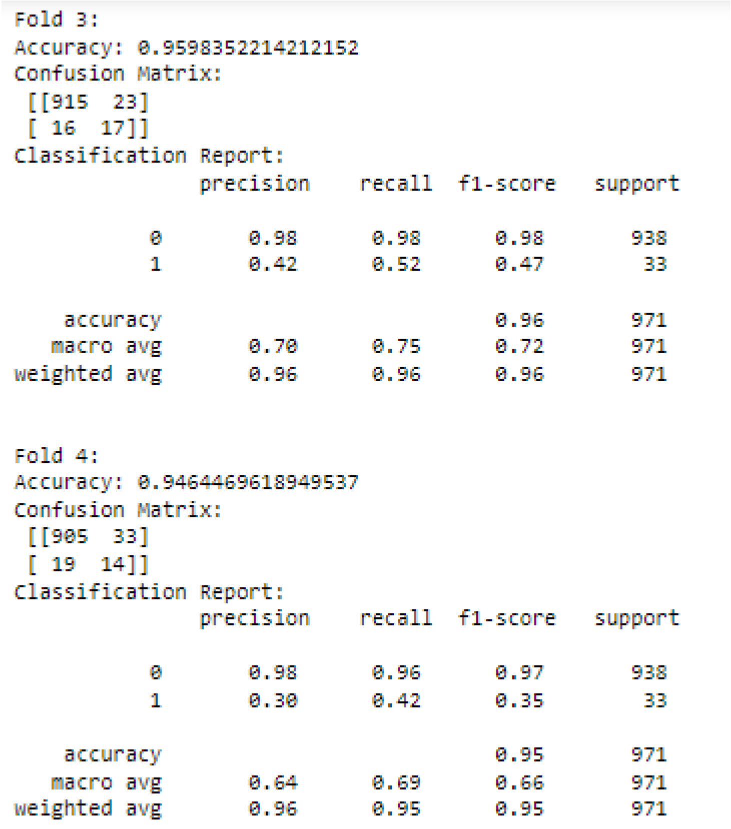
```
Average Cross-Validation Score: 0.9652062063207503
```

Figure 41

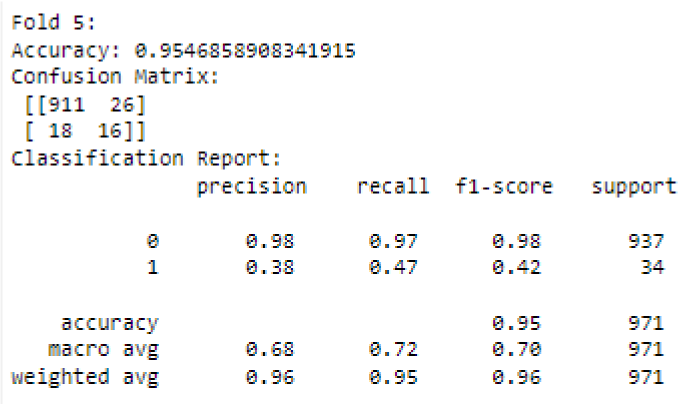
5-Fold Cross validation for Decision Tree Model



```
Fold 1:
Accuracy: 0.941358024691358
Confusion Matrix:
 [[902  36]
 [ 21  13]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.96      0.97       938
           1       0.27      0.38      0.31        34

    accuracy                           0.94       972
   macro avg       0.62      0.67      0.64       972
weighted avg       0.95      0.94      0.95       972


Fold 2:
Accuracy: 0.9393004115226338
Confusion Matrix:
 [[898  40]
 [ 19  15]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.96      0.97       938
           1       0.27      0.44      0.34        34

    accuracy                           0.94       972
   macro avg       0.63      0.70      0.65       972
weighted avg       0.95      0.94      0.95       972
```

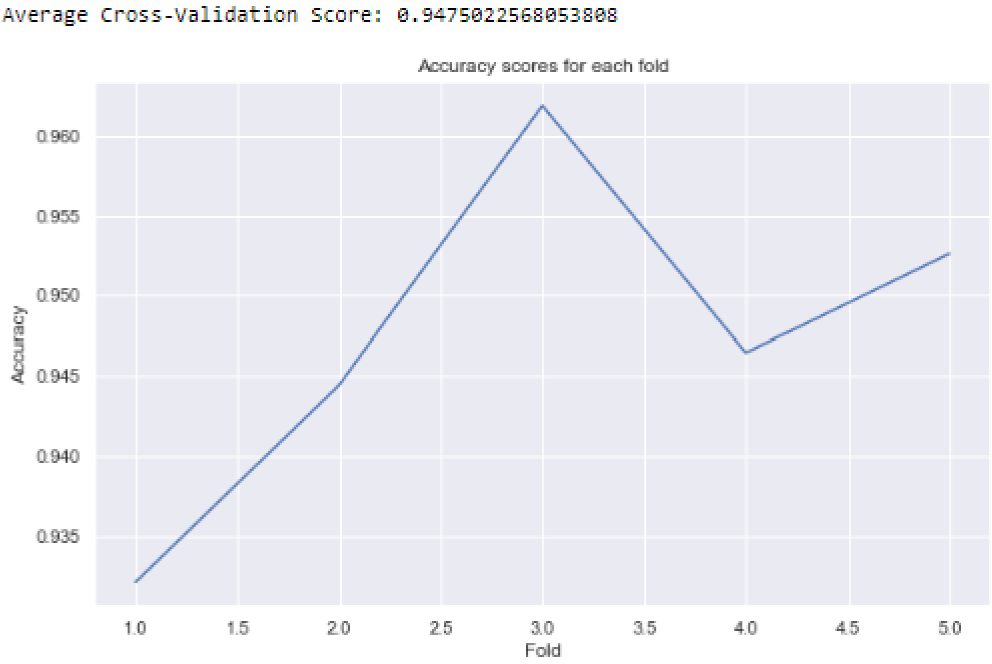


```
Fold 3:
Accuracy: 0.9598352214212152
Confusion Matrix:
 [[915  23]
 [ 16  17]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.98      0.98       938
           1       0.42      0.52      0.47        33

    accuracy                           0.96       971
   macro avg       0.70      0.75      0.72       971
weighted avg       0.96      0.96      0.96       971


Fold 4:
Accuracy: 0.9464469618949537
Confusion Matrix:
 [[905  33]
 [ 19  14]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.96      0.97       938
           1       0.30      0.42      0.35        33

    accuracy                           0.95       971
   macro avg       0.64      0.69      0.66       971
weighted avg       0.96      0.95      0.95       971
```



```
Fold 5:
Accuracy: 0.9546858908341915
Confusion Matrix:
 [[911  26]
 [ 18  16]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.97      0.98       937
           1       0.38      0.47      0.42        34

    accuracy                           0.95       971
   macro avg       0.68      0.72      0.70       971
weighted avg       0.96      0.95      0.96       971
```

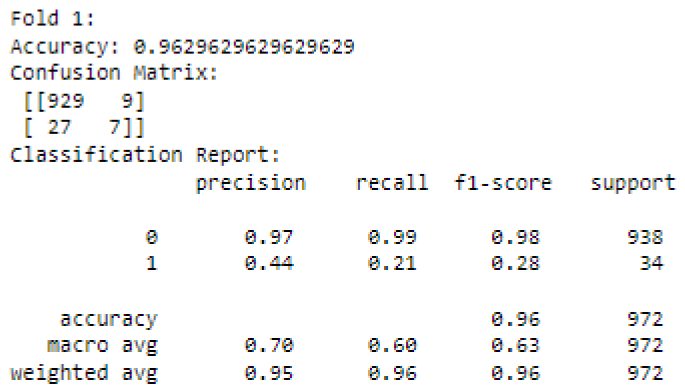
Figure 42



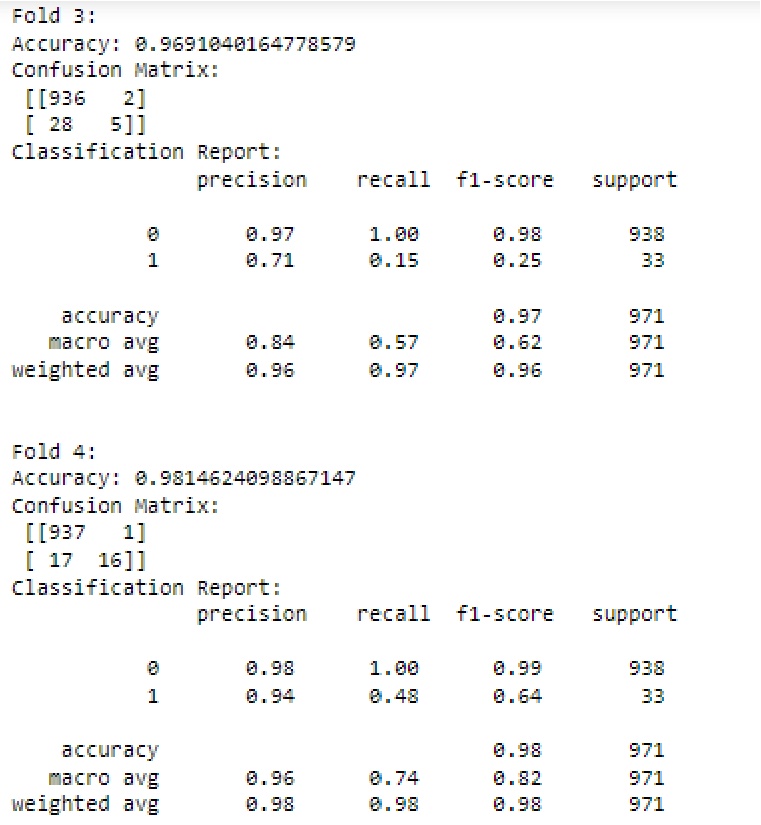
```
Average Cross-Validation Score: 0.9475022568053808
```

Figure 43

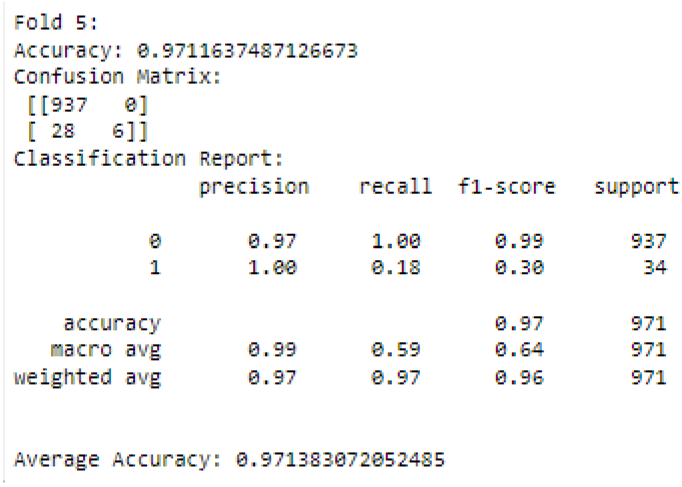
5-Fold Cross validation for Random Forest Tree Model



```
Fold 1:
Accuracy: 0.9629629629629629
Confusion Matrix:
 [[929   9]
 [ 27   7]]
Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.99      0.98       938
           1       0.44      0.21      0.28        34

    accuracy                           0.96       972
   macro avg       0.70      0.60      0.63       972
weighted avg       0.95      0.96      0.96       972


Fold 2:
Accuracy: 0.9722222222222222
Confusion Matrix:
 [[934   4]
 [ 23  11]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99       938
           1       0.73      0.32      0.45        34

    accuracy                           0.97       972
   macro avg       0.85      0.66      0.72       972
weighted avg       0.97      0.97      0.97       972
```

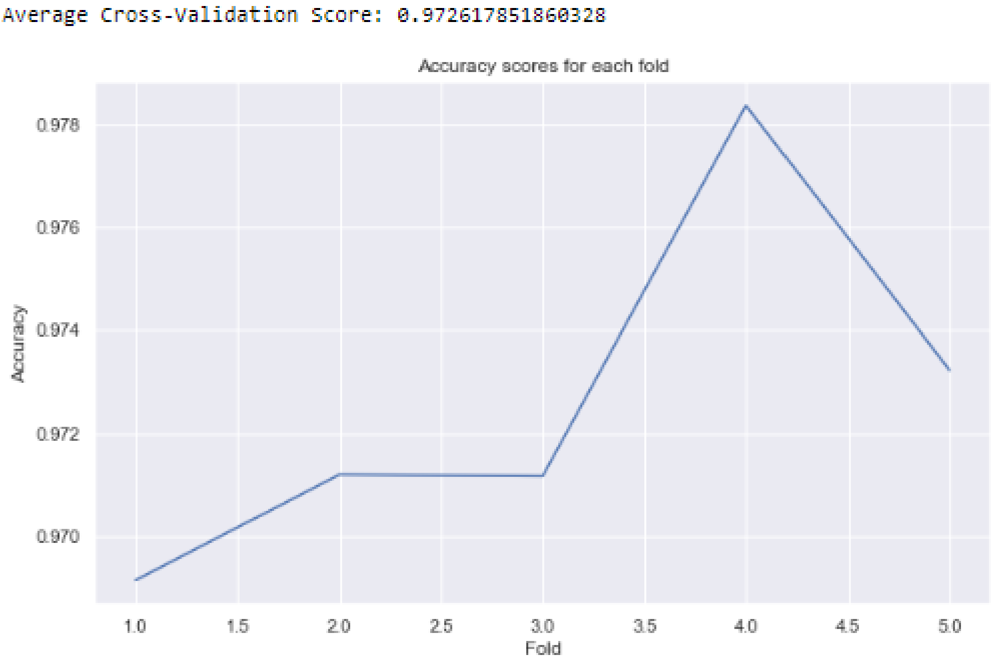


```
Fold 3:
Accuracy: 0.9691040164778579
Confusion Matrix:
 [[936   2]
 [ 28   5]]
Classification Report:
               precision    recall  f1-score   support

           0       0.97      1.00      0.98       938
           1       0.71      0.15      0.25        33

    accuracy                           0.97       971
   macro avg       0.84      0.57      0.62       971
weighted avg       0.96      0.97      0.96       971


Fold 4:
Accuracy: 0.9814624098867147
Confusion Matrix:
 [[937   1]
 [ 17  16]]
Classification Report:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99       938
           1       0.94      0.48      0.64        33

    accuracy                           0.98       971
   macro avg       0.96      0.74      0.82       971
weighted avg       0.98      0.98      0.98       971
```



```
Fold 5:
Accuracy: 0.9711637487126673
Confusion Matrix:
 [[937   0]
 [ 28   6]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       937
           1       1.00      0.18      0.30        34

    accuracy                           0.97       971
   macro avg       0.99      0.59      0.64       971
weighted avg       0.97      0.97      0.96       971


Average Accuracy: 0.971383072052485
```

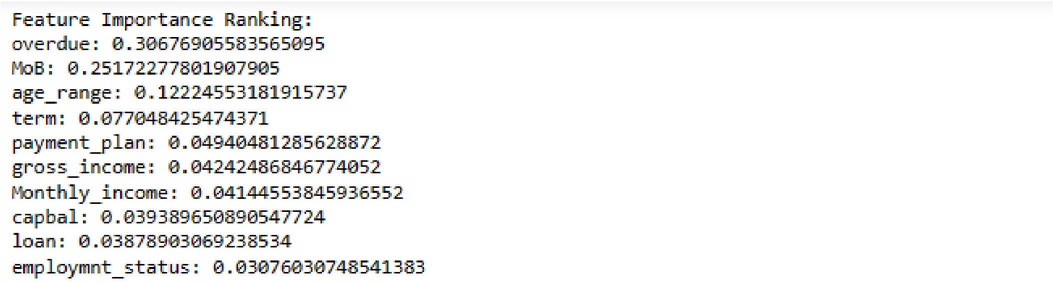
Figure 44



```
Average Cross-Validation Score: 0.972617851860328
```

Figure 45

Graph with Imbalance data



```
Feature Importance Ranking:
overdue: 0.30676905583565095
MoB: 0.25172277801907905
age_range: 0.12224553181915737
term: 0.077048425474371
payment_plan: 0.04940481285628872
gross_income: 0.04242486846774052
Monthly_income: 0.04144553845936552
capbal: 0.039389650890547724
loan: 0.03878903069238534
employmnt_status: 0.03076030748541383
```

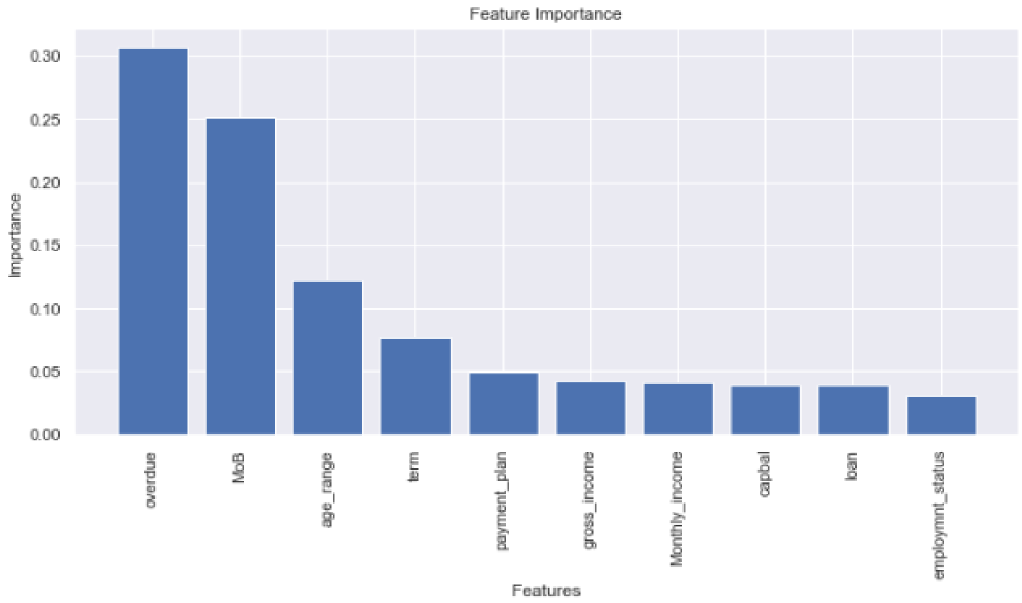
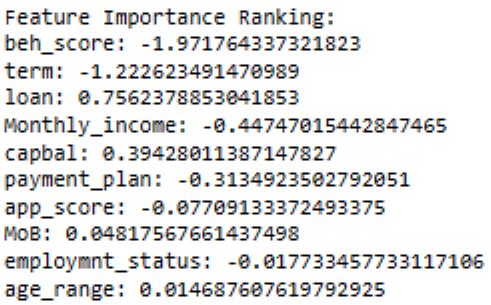


Figure 46

Feature Importance using new dataset for Logistic regression Model.



```
Feature Importance Ranking:
beh_score: -1.971764337321823
term: -1.222623491470989
loan: 0.7562378853041853
Monthly_income: -0.44747015442847465
capbal: 0.39428011387147827
payment_plan: -0.3134923502792051
app_score: -0.07709133372493375
MoB: 0.04817567661437498
employmnt_status: -0.017733457733117106
age_range: 0.014687607619792925
```

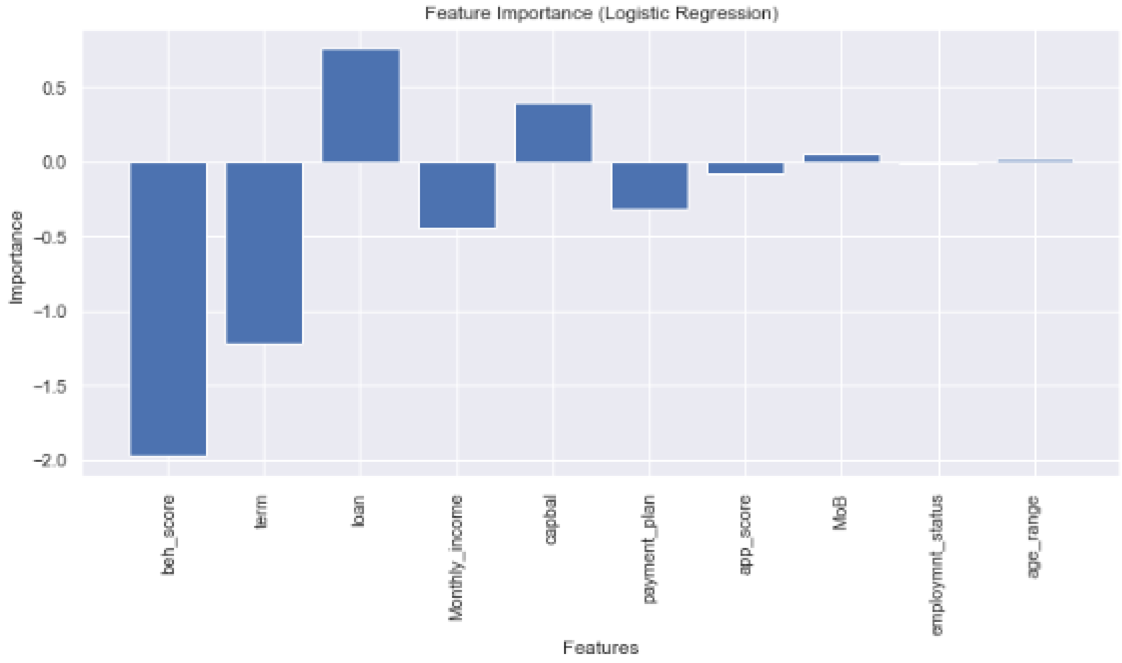
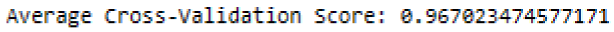


Figure 47

5-fold cross validation



```
Average Cross-Validation Score: 0.967023474577171
```

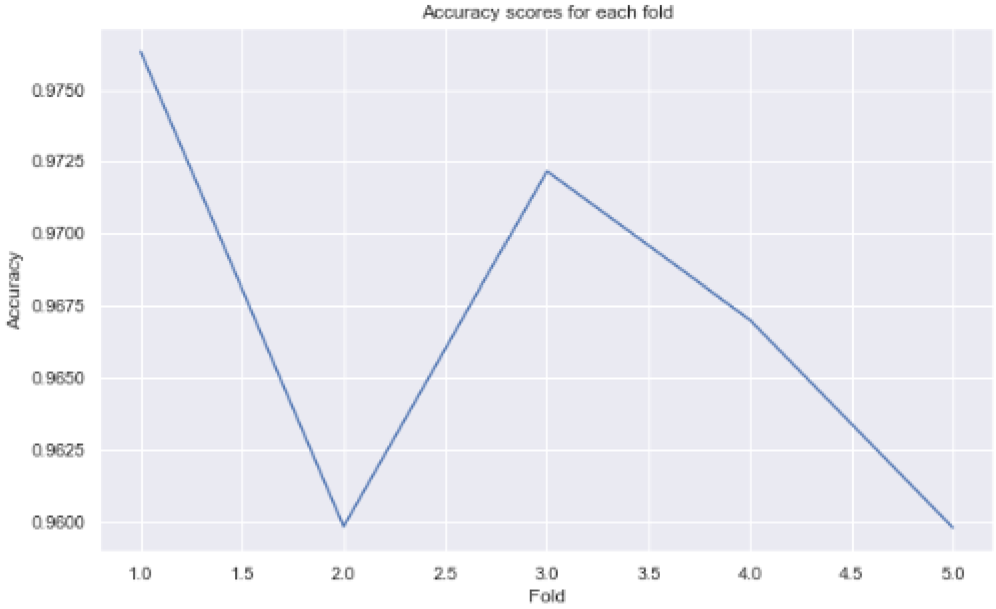
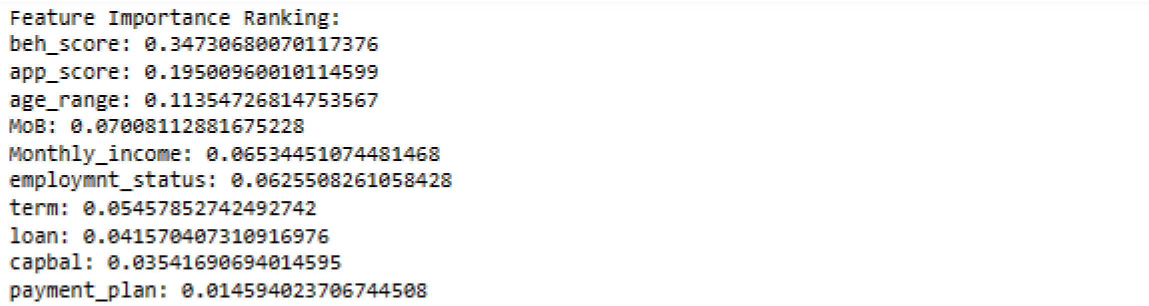


Figure 48

Decision Tree



```
Feature Importance Ranking:
beh_score: 0.34730680070117376
app_score: 0.19500960010114599
age_range: 0.11354726814753567
MoB: 0.07008112881675228
Monthly_income: 0.06534451074481468
employmnt_status: 0.0625508261058428
term: 0.05457852742492742
loan: 0.041570407310916976
capbal: 0.03541690694014595
payment_plan: 0.014594023706744508
```

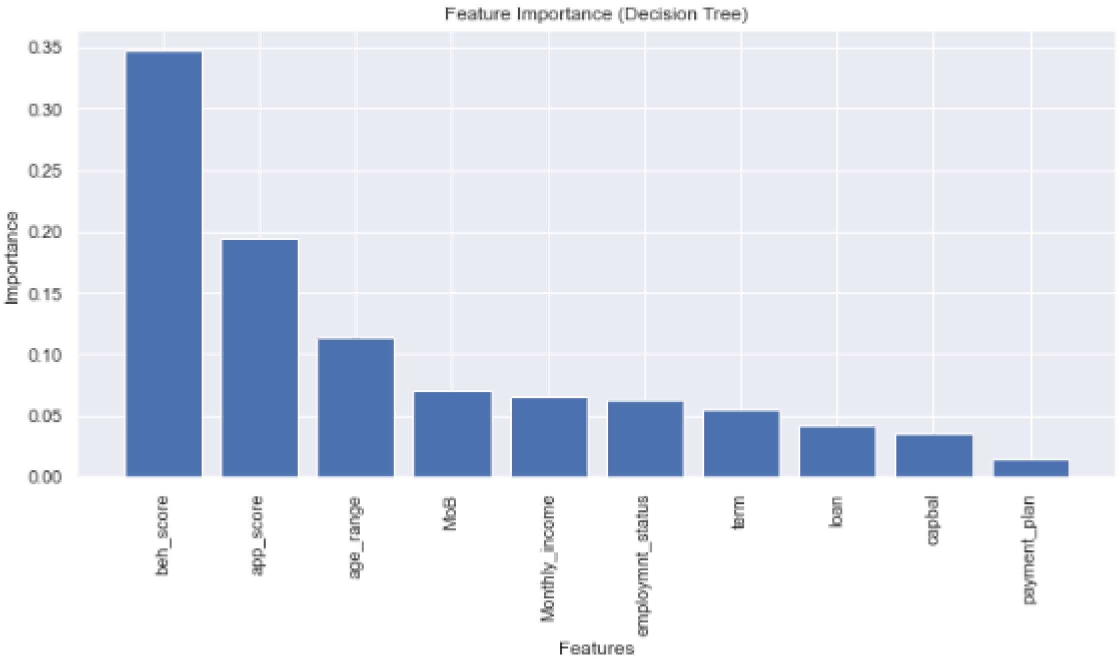
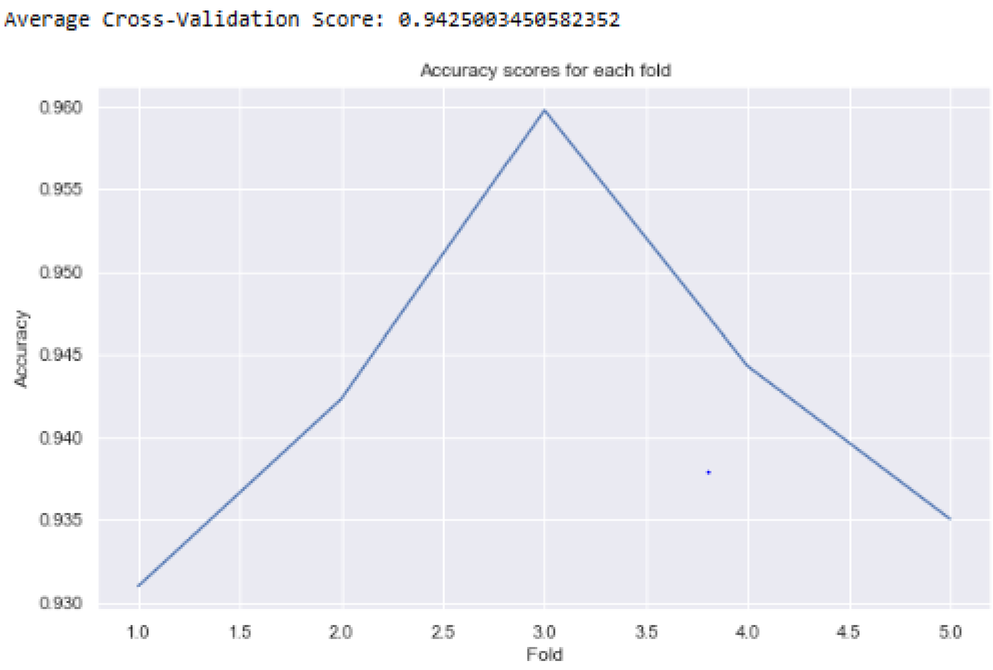


Figure 49

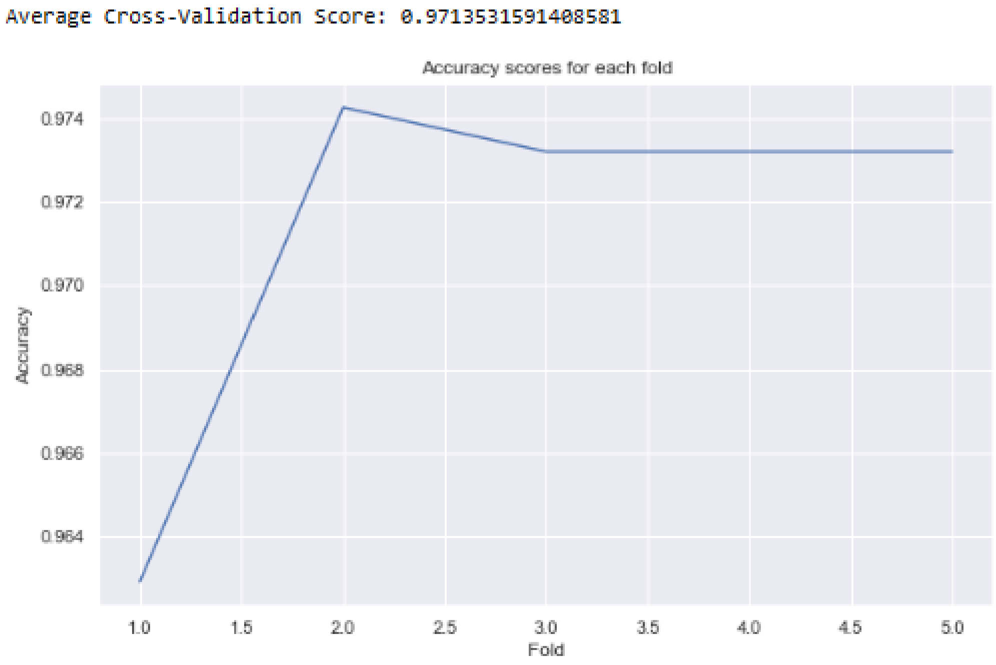
5-fold cross validation



```
Average Cross-Validation Score: 0.9425003450582352
```

Figure 50

Random Forest 5-fold Cross validation



```
Average Cross-Validation Score: 0.9713531591408581
```

Figure 51

15. APPENDIX D

(Code)

/* proc import */

```
/*                    datafile='/home/u63249191/Dissertation project/Data/sample.xlsx'  */
/*                    out=sam1 dbms=xlsx; */
/* run; */

/* proc import  */
/*                    datafile='/home/u63249191/Dissertation
project/Data/Data_behav/Data_Prep_MoB.xlsx'  */
/*                    out=sam1 dbms=xlsx; */
/* run; */

/* ---------------------- EDA -------------------------------- */
/* monthly income ratio */
/* debttoincome ratio */
/* Delinquency using SAS */
data sam1;
set MYDATA.Final_MoB_data;
run;

/* Removing Duplicates*/
proc sort data=sam1 out=nodup_emptable nodupkey
                    dupout=duplicate_table;
          by _all_;
run;

/* Missing Values*/
ods graphics on;
ods noproctitle;
title "Missing Values";

proc means data=nodup_emptable n nmiss missing;
          var snapshotdt key term app_score beh_score opendt_mth opendt_year woff_mth
woff_year woff_amount gross_income;
class status;
run;
```

```
proc print data=nodup_emptable (obs=5);
run;

/* Summary Statistics*/
ods graphics on;
ods noproctitle;
title "Summary Statistics";

proc means data=nodup_emptable mean median std min max n;
        var snapshotdt term app_score beh_score  woff_mth
woff_year woff_amount;
run;
title;

ods graphics / reset width=5.4in height=3.8in imagemap;
/* Observing Unique values*/
Title1 "Observing Unique values";

proc freq data=nodup_emptable;
        table status Employmnt_Status overdue capbal gross_income age_range loan category
payment_plan;
        label status="Number of Defaults" Employmnt_Status="Distribution of Employmnt_Status"
                overdue="Distribution of overdue" capbal="Distribution of Account Balance"
                gross_income="Distribution of Gross_Income"
                age_range="Distribution of Age Range"
                loan="Distribution of loan" category="Distribution of Delienquency Period"
                payment_plan= "Distribution of Payment Plan";
run;

Title;

/*  Outliers in Numerical Values */
ods graphics / reset width=5.4in height=3.8in imagemap;

%macro var_boxplot(data=, var=);
```

```
        proc sgplot data=&data;
                vbox &var / whiskerextent=1.5 outlierattrs=(symbol=CircleFilled size=10);
                yaxis grid;
                ods output outlier=extremes;
        run;

%mend;

%var_boxplot(data=nodup_emptable, var=term);
%var_boxplot(data=nodup_emptable, var=app_score);
%var_boxplot(data=nodup_emptable, var=beh_score);

/*  Count of snapshot for each key*/
proc sort data=nodup_emptable;
        by snapshotdt key;
run;

/* Look for some pattern in bad records */
data bads;
set nodup_emptable;
if status = 'B' then output;
run;

/* ----------------------- Data Transformation --------------------------------- */

/* Converting Category variable into numeric format */
/* After below step num_category becomes my main delienquency stage variable */
data cate_check;
set nodup_emptable;
if category = "Current" then category_2="0";
   else if category = "6+" then category_2="7";
   else if category = "Closed" then category_2= "";
       else category_2 =category;
num_category = input(category_2,2.);
run;
```

```
/*This is the last status - This will give you the "G" or "B" status*/
proc sql;
create table key_counts as
select key, count(snapshotdt) as SnapshotCount
from cate_check
group by key;
quit;

/* They have to have atleast 6 month of observation */
data keys_with_6_or_more;
set key_counts;
where SnapshotCount >= 6;
run;

/* Merge SnapshotCount column with the cate_check for counting total number of snaps in each key
*/
proc sql;
create table merged_data as
select a.*, b.SnapshotCount
from cate_check as a
right join keys_with_6_or_more as b
on a.key = b.key;
quit;

/* The 6 months of observation has to be between 34 to 40 */
DATA Between_34_to_40;
  SET merged_data;
  if 34 <= snapshotdt <= 40; * 6 months of obs;
RUN;

proc sort data=Between_34_to_40  out=Last_6_obs;
by key snapshotdt;
run;
```

```
data last_obs;
set Last_6_obs;
by key;
if last.key then output;
run;

/*this is to enable to take the first and last observations per account(key)*/
proc sort data=Between_34_to_40 out=sample_sorted;
by key snapshotdt;
run;

/*This will give all the variables when the loan was written (booked)
 at the point of the origination --- Compare category stage between firt_obs with last_obs*/
data first_obs;
set sample_sorted;
by key;
if first.key then output;
run;

proc freq data= last_obs;
table status;
run;

/*This will give you the worst arrears status by key (account)*/
/* proc freq data=last_obs;  */
/*          table category * status/missing nopercent nocol norow; */
/* run; */

proc sort data=cate_check  out=arrears_sorted;
by key num_category;
run;

/*This will give you the last observation per account (key_id) as it is sorted
in order of arrears (num_category)
*/
```

```
data max_arrears;
set arrears_sorted;
by key;
if last.key then output;
run;

/*This gives 1 observation (row) per account (key)
Most of the information is from the first observation
We also have new columns taken from 2 different dateset these include
Max ever arrears And the last status. If it ever defaulted then it will capture the "B" here.*/

proc sql;
create table First_obs_with_max_arrears as
select a. *
        , b. category as last_category
        , b. status as last_status_good_or_bad
        , c. num_category as max_arrears_status

/*This is now my base data. 1 observation per account (key)*/
from first_obs   as a

left join last_obs as b
on a. key = b. key

left join max_arrears as c
on a. key = c. key
;
quit;

proc freq data= First_obs_with_max_arrears;
table last_status_good_or_bad;
run;

/* Sample specific for Sampling & NOT for EDA */
/* Removing all closed accounts and Loan = 0 */
```

```
DATA No_closedacc;
  SET First_obs_with_max_arrears;
  IF last_status_good_or_bad = 'C' THEN Delete;
  else if loan = 0 then delete;
RUN;

data Goo_bad_only;
set No_closedacc;
/* Set the new default status based on the last status */
IF last_status_good_or_bad = "G" Then new_default_status = 0;
else if last_status_good_or_bad = "B" then new_default_status = 1;
run;

/* Instead of default we have to say 60 days past due */
data New_def_status;
/* This data step calculates the new default status */
set Goo_bad_only (keep=snapshotdt key term loan capbal overdue gross_income payment_plan
age_range employmnt_status MoB max_arrears_status new_default_status);
/* Set the new default status based on the maximum arrears status and the last status */
IF max_arrears_status > 2 THEN new_default_status = 1; * 60 days past due;
else if max_arrears_status = 2 THEN delete;
else if max_arrears_status <= 1 THEN new_default_status = 0;
Monthly_income = gross_income/12;
run;

proc freq data= New_def_status;
table new_default_status;
run;

/* ---------------------- Data visualisation -------------------------------- */

ods graphics / reset width=4.4in height=3.8 in imagemap;
/* status Employmnt_Status overdue capbal gross_income age_range loan category */
/*Univaraite Analysis for Numerical data */
```

```
/* Explain using the table it is actually normally distributed */
proc univariate data=nodup_emptable; * will use new variable for EDA;
	var snapshotdt key term app_score beh_score opendt_mth opendt_year woff_mth
woff_year woff_amount;
	hist;
run;

/*Univaraite Analysis for Categorical data */
proc freq data=nodup_emptable;
	tables status Employmnt_Status overdue capbal gross_income age_range loan category /
		plots=(freqplot);
run;

/* Only for EDA  */
DATA No_cloe_EDA;
  SET nodup_emptable;
  IF status = 'C' THEN Delete;
RUN;

/*Bivaraite Analysis for Numerical data */
proc univariate data=No_cloe_EDA;
	var snapshotdt term app_score woff_amount;
	class status;
	hist;
run;



/*Bivaraite Analysis for Categorical data */

/* Employmnt_Status */
proc freq data=cate_check;
	table Employmnt_Status * category/missing nopercent nocol norow plots=(freqplot);
run;

proc freq data=New_def_status;
```

```
        tables Employmnt_Status overdue capbal gross_income age_range loan term/
                plots=(freqplot);
run;

proc freq data=cate_check;
        table Employmnt_Status * status/missing nopercent nocol norow plots=(freqplot);
run;

/* overdue */
proc freq data=cate_check;
        table overdue * category/missing nopercent nocol norow plots=(freqplot);
run;

proc freq data=cate_check;
        table overdue * status/missing nopercent nocol norow plots=(freqplot);
run;

/* Account Balance */
proc freq data=cate_check;
        table capbal * category/missing nopercent nocol norow plots=(freqplot);
run;

proc freq data=cate_check;
        table capbal * status/missing nopercent nocol norow plots=(freqplot);
run;

/* Gross Income */
proc freq data=cate_check;
        table gross_income * category/missing nopercent nocol norow plots=(freqplot);
run;

proc freq data=cate_check;
        table gross_income * status/missing nopercent nocol norow plots=(freqplot);
run;
```

```
/* Age Range */
proc freq data=cate_check;
	table age_range * category/missing nopercent nocol norow plots=(freqplot);
run;

proc freq data=cate_check;
	table age_range * status/missing nopercent nocol norow plots=(freqplot);
run;

/* loan */
proc freq data=cate_check;
	table loan * category/missing nopercent nocol norow plots=(freqplot);
run;

proc freq data=cate_check;
	table loan * status/missing nopercent nocol norow plots=(freqplot);
run;

/* Comment On each stage of category */
/* proc freq data=cate_check;  */
/*	table category * category/missing nopercent nocol norow plots=(freqplot); */
/* run; */
/*  */
/* proc freq data=cate_check;  */
/*	table status * status/missing nopercent nocol norow plots=(freqplot); */
/* run; */



/* Data Description for the new sample */
proc freq data=First_obs_with_max_arrears;
	table max_arrears_status last_status_good_or_bad last_category;
run;

/* proc freq data=First_obs_with_max_arrears; */
/*	table max_arrears_status * last_status_good_or_bad/missing nocol nopercent norow; */
```

```
/* run; */
/*  */
/* proc freq data=cate_check; * why should we remove closed accounts; */
/*          table category * status/missing nopercent nocol norow; */
/* run; */
/*  */
/* proc freq data=New_def_status; */
/*          table loan*new_default_status; */
/* run; */

proc freq data=New_def_status;
            table new_default_status;
run;



/*  */
/* data train test; */
/*    set accepts_split; */
/*    if Selected = 1 then output train; */
/*    else output test; */
/* run; */

/* You may want to exclude num_category ="2" as these are neither good nor bad,
they are indeterminates (grey) */
/* neither black or white */



/*  Capbal taken from old one*/
/* IF capbal IN ('1-500') then WoE_capbal=0.363889277644613;Else */
/* IF capbal IN ('501-1000') then WoE_capbal=0.363889277644613;Else */
/* IF capbal IN ('1001-1500') then WoE_capbal=0.363889277644613;Else */
/* IF capbal IN ('1501-2000') then WoE_capbal=-0.23571038846107;Else */
/* IF capbal IN ('2001-2500') then WoE_capbal=-0.23571038846107;Else */
/* IF capbal IN ('>2501') then WoE_capbal=-0.23571038846107; */
/*  */
/* IF loan IN ('1-250') then WoE_loan =0.690196080028514;Else */
/* IF loan IN ('251-500') then WoE_loan =0.690196080028514;Else */
/* IF loan IN ('501-750') then WoE_loan =0.0757207139381183;Else */
/* IF loan IN ('751-1000') then WoE_loan =0.0757207139381183;Else */
/* IF loan IN ('>1000') then WoE_loan =-0.116338564846382; */
/*  */
/* IF payment_plan IN ('1 to 25') then WoE_payment_plan=-0.0561457157068355;Else */
/* IF payment_plan IN ('26-50') then WoE_payment_plan=-0.0561457157068355;Else */
/* IF payment_plan IN ('51-75') then WoE_payment_plan=-0.0561457157068355;Else */
/* IF payment_plan IN ('76-100') then WoE_payment_plan=-0.0561457157068355;Else */
/* IF payment_plan IN ('>100') then WoE_payment_plan=1.79239168949825; */
/*  */
/* IF gross_income IN ('2500') then WoE_gross_income=-0.421852166077449;Else */
/* IF gross_income IN ('6250') then WoE_gross_income=-0.421852166077449;Else */
/* IF gross_income IN ('8750') then WoE_gross_income=-0.421852166077449;Else */
/* IF gross_income IN ('14000') then WoE_gross_income=-0.421852166077449;Else */
/* IF gross_income IN ('22500') then WoE_gross_income=-0.244676279530792;Else */
/* IF gross_income IN ('27500') then WoE_gross_income=-0.244676279530792;Else */
/* IF gross_income IN ('35000') then WoE_gross_income=0.91315131299984;Else */
/* IF gross_income IN ('45000') then WoE_gross_income=0.91315131299984;Else */
/* IF gross_income IN ('60000') then WoE_gross_income=0.91315131299984;Else */
/* IF gross_income IN ('85000') then WoE_gross_income=0.91315131299984; */
/*  */
/* IF age_range IN ('18 to 20') then WoE_age_range=-0.66343167895703;Else */
/* IF age_range IN ('21 to 23') then WoE_age_range=-0.66343167895703;Else */
/* IF age_range IN ('24 to 30') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('31 to 35') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('36 to 40') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('41 to 50') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('51 to 60') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('61 to 70') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('71 to 80') then WoE_age_range=0.103575684992774;Else */
/* IF age_range IN ('Over 100') then WoE_age_range=0.103575684992774; */
/* Model 2 */
/* IF overdue IN ('0') then WoE_overdue=0.246723125047417;Else */
/* IF overdue IN ('1 to 25') then WoE_overdue=0.246723125047417;Else */
/* IF overdue IN ('26-50') then WoE_overdue=-2.34044411484012;Else */
/* IF overdue IN ('>100') then WoE_overdue=-2.34044411484012; */
/*  */
/* IF employmnt_status IN ('Full Time') then WoE_employmnt_status=0.0118721146275662;Else */
/* IF employmnt_status IN ('Homemaker') then WoE_employmnt_status=0.0118721146275662;Else */
/* IF employmnt_status IN ('Other') then WoE_employmnt_status=0.0118721146275662;Else */
/* IF employmnt_status IN ('Retired') then WoE_employmnt_status=-0.0804063370769311;Else */
/* IF employmnt_status IN ('Self Employed') then WoE_employmnt_status=-0.0804063370769311; */
/*  */
/* IF term <31 then WoE_term =0.382851337877814;Else */
/* IF term >=31 then WoE_term=-0.0581259678032846; */

/* Using New variables & New dataset */

Data Behav_training_WOE;
        Set New_def_status;

IF term <23 then WoE_term =-0.311014216570582;Else
IF term >=23 AND term  <46 then WoE_term =-0.0722482336531702;Else
IF term >=46 then WoE_term =0.157802473343372;

IF Monthly_income <1824.25006 then WoE_Monthly_income=-0.449683006372723;Else
IF Monthly_income >=1824.25006 AND Monthly_income  <2758.234721 then WoE_Monthly_income =-0.228097698893358;Else
IF Monthly_income >=2758.234721 then WoE_Monthly_income =0.717519207239748;

IF MoB <10 then WoE_MoB=0.605920412141201;Else
IF MoB >=10 AND MoB  <29 then WoE_MoB =0.0631413371825595;Else
IF MoB >=29 then WoE_MoB =-0.299316800219861;



IF capbal IN ('1-500') then WoE_capbal=0.0210523192606008;Else
IF capbal IN ('501-1000') then WoE_capbal=0.0210523192606008;Else
IF capbal IN ('1001-1500') then WoE_capbal=0.0210523192606008;Else
IF capbal IN ('1501-2000') then WoE_capbal=-0.0279086034274675;Else
IF capbal IN ('2001-2500') then WoE_capbal=-0.0279086034274675;Else
IF capbal IN ('>2501') then WoE_capbal=-0.0279086034274675;

IF loan IN ('1-250') then WoE_loan =-0.0500182975093999;Else
IF loan IN ('251-500') then WoE_loan =-0.0500182975093999;Else
IF loan IN ('501-750') then WoE_loan =0.255272505103306;Else
IF loan IN ('751-1000') then WoE_loan =0.255272505103306;Else
IF loan IN ('>1000') then WoE_loan =0.255272505103306;

IF overdue IN ('0') then WoE_overdue=0.301029995663981;Else
IF overdue IN ('1 to 25') then WoE_overdue=-2.4345689040342;Else
IF overdue IN ('26-50') then WoE_overdue=-2.4345689040342;Else
IF overdue IN ('>100') then WoE_overdue=-2.4345689040342;

IF payment_plan IN ('1 to 25') then WoE_payment_plan=-0.0656133698483868;Else
IF payment_plan IN ('26-50') then WoE_payment_plan=-0.0656133698483868;Else
IF payment_plan IN ('51-75') then WoE_payment_plan=-0.0656133698483868;Else
IF payment_plan IN ('76-100') then WoE_payment_plan=-0.0656133698483868;Else
IF payment_plan IN ('>100') then WoE_payment_plan=1.88649072517248;

IF employmnt_status IN ('Full Time') then WoE_employmnt_status=0.0045381236533386;Else
IF employmnt_status IN ('Homemaker') then WoE_employmnt_status=0.0045381236533386;Else
IF employmnt_status IN ('Other') then WoE_employmnt_status=0.0045381236533386;Else
IF employmnt_status IN ('Retired') then WoE_employmnt_status=-0.0336831132025726;Else
IF employmnt_status IN ('Self Employed') then WoE_employmnt_status=-0.0336831132025726;

IF age_range IN ('18 to 20') then WoE_age_range=-0.629118224061998;Else
IF age_range IN ('21 to 23') then WoE_age_range=-0.629118224061998;Else
IF age_range IN ('24 to 30') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('31 to 35') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('36 to 40') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('41 to 50') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('51 to 60') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('61 to 70') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('71 to 80') then WoE_age_range=0.110367490458345;Else
IF age_range IN ('Over 100') then WoE_age_range=0.110367490458345;

Run;

/************************************************WOE
Break****************************************/
TITLE;
TITLE1 "Correlation Analysis";
FOOTNOTE;
FOOTNOTE1;

PROC CORR DATA=Behav_training_WOE PLOTS=NONE PEARSON OUTP=Corr_logit VARDEF=DF;
        VAR WoE_capbal WoE_loan WoE_payment_plan WoE_Monthly_income WoE_age_range
WoE_overdue WoE_employmnt_status WoE_MoB
                ;
RUN;

/*  WoE_age_range WoE_loan*/

/************************************************Correlation Analysis
Break****************************************/
ODS GRAPHICS ON;
TITLE;
TITLE1 "Logistic Regression";
FOOTNOTE;
FOOTNOTE1 "scoring models";

/* WoE_capbal WoE_loan WoE_overdue WoE_payment_plan WoE_gross_income
WoE_employmnt_status */
/*              WoE_term WoE_age_range */
/*  */
PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
        MODEL new_default_status = WoE_capbal WoE_loan WoE_payment_plan WoE_gross_income
        WoE_age_range  WoE_overdue WoE_employmnt_status WoE_term
        / OUTROC=ROC SELECTION=stepwise SLE=0.1 SLS=0.1
                INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
                CLPARM=WALD CLODDS=WALD ALPHA=0.1 ;
RUN;

PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
        MODEL new_default_status = WoE_age_range WoE_gross_income WoE_capbal WoE_loan
WoE_term
        / OUTROC=ROC SELECTION=backward SLE=0.1 SLS=0.1
                INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
                CLPARM=WALD CLODDS=WALD ALPHA=0.1 ;
RUN;

PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
        MODEL new_default_status = WoE_age_range WoE_gross_income WoE_loan
        / OUTROC=ROC SELECTION=backward SLE=0.1 SLS=0.1
                INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
                CLPARM=WALD CLODDS=WALD ALPHA=0.05 ;
RUN;



PROC LOGISTIC DATA=Behav_training_WOE descending PLOTS(ONLY)=ALL;
        MODEL new_default_status = WoE_capbal WoE_loan WoE_payment_plan
        WoE_age_range WoE_employmnt_status WoE_term WoE_MoB WoE_Monthly_income
        / OUTROC=ROC SELECTION=backward SLE=0.1 SLS=0.1
                INCLUDE=0 CORRB CTABLE PPROB=(0.5) Scale=pearson RSQUARE LACKFIT LINK=LOGIT
                CLPARM=WALD CLODDS=WALD ALPHA=0.1 ;
RUN;

QUIT;
TITLE;
FOOTNOTE;
ODS GRAPHICS OFF;
```

```
proc export data=New_def_status
                outfile='/home/u63249191/Practice_sas/Dataset/EDA_data.csv' dbms=csv replace;
run;
```

(Delinquency code)

```
proc import
                datafile='/home/u63249191/Dissertation project/Data/Data_Prep_MoB.xlsx'
                out=sam1 dbms=xlsx;
run;

proc sort data=sam1 out=nodup_emptable nodupkey
                dupout=duplicate_table;
        by _all_;
run;

data cate_check;
set nodup_emptable;
if category = "Current" then category_2="0";
   else if category = "6+" then category_2="7";
   else if category = "Closed" then category_2= "";
       else category_2 =category;
num_category = input(category_2,2.);
run;

proc sql;
create table key_counts as
select key, count(snapshotdt) as SnapshotCount
from cate_check
group by key;
quit;

/* They have to have atleast 6 month of observation */
data keys_with_6_or_more;
set key_counts;
```

```
where SnapshotCount >= 6;
run;

proc sql;
create table merged_data as
select a.*, b.SnapshotCount
from sam1 as a
right join keys_with_6_or_more as b
on a.key = b.key;
quit;

/* The 6 months of observation has to be between 34 to 40 */
DATA Between_34_to_40;
  SET merged_data;
  if 34 <= snapshotdt <= 40; * 6 months of obs;
RUN;

proc sort data=Between_34_to_40  out=Last_6_obs;
by key snapshotdt;
run;

/* data last_obs; */
/* set Last_6_obs; */
/* by key; */
/* if last.key then output; */
/* run; */
/*  */
/* proc sort data=cate_check out=sample_sorted;  */
/* by key snapshotdt; */
/* run; */

/*This will give you all the variables when the loan was written (booked)
 at the point of the origination --- Compare category stage between firt_obs with last_obs*/
/* data first_obs; */
/* set sample_sorted; */
/* by key; */
/* if first.key then output; */
/* run; */
/*  */
/* proc freq data= last_obs; */
/* table status; */
/* run; */

proc sort data=cate_check  out=arrears_sorted;
by key num_category;
run;

/*This will give you the last observation per account (key_id) as it is sorted
in order of arrears (num_category)
*/
data max_arrears;
set arrears_sorted;
by key;
if last.key then output;
run;

/*This gives 1 observation (row) per account (key)
Most of the information is from the first observation
We also have new columns taken from 2 different dateset these include
Max ever arrears And the last status. If it ever defaulted then it will capture the "B" here.*/

proc sql;
create table First_obs_with_max_arrears as
select a. *
        , a. category as last_category
        , a. status as last_status_good_or_bad
        , c. num_category as max_arrears_status

/*This is now my base data. 1 observation per account (key)*/
from Between_34_to_40   as a
```

```
left join max_arrears as c
on a. key = c. key
;
quit;

proc freq data= First_obs_with_max_arrears;
table last_status_good_or_bad;
run;

/* Sample specific for Sampling & NOT for EDA */
/* Removing all closed accounts and Loan = 0 */
DATA No_closedacc;
  SET First_obs_with_max_arrears;
  IF last_status_good_or_bad = 'C' THEN Delete;
  else if loan = 0 then delete;
  else if app_score < 400 then delete; * outliers;
  else if beh_score > 743 then beh_score = 644;
RUN;

data Goo_bad_only;
set No_closedacc;
/* Set the new default status based on the last status */
IF last_status_good_or_bad = "G" Then new_default_status = 0;
else if last_status_good_or_bad = "B" then new_default_status = 1;
run;

data New_def_status_freq;
/* This data step calculates the new default status */
set Goo_bad_only (keep=snapshotdt key term app_score beh_score loan capbal overdue payment_plan
gross_income age_range employmnt_status MoB max_arrears_status new_default_status);
/* Set the new default status based on the maximum arrears status and the last status */
IF max_arrears_status > 1 THEN new_default_status = 1;
else if max_arrears_status <= 2 THEN new_default_status = 0;
Monthly_income = gross_income/12;
```

```
run;


proc freq data= New_def_status_freq;
table new_default_status;
run;


proc export data=New_def_status_freq
                    outfile='/home/u63249191/Practice_sas/Dataset/EDA_data_MOB.csv' dbms=csv
replace;
run;
```